RIGAