# SBFLEVO_OPT

A program to calibrate the crop growth model SBFLEVO for sugar beet in Flevoland on optical reflectance and/or radar backscatter data

B.A.M. Bouman

cabo-dlo

Report 164, August 1992

## Contents

## Abstract

A description is given of the (FORTRAN) program SBFLEVO_OPT that calibrates the crop growth model SBFLEVO for sugar beet in Flevoland on time series of remote sensing measurements. The growth model SBFLEVO consists of the growth model SUCROS, extended with remote sensing models to simulate optical reflectance and L-, S-, C-, X-, Ku1- and Ku2-band radar backscatter (HH polarization, several angles of incidence). The calibration procedure of SBFLEVO_OPT is based on a controlled random search algorithm as developed by Price (1979) and adapted to the calibration of complex simulation models by Klepper (1989) and Rouse (Klepper and Rouse, 1991).

The structure of the program SBFLEVO_OPT is described and an explanation is given how to operate the program and how to interpret and evaluate the calibration results.

SBFLEVO_OPT is applied to data of 10 fields of commercially grown sugar beets in Flevoland in the 1991 growing season. The simulated beet yield with SBFLEVO before and after calibration is compared with actual yields as supplied by the farmers. The calibration of SBFLEVO on measured time-series of WDVI with a portable reflectance meter, strongly increased the accuracy of the simulation of beet yield. Before calibration, the difference between simulated and actually obtained average yield (74.3 t/ha) was -14.3 t/ha. After calibration, this difference was reduced to -1.9 t/ha. Reflectance measurements after day number 250 (the first week of September) should not be included in the calibration procedure. The application of the calibration program SBFLEVO_OPT with airborne and/or satellite-borne remote sensing measurements has still to be tested.

The appendices give the complete listings of the program SBFLEVO_OPT and its input files, and of the GENSTAT5 program STATIS to evaluate calibration results.

# 1 Introduction

Crop growth simulation models as developed by de Wit and co-workers (de Wit, 1965; Penning de Vries and van Laar, 1982; Spitters et al., 1989) describe the relation between physiological processes in plants and environmental factors such as solar irradiation, air temperature and water and nutrient availability. The models compute the daily growth and development rate of a crop, simulating the production of dry matter from emergence to maturity/harvest and its partitioning between organs. At the end of a simulation run, the final yield of the harvested product is obtained. These models were developed as a research tool to synthesize available knowledge on crop growth and development gathered by researchers from different agronomic disciplines. Not so surprisingly, these models often appear to fail when applied to practical field conditions for the purpose of yield predictions. Therefore, attention has been paid to improve model simulations by incorporating some actual information on the growth and development of field crops, such as contained in remote sensing data. At the Centre for Agrobiological Research (CABO-DLO) a method has been developed to calibrate crop growth models on time series of optical reflectance and/or radar backscatter measurements. This method was validated on 11 fields of sugar beet in different years and locations, and it was shown that the seasonal-average error in simulated above-ground biomass decreased from 390-700 kg/ha to 225-475 kg/ha (with 'end-of-season' biomass values of 5500-7000 kg/ha) with model calibration on optical and/or X-band radar data (Bouman, 1991; pp. 119-139). X-band radar backscatter (ground-based) and optical reflectance data were very effective in the calibration of the growth model SUCROS on sowing date or emergence date (model initialization). The radar backscatter data further adjusted the growth model only during early crop growth, whereas optical data still adjusted the growth model until late in the growing season. The developed calibration procedure on remote sensing data, however, was only validated for the simulation of canopy biomass because of lack of data on tuber biomass. In a follow-up study, this calibration procedure for sugar beet was further elaborated into the program SBFLEVO_OPT and validated for the simulation of tuber yield. This report contains a description of SBFLEVO_OPT and presents the results of the 'tuber yield' validation experiment on data of the 1991 growing season in Flevoland, The Netherlands. The program SBFLEVO_OPT calibrates the crop growth simulation model SBFLEVO for sugar beet in Flevoland on time series of optical reflectance and/or radar backscatter measurements. SBFLEVO consists of the crop growth simulation model SUCROS (Spitters et al., 1989) extended with routines to calculate optical reflectance and L-, S-, C-, X-, Ku1- and Ku2-band radar backscatter of sugar beet (Bouman, 1992). SBFLEVO was especially adapted to conditions in Flevoland because this area is a main test site for international remote sensing campaigns and experiments (e.g. Agrisar-86, Agriscatt 87-88, Maestro-89, Maceurope-91, ERS-1, X-SAR/SIR-C). A description and user's guide of SBFLEVO is presented in CABO-DLO report 163 (Bouman, 1992) and users of SBFLEVO_OPT are advised to take note of the contents of this. [CABO-DLO report 163 also presents a similar simulation model for winter wheat, WWFLEVO, but the calibration of this model on time series of remote sensing measurements turned out to be too unreliable because of inconsistency in the simulation of leaf growth, leaf senescence and assimilate partitioning in the SUCROS component of WWFLEVO].

Chapter 2 briefly explains the theory of the controlled random search algorithm that is the basis of the calibration procedure of SBFLEVO_OPT. Chapter 3 describes the structure of SBFLEVO_OPT and explains how to operate the program. The in- and output files are described and some hints are given for the interpretation and evaluation of calibration results, including a small statistical GENSTAT5 program called STATIS. In Chapter 4, SBFLEVO_OPT *is applied to a data set of ten farmers of sugar beet in Flevoland in* the 1991 growing season. The simulated beet yield with SBFLEVO before and after calibration is compared with actually obtained yields by the farmers.

The appendices contain the complete listings of the program SBFLEVO_OPT, of the input files, and of the GENSTAT5 program STATIS to evaluate calibration results. SBFLEVO_OPT is written in the programming language FORTRAN 77; the program was developed on a DEC-µVAX 3600 (operating system VMS 5.4, compiler VAX-FORTRAN v 5.6-199).

## 2 The calibration procedure

The growth model SBFLEVO simulates crop variables, such as LAI, soil cover and biomass, and optical reflectance and radar backscatter of sugar beet in Flevoland in any specified year from sowing to harvest. The program SBFLEVO_OPT calibrates SBFLEVO on time series of measured optical reflectance and/or radar backscatter measurements [though it can also be used to calibrate SBFLEVO on measured crop variables, e.g. biomass, LAI]: selected model parameters of the SUCROS module of SBFLEVO are re-parameterized so that the simulated time series of remote sensing signals match as closely as possible the measured time series of remote sensing signals (Bouman, 1991; pp.119-139). This 'matching' of simulated output to the measurements (calibration) is based on a controlled random search algorithm for global optimization as developed by Price in 1979. This algorithm was adapted to the calibration of complex simulation models by Klepper in 1989. Klepper and Rouse (1991) demonstrated the applicability of the Price algorithm for the calibration of crop growth models on measured crop variables. The current program SBFLEVO_OPT is an adaptation of the program written in 1989 by Rouse for the calibration of a potato growth model. Recently, a general program for calibration and uncertainty analysis of simulation models with a user's manual was published by Stol et al. (1992), based on the same work by Klepper and Rouse.

The exact theory of the Price algorithm is described in the above references. Here, only a brief explanation of the calibration procedure as executed in SBFLEVO_OPT is given (partly from Stol et al., 1992).

First, the parameters of the module SUCROS that have to be re-parameterized are selected: calibration parameters. For each of these parameters, a so-called biological plausible range is established within which the new parameters values have to be found. Parameter values outside this range are supposed to be biologically unrealistic/impossible. Next, the simulation variable (calibration variable) is selected that will be compared to measured data (calibration data) for the calibration. More than one calibration variable can be chosen, e.g. both optical reflectance and radar backscatter in a certain frequency band.

Then, in the first part of SBFLEVO_OPT, a number of calibration parameter sets (NPS) are generated consisting of parameter values for each calibration parameter chosen at random from a uniform distribution between their biological plausible ranges. Let P(i,j) represent the parameter in the jth parameter set where i denotes the calibration parameter to be re-parameterized. The growth model SBFLEVO is run for each parameter set and the simulated calibration variable is stored for each day that calibration data are available. A goodness-of-fit measure C is calculated as an absolute difference between simulated and measured data, averaged over the time series:

$$C = \Sigma(|Yt-Y't|)/T \quad (\Sigma \text{ for } t=1,2,....T)$$

where Y is the simulated calibration variable, Y' is the measured calibration data and t denotes the number in the time-series. [Users of SBFLEVO_OPT may want to change the calculation of C from average absolute residuals to the sum of squares of residuals; see subroutine CTRION in Chapter 3]. When more than one calibration variable is used, the maximum C-value of the different variables is selected:

$$C = \max(C(k), k=1,2,...n)$$

where C(k) is the goodness-of-fit value for calibration variable k. Thus, after running the growth model SBFLEVO for NPS times, a goodness-of-fit measure C is obtained belonging to each initial calibration parameter set.

In the second part of SBFLEVO_OPT, a new parameter set is obtained and again the goodness-of-fit measure C is calculated. This new parameter set is obtained by choosing at random q+1 different parameter sets from the existing series of NPS ones, where q is the number of calibration parameters in each parameter set. For each calibration parameter, a new value P(i,*) is calculated by:

$$P(i,*) = 2.G(i) - P(i,q+1)$$

where G(i) is the average value of parameter i from the first q randomly chosen parameter values. The C-value of the new parameter set is compared to the largest C-value in the existing NPS parameter set: if the new C-value is smaller than the largest existing C-value, then the new parameter set replaces the parameter set with the largest C-value. This procedure is repeated a pre-determined number of times (NIT) until the parameter set is found with the best goodness-of-fit (that is, with the lowest C-value) between simulated calibration variables and measured calibration data.

In the last part of SBFLEVO_OPT, the model SBFLEVO is finally run to obtain simulated time-series of calibration variables using different calibration parameter values: the parameter set with the best goodness-of-fit value (lowest C-value), the parameter set with the worst goodness-of-fit value (highest C-value), the average parameter values from the final NPS parameter set, and the nominal parameter values that are standard in SBFLEVO.

The main outcome of the program SBFLEVO_OPT is a set of 'optimum' parameter values for the crop growth model SBFLEVO that leads to the best match between simulated remote sensing signals (optical and/or radar) and measured remote sensing signals. It was shown by Bouman (1991) that the 'optimum' parameter set thus obtained also resulted in a better correspondence between simulated canopy biomass and actual canopy biomass of beet than the standard parameter set. In Chapter 4 of this report, it is also shown that the calibration of SBFLEVO on a time series of optical data (more precisely: the Weighted Different Vegetation Index) improves the simulation of beet yield of 10 farmers in Flevoland in the 1991 growing season. It should be noted however, that the calibration procedure in SBFLEVO_OPT optimizes a number of calibration parameters at the same time and that, therefore, the obtained optimum values may deviate from the true physical values (within the biological plausible ranges). For yield prediction applications, the calibration should be performed each year again on actual remote sensing observations: optimum parameter values in a previous year are no better for yield predictions in a next year than the standard parameter values in SBFLEVO.

# 3 Running the model

*3.1 The structure of the model*

SBFLEVO_OPT consists of a short main module which calls a total of 10 different subroutines and functions (Figure 1). Subroutine CROP is the program SBFLEVO and the subroutines called by CROP are not given in Figure 1 (see CABO-DLO report 163). The in- and output files that are needed and generated respectively by the various subroutines are schematically indicated in Figure 2. A total of three input files are needed (of which two for SBFLEVO) and seven output files are generated.

The main functions of the subroutines and the in- and output files are briefly explained below. The most important abbreviations of variables used in the subroutines and functions are explained in the headers (see the listings in appendix III)

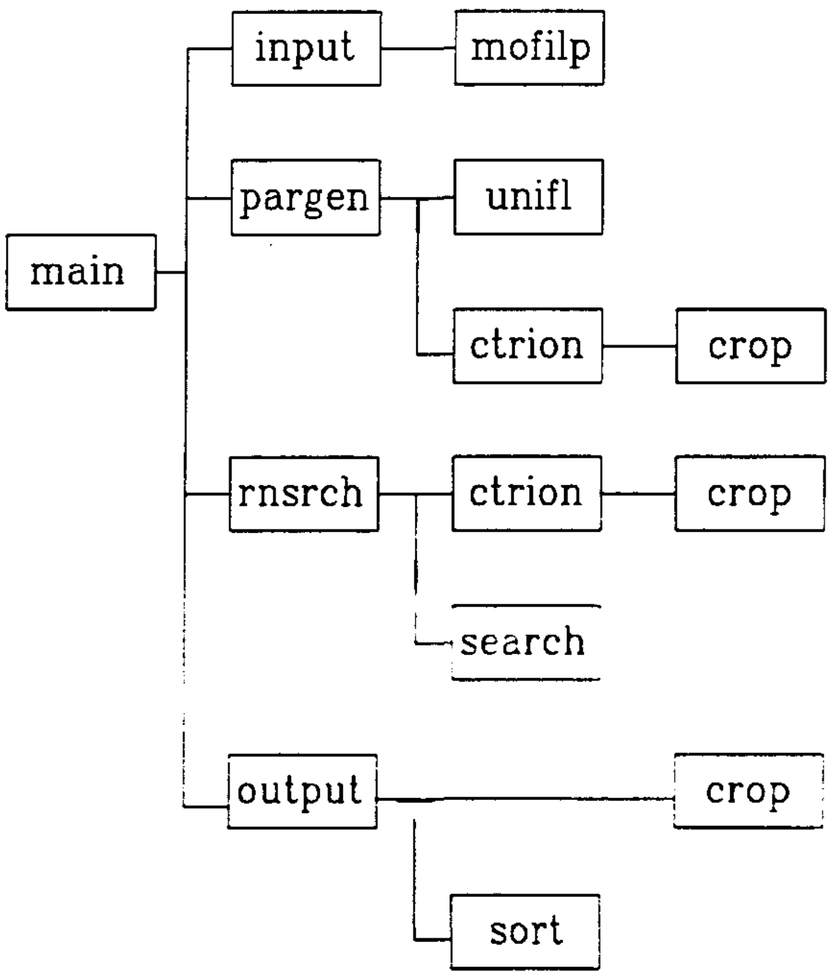


*Figure 1. Schematic illustration of the organization of SBFLEVO_OPT in main module and subroutines and functions. The subroutine CROP is the crop growth model SBFLEVO, and the subroutines of this model are not included in this diagram.*

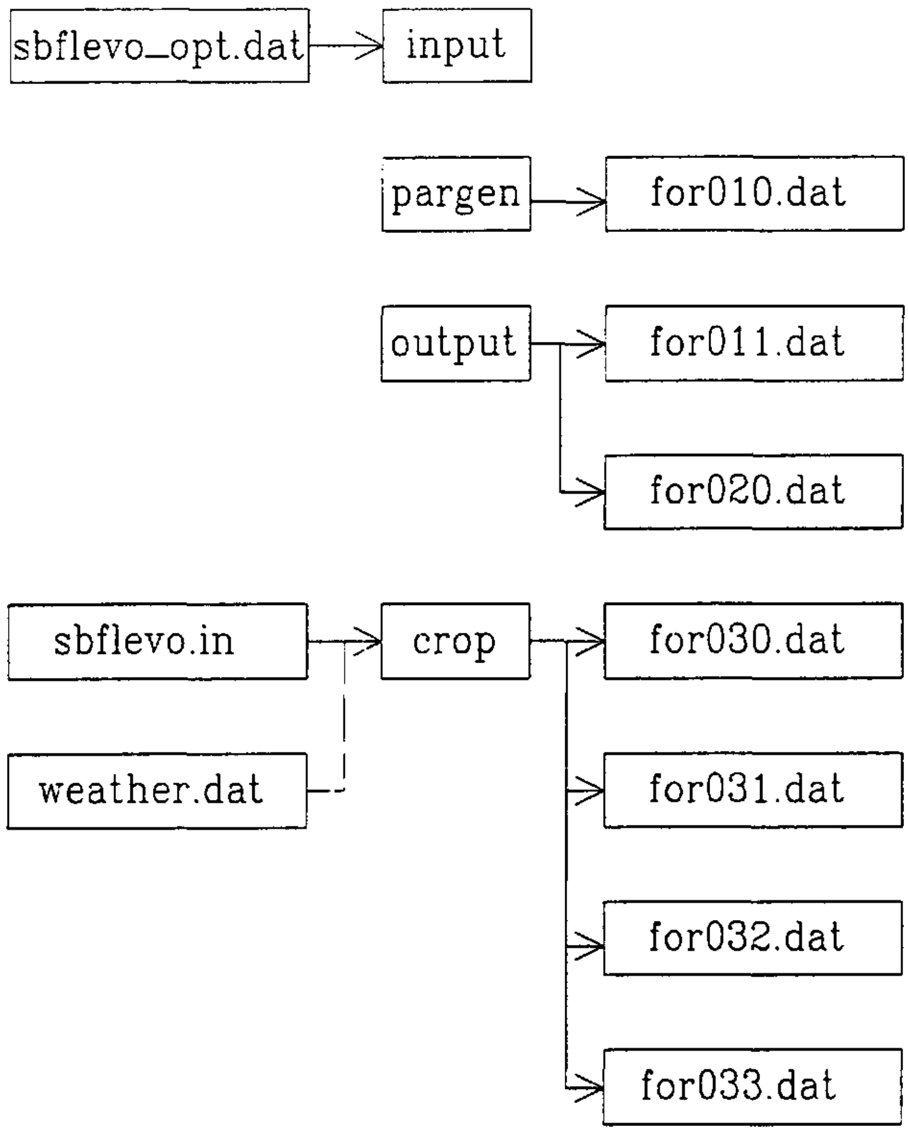


*Figure 2. In- and output files of the program SBFLEVO_OPT and their linkage to the subroutines.*

**MAIN**

The main module calls the program SBFLEVO_OPT.

**INPUT**

This subroutine reads the input data that control the calibration: the calibration parameters and their biological plausible ranges, the calibration data, the number of initial parameter sets (NPS), and the number of iterations of the calibration algorithm (NIT). The input data are read from the file SBFLEVO_OPT.DAT. More information on this subroutine is given in 3.2: Controlling the input and the calibration.

**MOFILP**
This subroutine is called in the subroutine INPUT to skip comment lines in the input file SBFLEVO_OPT.DAT.

**PARGEN**
This subroutine generates the initial (NPS) parameter sets and calculates the goodness-of-fit value C for each parameter set by calling CTRION. The initial parameter sets are generated by a call to UNIFL that chooses at random a number between 0-1 from a uniform distribution, and by subsequently transforming this random 0-1 number to a value between the biological plausible range for each calibration parameter. The randomly selected parameter values of the initial NPS parameter sets are written to the file FOR010.DAT.

**UNIFL**
This function is a high quality pseudo random number generator, constructed from the program UNIFL in Bratley et al., 1983.

**CTRION**
This subroutine calculates the goodness-of-fit value C to judge the correspondance between the simulated calibration variables and the measured calibration data for a given parameter set. The simulated calibration variables are obtained by calling the growth model SBFLEVO in subroutine CROP. The measured calibration data are passed from the subroutine INPUT to the MAIN module.

**RNSRCH**
This subroutine generates new parameter sets from randomly chosen subsets of the initial NPS parameter sets generated by PARGEN, and calculates their goodness-of-fit values C by calling CTRION. The C-value of each newly calculated parameter set is compared to the highest C-value from the existing parameter set (found by the subroutine SEARCH). If this new C-value is lower than the highest value, the newly calculated parameter set replaces the parameter set with the highest C-value in the existing set. This procedure is repeated NIT times (number of iterations).

**SEARCH**
This subroutine searches for the parameter set with highest C-value in the existing parameter sets.

**OUTPUT**
This subroutine calls SORT to sort the final parameter sets that are left after NIT iterations of RNSRCH to increasing C-value. The final NPS parameter values and the corresponding C-values are written to the file FOR011.DAT. The parameter values with the lowest C-value are written separately to the file FOR020.DAT (see also 3.3: The output and evaluation of the

calibration results). This subroutine also calls the crop growth model SBFLEVO (subroutine CROP) to get simulated calibration variables using different parameter sets.

**SORT**
This subroutine sorts an array in ascending order.

**CROP**
This subroutine is the crop growth model SBFLEVO (Bouman, 1992). It performs the crop growth simulation and calculates optical reflectance and radar backscatter of sugar beet grown in Flevoland. Crop data and remote sensing model parameters are read from the file SBFLEVO.IN, and weather data are read from the file specified in the last record of SBFLEVO.IN (WEER87.DAT,...., WEER91.DAT). Time series of calibration variables are written to file when CROP is called from the subroutine output: FOR030.DAT using the parameter set with the lowest C-value; FOR031.DAT using the parameter set with the highest C-value; FOR032.DAT using the average parameter values from the final NPS parameter set; FOR033.DAT using the nominal parameter values that are standard in SBFLEVO.

## *3.2 Controlling the input and the calibration*

The calibration is controlled by the input supplied in the file SBFLEVO_OPT.DAT that is read by the subroutine INPUT. The following definitions are important:
- Calibration parameter: model parameter of the module SUCROS that is to be re-parameterized. Examples are DAYSOW = date of sowing, RGRL = relative growth rate.
- Calibration variable: (remote sensing) state variable simulated by SBFLEVO that is the same as the measured (remote sensing) variable used for the calibration. Examples: simulated Weighted Difference Vegetation Index (WDVI; Clevers, 1988, 1989), simulated C-band radar backscatter at 40° incidence angle, in HH polarisation.
- Calibration data: measured (remote sensing) variable used to calibrate SBFLEVO upon. Examples: measured Weighted Difference Vegetation Index (WDVI), measured C-band radar backscatter at 40° incidence angle, in HH polarisation.

The parameters and data that control the calibration can be divided into three groups: the optimization parameters that specify the numbers of iterations and number of initial parameter sets; the calibration parameters; and the calibration data and variables.

*Optimization parameters*
The first parameters supplied in the file SBFLEVO_OPT.DAT are the number of (initial) parameter sets NPS and the number of iterations NIT (see 2: The calibration procedure). As standard NPS is set to 200 and NIT to 3000, but these values can be lowered or increased, depending on the evaluation of the results of the calibration (see 3.3 The output and evaluation of the calibration results).

*Calibration parameters*

Next in SBFLEVO_OPT.DAT the number of calibration parameters NOP have to be given. When the calibration is performed on optical data, this number is standard 4, and when the calibration is performed on radar data this number is standard 3. More calibration parameters can be included when the evaluation of the results indicates that the match between simulated and measured remote sensing data can still be better (extra parameters; see also 3.3 The output and evaluating the calibration results). The calibration parameters are the following:

| Standard parameters | | Optical | Radar |
|---|---|---|---|
| DAYSOW | sowing date | x | x |
| RGRL | relative growth rate | x | x |
| EFF | light use efficiency | x | x |
| LSHAD | maximum leaf area | x | |
| Extra parameters | | | |
| NPL | number of plants | x | x |
| AMX | potential $CO_2$ ass. rate ($\mu g$ $CO_2$ $m^{-2}s^{-1}$) | x | x |

Note: the crosses in the columns 'optical' and 'radar' indicate whether the mentioned parameters can be calibrated using optical or radar remote sensing data.

After the number of parameters NOP is given, the calibration parameters are specified together with their biological plausible ranges. The following ranges are standard values:

| Standard parameter | Lower limit | Upper limit |
|---|---|---|
| DAYSOW | 80 | 120 |
| RGRL | 0.0130 | 0.0170 |
| EFF | 0.0100 | 0.0150 |
| LSHAD | 3.5 | 5.5 |
| Extra parameter | | |
| NPL | 7 | 11 |
| AMX | 1.1000 | 1.4000 |

During the calibration procedure, parameter values are chosen from between these biological plausible ranges and passed to the growth model SBFLEVO (subroutine CROP) via the array PASP2(I), where I indexes the calibration parameter. In the subroutine CROP, the actual values of the calibration parameters are reset to the chosen values stored in PASP2 after *reading all model parameters from the file* SBFLEVO.IN. Therefore, the model parameters that are reset in CROP should be equal to and in the same order as the calibration parameters specified in the file SBFLEVO_OPT.DAT. In subroutine CROP:

```
DAYSOW = PASP2(1)
RGRL   = PASP2(2)
EFF    = PASP2(3)
```

etc. according to the calibration parameters specified in SBFLEVO_OPT.DAT.

Finally, in the subroutine OUTPUT the calibration parameter values with the lowest goodness-of-fit values C are written to the file FOR020.DAT via the array PASP3(I). Note that the names of the calibration parameters again have to match the specified calibration parameters in SBFLEVO_OPT.DAT:

```
WRITE(20,*) 'DAYSOW, PASP3(1)
WRITE(20,*) 'RGRL',  PASP3(2)
WRITE(20,*) 'EFF',   PASP3(3)
```

etc. according to the calibration parameters specified in SBFLEVO_OPT.DAT.

*Calibration data and variables*

The last input supplied in SBFLEVO_OPT.DAT relate to the measured calibration data. A 'data set' is a time series of remote sensing data of a given field. When replicate measurements are made of the same field (e.g. another plot in the same field, or a replicate in an agricultural trial), these replicates belong to the same data set. In most remote sensing observations, however, whole fields are measured and no replicates are given. In SBFLEVO_OPT.DAT, a number of data sets (NDS) can sequentially be given up to a maximum of five. The switch IDSNK controls the data set(s) to be used for calibration. The data are entered in the following order:

- First, the number of data sets is given: NDS (up to 5)
- Then, the data sets that will be used for calibration: IDSNK. The value 1 means that the data set will be used; the value 0 that it will not be used. The example 1 0 0 0 0 means that the first data set will be used.
- Next the data sets are sequentially entered. For each data set the following information is supplied:
    - Number of data points: NDP; i.e. the number of remote sensing observation. In the example below: 7
    - Number of replicates: NREP (usually 1)
    - The day numbers and the measured (remote sensing) calibration data. The calibration can be performed to 1, 2 or 3 data variables (i.e type of remote sensing measurements). All three variables have to be supplied and are stored in the arrays FIX1(I), FIX2(I) and FIX3(I), where I denotes the time-sequence. When only 1 data variable is used, then FIX2 and FIX3 have to be filled with 999.9; when only 2 data variables are used, then FIX3 has to be filled with 999.9. In an example, the calibration is only performed on the data variable WDVI:

| Day number | FIX1 | FIX2 | FIX3 |
|---|---|---|---|
| 089 | 0.0 | 999.9 | 999.9 |
| 103 | 1.0 | 999.9 | 999.9 |
| 117 | 7.4 | 999.9 | 999.9 |
| 131 | 24.3 | 999.9 | 999.9 |
| 145 | 40.1 | 999.9 | 999.9 |
| 170 | 44.9 | 999.9 | 999.9 |
| 200 | 45.0 | 999.9 | 999.9 |

Note that the maximum length of the array FIX1, FIX2 and FIX3 is limited to 100.

During the calibration procedure, the simulated calibration variables are stored in the arrays MPAR1, MPAR2 and MPAR3 (and also PAR1M, PAR2M and PAR3M) at the end of the main module of the subroutine CROP for comparison with the measured calibration data that are stored in FIX1, FIX2 and FIX3. Therefore, the simulation variables stored in MPAR have to be of the same type and in the same order as the calibration data given in the data sets in SBFLEVO_OPT.DAT. When FIX2 and/or FIX3 were given as 999.9, then MPAR2 (PAR2M) and/or MPAR3 (PAR3M) should also be given as 999.9. In the example from above:

```
IF (IID.EQ.JDAY(IDSN,K)) THEN
      MPAR1(IDSN,K) = WDVI
      MPAR2(IDSN,K) = 999.9
      MPAR3(IDSN,K) = 999.9
      ....
END IF

IF (IID.EQ.JDAY(IDSN,IFD)) THEN
      PAR1M(IDSN,K2) = WDVI
      PAR2M(IDSN,K2) = 999.9
      PAR3M(IDSN,K2) = 999.9
      ....
END IF
```

The IF-statement is meant to pass the simulated calibration variable to the arrays PASM and MPASP synchronous with the measured calibration data stored in FIX.

The calibration variables and calibration data can be of any (remote sensing) kind that is simulated by the growth model SBFLEVO (see CABO-DLO report 163, Bouman, 1992). The simulated remote sensing variables are:

<u>I. Optical reflectance</u>
- Nadir green reflectance NADG
- Nadir red reflectance NADR

- Nadir infra-red reflectance NADIR
- Infrared/green ratio RATIO
- Normalized Difference Vegetation Index NDVI
- Weighted Difference Vegetation Index WDVI (calculated with EXTRAD)
- Weighted Difference Vegetation Index WDVIM (calculated with metamodel)

<u>II. Radar backscatter</u>

- X-band radar backscatter (calibrated on ground based radar measurements) at vertical like-polarization VV: GAMMA(I) where I denotes the incidence angle:

| I | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Incidence angle | 10° | 20° | 30° | 40° | 50° | 60° | 70° | 75° |

- L-, S-, C-, X-, Ku1- and Ku2-band radar backscatter (calibrated on airborne DUTSCAT measurements) at horizontal like-polarization HH: GAMMAA(I,J), where I denotes the incidence angle and J denotes the frequency:

| I | 1 | 2 | 3 | 4 | 5 | |
|---|---|---|---|---|---|---|
| Incidence angle | 20° | 30° | 40° | 50° | 60 | |
| J | 1 | 2 | 3 | 4 | 5 | 6 |
| Frequency | L | S | C | X | Ku1 | Ku2 |

(Thus: GAMMAA(1,3) simulates the C-band ERS-1 satellite data at 20° incidence angle, and GAMMAA(2,1) to GAMMAA(3,1) simulates the L-band JERS-1 satellite data at 30-40° incidence angle).

### *3.3 The output and evaluation of the calibration results*

SBFLEVO_OPT.DAT generates a number of output files with the results of the calibration (Figure 2) that can be evaluated on the effectiveness of the calibration.

The two output files FOR010.DAT and FOR011.DAT contain the initial and the final sets of calibration parameter values together with the corresponding C-values (goodness-of-fit measures). In the columns are the values of the selected calibration parameters and of the corresponding goodness-of-fit measure C; the rows are the NPS parameter sets. The file FOR011.DAT is sorted on ascending C-values. The C-values of both files can be compared to check if the calibration procedure has indeed minimized the goodness-of-fit measure C (C-values in FOR011.DAT should be lower than C-values in FOR010.DAT). Note that no column-headings are given; the columns of calibration parameters are in the same order as supplied by the user in SBFLEVO_OPT.DAT; the goodness-of-fit measure C is the last column.

A small program STATIS.GEN, written with the statistical package GENSTAT5 (GENSTAT5 Reference Manual, 1988) makes histograms of the calibration parameters in both FOR010.DAT and FOR011.DAT and calculates the coefficient of correlation between the calibration parameters. These histograms show whether the initial parameter values (FOR010.DAT) were indeed uniformly distributed between the biological plausible ranges; and whether the calibration procedure has converged to certain parameter values (FOR011.DAT). If this latter is not the case for some calibration parameter, two conclusions can be drawn (provided that SBFLEVO_OPT is implemented and run correctly):

1. The number of parameter sets (NPS) and/or the number of iterations (NIT) should be increased so that covergence is finally reached, or
2. The simulation of the chosen calibration variable is not sensitive to that specific calibration parameter in the time-frame of the measured calibration data. The time-frame of the remote sensing measurements should be expanded (more measurements earlier or later in the growing season), or, in a next run with SBFLEVO_OPT, that specific parameter can be dropped as calibration parameter for that specific calibration run.

The output file FOR020.DAT lists the calibration parameter values that lead to the lowest C-value, i.e. that lead to the best fit between simulated and measured calibration variables.

The output files FOR030.DAT to FOR033.DAT contain time series of the simulated calibration variables. Note that no column-headings are given: the first column gives the day number and the second to fourth columns give the calibration variables as stored in PASP1M to PASP3M (see 3.2 Controlling the input and the calibration). FOR030.DAT contains simulation variables using the parameter set with the lowest C-value; FOR031.DAT using the parameter set with the highest C-value; FOR032.DAT using the average parameter values from the final NPS parameter set; FOR033.DAT using the nominal parameter values that are standard in SBFLEVO. The simulated (remote sensing) variables in FOR030.DAT should match the corresponding measured calibration data supplied by the user in SBFLEVO_OPT.DAT. If this is not the case (or not sufficiently), two options can be used to improve the calibration:

1. The number of parameter sets (NPS) and/or the number of iterations (NIT) can be increased so that a better convergence is reached (check the degree of convergence with the histograms of the final parameter sets of FOR011.DAT, and with the distributions of the C-values), or
2. The selected calibration parameters in SBFLEVO_OPT.DAT can be extended with some extra parameters (see 3.1 Controlling the input and the calibration).

Though tempting as it might be, the biological plausible ranges should not be extended over the given limits (see 3.1: Controlling the input and the calibration) in order to try to improve the calibration. When, after options 1 and 2 have been executed, the deviations between simulated and measured (remote sensing) variables are still too great, it should be concluded that the environmental conditions for that specific crop were outside the validity range of the model SBFLEVO. [This can indicate the occurrence of some severe growth limitations which, in itself, is a valuable observation for qualitative growth monitoring!].

## 4 Model results for 1991

### *4.1 Materials*

In 1991, Flevoland was one of the test sites for the international Maceurope campaign and for a Dutch study on monitoring possibilities with the ERS-1 satellite (ESA AO project). During Maceurope, multi-temporal radar data were acquired with the NASA/JPL SAR system in the L-, C- and P-band; optical scanner data were acquired with the AVIRIS, the CAESAR and the GER; and aerial photographs were obtained with an ultra-light system. At the time of this study, however, these data were not yet fully processed and calibrated to test the model SBFLEVO_OPT on aerial remote sensing data. The ERS-1 was launched only halfway the growing season and, therefore, this incomplete data set was not suitable for use in SBFLEVO_OPT.

As part of the ground-truth collection in 1991, optical reflectance was measured of sugar beet fields of 10 farmers in the region with a portable field reflectance meter CROPSCAN (Uenk et al., 1991). From the reflectance measurements in the infra-red (IR; ~844-856 nm) and green (GR; ~545-555 nm) channels, the Weighted Difference Vegetation Index WDVI (Clevers, 1988, 1989) was calculated:

$$WDVI = IR - 1.3*GR \quad (\%)$$

These time series of WDVI data were suitable for the application of SBFLEVO_OPT to calibrate the growth model SBFLEVO and to test whether the calibrated model gave more accurate simulations of final beet yield than the model using standard parameter values. Data on the actual beet yield of the farmers was obtained by the Agricultural University of Wageningen, Department of Landsurveying and Remote Sensing through questionnaires to the farmers (Bücker et al., 1992).

### *4.2 Calibration results*

First, SBFLEVO was run to simulate crop growth with standard input parameter values (Bouman, 1992). Since in most practical situations of (regional) yield prediction, no actual information on sowing and harvesting dates is available, the simulation was performed for a hypothetical average sugar beet crop with a sowing date of day number 105 (April 15) and a harvesting date of day number 295 (October 22). Simulated beet yields are given together with the actually obtained yields by the farmers in Table 1. The average yield was underestimated by 14.3 t/ha (19% error). Next, the model SBFLEVO was run with actual sowing and harvesting dates of all 10 farmers. Now, the underestimation of the average beet yield was reduced to 6.2 t/ha (8.3% error). It should be repeated, however, that for practical yield predictions, no information on actual sowing and harvesting dates is generally available.
SBFLEVO_OPT was then used to calibrate SBFLEVO on the measured time-series of WDVI for all 10 fields separately. The sowing date was one of the calibration parameters and the harvesting date was again taken as day number 295. Table 2 gives the calibration parameter values that resulted in the best correspondence between simulated and measured time-series of WDVI. The results of the calibration are illustrated in Figure 3 for three sugar beet fields. For field 6 and 9 (Figure 3a and 3b), the last two WDVI measurements were not

included in the calibration procedure. The WDVI of sugar beet often tends to decrease in the last phase of the growing season due to some yellowing and changes in the architecture of the canopy. Since these changes are not accounted for by the remote sensing models in SBFLEVO (namely EXTRAD and a semi-empirical metamodel), this decrease in WDVI can not be simulated by SBFLEVO. Therefore, as a rule of thumb, it is advised not to use optical reflectance measurements after about day number 250 (the first week of September) in the calibration procedure. In the case of field 6, including the last two WDVI measurements led to a sugar beet yield of 63.3 t/ha (instead of 65.3 t/ha, see Table 1), whereas in the case of field 10, it did not make any marked difference.

The simulated beet yields after calibration of SBFLEVO are given in Table 1, and the simulated versus actually obtained yields is plotted in Figure 4. After calibration on WDVI, the difference between realized average yield and simulated average yield was only - 1.9 t/ha (2.6% error). *[For comparison: in a data set of three sugar beet fields in Flevoland in 1988, the calibration procedure resulted in a difference between realized average yield (68.1 t/ha) and simulated average yield of + 2.3 t/ha (3.4% error).]*

Figures 5 to 9 give the development of the simulated WDVI, LAI, soil cover, dry canopy biomass and beet weight respectively for fields 6, 9 and 10 after calibration and with the standard parameter values.

### *4.3 Conclusion and discussion*

It is concluded that the calibration of SBFLEVO on measured time-series of WDVI with a field reflectance meter increased the accuracy of the simulation of beet yield of ten farmers in Flevoland. Before calibration, the difference between actually obtained average yield (74.3 t/ha) was -14.3 t/ha, and after calibration, this difference was only -1.9 t/ha. Reflectance measurements after day number 250 (the first week of September) should not be included in the calibration procedure.

The application of the calibration program SBFLEVO_OPT with airborne and/or satellite-borne remote sensing measurements has still to be tested. Especially the suitability of radar data from the ERS-1 (C-band) and the JERS-1 (L-band) has to be evaluated in the coming years.

*Table 1. Actually obtained and simulated beet yields (ton/ha) of ten farmers in Flevoland in the 1991 growing season. Yactual = actually obtained yield; Ysim1 = simulated yield with sowing date=95 and harvest date=295; Ysim2 = simulated yield with actual sowing and harvest dates; Ycal = simulated yields after model calibration on measured time-series of the Weighted Difference Vegetation Index WDVI. The number N refers to the number of WDVI measurements. Note that no reflectance measurements were made on the second field of farmer 4 and of farmer 10.*

| Farmer | Yactual | Ysim1 | Ysim2 | Ycal | N |
|---|---|---|---|---|---|
| 1 | 87 | 60.0 | 69.4 | 80.0 | 9 |
| 2 | 79 | 60.0 | 70.6 | 76.7 | 9 |
| 3 | 75 | 60.0 | 70.3 | 74.3 | 9 |
| 4 | 77 | 60.0 | 70.6 | 71.5 | 9 |
|  | 81 | 60.0 | 74.6 | - | - |
| 5 | 69.5 | 60.0 | 63.5 | 70.5 | 8 |
| 6 | 68 | 60.0 | 63.3 | 65.3 | 11 |
| 7 | 70 | 60.0 | 65.6 | 69.9 | 12 |
| 8 | 61 | 60.0 | 58.2 | 68.1 | 11 |
| 9 | 70 | 60.0 | 63.6 | 72.2 | 15 |
| 10 | 77 | 60.0 | 72.8 | 75.7 | 12 |
|  | 77.5 | 60.0 | 74.8 | - | - |
| Average | 74.3 | 60.0 | 68.1 | 72.2 | 11 |

*Table 2. Parameter values that resulted in the best correspondence between simulated and measured WDVI-time series (lowest C-value) for 10 fields in Flevoland in the 1991 growing season. The standard (nominal) values of the parameters are also given for comparison.*

| Farmer | DAYSOW | RGRL | EFF | LSHAD |
|---|---|---|---|---|
| 1 | 96 | 0.0169 | 0.0150 | 5.118 |
| 2 | 92 | 0.0150 | 0.0150 | 4.649 |
| 3 | 102 | 0.0168 | 0.0149 | 5.021 |
| 4 | 104 | 0.0169 | 0.0142 | 3.907 |
| 5 | 122 | 0.0170 | 0.0147 | 4.844 |
| 6 | 124 | 0.0149 | 0.0145 | 5.372 |
| 7 | 119 | 0.0166 | 0.0150 | 5.130 |
| 8 | 124 | 0.0167 | 0.0146 | 5.489 |
| 9 | 105 | 0.0166 | 0.0148 | 5.181 |
| 10 | 98 | 0.0163 | 0.0148 | 5.426 |
| Average | 109 | 0.0164 | 0.0146 | 4.828 |
| Nominal | 95 | 0.0156 | 0.0125 | 4.500 |

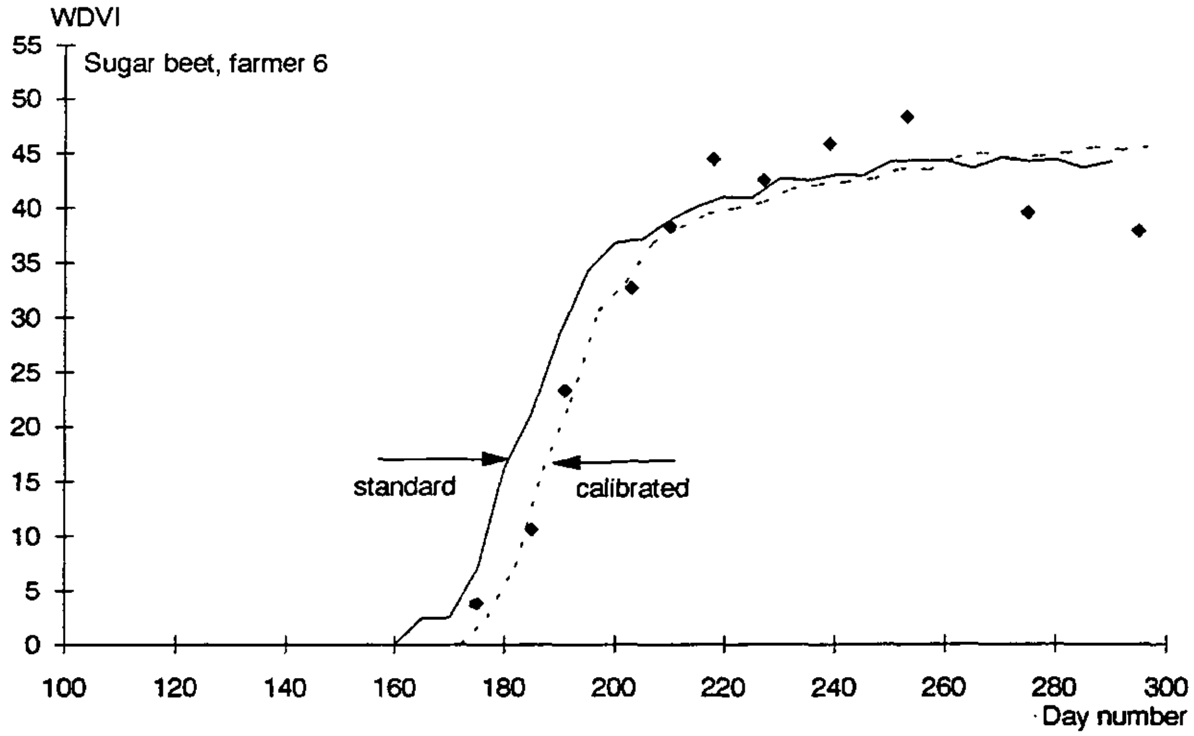


*Figure 3a. Measured (◆) and simulated WDVI before calibration (standard: __________) and after calibration (- - - - -) of sugar beet, farmer 6, Flevoland 1991.*

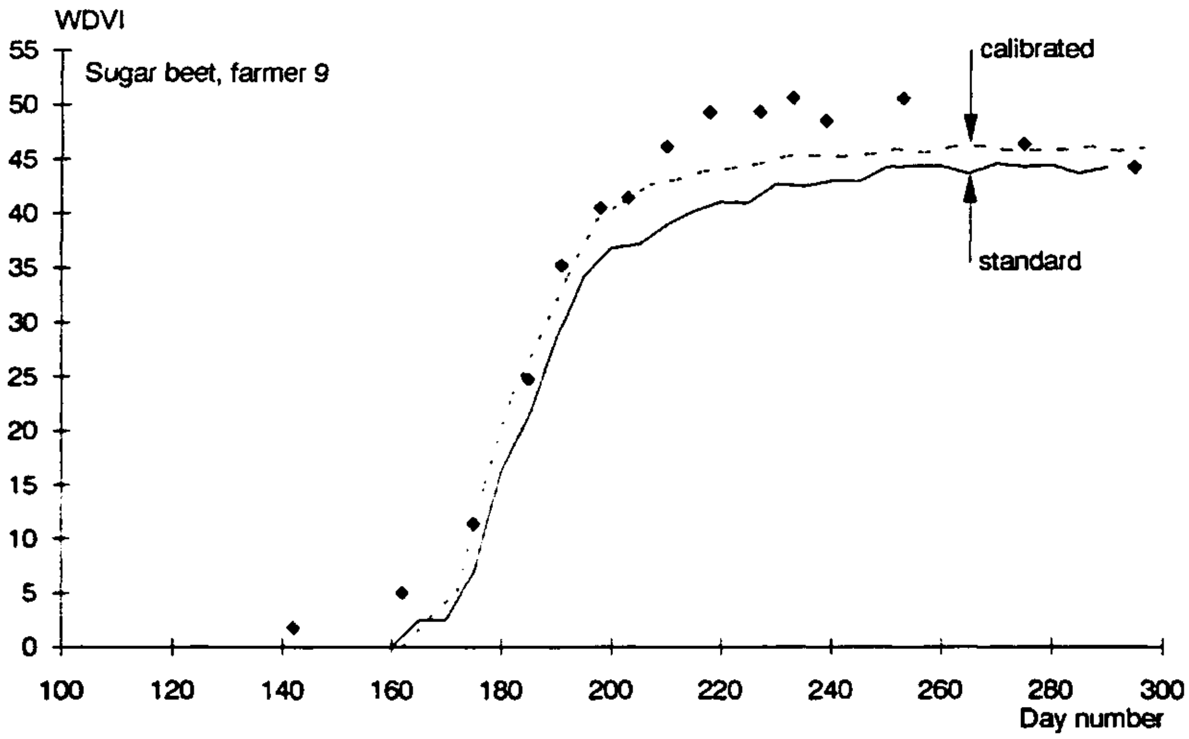


*Figure 3b. Measured (◆) and simulated WDVI before calibration (standard: __________) and after calibration (- - - - -) of sugar beet, farmer 9, Flevoland 1991.*

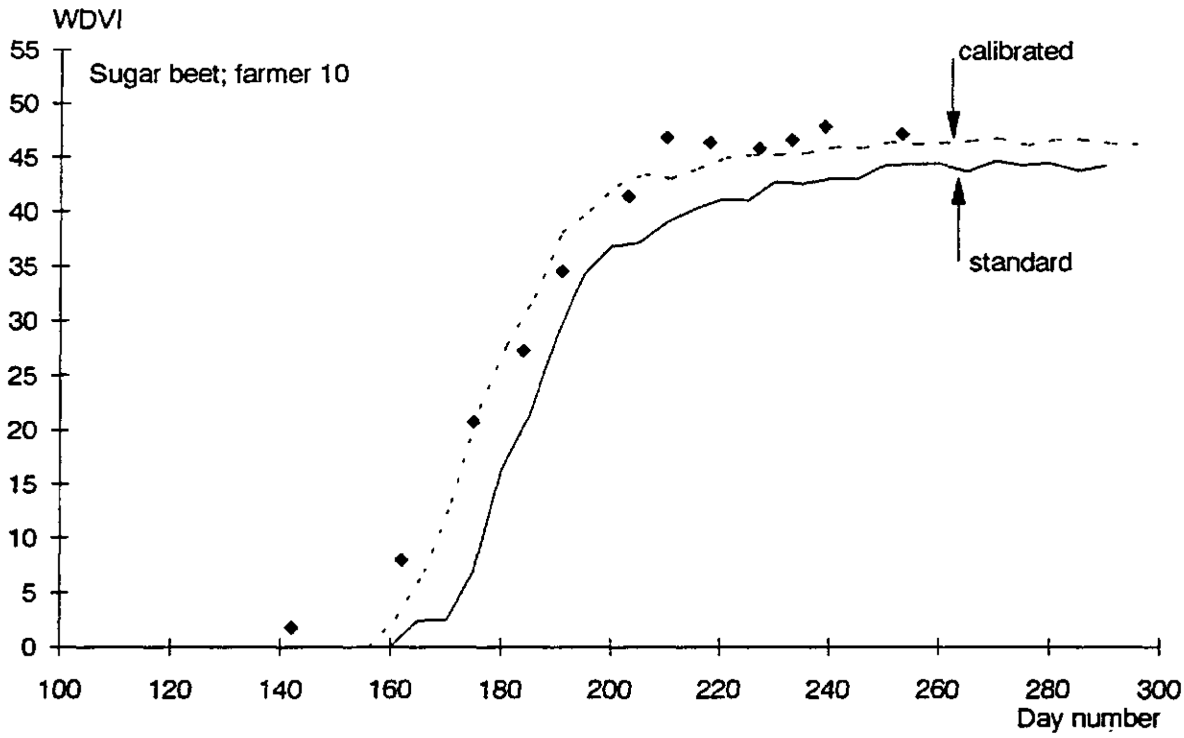


*Figure 3c. Measured (◆) and simulated WDVI before calibration (standard: __________) and after calibration (- - - - -) of sugar beet, farmer 10, Flevoland 1991.*

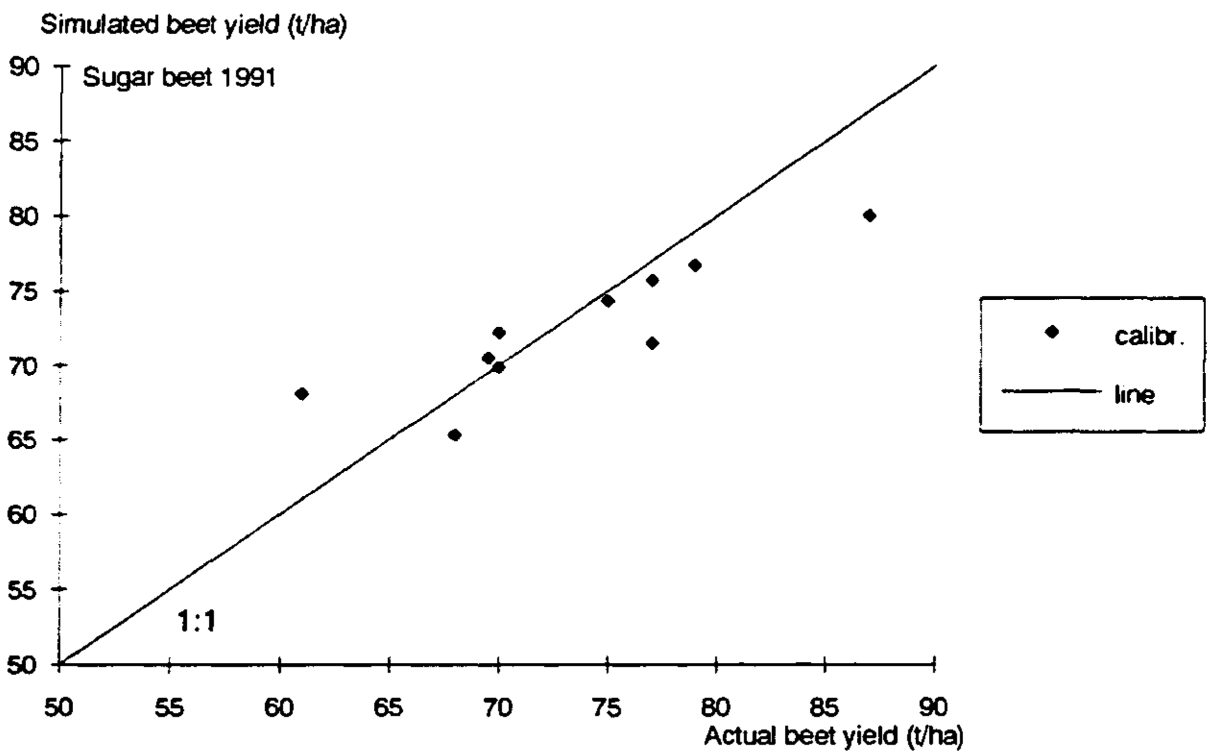


*Figure 4. Simulated beet yield after calibration versus actually obtained beet yield for ten farmers in Flevoland, 1991.*

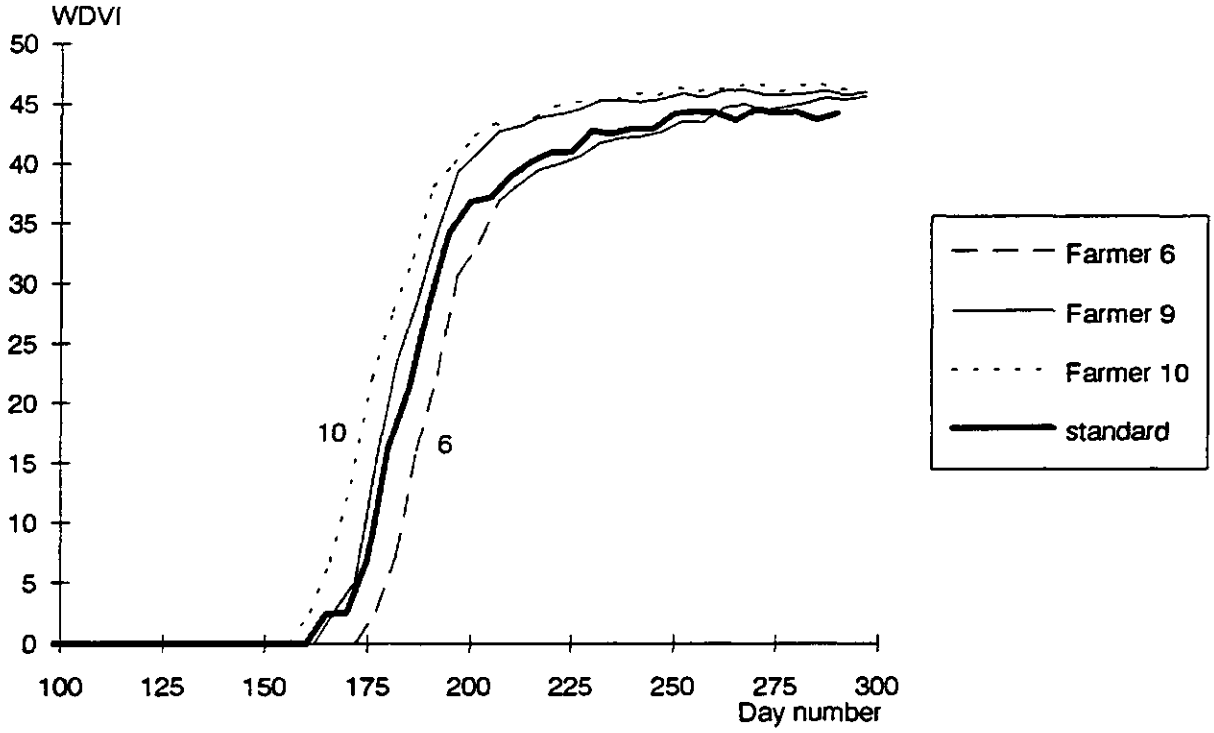


*Figure 5. Simulated WDVI of sugar beet before calibration (standard) and after calibration for farmers 6,9 and 10 in Flevoland, 1991.*

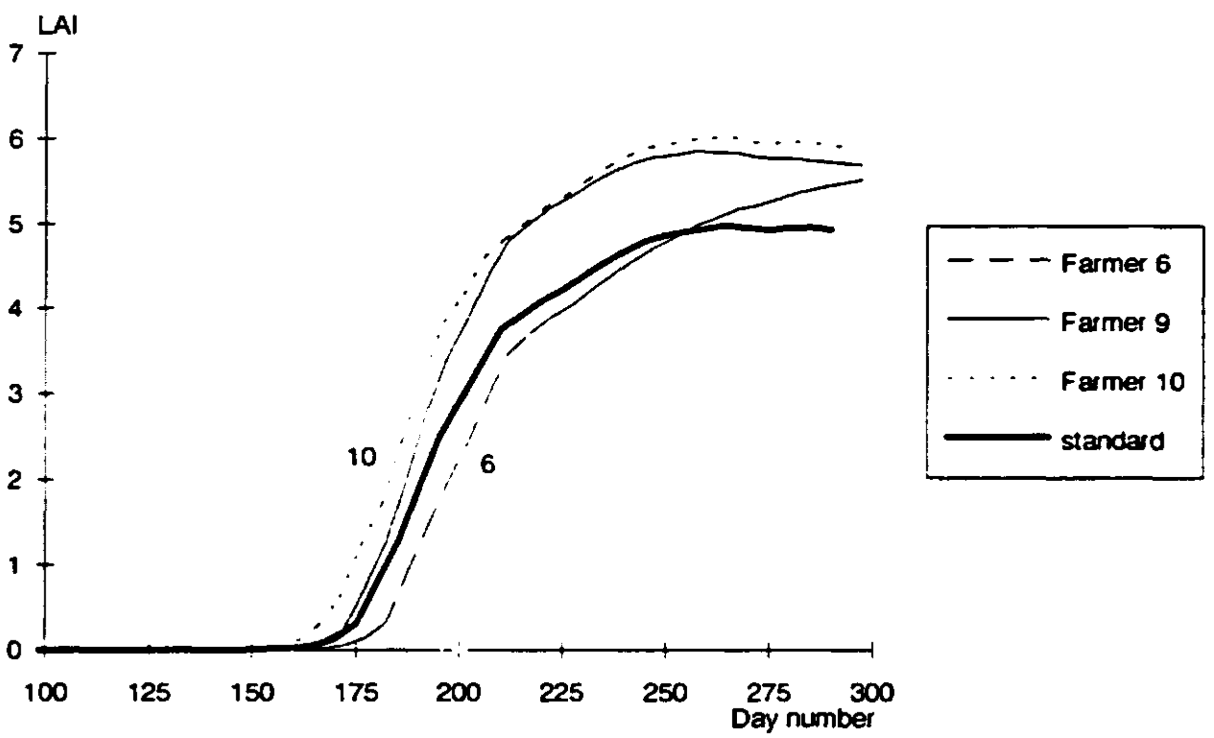


*Figure 6. Simulated LAI of sugar beet before calibration (standard) and after calibration for farmers 6,9 and 10 in Flevoland, 1991.*

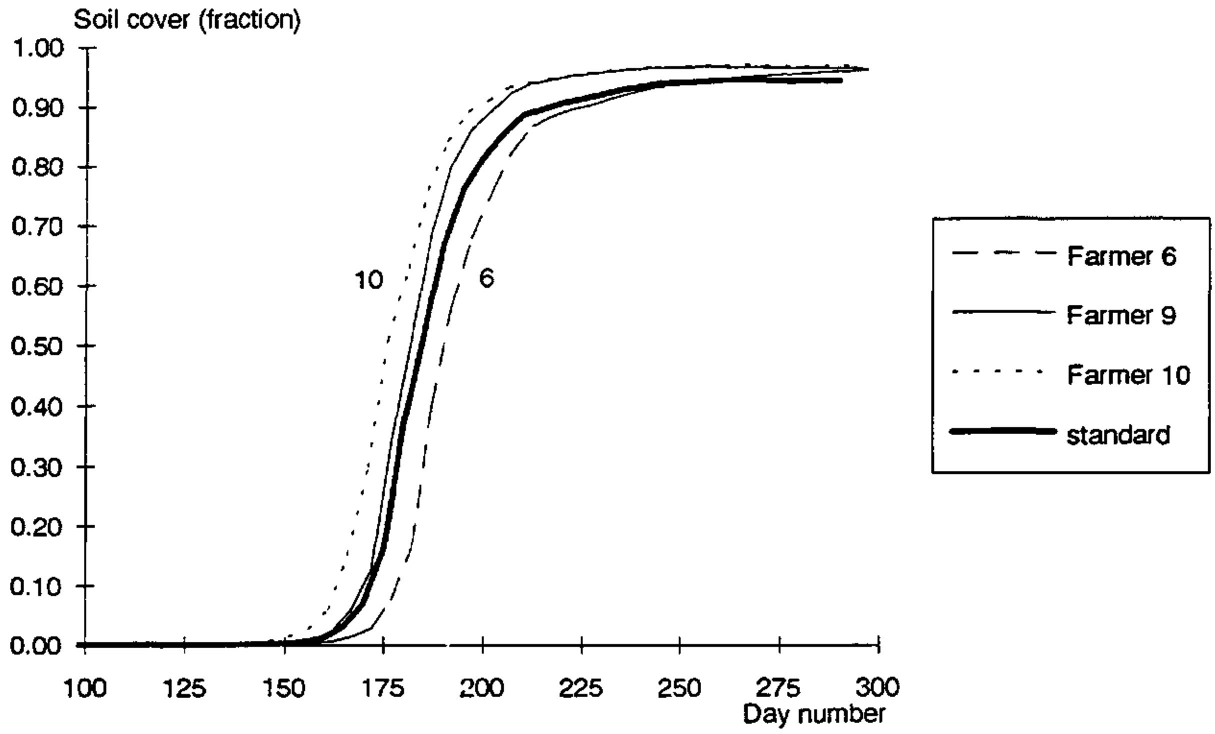


*Figure 7. Simulated soil cover of sugar beet before calibration (standard) and after calibration for farmers 6,9 and 10 in Flevoland, 1991.*

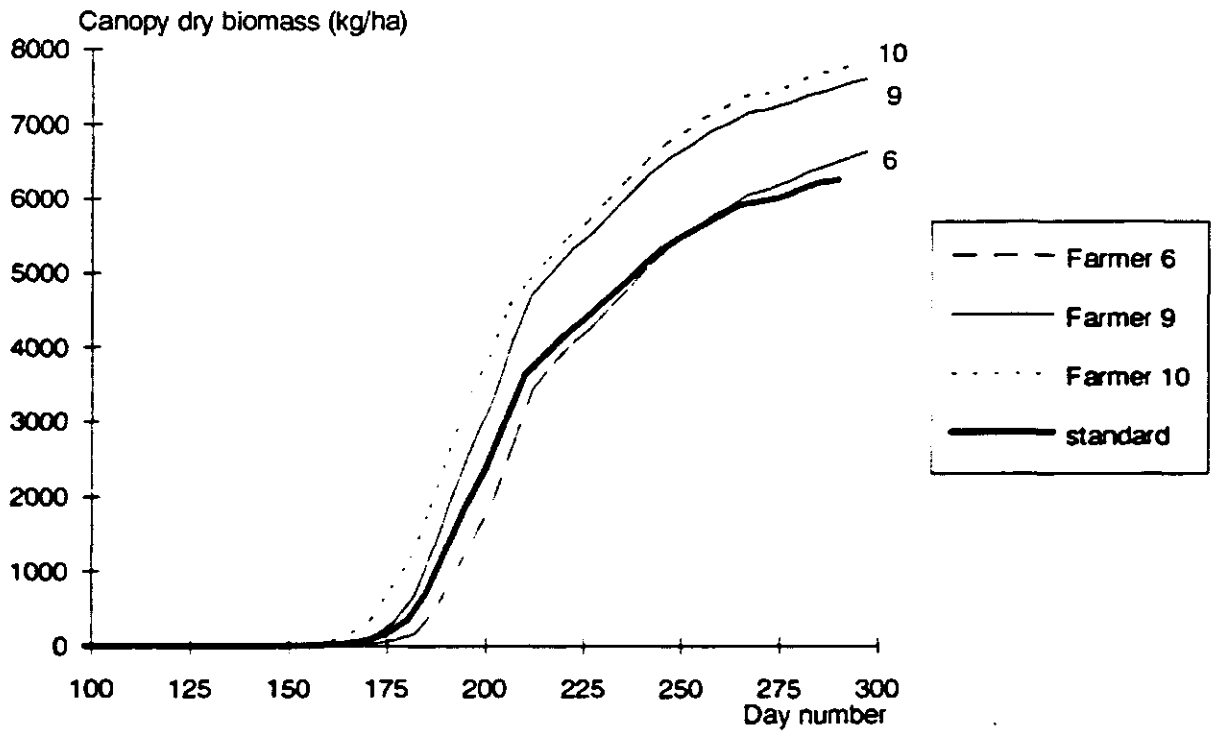


*Figure 8. Simulated dry canopy biomass of sugar beet before calibration (standard) and after calibration for farmers 6,9 and 10 in Flevoland, 1991.*

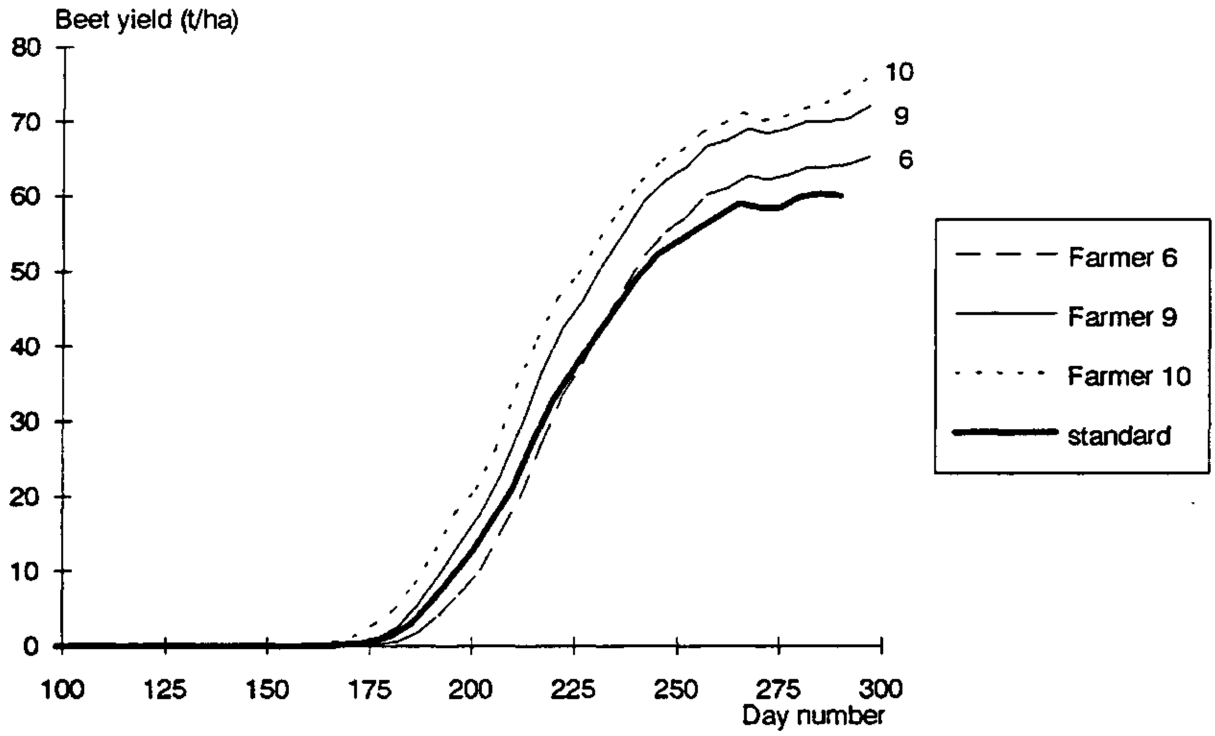


*Figure 9. Simulated yield of sugar beet before calibration (standard) and after calibration for farmers 6,9 and 10 in Flevoland, 1991.*

**Acknowledgements**

W. Stol is thanked for his critical review of the manuscript of this report.

## References

Bouman, B.A.M. (1991), Linking X-band radar backscattering and optical reflectance with crop growth models, Thesis Agricultural University Wageningen, The Netherlands; 169 pp.

Bouman, B.A.M. (1992), SBFLEVO and WWFLEVO, Growth models to simulate crop growth, optical reflectance and radar backscatter of sugar beet and winter wheat, calibrated for Flevoland, CABO-DLO report 163, CABO-DLO, Wageningen, The Netherlands; 61 pp plus appendices.

Bratley, P., B.L. Fox & L.E. Schrage, (1983), A guide to simulation, Springer-Verlag, New York, USA

Bücker, C., J.G.P. Clevers, H.J.C. van Leeuwen, B.A.M. Bouman & D. Uenk, 1992, Optical component Mac Europe, Ground truth report Flevoland 1991, Agricultural University Wageningen - Department of Landsurveying and Remote Sensing, Report LUW-LMK-199204, Wageningen, The Netherlands; 55 pp.

Clevers, J.G.P.W. (1988), The derivation of a simplified reflectance model for the estimation of leaf area index, Remote Sensing of Environment, 25:53-69.

Clevers, J.G.P.W. (1989), The application of a weighted infrared-red vegetation index for estimating leaf area index by correcting for soil moisture, Remote Sensing of Environment, 29:25-37

GENSTAT5 Reference Manual, (1988), Oxford Science Publication, UK.

Klepper, O. (1989), A model of carbon flows in relation to macrobenthic food supply in the Oosterschelde estuary (S.W. Netherlands), Thesis, Agricultural University Wageningen, The Netherlands; 270 pp.

Klepper, O. & D.I. Rouse, (1991), A procedure to reduce parameter uncertainty for complex models by comparison with real system output illustrated on a potato growth model, Agricultural Systems 36: 375-395.

Penning de Vries, F.W.T. & H.H. van Laar (Eds.), (1982), Simulation of plant growth and crop production, PUDOC, Wageningen, The Netherlands, 308 pp.

Price, W.L. (1979), A controlled random search procedure for global optimisation, The Computer Journal, 20:367-370.

Spitters, C.J.T., H. van Keulen & D.W.G. van Kraalingen, (1989), A simple and universal crop growth simulator: SUCROS87, In Simulation and systems management in crop protection: 147-181, edited by R. Rabbinge, S.A. Ward and H.H. van Laar (PUDOC Wageningen, The Netherlands).

Stol, W., D.I. Rouse, D.W.G. van Kraalingen & O. Klepper, (1992), FSEOPT, a FORTRAN program for calibration and uncertainty analysis of simulation models, Simulation report CABO-TT 24, CABO-DLO and Agricultural University Wageningen, Wageningen, The Netherlands; 24 pp. plus appendices.

Wit, C.T. de, (1965), Photosynthesis of leaf canopies, Agricultural Research Report 663, Centre for Agricultural Publications and Documentation (PUDOC), Wageningen, The Netherlands; 57 pp.

Uenk, D., B.A.M. Bouman & H.W.J. van Kasteren, (1992), Reflectiemetingen aan landbouwgewassen, Handleiding voor het meten van gewasreflectie, Standaardlijnen voor de bepaling van bodembedekking en LAI (in Dutch), CABO-DLO report 156, CABO-DLO, Wageningen, The Netherlands; 56 pp.

## APPENDICES

The appendices contain the listings of the following files and programs:

I. SBFLEVO_OPT.DAT (input file SBFLEVO_OPT)
II. SBFLEVO.IN (input file subroutine CROP = growth model SBFLEVO)
III. SBFLEVO_OPT.FOR (calibration program of growth model SBFLEVO)
IV. STATIS.GEN (GENSTAT5 program for evaluation output SBFLEVO_OPT)

I

# APPENDIX I SBFLEVO_OPT.DAT

**Input datafile for the calibration program SBFLEVO_OPT**

```
*-----------------------------------------------------------*
*  SBFLEVO_OPT.DAT                                           *
*  Input datafile for the calibration program SBFLEVO_OPT   *
*  This file contains parameters and data that control      *
*  the calibration: parameters that specify the number of   *
*  iterations, the calibration parameters with their bio-   *
*  logical plausible ranges, and the calibration data.      *
*                                                            *
*  The data in this file should be altered/adapted for any *
*  calibration with SBFLEVO_OPT.                             *
*                                                            *
*  This example-file contains calibration data of 5         *
*  farmers of sugar beet in Flevoland in the 1991 growing   *
*  season (calibration data are WDVI values measured with   *
*  a portable reflectance meter                             *
*------------------------------------------------------------*
*
*-----number of parameter sets; NPS
*   number of iterations of algorithm; NIT
*   units: -
200
3000
*
*-----number of parameters to be calibrated is read; NOP
*   units: -
4
*
*-----format for input of parameter bounds is:
*   (A6,1X,F10.5,1X,F10.5)
*   units: variable
DAYSOW  80.00000 125.00000
RGRL    0.01300  0.01700
EFF     0.01000  0.015000
LSHAD   3.50000  5.50000
*
*-----the number of data sets; NDS
*   (meaning number of year / locations
*   of data is entered)
*   units: -
5
*
*-----the data sets (years) which will be run; IDSNK
*   units: -
0 0 0 1 0
```

```
*
*-----number of data points in the first set are
*    entered; NDP
*    units: -
9
*
*-----number of replicates; NREP
*    units: -
1
*
*-----experimental field data of first data set (farmer 3; 1991)
*    free format reading
*    date, wdvi, dummy1, dummy2
185 30.404 999.9 999.9
192 38.043 999.9 999.9
204 43.734 999.9 999.9
210 44.74 999.9 999.9
218 45.95 999.9 999.9
227 40.546 999.9 999.9
239 46.357 999.9 999.9
253 46.718 999.9 999.9
295 39.725 999.9 999.9
*
*-----number of data points in the second set are
*    entered; NDP
*    units: -
9
*
*-----number of replicates; NREP
*    units: -
1
*
*-----experimental field data of second data set (farmer 2; 1991)
*    free format reading
*    date, wdvi, dummy1,dummy2
186 32.691 999.9 999.9
192 35.694 999.9 999.9
204 38.093 999.9 999.9
210 42.366 999.9 999.9
218 45.02 999.9 999.9
227 46.963 999.9 999.9
239 44.703 999.9 999.9
253 45.579 999.9 999.9
295 36.136 999.9 999.9
*
*-----number of data points in the third set are
*    entered; NDP
*    units,-
```

```
9
*
*-----number of replicates; NREP
*    units: -
1
*
*-----experimental field data of third data set (farmer 4; 1991)
*    free format reading
*    date, dummy1,dummy2
185 29.573 999.9 999.9
192 38.483 999.9 999.9
204 40.87 999.9 999.9
210 44.264 999.9 999.9
218 45.906 999.9 999.9
227 44.019 999.9 999.9
239 43.938 999.9 999.9
253 43.293 999.9 999.9
295 38.757 999.9 999.9
*
*-----number of data points in the fourth set are
*    entered; NDP
*    units: -
12
*
*-----number of replicates; NREP
*    units: -
1
*
*-----experimental field data of fourth data set (farmer 7; 1991)
*    free format reading
*    date, wdvi,dummy1,dummy2
175 5.727 999.9 999.9
184 18.177 999.9 999.9
190 26.485 999.9 999.9
198 36.641 999.9 999.9
203 39.431 999.9 999.9
210 44.038 999.9 999.9
218 46.472 999.9 999.9
227 47.022 999.9 999.9
239 46.129 999.9 999.9
253 46.748 999.9 999.9
275 49.926 999.9 999.9
295 41.615 999.9 999.9
*-----number of data points in the fifth set are
*    entered; NDP
*    units: -
8
*
```

```
*-----number of replicates; NREP
*    units: -
1
*
*-----experimental field data of fifth data set (farmer 5; 1991)
*    free format reading
*    date, wdvi,dummy1,dummy2
185 20.954 999.9 999.9
204 44.934 999.9 999.9
210 46.933 999.9 999.9
218 50.096 999.9 999.9
227 49.236 999.9 999.9
239 46.227 999.9 999.9
253 45.667 999.9 999.9
295 39.082 999.9 999.9
*
```

**APPENDIX II SBFLEVO.IN**

**Inputfile with crop data and remote sensing-model data (optical and radar) for the growth model SBFLEVO for sugar beet in Flevoland.**

*The text in italic does not make part of the real inputfile, but are comments for clarification.*

** General crop data, sugar beet*
52.000 LATitude (The Netherlands)
9.000 NPL, number of plants per $m^2$
.845 LA0, initial leaf area per plant in $cm^2$
105.000 TEMERG, tempsum before emergence
.0156 RGRL, relative growth rate per degreeday of leaf area
4.500 LSHAD, LAI above which death of leaf due to shading
2.000 TBASE, base temperature
.0020 SLA, specific leaf area
1.2500 AMX, Amax in $\mu g\ CO_2\ m^{-2}s^{-1}$
.0125 EFF, light use efficiency in $\mu g\ J^{-1}$(PAR absorbed)
.690 KDIF, extinction coefficient diffuse PAR
.58 KCOVER, extinct. coeff. to calculate soil cover
.200 SCV, scattering coefficient visible light
2.000 Q10
.003 MAINSO, maintenance coefficient of storage organs ($d^{-1}$)
1.290 ASRQSO, assimilate requirement to grow storage organs
.130 RDSTLV, fraction redistributed of dying leaves
0.36 RDSTST, fraction redistributed of dying petioles

** Soil moisture content data for 'Cloud' model for radar backscatter: first number=daynr, second=moist.content(%), third=daynr, fourth = moist. content(%), etc. Number of data is not limited, number is given as first figure (here: 4).*
4 data IMST, soil moisture content
0. 5.0 365. 5.0

** Crop data on temperature effects and assimilate partitioning*
10 data AMDVST, relative effect of temperature sum on Amax
.000 .500 500.000 1.000 700.000 1.000 1700.000 .800
3000.000 .600
16 data AMTMPT, relative effect of temperature on Amax
-10.000 .010 3.000 .010 10.000 .750
15.000 1.000 20.000 1.000 26.000 .750
33.000 .010 45.000 .010
10 data FSHTB, partitioning to shoot growth
.000 .800 400.000 .700 900.000 .520
901.000 .220 3000.000 .220
10 data FLVTB, partitioning of shoot growth to leaves
.000 .650 370.000 .650 665.000 .440
820.000 .290 3000.000 .290
10 data FSTTB, partitioning of shoot growth to petioles

```
   .000     .300  370.000     .300   665.000     .470
   820.000  .610  3000.000    .610
10  data    FCRTB, partitioning of shoot growth to crown
   .000     .050  370.000     .050   665.000     .090
  820.000    .100  3000.000    .100
12  data    FRTTB, partitioning of below ground growth to fibrous roots
   0.000    1.0    400.     1.     500.      0.5
  1000.     0.1   2000.     0.03   3000.     0.03
```
** further crop data*
```
10  data    RDRT, relative death rate
   0.     0.    600.     0.    1000.     .00022
  1500.    .00050  2500.    .00075
10  data    BDMPT, dry matter percentage of beets
   0.     .135    800.     .135   1150.     .160
  2000.    .242   2500.     .242
```

** RADAR BACKSCATTER*

** Input data for 'Cloud' model for X-band radar backscatter. First the angles of incidence are given; then the G-parameters per angle of incidence; and then the C-parameters per angle of incidence. Finally K, D and the plant water content. Data derived from ground-based X-band, VV polarisation, ROVE measurements 1980 on De Schreef in Flevoland.*

```
8 data INC, incidence angles measured radar backscatter NC
10. 20. 30. 40. 50. 60. 70. 75.
8 data IGS, soil parameter G cloud ROVE-1980, VV
0.525 0.174 0.120 0.095 0.076 0.065 0.055 0.042
8 data ICPL, plant parameter C cloud CPL ROVE-1980, VV
1.060 1.190 1.200 1.150 1.170 1.150 0.980 0.930
0.06 KS
0.46 DPL-1980 (ROVE, VV)
90.83 PLWC
```

** Input data for 'Cloud' model for L-,S-,C-,X-,Ku1- and Ku2-band radar backscatter. Data derived from airborne, HH polarisation, DUTSCAT measurements during Agriscatt 1988 in Flevoland. First the angles of incidence are given:*

```
20. 30. 40. 50. 60. data incidence angles Agriscatt
```

** then, on a line the G-parameters per frequency from L- to Ku2-band, the first line being the parameters for the first angle of incidence (here 20°), the second line for the second angle of incidence, ....*

```
0.00190 0.013024 0.019637 0.125781 0.163298 0.077177
0.00085 0.009141 0.011464 0.067289 0.102148 0.074368
0.00070 0.004448 0.049184 0.034643 0.120707 0.076001
0.00082 0.003735 0.004577 0.021858 0.065574 0.070116
0.00079 0.002933 0.003907 0.025829 0.107194 0.029653
```

** then, on a line the C-parameters per frequency from L- to Ku2-band, the first line being the parameters for the first angle of incidence (here 20°), the second line for the second angle of incidence, ....*

```
.19115 .32000 .58845 1.04065 1.94800 1.37346
.12776 .27578 .50725 .83942 2.66803 1.14837
.11424 .30995 .44199 1.22765 2.57024 1.14212
```

.11386 .34963 .32138 1.13502 2.59050 1.33077
.10328 .36640 .49978 .96138 1.95570 1.20513
** then the K-parameters, per frequency,*
0.100 0.069 0.058 0.048 0.044 0.041
** and finally, the D-parameters per frequency.*
1.1025 0.2314 0.1009 0.2099 1.0000 1.0000

** CANOPY REFLECTANCE*
** Input data for the EXTRAD model to calculate canopy reflectance,*
** data derived from reflectance measurements during Agriscatt 1987*
** and 1988 in Agriscatt campaign.*
0.146 RHOSG green reflection coeff. soil ASCAT
0.166 RHOSR red reflection coefficient soil;
0.199 RHOSIR ir reflection coefficient soil; ASCAT
0.294 SCATG green scattering coeff. leaves ASCAT
0.079 SCATR red scattering coefficient leaves
0.974 SCATIR ir scattering coefficient leaves ASCAT
60. BETA solar height
0.015 0.045 0.074 0.1 0.123 0.143 0.158 0.168 0.174 F leaf angle distribution
** Input data for metamodel WDVI-LA (data also from Agriscatt):*
48.64 WMAX maximum WDVI in metamodel WDVI-LAI
0.485 WEXT extinction coefficient in metamodel WDVI-LAI

** Initialization of crop variables*
0 WLVG
0 WLVD
0 WSTG
0 WSTD
0 WSO
0 WRT
0 WCR
0 LAI
0 TSUM2
0. TSUMEM
0 EMERG
** Timer parameters*
95 DAYSOW
90. TIME
295. FINTIM
5 PRDEL
** Data file that contains the weather data*
'WEER90'

## APPENDIX III SBFLEVO_OPT.FOR

I

## APPENDIX III SBFLEVO_OPT.FOR

**FORTRAN listing of the program SBFLEVO_OPT for the calibration of the growth model SBFLEVO for sugar beet in Flevoland on measurements of optical reflectance and/or radar backscatter.**

```
*-----------------------------------------------------------------*
*  PROGRAM SBFLEVO_OPT.FOR                                        *
*                                                                 *
*  Author : B.A.M. Bouman; 1989, 1992                             *
*  Based on model developed by Doug Rouse & Willem Stol           *
*  (24-FEB-1989)                                                  *
*                                                                 *
*    This program executes an algorithm to calibrate the          *
*    combined crop growth and remote sensing (optical+radar)      *
*    model SBFLEVO for sugar beet in Flevoland, The Netherlands.  *
*    The calibration can be executed on radar and optical         *
*    reflectance data, as well as on crop data (ground-truth)     *
*                                                                 *
*  DOCUMENTATION:                                                 *
*  Bouman, B.A.M., 1992, SBFLEVO_OPT, A program to calibrate the  *
*  crop growth model SBFLEVO for sugar beet in Flevoland on       *
*  optical reflectance and/or radar backscatter data, CABO-DLO    *
*  report 164, (CABO-DLO, P.O. box 14, 6700 AA Wageningen,        *
*  The Netherlands)                                               *
*                                                                 *
*  SUBROUTINES and FUNCTIONS called :                             *
*                                                                 *
*  INPUT   -  gets field data and initial conditions              *
*  PARGEN  -  generates parameter sets and calculates Criterion   *
*             values C                                            *
*  RNSRCH  -  performs random search and optimization algorithm   *
*  OUTPUT  -  writes final values of parameter sets and           *
*             calculates statistics on them                       *
*                                                                 *
*  FILE usage : none                                              *
*-----------------------------------------------------------------*

      PROGRAM SBFLEVO_OPT

*-----MAIN module

      REAL C(500), FIX1(5,100), FIX2(5,100), FIX3(5,100)
      REAL LB(50), P(50,500), PASP(50), UB(50), WFIX1, WFIX2, WFIX3

      INTEGER IDSNK(5)
      INTEGER JDAY1(5,100), NDP(5), NDS, NIT, NOP, NPS, NRP(5)
      LOGICAL RESET
```

```
*-----the subroutine INPUT contains the calibration data (radar, optical
*     or crop data) and initial values of the model parameters to be
*     calibrated

      CALL INPUT (NPS,NOP,NDP,NDS,NIT,NRP,LB,UB,FIX1,FIX2,FIX3,
     $            WFIX1,WFIX2,WFIX3,IDSNK,JDAY1)

*-----the subroutine PARGEN generates the (calibration) model parameter
*     sets and calls the CROP model (SBFLEVO) for the purpose
*     of calculating the criterion value

      CALL PARGEN (NPS,NOP,NDP,NDS,P,C,LB,UB,PASP,FIX1,FIX2,FIX3,
     $            WFIX1,WFIX2,WFIX3,NRP,IDSNK,JDAY1)

*-----the subroutine RNSRCH performs the random search of model parameter
*     values and the optimization procedure

      CALL RNSRCH (NPS,NOP,NDP,NDS,NIT,P,C,LB,UB,PASP,FIX1,FIX2,
     $            FIX3,WFIX1,WFIX2,WFIX3,NRP,IDSNK)

*-----the subroutine OUTPUT is used to control the output (Note: output
*     is written mainly in the subroutine CROP (SBFLEVO).

      CALL OUTPUT(NPS,NOP,NDS,P,C,IDSNK,LB,UB)

      STOP
      END
```

```
*---------------------------------------------------------------------*
*  SUBROUTINE PARGEN                                                  *
*                                                                     *
*  Authors: Doug Rouse & Willem Stol                                  *
*  Date   : 01-FEB-1989                                               *
*  Adapted: B.A.M. Bouman; 1989, 1992                                 *
*  Purpose: This subroutine generates NPS parameter sets.  Each       *
*           parameter in each parameter set is choosen at random      *
*           using the random number generating function UNIFL().      *
*                                                                     *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)    *
*  name   type meaning                               units  class     *
*  ----   ---- -------                               -----  -----     *
*  NPS     I4  Number of Parameter Sets                -     I,O      *
*  NOP     I4  Number of Parameters                    -     I,O      *
*  NDP     I4  Number of Data Points                   -     I,O      *
*  NDS     I4  Number of Data Sets                     -     I,O      *
*  P       R4  Parameter space                         -      O       *
*  C       R4  Criterion value for each parameter set  -      O       *
*  LB      R4  Lower Bound of parameter(s)        variable    I       *
*  UB      R4  Upper Bound of parameter(s)        variable    I       *
*  PASP    R4  Parameter set array                variable    O       *
*  FIX1    R4  Calibration data 1                 variable    I       *
*  FIX2    R4  Calibration data 2                 variable    I       *
*  FIX3    R4  Calibration data 3                 variable    I       *
*  WFIX1   R4  Summation of calibration data 1    variable    O       *
*  WFIX2   R4  Summation of calibration data 2    variable    O       *
*  WFIX3   R4  Summation of calibration data 3    variable    O       *
*  NRP     I4  Number of RePlicates                    -      I       *
*  IDSNK   I4  Data Set Number                         -      I       *
*                                                                     *
*  FATAL ERROR CHECKS (execution terminated, message): none           *
*                                                                     *
*  WARNINGS: none                                                     *
*                                                                     *
*  SUBROUTINES and FUNCTIONS called :                                 *
*                                                                     *
*       - CTRION()    - calculates Criterionn value C for each        *
*                       parameter set                                 *
*       - UNIFL       - pseudo random number generator                *
*                                                                     *
*  FILE usage : none                                                  *
*---------------------------------------------------------------------*

      SUBROUTINE PARGEN (NPS,NOP,NDP,NDS,P,C,LB,UB,PASP,FIX1,FIX2,FIX3,
     $             WFIX1,WFIX2,WFIX3,NRP,IDSNK,JDAY1)

      INTEGER NPS,NOP,NDP(5),NDS,TB,NRP(5),IDSNK(5)
```

```
      INTEGER IJ,I,JDAY1(5,100),J,JJ,IJK

      REAL C(500),MULTR(50),LB(50),UB(50),P(50,500),PASP(50),CT
      REAL FIX1(5,100),FIX2(5,100),FIX3(5,100),WFIX1,WFIX2
      REAL WFIX3,PAR1M,PAR2M,PAR3M,APAR1T,APAR2T,APAR3T
      REAL SMALF1(5,365),SMALF2(5,365),SMALF3(5,365),LARGF1(5,365)
      REAL LARGF2(5,365),LARGF3(5,365),UNIFL

*-----common block is common with subroutine CROP (growth model SBFLEVO)
*     passing the final value of simulated calibration variables
*     back for printing

      COMMON/T/PAR1M(5,25),PAR2M(5,25),PAR3M(5,25)

*-----common block CONFIN is common with subroutine CROP (model SBFLEVO)

      COMMON/CONFIN/APAR1T(5,365),APAR2T(5,365),APAR3T(5,365)

*-----end of declarations

*-----variables retain their values between subsequent calls of
*     this subroutine

      SAVE

*-----initialize array used to find smallest and largest
*     values of weights of output variables through time

      DO 20 IJ = 1, NDS
        DO 10 I = 1, 160
          SMALF1 (IJ,I) = 999999.
          SMALF2 (IJ,I) = 999999.
          SMALF3 (IJ,I) = 999999.
          LARGF1 (IJ,I) = -99999.
          LARGF2 (IJ,I) = -99999.
          LARGF3 (IJ,I) = -99999.
10      CONTINUE
20    CONTINUE

*-----create array containing the multiplication factor
*     used to change the uniform random deviate on the
*     interval 0-1 into a value on the range of the parameter

      DO 30 I = 1, NOP
        MULTR (I) = UB (I) - LB (I)
30    CONTINUE

*-----Create N parameter sets
```

```
*    Note: remember that NPS = number of parameter sets

      DO 80 J = 1, NPS

*-----Create each parameter set
*    Note: remember that NOP = number of parameters in model (SBFLEVO)

39          DO 40 I = 1, NOP
             P (I,J) = UNIFL() * MULTR (I) + LB (I)
*-----the array PSAP() is used to pass the individual parameter
*    sets into the plant growth model (subroutine CROP; SBFLEVO)

             PASP (I) = P (I,J)
40          CONTINUE

*-----Calculate the criterion value C for each parameter set

        CALL CTRION (PASP,CT,NDP,NDS,NRP,FIX1,FIX2,FIX3,
     $              WFIX1,WFIX2,WFIX3,IDSNK)

        C (J) = CT

*------Write the random chosen initial parameter values with the
*     criterion values to file

      WRITE (10,45) (PASP (I),I=1,NOP),C(J)
45    FORMAT (F10.5,7(1X,F10.5))

55    CONTINUE

80    CONTINUE

      RETURN
      END
```

```
*-------------------------------------------------------------------*
*  SUBROUTINE RNSRCH                                                *
*                                                                   *
*  Authors: Doug Rouse & Willem Stol                                *
*  Date   : 01-FEB-1989                                             *
*  Adapted: B.A.M. Bouman; 1989, 1992                               *
*  Purpose: This subroutine is meant to randomly choose subsets of  *
*           the parameter sets generated previously by subroutine   *
*           PARGEN for the purpose of generating new parameter      *
*           sets that converge on optimal parameter values.         *
*                                                                   *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)  *
*  name     type meaning                            units  class    *
*  ----     ---- -------                            -----  -----    *
*  NPS      I4   Number of Parameter Sets             -     I,O     *
*  NOP      I4   Number of Parameters                 -     I,O     *
*  NDP      I4   Number of Data Points                -     I,O     *
*  NDS      I4   Number of Data Sets                  -     I,O     *
*  NIT      I4   Number of ITerations                 -     I,O     *
*  P        R4   Parameter space                      -       O     *
*  C        R4   Criterion value for each parameter set -     O     *
*  LB       R4   Lower Bound of parameter(s)       variable   I     *
*  UB       R4   Upper Bound of parameter(s)       variable   I     *
*  PASP     R4   Parameter set array               variable   O     *
*  FIX1     R4   Calibration data 1                variable   I     *
*  FIX2     R4   Calibration data 2                variable   I     *
*  FIX3     R4   Calibration data 3                variable   I     *
*  WFIX1    R4   Summation of calibration data 1   variable   O     *
*  WFIX2    R4   Summation of calibration data 2   variable   O     *
*  WFIX3    R4   Summation of calibration data 3   variable   O     *
*  NRP      I4   Number of RePlicates                 -       I     *
*  IDSNK    I4   Data Set Number                      -       I     *
*                                                                   *
*  FATAL ERROR CHECKS (execution terminated, message): none         *
*                                                                   *
*  WARNINGS:        value of IWAR returned                          *
*    -                                                              *
*                                                                   *
*  SUBROUTINES and FUNCTIONS called : none                          *
*                                                                   *
*      - CRTION  - calculate Criterion value C for PASP             *
*      - SEARCH  - search all the parameter sets for those with C   *
*                  greater than CSTAR                               *
*                                                                   *
*  FILE usage : none                                                *
*-------------------------------------------------------------------*

      SUBROUTINE RNSRCH (NPS,NOP,NDP,NDS,NIT,P,C,LB,UB,PASP,
```

```
$            FIX1,FIX2,FIX3,WFIX1,WFIX2,WFIX3,
$            NRP,IDSNK)

      REAL C(500), CT, FIX1(5,100), FIX2(5,100), FIX3(5,100)
      REAL G(50), LB(50), P(50,500), PASP(50), PSUM, UB(50)
      REAL UNIFL, WFIX1, WFIX2, WFIX3

      INTEGER BOPT, I, IDSNK(5), IFLAG, II
      INTEGER IPOINT, J, K, L, NDP(5), NDS, NIT, NOP, NPS
      INTEGER NRP(5), PS(51), PST, TB

      LOGICAL MORE,LESS

*-----end of declarations

*-----variables retain their values between
*     subsequent calls of this subroutine

      SAVE

*-----Remember that NIT = number of iterations of algorithm
      DO 90 L = 1, NIT

*-----Selection of NOP+1 parameter sets at random
*     Note: remember that in the array P (I,J), J indexes
*     the parameter sets

 10   PS (1) = NINT (UNIFL () * (NPS -1) + 1.)
      I = 1
 20   IF (I.LE.(NOP+1)) THEN
        I = I + 1
 30     IFLAG = 0
        PST = NINT (UNIFL() * (NPS - 1) + 1.)

*-----Check to make sure the parameter set just choosen
*     is not the same as any of the previously choosen
*     parameter sets

      DO 40 J = 1, I-1
        IF (PST.EQ.PS(J)) IFLAG = 1
 40   CONTINUE

      IF (IFLAG.EQ.1) GO TO 30
        PS(I) = PST
        GO TO 20
      END IF

*-----Calculate new parameter set
```

```
      MORE  = .FALSE.
      LESS  = .FALSE.
      BOPT  = 1

*-----First calculate array G () containing the average
*     values of each parameter for the NOP randomly
*     choosen parameter sets as follows

      DO 60 I = 1,NOP
        PSUM = 0.0
        DO 50 K = 1, NOP
          J = PS (K)
          PSUM = PSUM + P(I,J)
 50     CONTINUE
        G (I) = PSUM/NOP
 60   CONTINUE

*-----Use formula: PASP (I) = 2.0 * G (I) - P(I,PS(NOP+1)),
*     where P is a parameter set choosen from the NPS - NOP
*     remaining unchoosen sets; that is P is the NOP+1
*     parameter set selected at random in array PS

      I = 1
 70   IF (I.LE.NOP) THEN

        PASP (I) = 2.0 * G (I) - P (I,PS (NOP+1))

*-----Check to see if the calculated parameter value falls
*     outside the bounds of possible values, if so set BOUND
*     to false

        MORE = PASP (I).GT.UB (I)
        LESS = PASP (I).LT.LB (I)

        IF (MORE) THEN
          PASP (I) = UB (I)
        ELSE IF (LESS) THEN
          PASP (I) = LB (I)
        END IF

*-----Check to see if parameter sets with parameters out of bounds are
*     going to be discarded (BOPT=1) or set to the bound (BOPT=0)

        IF (BOPT.EQ.1) THEN
          IF (MORE.OR.LESS) THEN
            GO TO 10
          END IF
```

```
        END IF

        I = I + 1
        GO TO 70
      END IF

*-----Calculate C for PASP

      CALL CTRION (PASP,CT,NDP,NDS,NRP,FIX1,FIX2,FIX3,
     $           WFIX1,WFIX2,WFIX3,IDSNK)

*-----Search all the parameter sets for those with C greater than
*     CSTAR; discard the parameter set with the largest C from among
*     those that have a C value greater then CSTAR

      CALL SEARCH (C,IPOINT,NPS)

*-----Replace the discarded parameter set with parameter set PR( )

      IF (CT.LE.C(IPOINT)) THEN
        DO 80 II = 1, NOP
          C(IPOINT) = CT
          P(II,IPOINT) = PASP(II)
80      CONTINUE
      END IF

90    CONTINUE

      RETURN
      END
```

```
*------------------------------------------------------------------*
* SUBROUTINE SEARCH                                                *
*                                                                  *
* Authors: Doug Rouse & Willem Stol                                *
* Date    : 01-FEB-1989                                            *
* Purpose: This subroutine searches for the largest value of C().  *
*                                                                  *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)  *
* name    type meaning                              units  class   *
* ----    ---- -------                              -----  -----   *
* C        R4  Criterion values for each parameter set  -     I    *
* IPOINT   I4  Array index pointer                       -     O    *
* NPS      I4  Number of parameter sets                  -     I,O  *
*                                                                  *
* FATAL ERROR CHECKS (execution terminated, message): none         *
*                                                                  *
* WARNINGS: none                                                   *
*                                                                  *
* SUBROUTINES and FUNCTIONS called : none                          *
*                                                                  *
* FILE usage : none                                                *
*------------------------------------------------------------------*

      SUBROUTINE SEARCH(C,IPOINT,NPS)

      REAL C(500), OLDC
      INTEGER I, IPOINT, NPS

*-----Variables retain their values between
*     subsequent calls of this subroutine

      SAVE

      OLDC = 0.0
      DO 10 I = 1, NPS
        IF (C(I).GT.OLDC) THEN
          IPOINT = I
          OLDC = C(I)
        END IF
10    CONTINUE

      RETURN
      END
```

```
*----------------------------------------------------------------*
* SUBROUTINE CTRION                                               *
*                                                                 *
* Authors: Doug Rouse & Willem Stol                               *
* Date   : 01-FEB-1989                                            *
* Adapted: B.A.M. Bouman; 1989, 1992                              *
* Purpose: This subroutine calculates the criterion value C for   *
*          judging the closeness of correspondence between the    *
*          simulated calibration variables (model output) and the *
*          calibration data (measurements of radar backscatter,   *
*          optical reflectance or crop parameters) for a given    *
*         parameter set.                                          *
*                                                                 *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time) *
* name     type meaning                           units  class    *
* ----     ---- -------                           -----  -----    *
* PASP     R4   Parameter set array            variable    I      *
* CT       R4   Criterion value                    -       O      *
* NDP      I4   Number of Data Points              -       I      *
* NDS      I4   Number of Data Sets                -       I      *
* NRP      I4   Number of RePlicates               -       I      *
* FIX1     R4   Calibration data 1             variable    I      *
* FIX2     R4   Calibration data 2             variable    I      *
* FIX3     R4   Calibration data 3             variable    I      *
* WFIX1    R4   Summation of calibration data 1 variable   O      *
* WFIX2    R4   Summation of calibration data 2 variable   O      *
* WFIX3    R4   Summation of calibration data 3 variable   O      *
* IDSNK    I4   Data Set Number                    -       I      *
*                                                                 *
* FATAL ERROR CHECKS (execution terminated, message): none        *
*                                                                 *
* WARNINGS: none                                                  *
*                                                                 *
* SUBROUTINES and FUNCTIONS called :                              *
*                                                                 *
*    CROP: crop growth model SBFLEVO for sugar beet in Flevoland  *
*                                                                 *
* FILE usage : none                                               *
*----------------------------------------------------------------*
      SUBROUTINE CTRION (PASP,CT,NDP,NDS,NRP,FIX1,FIX2,FIX3,
     $           WFIX1,WFIX2,WFIX3,IDSNK)

      REAL CPFIX1, CPFIX2, CPFIX3, CT, FIX1(5,100), FIX2(5,100)
      REAL FIX3(5,100), MPAR1, MPAR2, MPAR3, PASP(50), PASP2
      REAL RFIX1, RFIX2, RFIX3, SRFIX1, SRFIX2, SRFIX3, WFIX1
      REAL WFIX2, WFIX3

      INTEGER I, IDSN, IDSNK(5), II, KCI, KCII, NDP(5), NDS
```

```
      INTEGER NRP(5), TNDP

*-----The common block 'A' is common with subroutine CROP

      COMMON /A/ MPAR1(5,50),MPAR2(5,50),MPAR3(5,50),PASP2(50),IDSN

*-----Variables retain their values between
*     subsequent calls of this subroutine

      SAVE

      DO 10 I = 1, 20
        PASP2(I) = PASP(I)
10    CONTINUE

      IF (IDSNK(1).EQ.1) THEN
         IDSN = 1
         CALL CROP
      END IF

      IF (IDSNK(2).EQ.1) THEN
         IDSN = 2
         CALL CROP
      END IF

      IF (IDSNK(3).EQ.1) THEN
         IDSN = 3
         CALL CROP
      END IF

      IF (IDSNK(4).EQ.1) THEN
         IDSN = 4
         CALL CROP
      END IF

      IF (IDSNK(5).EQ.1) THEN
         IDSN = 5
         CALL CROP
      END IF

*-----The residuals will be calculated for each simulated calibration
*     variable (simulated FIX1 to FIX3). Remember that NDP = number
*     of data points in field data set

      SRFIX1 = 0.0
      SRFIX2 = 0.0
      SRFIX3 = 0.0
      TNDP  = 0
```

```
      DO 20 II = 1, NDS
        KCI  = 1
        KCII = 1
        IF (IDSNK(II).EQ.0) GO TO 20

        DO 40 I = 1, NDP(II)*NRP(II)
          IF (KCI.GT.NRP(II)) THEN
            KCI = 1
            KCII = KCII + 1
          END IF
          KCI = KCI + 1

*-----     Sum of residuals for simulated FIX1
          RFIX1  = ABS(MPAR1(II,KCII) - FIX1(II,I))
          SRFIX1 = SRFIX1 + RFIX1

*-----     Sum of residuals for simulated FIX2
          RFIX2  = ABS(MPAR2(II,KCII) - FIX2(II,I))
          SRFIX2 = SRFIX2 + RFIX2

*-----     Sum of residuals for simulated FIX3
          RFIX3  = ABS(MPAR3(II,KCII) - FIX3(II,I))
          SRFIX3 = SRFIX3 + RFIX3

40      CONTINUE
        TNDP = TNDP + NDP(II)*NRP(II)
20    CONTINUE

*-----Calculate the average residual for each simulated calibration variable

      CPFIX1 = SRFIX1 / TNDP
      CPFIX2 = SRFIX2 / TNDP
      CPFIX3 = SRFIX3 / TNDP

*-----Apply the appropriate weighting factor to each average residual

      CPFIX1 = CPFIX1 / WFIX1
      CPFIX2 = CPFIX2 / WFIX2
      CPFIX3 = CPFIX3 / WFIX3

*-----Find the maximum of the weighted average residuals

      CT = AMAX1 (CPFIX1, CPFIX2, CPFIX3)

      RETURN
      END
```

```
*-----------------------------------------------------------------*
* SUBROUTINE INPUT                                                *
*                                                                 *
* Authors: Doug Rouse & Willem Stol                               *
* Date    : 01-FEB-1989                                           *
* Adapted: B.A.M. Bouman; 1989, 1992                              *
* Purpose: This subroutine contains all the input values necessary*
*    to execute the calibration program (mainly the calibration   *
*    data: measurements of radar backscatter, optical reflectance *
*    or crop parameters.                                          *
*                                                                 *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time) *
* name    type meaning                             units  class  *
* ----    ---- -------                             -----  -----  *
* NPS      I4  Number of Parameter Sets              -      O    *
* NOP      I4  Number Of Parameters                  -      O    *
* NDP      I4  Number of Data Points                 -      O    *
* NDS      I4  Number of Data Sets                   -      O    *
* NIT      I4  Number of Iterations                              *
*              of calibration algorithm              -      O    *
* NRP      I4  Number of RePlicates                  -      O    *
* LB       R4  Lower Bound of parameter(s)       variable   O    *
* UB       R4  Upper Bound of parameter(s)       variable   O    *
* FIX1     R4  Calibration data 1                variable   I    *
* FIX2     R4  Calibration data 2                variable   I    *
* FIX3     R4  Calibration data 3                variable   I    *
* WFIX1    R4  Summation of calibration data 1   variable   O    *
* WFIX2    R4  Summation of calibration data 2   variable   O    *
* WFIX3    R4  Summation of calibration data 3   variable   O    *
* IDSNK    I4  Index of Data Set Number              -      O    *
*                                                                 *
* FATAL ERROR CHECKS (execution terminated, message): none        *
*                                                                 *
* WARNING:                                                        *
*                                                                 *
*     -  on errors during reading inputfile                       *
*                                                                 *
* SUBROUTINES and FUNCTIONS called :                              *
*                                                                 *
*     -  MOFILP   - to skip comment lines in the input file       *
*                                                                 *
* FILE usage :                                                    *
*                                                                 *
*     SBFLEVO_OPT.DAT - (UNIT:1)                                  *
*-----------------------------------------------------------------*

      SUBROUTINE INPUT (NPS,NOP,NDP,NDS,NIT,NRP,LB,UB,FIX1,FIX2,FIX3,
     $           WFIX1,WFIX2,WFIX3,IDSNK,JDAY1)
```

```
      REAL FIX1(5,100), FIX2(5,100), FIX3(5,100), LB(50), TWFIX1
      REAL TWFIX2, TWFIX3, UB(50), WFIX1, WFIX2, WFIX3

      INTEGER DUM, I, IDSNK(5)
      INTEGER J, JDAY(5,100), JDAY1(5,100), JK, NDP(5), NRP(5)
      INTEGER NPS, NDS, NIT, NOP

*-----The common block 'C' is common with subroutine CROP

      COMMON/C/ DUM(5), JDAY

*-----Variables retain their values between
*     subsequent calls of this subroutine

      SAVE

*-----Open the input data file

      OPEN (UNIT=1,FILE='SBFLEVO_OPT.DAT',STATUS='OLD')

*-----Initialize array nrep

      DO 10 I = 1, 2
        NRP (I) = 0
10    CONTINUE

*-----Read NPS, NIT, and lower and upper bounds for parameters
*     that are to be calibrated

      CALL MOFILP (1)
      READ (1,*,ERR=70,END=70) NPS
      READ (1,*,ERR=70,END=70) NIT
      CALL MOFILP (1)
      READ (1,*,ERR=70,END=70) NOP
      CALL MOFILP (1)

      DO 20 I = 1, NOP
        READ (1,'(6X,F11.5,F11.5)',ERR=70,END=70) LB (I), UB (I)
20    CONTINUE

      CALL MOFILP (1)
      READ (1,*,ERR=70,END=70) NDS
      CALL MOFILP (1)
      READ (1,*,ERR=70,END=70) (IDSNK(I),I=1,NDS)

      TWFIX1 = 0.0
      TWFIX2 = 0.0
```

```
      TWFIX3 = 0.0

      DO 40 I = 1, NDS
        CALL MOFILP (1)
        READ (1,*,ERR=70,END=70) NDP(I)
        CALL MOFILP (1)
        READ (1,*,ERR=70,END=70) NRP(I)
        CALL MOFILP (1)

        DO 30 J = 1, NDP(I)*NRP(I)
          READ (1,*,ERR=70,END=70)
     $     JDAY1(I,J),FIX1(I,J),FIX2(I,J),FIX3(I,J)
          TWFIX1 = TWFIX1 + FIX1 (I,J)
          TWFIX2 = TWFIX2 + FIX2 (I,J)
          TWFIX3 = TWFIX3 + FIX3 (I,J)
30      CONTINUE
40    CONTINUE

      WFIX1 = TWFIX1
      WFIX2 = TWFIX2
      WFIX3 = TWFIX3

      DO 60 I = 1, NDS
        DUM (I) = NDP (I)
        DO 50 J = 1, NDP (I)
          JK = (J-1) * NRP (I) + 1
          JDAY (I,J) = JDAY1 (I,JK)
50      CONTINUE
60    CONTINUE

      GO TO 80

70      CONTINUE

*----- Error message due to reading error
       WRITE (*,*) ' ERROR DURING READING INPUT FILE',
     $           ' IN SUBROUTINE: INPUT'

80    CONTINUE
      CLOSE (1)

      RETURN
      END
```

```
*******************************************************************
* SUBROUTINE MOFILP (IUNIT)                                       *
* Author: Daniel van Kraalingen                                   *
* Date: Aug 87                                                    *
*                                                                 *
* This subroutine moves the filepointer across comment lines of   *
* datafiles and puts the file pointer at the first non comment    *
* record.                                                         *
* Comment lines have an asterisk (*) in their first column        *
*                                                                 *
* IUNIT, unit number of file, (I)                                 *
*******************************************************************

      SUBROUTINE MOFILP (IUNIT)

      CHARACTER LINE*80,CHR*1

      CHR = '*'
10    IF (CHR.NE.'*') GOTO 20
        READ (IUNIT,'(A)') LINE
        CHR = LINE(1:1)
      GOTO 10

20    CONTINUE
      BACKSPACE (IUNIT)

      RETURN
      END
```

```
*----------------------------------------------------------------------*
* SUBROUTINE OUTPUT                                                    *
*                                                                      *
* Authors: Doug Rouse & Willem Stol                                    *
* Date   : 01-FEB-1989                                                 *
* Adapted: B.A.M. Bouman; 1989, 1992                                   *
* Purpose: This subroutine outputs the final array values              *
*     (of simulated calibration variables FIX1-FIX3)                   *
*     AVERUN: model SBFLEVO is run with average parameter values       *
*     BESTR : model SBFLEVO is run with 'best' parameter values        *
*     WORSTR: model SBFLEVO is run with 'worst' parameter values       *
*     NOMINL: model SBFLEVO is run with nominal parameter values       *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name    type meaning                                 units  class    *
* ----    ---- -------                                 -----  -----    *
* NPS      I4  Number of Parameter Sets                  -     I,O     *
* NOP      I4  Number of Parameters                      -     I,O     *
* NDS      I4  Number of Data Sets                       -     I,O     *
* NIT      I4  Number of ITerations                      -     I,O     *
* P        R4  Parameter space                           -      O      *
* C        R4  Criterion value for each parameter set    -      O      *
* IDSNK    I4  Data Set Number                           -      I      *
*                                                                      *
* FATAL ERROR CHECKS (execution terminated, message): none             *
*                                                                      *
* WARNINGS: none                                                       *
*                                                                      *
* SUBROUTINES and FUNCTIONS called :                                   *
*                                                                      *
*     - SORT - sorts the criterion values                              *
*     - CROP - calculates the model outcome for dataset                *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*

      SUBROUTINE OUTPUT (NPS,NOP,NDS,P,C,IDSNK,LB,UB)

      REAL APAR1T, APAR2T, APAR3T, C(500), LRGF1(5,365), LRGF2(5,365)
      REAL LRGF3(5,365), P(50,500), PASP3, SMLF1(5,365), SMLF2(5,365)
      REAL SMLF3(5,365), SUM(50), LB(50), UB(50)

      INTEGER I, IDSN2, IDSNK(5), IH, II, IJ, IJK, ISORTP(500)
      INTEGER J, K, NDS, NOP, NPS

      LOGICAL AVERUN,BESTR,WORSTR,NOMINL

      COMMON/FINRUN/PASP3(50),AVERUN,BESTR,WORSTR,NOMINL,IDSN2
```

```
*-----The common block CONFIN is common with subroutine CROP

      COMMON/CONFIN/APAR1T(5,365),APAR2T(5,365),APAR3T(5,365)

*-----Variables retain their values between
*     subsequent calls of this subroutine

      SAVE

      AVERUN = .FALSE.
      BESTR  = .FALSE.
      WORSTR = .FALSE.
      NOMINL = .FALSE.

*-----I. Run the subroutine CROP (model SBFLEVO) to get simulated calibration
*     variables (SBFLEVO model output) for the average values from the
*     final parameter sets

      DO 10 I = 1, NOP
        SUM (I) = 0.
10    CONTINUE

      DO 30 J = 1, NPS
        DO 20 K = 1,NOP
        SUM (K) = SUM (K) + P (K,J)
20      CONTINUE
30    CONTINUE

      DO 40 I = 1, NOP
        PASP3 (I) = SUM (I) / NPS
40    CONTINUE

      AVERUN = .TRUE.

      IF (IDSNK(1).EQ.1) THEN
         IDSN2=1
         CALL CROP
      END IF

      IF (IDSNK(2).EQ.1) THEN
         IDSN2 = 2
         CALL CROP
      END IF

      IF (IDSNK(3).EQ.1) THEN
         IDSN2 = 3
         CALL CROP
```

```
      END IF

      IF (IDSNK(4).EQ.1) THEN
         IDSN2 = 4
         CALL CROP
      END IF

      IF (IDSNK(5).EQ.1) THEN
         IDSN2 = 5
         CALL CROP
      END IF

      AVERUN = .FALSE.

*-----Sort criterion values so the lowest and highest can be found

      CALL SORT (NPS,C,ISORTP)

*-----Write finally obtained, sorted parameter and Criterion
*     values to file

      DO 333, J=1,NPS
       WRITE (11,444) (P(I,ISORTP(J)),I=1,NOP),C(ISORTP(J))
444    FORMAT(F10.5,7(1X,F10.5))
333   CONTINUE

*-----II. Run the subroutine CROP (model SBFLEVO) to get simulated calibration
*     variables for the parameter set with the lowest criterion value

      DO 50 I = 1, NOP
        PASP3 (I) = P (I,ISORTP(1))
 50   CONTINUE

*-----Write the parameters with the lowest criterion value to file

      WRITE(20,*) ' DAYSOW', PASP3(1)
      WRITE(20,*) ' RGRL', PASP3(2)
      WRITE(20,*) ' EFF', PASP3(3)
      WRITE(20,*) ' LSHAD', PASP3(4)

      BESTR = .TRUE.

      IF (IDSNK(1).EQ.1) THEN
         IDSN2=1
         CALL CROP
      END IF

      IF (IDSNK(2).EQ.1) THEN
```

```
        IDSN2 = 2
        CALL CROP
      END IF

      IF (IDSNK(3).EQ.1) THEN
        IDSN2 = 3
        CALL CROP
      END IF

      IF (IDSNK(4).EQ.1) THEN
        IDSN2 = 4
        CALL CROP
      END IF

      IF (IDSNK(5).EQ.1) THEN
        IDSN2 = 5
        CALL CROP
      END IF

      BESTR = .FALSE.

*-----III. Run the subroutine CROP (model SBFLEVO) to get simulated calibration
*     variables for the parameter set with the worst criterion value

      DO 60 I = 1,NOP
        PASP3 (I) = P (I,ISORTP(NPS))
60    CONTINUE

      WORSTR = .TRUE.

      IF (IDSNK(1).EQ.1) THEN
        IDSN2=1
        CALL CROP
      ENDIF

      IF (IDSNK(2).EQ.1) THEN
        IDSN2 = 2
        CALL CROP
      ENDIF

      IF (IDSNK(3).EQ.1) THEN
        IDSN2 = 3
        CALL CROP
      ENDIF

      IF (IDSNK(4).EQ.1) THEN
        IDSN2 = 4
        CALL CROP
```

```
      ENDIF

      IF (IDSNK(5).EQ.1) THEN
        IDSN2 = 5
        CALL CROP
      END IF

      WORSTR = .FALSE.

*-----IV. Run the subroutine CROP (model SBFLEVO) to get simulated calibration
*     variables for the parameter set with the nominal values of the parameters

      NOMINL    = .TRUE.

      DO 65, I = 1,NOP
       PASP3 (I)  = (LB(I)+UB(I)) / 2.
65    CONTINUE

      IF (IDSNK(1).EQ.1) THEN
        IDSN2=1
        CALL CROP
      ENDIF

      IF (IDSNK(2).EQ.1) THEN
        IDSN2 = 2
        CALL CROP
      ENDIF

      IF  (IDSNK(3).EQ.1) THEN
        IDSN2 = 3
        CALL CROP
      ENDIF

      IF (IDSNK(4).EQ.1) THEN
        IDSN2 = 4
        CALL CROP
      ENDIF

      IF (IDSNK(5).EQ.1) THEN
        IDSN2 = 5
        CALL CROP
      END IF

      NOMINL = .FALSE.

       WRITE (*,*) ' END OF PROGRAM IN SUBROUTINE OUTPUT'

       RETURN
```

```
*----------------------------------------------------------------------*
* SUBROUTINE SORT                                                      *
*                                                                      *
* Authors: Doug Rouse & Willem Stol                                    *
* Date   : 01-FEB-1989                                                 *
* Purpose: This subroutine sorts an array.                             *
*                                                                      *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                  units  class    *
* ----   ---- -------                                  -----  -----    *
* NPS     I4  Number of Parameter Sets                   -      I      *
* P       R4  Parameter space                            -      O      *
* C       R4  Criterion value for each parameter set     -      O      *
* ISORTP  I4  Sorted parameter sets                variable     O      *
*                                                                      *
* FATAL ERROR CHECKS (execution terminated, message): none             *
*                                                                      *
* WARNINGS: none                                                       *
*                                                                      *
* SUBROUTINES and FUNCTIONS called : none                              *
*                                                                      *
* FILE usage : none                                                    *
*----------------------------------------------------------------------*
   SUBROUTINE SORT (N,ARRIN,INDX)
   DIMENSION ARRIN(N), INDX (N)

   DO 11 J=1,N
    INDX(J)=J
11  CONTINUE

   L=N/2+1
   IR=N
10  CONTINUE

    IF(L.GT.1)THEN
     L=L-1
     INDXT=INDX(L)
     Q=ARRIN(INDXT)
    ELSE
     INDXT=INDX(IR)
     Q=ARRIN(INDXT)
     INDX(IR)=INDX(1)
     IR=IR-1
     IF(IR.EQ.1)THEN
      INDX(1)=INDXT
      RETURN
     ENDIF
    ENDIF
```

```
      I=L
      J=L+L

20     IF(J.LE.IR)THEN
       IF(J.LT.IR)THEN
        IF(ARRIN(INDX(J)).LT.ARRIN(INDX(J+1)))J=J+1
       ENDIF
       IF(Q.LT.ARRIN(INDX(J)))THEN
        INDX(I)=INDX(J)
        I=J
        J=J+J
       ELSE
        J=IR+1
       ENDIF
      GO TO 20
      ENDIF

      INDX(I)=INDXT
     GO TO 10
     END
```

```
*-----------------------------------------------------------------------*
* REAL FUNCTION UNIFL                                                   *
*                                                                       *
* Authors: Kees Rappoldt                                                *
* Date   : 14 October 1987                                              *
* Purpose: This function is a high quality pseudo random number        *
*          generator.                                                   *
*                                                                       *
*    This generator is of the so called combined type. This             *
*    generator does not behave pathologically with the Box-Muller       *
*    method for the generation of normal variates, as do the            *
*    commonly used linear congruential generators (see also             *
*    comments in FUNCTION BOXMUL).                                      *
*                                                                       *
*    The algorithm is:  X(i+1) = 40014 * X(i) mod (2147483563)          *
*                       Y(i+1) = 40692 * Y(i) mod (2147483399)          *
*                       Z(i+1) = (X(i+1)+Y(i+1)) mod (2147483563)       *
*    The random number returned is constructed dividing Z by its       *
*    range. The period of the generator is about 2.30584E+18.           *
*                                                                       *
*    The program below has been written using a program UNIFL in       *
*    Bratley et.al. (1987). A logical INIT has been added in           *
*    order to include seeds in the program. The algorithm              *
*    originates from L'Ecuyer (1986). In Bratley et.al. (page 332)     *
*    more information can be found on seeds and periods of X and       *
*                                                                       *
*    References:                                                        *
*                                                                       *
*    Bratley,P., B.L.Fox, L.E.Schrage. 1983. A guide to simulation     *
*        Springer-Verlag New York Inc. 397 pp.                          *
*    L'Ecuyer,P. (1986). Efficient and portable combined pseudo-       *
*        random number generators. Commun. ACM (to appear).            *
*                                                                       *
* FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
* name   type meaning                                    units  Class  *
* ----   ---- -------                                    -----  -----  *
* UNIFL   R4  Value of returning random number             -      O    *
*                                                                       *
* FATAL ERROR CHECKS (execution terminated, message): None             *
*                                                                       *
* WARNINGS: none                                                        *
*                                                                       *
* SUBROUTINES and FUNCTIONS called : none                               *
*                                                                       *
* FILE usage : none                                                     *
*-----------------------------------------------------------------------*

      REAL FUNCTION UNIFL()
```

```
      DIMENSION JX(3)

      LOGICAL INIT

      DATA INIT/.FALSE./

*-----end of declarations

      IF (.NOT.INIT) THEN
*-----   initialize generator ; see for seeds Bratley (1983)
        JX(2) = 1122334455
        JX(3) = 1408222472
        INIT = .TRUE.
      END IF

*-----get next term in first stream = 40014 * JX(2) mod 2147483563

      K = JX(2) / 53668
      JX(2) = 40014 * (JX(2) - K * 53668)  -  K * 12211
      IF (JX(2).LT.0)  JX(2) = JX(2) + 2147483563

*-----get next term in the second stream = 40692 * JX(3) mod 2147483399

      K = JX(3) / 52774
      JX(3) = 40692 * (JX(3) - K * 52774)  -  K * 3791
      IF (JX(3).LT.0)  JX(3) = JX(3) + 2147483399

*-----set JX(1) = ((JX(3) + 2147483562 - JX(2)) mod 2147483562) + 1

      K = JX(3) - JX(2)
      IF (K.LE.0)  K = K + 2147483562
      JX(1) = K

*-----put it on the interval (0,1)

      UNIFL = K * 4.656613E-10

      RETURN
      END
```

```
*----------------------------------------------------------------
* SUBROUTINE CROP                   (Bouman, 1992)              *
*                                                               *
* Author: B.A.M. Bouman       July 1992                         *
* This subroutine is the crop growth model SBFLEVO that         *
* simulates crop growth, optical reflectance and L-, S-, C-,    *
* X-, Ku1- and Ku2-band radar backscatter of sugar beet,        *
* calibrated for Flevoland in The Netherlands                   *
* (Bouman, 1992; CABO-DLO report 163)                           *
*                                                               *
* The crop growth module of SBFLEVO is SUCROS87 (Simple and     *
* Universal CROp growth Simulator) for sugar beet (Spitters     *
* et al., 1987, 1989) as changed by Goudriaan (1988, 1989)      *
* and calibrated for Flevoland by Bouman (1992).                *
*                                                               *
* The calculations of optical reflectance are based on the      *
* model EXTRAD (Goudriaan, 1977), the calculations of the radar *
* backscatter on the 'Cloud' model (Attema & Ulaby, 1978).      *
* The remote sensing models were calibrated on data from        *
* Flevoland (Bouman 1992)                                       *
*                                                               *
* The mentioned references are given in the documentation on    *
* SBFLEVO:                                                      *
*  Bouman, B.A.M., 1992, SBFLEVO and WWFLEVO; Growth models to  *
*  simulate crop growth, optical reflectance and radar          *
*  backscatter of sugar beet and winter wheat, calibrated for   *
*  Flevoland, CABO-DLO report 164, CABO-DLO, Wageningen,        *
*  The Netherlands.                                             *
*                                                               *
* SUBROUTINES AND FUNCTIONS CALLED:                             *
*   - CLOUDX, CLOUDM, REFLEX, PRINT, ASTRO, TOTAS, ASSIM,       *
*     GLA, LINT                                                 *
*                                                               *
* FILE usage:                                                   *
*   - SBFLEVO.IN; data file                                     *
*   - WEER87,.... WEER91: weather data files                    *
*---------------------------------------------------------------*

      SUBROUTINE CROP

      IMPLICIT REAL (A-H,J-Z)
      IMPLICIT INTEGER (I)

      REAL AMDVST(50), AMTMPT(50), BDMPT(50)
      REAL DVRVT(50), DVRRT(50), FRTTB(50)
      REAL FSHTB(50), FLVTB(50), FSTTB(50), FCRTB(50)
      REAL RDRT(50), MST(100)
      REAL DTRT(365),TMAXT(365),TMINT(365)
```

```
      REAL NC(100),CPL(10),GS(10),ATT(10),SOIL(10),PLANT(10)
      REAL GAMMA(10)
      REAL CPLA(5,6),GSA(5,6),ATTA(5,6),SOILA(5,6),PLANTA(5,6)
      REAL GAMMAA(5,6)
      REAL NCA(5), KSA(6),DPLA(6)
      REAL F(9),BU(9)

      CHARACTER*9 FLEVOYEAR

      PARAMETER (PI=3.1415926)

*-----extra declarations in simulation model to combine
*     with calibration procedure

      REAL MPAR1,MPAR2,MPAR3,PASP2,PASP3
      REAL PAR1M,PAR2M,PAR3M,APAR1T,APAR2T,APAR3T

      INTEGER NDP,JDAY,K,IK,K2,IFD
      LOGICAL AVERUN,BESTR,WORSTR,NOMINL

*-----common block T is common with subroutine PARGEN for the purpose
*     of passing back the final value of tuber weight for printing
      COMMON/T/PAR1M(5,25),PAR2M(5,25),PAR3M(5,25)

*-----the COMMON BLOCK A is common with subroutine CTRION.  It passes
*     the model output data used in calculating residuals to CTRION.
      COMMON/A/ MPAR1(5,50),MPAR2(5,50),MPAR3(5,50),PASP2(50),IDSN

*-----the COMMON BLOCK C is common with subroutine INPUT.  The array
*     JDAY contains the julian days for which there are field data
*     and therefore determines the days for which model output is
*     to be saved and passed to CTRION
      COMMON/C/ NDP(5), JDAY(5,100)

*-----the COMMON BLOCK FINRUN is common with subroutine OUTPUT
      COMMON/FINRUN/PASP3(50),AVERUN,BESTR,WORSTR,NOMINL,IDSN2

*-----the common block CONFIN is common with subroutines PARGEN and
*     OUTPUT
      COMMON/CONFIN/ APAR1T(5,365),APAR2T(5,365),APAR3T(5,365)

*-----check to see if this is the final run for output, if it is then
*     set the parameter values to the average values from the final
*     parameter set array as calculated and passed from OUTPUT
      IF (AVERUN.OR.BESTR.OR.WORSTR.OR.NOMINL) THEN
        IDSN = IDSN2
        DO 5 IK = 1,20
          PASP2(IK) = PASP3(IK)
```

```
5     CONTINUE
    END IF

    DATA BU/.03015,.08682,.13302,.16318,.17365,
   $      .16318,.13302,.08682,.03015/


*****************************************************************
*   INITIALISATION                                  *
*****************************************************************


*-----open datafile crop parameters
    OPEN(20,FILE='SBFLEVO.IN',STATUS='OLD')
    READ(20,*) LAT
    READ(20,*) NPL
    READ(20,*) LA0
    READ(20,*) TEMERG
    READ(20,*) RGRL
    READ(20,*) LSHAD
    READ(20,*) TBASE
    READ(20,*) SLA
    READ(20,*) AMX
    READ(20,*) EFF
    READ(20,*) KDIF
    READ(20,*) KCOVER
    READ(20,*) SCV
    READ(20,*) Q10
    READ(20,*) MAINSO
    READ(20,*) ASRQSO
    READ(20,*) RDSTLV
    READ(20,*) RDSTST
*-----soil moisture data
    READ(20,*) IMST
    READ(20,*) (MST(I),I=1,IMST)
*-----further crop data
    READ(20,*) IAMDVN
    READ(20,*) (AMDVST(I),I=1,IAMDVN)
    READ(20,*) IAMTMN
    READ(20,*) (AMTMPT(I),I=1,IAMTMN)
    READ(20,*) IFSHN
    READ(20,*) (FSHTB(I),I=1,IFSHN)
    READ(20,*) IFLVN
    READ(20,*) (FLVTB(I),I=1,IFLVN)
    READ(20,*) IFSTN
    READ(20,*) (FSTTB(I),I=1,IFSTN)
    READ(20,*) IFCRN
    READ(20,*) (FCRTB(I),I=1,IFCRN)
    READ(20,*) IFRTN
```

```
      READ(20,*) (FRTTB(I),I=1,IFRTN)
      READ(20,*) IRDRN
      READ(20,*) (RDRT(I),I=1,IRDRN)
      READ(20,*) IBDMPN
      READ(20,*) (BDMPT(I),I=1,IBDMPN)
*-----parameters for X-band cloud model (ROVE)
      READ(20,*) INC
      READ(20,*) (NC(I),I=1,INC)
      READ(20,*) IGS
      READ(20,*) (GS(I),I=1,IGS)
      READ(20,*) ICPL
      READ(20,*) (CPL(I),I=1,ICPL)
      READ(20,*) KS
      READ(20,*) DPL
      READ(20,*) MCCROP
*-----parameters for multi-freq. cloud model
*-----Agriscatt-88; 5 inci. angles; 6 frequencies
*-----first: read incidence angles
      READ(20,*) (NCA(I),I=1,5)
*-----bare soil G-values; frequencies in line, inci. angles by row
      READ(20,*) (GSA(1,J),J=1,6)
      READ(20,*) (GSA(2,J),J=1,6)
      READ(20,*) (GSA(3,J),J=1,6)
      READ(20,*) (GSA(4,J),J=1,6)
      READ(20,*) (GSA(5,J),J=1,6)
*-----Crop C-values; frequencies in line, inci. angles by row
      READ(20,*) (CPLA(1,J),J=1,6)
      READ(20,*) (CPLA(2,J),J=1,6)
      READ(20,*) (CPLA(3,J),J=1,6)
      READ(20,*) (CPLA(4,J),J=1,6)
      READ(20,*) (CPLA(5,J),J=1,6)
*-----K-values per frequency; D-values per frequency
      READ(20,*) (KSA(J),J=1,6)
      READ(20,*) (DPLA(J),J=1,6)
*-----parameters for EXTRAD reflection calculations
      READ(20,*)RHOSG
      READ(20,*)RHOSR
      READ(20,*)RHOSIR
      READ(20,*)SCATG
      READ(20,*)SCATR
      READ(20,*)SCATIR
      READ(20,*)BETA
      READ(20,*) (F(I), I=1,9)
*-----parameters for metamodel WDVI-LAI
      READ(20,*) WMAX
      READ(20,*) WEXT
*-----Initialization of crop status; weights in kg/ha
      READ(20,*) WLVG
```

```
      READ(20,*) WLVD
      READ(20,*) WSTG
      READ(20,*) WSTD
      READ(20,*) WSO
      READ(20,*) WRT
      READ(20,*) WCR
      READ(20,*) LAI
      READ(20,*) TSUM2
      READ(20,*) TSUMEM
      READ(20,*) EMERG
      READ(20,*) DAYSOW
      READ(20,*) TIME
      READ(20,*) FINTIM
      READ(20,*) PRDEL
*-----input weather file
      READ(20,*) FLEVOYEAR
      CLOSE (20)

*-----read weather data from file
      OPEN(50,FILE=FLEVOYEAR,STATUS='OLD')
      READ(50,*) IDAYNR
      DO 8 I=1,IDAYNR
       READ(50,*) DTRT(I),TMAXT(I),TMINT(I)
8      CONTINUE
      CLOSE(50)

*-----growth rates are set to 0
      GLV   =0.
      DLV   =0.
      FDLV  =0.
      WLVD  =0.
      GST   =0.
      DST   =0.
      FDST  =0
      GSO   =0.
      GCR   =0.
      GRT   =0.
      GLAI  =0.
      DTSUM2=0.
      DTSUMM=0.
      DEMERG=0.
      DELT = 1.
*-----data to use in print subroutine
      PRTIME=TIME

*-----in array PASP2(I) remember that I indexes the parameters in a set
*     let the model parameters being optimized or be set to the
*     current values of PASP(I) or if final run PASP3(I)
```

```
      DAYSOW = PASP2(1)
      RGRL   = PASP2(2)
      EFF    = PASP2(3)
      LSHAD  = PASP2(4)

*-----only now initialization of LAI
      LAI=NPL*LA0*1.E-4

*-----communication with optimize program
      K   = 1
      K2  = 1
      IFD = 1
      ISTART = INT(TIME)-1
      ICOUNT = 1


*************DYNAMIC PART***********************************

10    CONTINUE

*-----go to end of loop if simulation is complete
      IF(TIME .GT. FINTIM-0.5*DELT) GOTO 1000

      DAY = MOD(TIME,365.)
      IDAY=INT(DAY)

*-----daily temperature ( C): maximum, minimum, average, daytime and
*     effective
      DTR=DTRT(IDAY)
      DTMAX=TMAXT(IDAY)
      DTMIN=TMINT(IDAY)

      DTR=DTR*1.E3
      DAVTMP= 0.5 * (DTMAX+DTMIN)
      DDTMP = DTMAX - 0.25 * (DTMAX-DTMIN)
      DTEFF = MAX(0.,DAVTMP-TBASE)
      TEFF  = Q10**((DAVTMP-25.)/10.)

*-----moisture content of topsoil
      MS=LINT(MST,IMST,DAY)

*-----integration of rate variables (kg/ha/day)
      WLVG  = WLVG  + (GLV-DLV)*DELT
      WLVD  = WLVD  + FDLV*DELT
      WSTG  = WSTG  + (GST-DST)*DELT
      WSTD  = WSTD  + FDST*DELT
      WSO   = WSO   + GSO*DELT
      WCR   = WCR   + GCR*DELT
```

```
      WRT   = WRT   + GRT*DELT
      LAI   = LAI   + GLAI*DELT
      TSUM2 = TSUM2 + DTSUM2*DELT
      TSUMEM= TSUMEM+ DTSUMM*DELT
      EMERG = EMERG + DEMERG*DELT

*-----summation of some state variables (weights in kg/ha)
      WLV   = WLVG + WLVD
      WST   = WSTG + WSTD
      TADRW = WLV + WST
      TUBER = WSO+WCR+WRT
      SUGAR = 0.75*WSO

*-----Beet yield (ton fresh/ha) from dry matter yield and d.m. percentage
*-----WTUBER is total 'underground'+crown; WBEET is only storage organ,
*-----in this SUCROS version, the harvested product is taken to be WTUBER!!
      BDMP  = LINT(BDMPT,IBDMPN,TSUM2)
      WTUBER = TUBER/BDMP * 1.E-3
      WBEET = WSO / BDMP  * 1.E-3

*-----Calculation of soil cover
      COVER = 1-EXP(-KCOVER*LAI)

*-----calculation of radar backscatter and reflection for dates to optimize:
      IF(IDAY.EQ.JDAY(IDSN,ICOUNT)) THEN
       CALL CLOUDX(PI,MCCROP,DPL,KS,NC,GS,CPL,INC,MS,
     $         TADRW,PLWCROP,ATT,SOIL,PLANT,GAMMA)
       CALL CLOUDM(PI,MCCROP,DPLA,KSA,NCA,GSA,CPLA,MS,
     $         TADRW,PLWCROP,ATTA,SOILA,PLANTA,GAMMAA)
       CALL REFLEX(RHOSG,RHOSR,RHOSIR,SCATG,SCATR,SCATIR,BETA,
     $         F,BU,LAI,FRDIF,NADG,NADR,NADIR,NDVI,WDVI,RATIO)
       ICOUNT = ICOUNT + 1
      END IF

*-----calculation of WDVI via metamodel
*     is WDVIM (to distinguish from WDVI calculated above)
      WDVIM=WMAX*(1-EXP(-WEXT*LAI))


******CALCULATION OF RATE VARIABLES**********************************

*-----emergence process begins after sowing
      IF(DAY.LT.DAYSOW)GOTO 100
      DEMERG=MAX(0.,DAVTMP-3.)

*-----start simulation of growth after temp. sum TEMERG is reached
      IF(EMERG.LT.TEMERG)GO TO 100
```

```
*-----temperatures after emergence
      DTSUM2= MAX(0.,MIN(19.,DAVTMP-2.))
      DTSUMM= DTEFF

*-----relative death rates, due to temperature and shading
*-----(above LAI=LSHAD)
*     RDRV = LINT(RDRT,IRDRN,TSUM2)*DTSUM2
*-----specifically in Flevoland:
      RDRV = 0.
      RDRSH= 0.03*MAX(0.,(LAI-LSHAD)/LSHAD)
      RDR=   MAX(RDRV,RDRSH)

*-----leaf photosynthesis rate at light saturation (kg CO2/ha leaf/h)
      AMDVS = LINT(AMDVST,IAMDVN,TSUM2)
      AMTMP = LINT(AMTMPT,IAMTMN,DDTMP)
*     AMAX  = AMX * AMDVS * AMTMP
*-----specifically in Flevoland:
      AMAX = AMX * AMTMP

*-----subroutine ASTRO computes  day length and daily radiation
*     characteristics from Julian date, latitude and measured daily total
      CALL ASTRO(DAY,LAT,DTR,
     $ SC,SINLD,COSLD,DAYL,DSINB,DSINBE,ATMTR,FRDIF)

*-----subroutine TOTASS computes daily total gross assimilation (DTGA)
      CALL TOTASS(SC,DAYL,SINLD,COSLD,DSINBE,DTR,ATMTR,FRDIF,
     $  SCV,AMAX,EFF,KDIF,LAI,DTGA)

*-----conversion from assimilated CO2 to CH2O
      GPHOT = DTGA * 30./44.

*-----maintenance respiration (kg CH2O/ha/d)
      IF(WLV.GT.0) THEN
         MNDVS = WLVG / WLV
      ELSE
         MNDVS=1.
      ENDIF
      MAINTS = 0.03*WLV + 0.015*WST + 0.015*WRT + MAINSO*WSO
      MAINT  = MIN(GPHOT, MAINTS * TEFF * MNDVS)

      DLV = WLVG * RDR
      DST = WSTG * RDR

* Rate of fall of leaf blades (FDLV) and petioles (FDST) (kgDM/ha/d)
      FDLV = (1.-RDSTLV) * DLV
      FDST = (1.-RDSTST) * DST

* Redistribution of matter from dying leaves (kg CH2O/ha/d)
```

```
      REDIST = RDSTLV * DLV + RDSTST * DST

* Available carbohydrates for growth (kg CH2O/ha/d)
      AVASS = GPHOT - MAINT + REDIST

*-----fraction of dry matter growth occurring in shoots, leaves, stems,
*     storage organs and roots, petioles, crowns, fibrous roots and beets
      FSH = LINT(FSHTB,IFSHN,TSUM2)

      FLV = LINT(FLVTB,IFLVN,TSUM2)
      FST = LINT(FSTTB,IFSTN,TSUM2)
      FCR = 1. - FLV - FST

      FRT = LINT(FRTTB,IFRTN,TSUM2)
      FSO = 1. -FRT

*-----assimilate requirements for dry matter conversion (kgCH2O/kgDM)
      ASRQ = FSH *(1.46*FLV + 1.51*(FST+FCR)) +
     $        (1.-FSH)*(ASRQSO*FSO+1.44*FRT)

* Total growth rate (kgDM/ha/d) and growth rates of
* shoots (leaf blades,petioles,crowns) and below-ground parts (roots,beets)
      GTW = AVASS / ASRQ
      GSH = FSH * GTW
      GLV = FLV * GSH
      GST = FST * GSH
      GCR = FCR * GSH
      GBLW= (1.-FSH) * GTW
      GSO = FSO * GBLW
      GRT = FRT * GBLW

      GLAI = GLA (DTEFF,TSUM2,LAI,RGRL,TSUMEM,SLA,GLV,DLV,LAII,DELT)

100   CONTINUE

**********************************************************************
*------communication with the optimize program PARAM

      IID = NINT(DAY)

      IF (IID.EQ.JDAY(IDSN,K)) THEN
        MPAR1(IDSN,K) = WDVI
        MPAR2(IDSN,K) = 999.9
        MPAR3(IDSN,K) = 999.9
        K = K + 1
        KSTOP = K-1
      END IF
```

```
      IF (IID.EQ.JDAY(IDSN,IFD)) THEN
         PAR1M(IDSN,K2) = WDVI
         PAR2M(IDSN,K2) = 999.9
         PAR3M(IDSN,K2) = 999.9
         IFD = IFD + 2
         K2 = K2 + 1
      END IF

      TIME  = TIME  + DELT

      GO TO 10

1000  CONTINUE

******END OF DYNAMIC PART*************************************************

*-----This causes model output to be sent to a file on the last
*     runs from subroutine OUTPUT. A time course is written of simulated
*     calibration variables for the parameter sets with average values,
*     best value of criterion, worst value of criterion, and with
*     nominal values:
*         AVERUN: model SBFLEVO is run with average parameter values
*         BESTR : model SBFLEVO is run with 'best' parameter values
*         WORSTR: model SBFLEVO is run with 'worst' parameter values
*         NOMINL: model SBFLEVO is run with nominal parameter values

*----- AVERUN: model SBFLEVO is run with average parameter values
      IF (AVERUN) THEN
       DO 25 K = 1, KSTOP
       WRITE(32,*) JDAY(IDSN,K),MPAR1(IDSN,K),MPAR2(IDSN,K),
     $       MPAR3(IDSN,K)
25     CONTINUE
       WRITE(32,'(A)')
      END IF

*----- BESTR : model SBFLEVO is run with 'best' parameter values
      IF (BESTR) THEN
       DO 30 K = 1, KSTOP
       WRITE(30,*) JDAY(IDSN,K),MPAR1(IDSN,K),MPAR2(IDSN,K),
     $       MPAR3(IDSN,K)
30     CONTINUE
       WRITE(30,'(A)')
      END IF

*----- WORSTR: model SBFLEVO is run with 'worst' parameter values
      IF (WORSTR) THEN
       DO 35 K = 1, KSTOP
       WRITE(31,*) JDAY(IDSN,K),MPAR1(IDSN,K),MPAR2(IDSN,K),
```

```
   $      MPAR3(IDSN,K)
35     CONTINUE
      WRITE(31,'(A)')
     END IF

*----- NOMINL: model SBFLEVO is run with nominal parameter values
     IF (NOMINL) THEN
      DO 40 K = 1, KSTOP
      WRITE(33,*) JDAY(IDSN,K),MPAR1(IDSN,K),MPAR2(IDSN,K),
   $      MPAR3(IDSN,K)
40     CONTINUE
      WRITE(33,'(A)')
     END IF

     RETURN
     END
```

```
* ----------------------------------------------------------------
* Function GLA:
* computes daily increase of leaf area index (ha leaf/ ha ground/ d)
* ----------------------------------------------------------------
      FUNCTION GLA (DTEFF,TSUM2,LAI,RGRL,TSUMEM,SLA,
     $              GLV,DLV,LAII,DELT)
      IMPLICIT REAL (A-Z)
*     during juvenile growth:
      IF ((TSUM2.LT.450.).AND.(LAI.LT.0.75)) THEN
        GLA = LAI * (EXP(RGRL*DTEFF*DELT)-1.)
*         GLA = LAII * RGRL*DTEFF*EXP(RGRL * TSUMEM)
      ELSE
*     during mature plant growth:
        GLA = SLA * (GLV - DLV)
      ENDIF
      RETURN
      END
```

```
*----------------------------------------------------------------------
*  REAL FUNCTION LINT
*  Authors: Daniel van Kraalingen
*  Date    : 28-JAN-1987
*  Purpose: This function is a linear interpolation function.  The
*           function does not extrapolate : in case of X below or
*           above the region defined by TABLE, the first
*           respectively the last Y-value is returned and a message
*           is generated.
*
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
*  name   meaning                                      units  class
*  ----   -------                                      -----  -----
*  LINT   function name, result of the interpolation     =      O
*  TABLE  A one-dimensional array with paired            =      I
*         data: x,y,x,y, etc.
*  ILTAB  The number of elements of the array            -      I
*         TABLE
*  X      The value at which interpolation should        =      I
*         take place
*
*  FATAL ERROR CHECKS (execution terminated, message)
*  condition
*  ---------
*  TABLE(I) < TABLE(I-2) , for I odd
*  ILTAB odd
*
*  No WARNINGS using the control variable IWAR are generated since
*  nobody will check IWAR after each LINT  call ; instead an X-valu
*  below TABLE(1) or above TABLE(ILTAB-1) is reported on screen
*  with a message containing the value of ILTAB and X. Further
*  information on the error is not available within this function.
*
*  No other SUBROUTINES and FUNCTIONS are called
*  No FILE's are used (error message with WRITE(*,...)... )
*----------------------------------------------------------------------
      REAL FUNCTION LINT (TABLE,ILTAB,X)
      IMPLICIT REAL (A-Z)
      INTEGER I, IUP, ILTAB
      DIMENSION TABLE(ILTAB)

*     check on odd ILTAB
      IF (MOD(ILTAB,2).NE.0) THEN
        WRITE (*,'(A,I4/,A)')
     $    ' ERROR in function LINT: ILTAB=',ILTAB,
     $    ' ILTAB must be even !'
       STOP
      END IF
```

```
      IUP = 0
      DO 10 I=3,ILTAB,2
*        check on ascending order of X-values in function
         IF (TABLE(I).LE.TABLE(I-2)) THEN
            WRITE (*,'(A,I4/,A,2F12.4,/,A,I4,A/,A)')
     $         ' X-coordinates not in ascending order at element',I,
     $         ' elements I-2 and I are',TABLE(I-2),TABLE(I),
     $         ' LINT-function contains',ILTAB,' points',
     $         ' Run deleted!'
       PAUSE
       STOP
         END IF
         IF (IUP.EQ.0.AND.TABLE(I).GE.X) IUP = I
10    CONTINUE

      IF (X.LT.TABLE(1)) THEN
         WRITE (*,'(A/A,I4,A/A,G12.4)')
     $         ' Interpolation below defined region!!',
     $         ' LINT-function contains ',ILTAB,' points,',
     $         ' Interpolation at X=',X
         LINT = TABLE(2)
         PAUSE
            GOTO 40
      END IF

      IF (X.GT.TABLE(ILTAB-1)) THEN
         WRITE (*,'(A/A,I4,A/A,G12.4)')
     $         ' Interpolation above defined region!!',
     $         ' LINT-function contains ',ILTAB,' points,',
     $         ' Interpolation at X=',X
         LINT = TABLE(ILTAB)
         PAUSE
            GO TO 40
      END IF

*     normal interpolation
      SLOPE = (TABLE(IUP+1)-TABLE(IUP-1))/(TABLE(IUP)-TABLE(IUP-2))
      LINT = TABLE(IUP-1)+(X-TABLE(IUP-2))*SLOPE

40    RETURN
      END
```

```
*------------------------------------------------------------------
*  SUBROUTINE ASTRO
*  Authors: Daniel van Kraalingen
*  Date   : 9-Aug-1987
*  Modified by Jan Goudriaan 4 Febr 1988
*  Purpose: This subroutine calculates astronomic daylength and
*           photoperiodic daylength.
*           and diurnal radiation characteristics such as daily
*           integral of sine of solar elevation, solar constant
*           Measured daily total of global radiation is used to find
*           atmospheric transmissivity and fraction diffuse
radiation
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
*  name    meaning                                    units  class
*  ----    -------                                    -----  -----
*  DAY     Day number (Jan 1st = 1)                     -      I
*  LAT     Latitude of the site                      degrees   I
*       Measured daily total global radiation      J m-2 d-1 I
*  SC      Solar constant                            J m-2 s-1 O
*  SINLD   Seasonal offset of sine of solar height      -      O
*  COSLD   Amplitude of sine of solar height            -      O
*  DAYL    Astronomical daylength (base = 0 degrees)    h      O
*  DSINB   Daily total of sine of solar height          s      O
*  DSINBE  Daily total of effective solar height        s      O
*  ATMTR   Atmospheric transmissivity                   -      O
*  FRDIF   Fraction diffuse in global radiation         -      O
*
*  FATAL ERROR CHECKS (execution terminated, message)
*  condition
*
*  LAT > 67, LAT < -67
*
*  SUBROUTINES and FUNCTIONS called : none
*
*  FILE usage : none
*-----------------------------------------------------------------

      SUBROUTINE ASTRO (DAY,LAT,DTR,
     &  SC,SINLD,COSLD,DAYL,DSINB,DSINBE,ATMTR,FRDIF)
      IMPLICIT REAL (A-Z)

*-----PI and conversion factor from degrees to radians
      PARAMETER (PI=3.141592654, RAD=0.017453292)

*-----check on input range of parameters
      IF (LAT.GT.67.)  STOP 'ERROR IN ASTRO: LAT > 67'
      IF (LAT.LT.-67.) STOP 'ERROR IN ASTRO: LAT <-67'
```

```
*-----declination of the sun as function of daynumber (DAY)
      DEC = -ASIN(SIN(23.45*RAD)*COS(2.*PI*(DAY+10.)/365.))

*-----SINLD, COSLD and AOB are intermediate variables

      SINLD = SIN(RAD*LAT)*SIN(DEC)
      COSLD = COS(RAD*LAT)*COS(DEC)
      AOB = SINLD/COSLD

*-----daylength (DAYL)

      DAYL  = 12.0*(1.+2.*ASIN(AOB)/PI)

      DSINB = 3600.*(DAYL*SINLD+24.*COSLD*SQRT(1.-AOB*AOB)/PI)
      DSINBE= 3600.*(DAYL*(SINLD+0.4*(SINLD*SINLD+COSLD*COSLD*0.5))+
     $       12.0*COSLD*(2.0+3.0*0.4*SINLD)*SQRT(1.-AOB*AOB)/PI)

*-----solar constant (SC) and daily extraterrestrial radiation (ANGOT)

      SC = 1370.*(1.+0.033*COS(2.*PI*DAY/365.))
      DSO = SC*DSINB

*-----diffuse light fraction (FRDIF) from atmospheric transmission (ATMTR)

      ATMTR = DTR/DSO
      IF (ATMTR.GT.0.75) THEN
        FRDIF = 0.23
      ELSE IF (ATMTR.LE.0.75.AND.ATMTR.GT.0.35) THEN
        FRDIF = 1.33-1.46*ATMTR
      ELSE IF (ATMTR.LE.0.35.AND.ATMTR.GT.0.07) THEN
        FRDIF = 1.-2.3*(ATMTR-0.07)**2
      ELSE
        FRDIF=1.
      END IF
      RETURN
      END
```

```
*----------------------------------------------------------------------
*  SUBROUTINE TOTASS
*  Authors: Daniel van Kraalingen
*  Date   : 10-Dec-1987
*  Modified by Jan Goudriaan 5-Febr-1988, to seconds 29-June-89
*  Purpose: This subroutine calculates daily total gross
*           assimilation (DTGA) by performing a Gaussian integration
*           over time. At three different times of the day,
*           radiation is computed and used to determine assimilation
*           whereafter integration takes place.
*
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
*  name    meaning                                        units  class
*  ----    -------                                        -----  -----
*  SC      Solar constant                               J m-2 s-1 I
*  DAYL    Astronomical daylength (base = 0 degrees       h       I
*  SINLD   Seasonal offset of sine of solar height        -       I
*  COSLD   Amplitude of sine of solar height              -       I
*  DSINBE  Daily total of effective solar height          s       I
*  DTR     Daily total of global radiation              J/m2/d    I
*  ATMTR   Atmospheric transmissivity                     -       I
*  FRDIF   Fraction diffuse in global radiation           -       I
*  SCV     Scattering coefficient of leaves for visible
*             radiation (PAR)                             -       I
*  AMAX    Assimilation rate at light saturation       kg CO2/    I
*                                                     ha leaf/h
*  EFF     Initial light use efficiency               kg CO2/J/   I
*                                                     ha/h m2 s
*  KDIF    Extinction coefficient for diffuse light               I
*  LAI     Leaf area index                              ha/ha     I
*  DTGA    Daily total gross assimilation            kg CO2/ha/d O
*
*
*  SUBROUTINES and FUNCTIONS called : ASSIM
*
*  FILE usage : none
*----------------------------------------------------------------------

      SUBROUTINE TOTASS (SC,DAYL,SINLD,COSLD,DSINBE,
     $ DTR,ATMTR,FRDIF,SCV,AMAX,EFF,KDIF,LAI,DTGA)
      IMPLICIT REAL(A-Z)
      REAL XGAUSS(3), WGAUSS(3)
      INTEGER I, IGAUSS

      PARAMETER (PI=3.141592654)

      DATA IGAUSS /3/
      DATA XGAUSS /0.1127, 0.5000, 0.8873/
```

```
      DATA WGAUSS /0.2778, 0.4444, 0.2778/

*-----assimilation set to zero and three different times of the day (HOUR)

      DTGA = 0.
      DO 10 I=1,IGAUSS
       HOUR = 12.0+DAYL*0.5*XGAUSS(I)

*-------at the specified HOUR, radiation is computed and used to compute
*       assimilation

*-----sine of solar elevation

      SINB  = AMAX1(0.,SINLD+COSLD*COS(2.*PI*(HOUR+12.)/24.))

*-----diffuse PAR (PARDIF) and direct PAR (PARDIR)

      PAR    = 0.5*DTR*SINB*(1.+0.4*SINB)/DSINBE
      PARDIF =  MIN (PAR,SINB*FRDIF*ATMTR*0.5*SC)
      PARDIR = PAR-PARDIF
       CALL ASSIM (SCV,AMAX,EFF,KDIF,LAI,SINB,PARDIR,PARDIF,FGROS)

*-------integration of assimilation rate to a daily total (DTGA)

       DTGA = DTGA+FGROS*WGAUSS(I)
10     CONTINUE

*-------to kg ha-1 d-1:
      DTGA = DTGA*DAYL*36.

      RETURN
      END
```

```
*-------------------------------------------------------------------
*  SUBROUTINE ASSIM
*  Authors: Daniel van Kraalingen
*  Date   : 10-Dec-1987
*  Modified by Jan Goudriaan 5-Febr-1988
*  Purpose: This subroutine performs a Gaussian integration over
*           depth of canopy by selecting three different LAI's and
*           computing assimilation at these LAI levels. The
*           integrated variable is FGROS.
*
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)
*  name    meaning                                     units  class
*  ----    -------                                     -----  -----
*  SCV     Scattering coefficient of leaves for visible
*             radiation (PAR)                            -      I
*  AMAX    Assimilation rate at light saturation     kg CO2/   I
*                                                   ha leaf/h
*  EFF     Initial light use efficiency             kg CO2/J/  I
*                                                   ha/h m2 s
*  KDIF    Extinction coefficient for diffuse light             I
*  LAI     Leaf area index                             ha/ha    I
*  SINB    Sine of solar height                          -      I
*  PARDIR  Instantaneous flux of direct radiation (PAR) W/m2    I
*  PARDIF  Instantaneous flux of diffuse radiation(PAR) W/m2    I
*  FGROS   Instantaneous assimilation rate of        kg CO2/   O
*          whole canopy                             ha soil/h
*
*  SUBROUTINES and FUNCTIONS called : none
*
*  FILE usage : none
*-------------------------------------------------------------------
      SUBROUTINE ASSIM (SCV,AMAX,EFF,KDIF,LAI,SINB,PARDIR,
     $PARDIF,FGROS)
      IMPLICIT REAL(A-Z)
      REAL XGAUSS(3), WGAUSS(3)
      INTEGER I1, I2, IGAUSS

*-----Gauss weights for three point Gauss
      DATA IGAUSS /3/
      DATA XGAUSS /0.1127, 0.5000, 0.8873/
      DATA WGAUSS /0.2778, 0.4444, 0.2778/

*-----reflection of horizontal and spherical leaf angle distribution
      SQV=SQRT(1.-SCV)
      REFH = (1.-SQV)/(1.+SQV)
C     REFS = REFH*2./(1.+1.6*SINB)
      REFS=REFH
*-----extinction coefficient for direct radiation and total direct flux
```

```
      CLUSTF=KDIF/(0.8*SQV)
      KDIRBL = (0.5/SINB)*CLUSTF
      KDIRT=KDIRBL*SQV

*-----selection of depth of canopy, canopy assimilation is set to zero

      FGROS = 0.
      DO 10 I1=1,IGAUSS
        LAIC = LAI*XGAUSS(I1)

*--------absorbed fluxes per unit leaf area: diffuse flux, total direct
*        flux, direct component of direct flux.

        VISDF = (1.-REFS)*PARDIF*KDIF*EXP(-KDIF*LAIC)
        VIST  = (1.-REFS)*PARDIR*KDIRT*EXP(-KDIRT*LAIC)
        VISD  = (1.-SCV)*PARDIR*KDIRBL*EXP(-KDIRBL*LAIC)

*--------absorbed flux (J/M2 leaf/s) for shaded leaves and assimilation of
*        shaded leaves

        VISSHD = VISDF+VIST-VISD
        FGRSH  = AMAX*(1.-EXP(-VISSHD*EFF/AMAX))

*--------direct flux absorbed by leaves perpendicular on direct beam and
*        assimilation of sunlit leaf area

        VISPP = (1.-SCV)*PARDIR/SINB
        FGRSUN = 0.
        DO 20 I2=1,IGAUSS
          VISSUN = VISSHD+VISPP*XGAUSS(I2)
          FGRS = AMAX*(1.-EXP(-VISSUN*EFF/AMAX))
          FGRSUN = FGRSUN+FGRS*WGAUSS(I2)
20      CONTINUE

*--------fraction sunlit leaf area (FSLLA) and local assimilation rate (FGL)

        FSLLA = CLUSTF*EXP(-KDIRBL*LAIC)
        FGL =FSLLA*FGRSUN+(1.-FSLLA)*FGRSH

*--------integration of local assimilation rate to canopy assimilation (FGROS)

        FGROS = FGROS+FGL*WGAUSS(I1)
10    CONTINUE
      FGROS = FGROS*LAI

      RETURN
      END
```

```
*------------------------------------------------------------------
* SUBROUTINE CLOUDX
* Calculates X-band radar backscatter from 'Cloud' model
* one-layer version for sugar beet.
*
* FORMAL PARAMETERS:
* name:        meaning:
* TADRW        above-ground dry biomass               kg/ha      I
* MS           topsoil moisture content               %(vol.)    I
* MCCROP       plant water content                    %
* GAMMA(I)     radar backscatter crop at I degrees
*              incidence angle                        dB         O
*
* LOCAL PARAMETERS
* PLWCROP      plant water of canopy                  kg/m2      O
* INC(I)       I/10 incidence angle                   degree
* CPL(I)       Cloud parameter C at I/10 incidence    -
* GS(I)        Cloud parameter G at I/10 incidence    -
* DPL          Attenuation factor D (plant)           -
* KS           Moisture coefficient K (soil)          -
*---------------------------------------------------------------------
      SUBROUTINE CLOUDX(PI,MCCROP,DPL,KS,NC,GS,CPL,INC,MS,
     $           TADRW,PLWCROP,ATT,SOIL,PLANT,GAMMA)

      IMPLICIT REAL (A-H,J-Z)
      IMPLICIT INTEGER (I)

      REAL NC(100),CPL(10),GS(10),ATT(10),SOIL(10),PLANT(10)
      REAL GAMMA(10)

      RADC=PI/180.

* calculation of plant water in kg/m2
      PLWCROP = 0.0001*TADRW*MCCROP/(100.-MCCROP)

* calculation of radar backscatter
      DO 10 I=1,INC
       ATT(I)=DPL*PLWCROP/COS(RADC*NC(I))
       SOIL(I)=GS(I)*EXP(KS*MS-ATT(I))
       PLANT(I)=CPL(I)*(1.-EXP(-ATT(I)))
       GAMMA(I)=10*ALOG10(PLANT(I)+SOIL(I))
10     CONTINUE
      RETURN
      END
```

```
*-----------------------------------------------------------------
* SUBROUTINE CLOUDM
* Calculates L- to Ku2-band radar backscatter from 'Cloud' model
* one-layer version for sugar beet.
*
* FORMAL PARAMETERS:
* name:       meaning:
* TADRW       above-ground dry biomass                 kg/ha      I
* MS          topsoil moisture content                 %(vol.)    I
* MCCROP      plant water content                      %
* GAMMA(I,J) radar backscatter crop at I degrees
*             incidence angle, and at frequency J      dB         O
*             J=1 = L-band
*               2 = S-band
*               3 = C-band
*               4 = X-band
*               5 = Ku1-band
*               6 = Ku2-band
* LOCAL PARAMETERS
* Note that J stands for frequency!
* PLWCROP     plant water of canopy                    kg/m2      O
* NCA(I)        I/10 incidence angle                     degree
* CPLA(I,J)      Cloud parameter C at I/10 incidence    -
* GSA(I,J)       Cloud parameter G at I/10 incidence    -
* DPLA(J)         Attenuation factor D (plant)          -
* KSA(J)          Moisture coefficient K (soil)         -
*-----------------------------------------------------------------
      SUBROUTINE CLOUDA(PI,MCCROP,DPLA,KSA,NCA,GSA,CPLA,MS,
     $         TADRW,PLWCROP,ATTA,SOILA,PLANTA,GAMMAA)

      IMPLICIT REAL (A-H,K-Z)
      IMPLICIT INTEGER (I)
      IMPLICIT INTEGER (J)

      REAL CPLA(5,6),GSA(5,6),ATTA(5,6),SOILA(5,6),PLANTA(5,6)
      REAL GAMMAA(5,6)

*-----NB: 5 incidence angles and 6 frequencies in Agriscatt

      REAL NCA(5)
      REAL KSA(6),DPLA(6)

      RADC=PI/180.

* calculation of plant water in kg/m2
      PLWCROP = 0.0001*TADRW*MCCROP/(100.-MCCROP)

* calculation of radar backscatter
```

```
      DO 10 J=1,6
       DO 5 I=1,5
        ATTA(I,J)=DPLA(J)*PLWCROP/COS(RADC*NCA(I))
        SOILA(I,J)=GSA(I,J)*EXP(KSA(J)*MS-ATTA(I,J))
        PLANTA(I,J)=CPLA(I,J)*(1.-EXP(-ATTA(I,J)))
        GAMMAA(I,J)=10*ALOG10(PLANTA(I,J)+SOILA(I,J))
5      CONTINUE
10     CONTINUE
      RETURN
      END
```

```
*******************************************************************
* Radiation programme EXTRAD by Jan Goudriaan (1977)               *
* Version adapted by Bouman 1990, for calculations of reflectance  *
* Calculation of nadir reflectance in G, R and IR                  *
* and calculation of reflectance ratio's  IR/GR, WDVI and NDVI     *
*                                                                  *
* FORMAL PARAMETERS:                                               *
* name:        meaning:                                            *
* RHOSG        green hemispherical reflectance coeff. soil         *
* RHOSR        red hemispherical reflectance coeff. soil           *
* RHOSIR       infrared hemispherical reflectance coeff. soil      *
* SCATG        green scattering coeff. leaves                      *
* SCATR        red scattering coeff. leaves                        *
* SCATIR       infrared scattering coeff. leaves                   *
* BETA         solar elevation angle                               *
* F            leaf angle distribution                             *
* FRDIF        fraction diffuse sky  irradiance                    *
* LAI          leaf area index                                     *
*                                                                  *
* LOCAL PARAMETERS:                                                *
* name:        meaning:                                            *
* NADG         nadir green canopy reflectance                      *
* NADR         nadir red canopy reflectance                        *
* NADIR        nadir infrared canopy reflectance                   *
* RATIO        infrared/green reflectance ratio                    *
* WDVI         weighted difference vegetation index                *
* NDVI         normalized difference vegetation index              *
*******************************************************************

      SUBROUTINE REFLEX(RHOSG,RHOSR,RHOSIR,SCATG,SCATR,SCATIR,BETA,
     $         F,BU,LAI,FRDIF,NADG,NADR,NADIR,NDVI,WDVI,RATIO)

      IMPLICIT REAL(A-Z)
      INTEGER I,J,K,IS,ITER,ITERM,N,N1
      REAL BL(9),MI(9),MT(9),OAV(9),SM(9),F(9),BU(9)
      REAL PHID(9,101),PHIU(9,101)
      REAL RN(101),TPHIU(1010),TPHID(101)
      PARAMETER (PI=3.1415926)

*****************************************************************
*-----initialisation; calculations for leaf angle distribution

      FRDIR=1.-FRDIF
      LS=0.1
      IS=1+ INT(BETA*0.1)
      RAD=PI/180.
      SBL=0.
```

```
      DO 20 K=1,9
      SINB=SIN(RAD*(10.*(FLOAT(K)-0.5)))
      COSB=COS(RAD*(10.*(FLOAT(K)-0.5)))
      OAV(K)=0.
      DO 25 I=1,9
        SINL=SIN(RAD*(10.*(FLOAT(I)-0.5)))
          COSL=COS(RAD*(10.*(FLOAT(I)-0.5)))
          IF(K.GE.I)THEN
            O=SINB*COSL
          ELSE
            O=2.*(SINB*COSL*ASIN(SINB*COSL/(SINL*COSB))+
     $          SQRT(SINL*SINL-SINB*SINB))/PI
        ENDIF
*-----OAV(K) is the leaf projection into direction K:
      OAV(K)=OAV(K)+O*F(I)
25    CONTINUE
*-----Fraction intercepted MI and transmitted MT:
      MI(K)=OAV(K)*LS/SINB
      MT(K)=1.-MI(K)
*-----SBL is needed for normalization of the view factors BL(K):
      SBL=SBL+BU(K)*MI(K)
20    CONTINUE


*-----initialsiation according to leaf layers
      N=INT(LAI/LS  +0.5)
      N1=N+1

*************************************************************************
*-----loop for calculating red and IR parts of the profile

      SCAT=SCATG
      RHOS=RHOSG

10    CONTINUE

*-----initialization of radiation profile:
      DO 30 K=1,9
      DO 40 J=1,N1
      PHIU(K,J)=0.
      PHID(K,J)=0.
40    CONTINUE
      BL(K)=BU(K)*MI(K)/SBL
* diffuse radiation is distributed according to BU(K):
      PHID(K,1)=100.*BU(K)*FRDIF
30    CONTINUE
* the direct incoming component is added:
      PHID(IS,1)=PHID(IS,1)+100.*FRDIR
```

```
***********************************************************************
*----main routine for the profile calculation

*-----number of iterations depends on scattering coefficient:
     ITERM=1
     IF(SCAT.GT.0.1)ITERM=2
     IF(SCAT.GT.0.5)ITERM=5
     IF(SCAT.GT.0.9)ITERM=10
     IF(SCAT.GT.0.99)ITERM=20
     IF(SCAT*RHOS.GT.0.99)ITERM=50

     DO 2000 ITER=1,ITERM

*-----first a calculation from top to bottom:
     DO 220 J=2,N1
     INTER=0.
       DO 205 K=1,9
205       INTER=INTER + MI(K)*(PHID(K,J-1)+PHIU(K,J))
       DO 215 K=1,9
215      PHID(K,J)=PHID(K,J-1)*MT(K) +0.5*SCAT*INTER*BL(K)
220   CONTINUE

* reflected radiation at the soil surface:
     INTER=0.
     DO 225 K=1,9
225   INTER=INTER + PHID(K,N1)
     DO 230 K=1,9
230   PHIU(K,N1)=RHOS*INTER*BU(K)

* now the calculation from bottom to top:
     DO 250 J=N,1,-1
     INTER=0.
       DO 235 K=1,9
235      INTER=INTER +MI(K)*(PHID(K,J)+PHIU(K,J+1))
       DO 250 K=1,9
250      PHIU(K,J)=PHIU(K,J+1)*MT(K)+0.5*SCAT*INTER*BL(K)


2000  CONTINUE

***********************************************************************
*-----calculating the nadir reflection and reflectance, writing output

     DO 2500 LAIC=0,LAI,0.5
     J=1+INT(LAIC/LS)
     TPHIU(J)=0.
     TPHID(J)=0.
```

```
      DO 2150 K=1,9
      TPHIU(J)=TPHIU(J)+PHIU(K,J)
      TPHID(J)=TPHID(J)+PHID(K,J)
2150  CONTINUE
2500  CONTINUE

*-----calculation of reflectance values for the green, red and IR part
      IF (SCAT .EQ. SCATG .AND. RHOS .EQ. RHOSG) THEN
       RHOMG=100*TPHIU(1)/TPHID(1)
       NADG=PHIU(9,1)/BU(9)
       SCAT=SCATR
       RHOS=RHOSR
       GOTO 10
      END IF

      IF (SCAT .EQ. SCATR .AND. RHOS .EQ. RHOSR) THEN
       RHOMR=100*TPHIU(1)/TPHID(1)
       NADR=PHIU(9,1)/BU(9)
       SCAT=SCATIR
       RHOS=RHOSIR
       GOTO 10
      ELSE
       RHOMIR=100*TPHIU(1)/TPHID(1)
       NADIR=PHIU(9,1)/BU(9)
      ENDIF

*-----calculation of some reflectance ratio's
*-----1)NDVI (Tucker)
      NDVI=(NADIR-NADG)/(NADIR+NADG)

*-----2) IR/GREEN ratio
      RATIO=NADIR/NADG

*-----3)WDVI (Clevers)
      WDVI=NADIR-(RHOSIR/RHOSG)*NADG
**********************************************************************


      RETURN
      END
```

## APPENDIX IV STATIS.GEN

**Genstat5 program for plotting histograms of initial and final NPS parameter values.**

```
"***************STATIS.GEN*****************"
" GENSTAT5 program                          "
" calculates some statistics on output from "
" SBFLEVO_OPT: histograms of the initial and "
" final NPS parameter sets; and correlations "
" between the parameters.                   "
"                                           "
" Bouman, 1992                              "
"*******************************************"
VARIATE[NVALUES=200] PAR1,PAR2,PAR3,PAR4,PAR5
OPEN 'FOR010.DAT';CHANNEL=2
READ[CHANNEL=2] PAR1,PAR2,PAR3,PAR4,PAR5
HISTOGRAM PAR1
HISTOGRAM PAR2
HISTOGRAM PAR3
HISTOGRAM PAR4
HISTOGRAM PAR5
OPEN 'FOR011.DAT';CHANNEL=3
READ[CHANNEL=3] PAR1,PAR2,PAR3,PAR4,PAR5
HISTOGRAM PAR1
HISTOGRAM PAR2
HISTOGRAM PAR3
HISTOGRAM PAR4
HISTOGRAM PAR5
MODEL PAR1
TERMS[PRINT=CORRELATIONS] PAR2,PAR3,PAR4
STOP
```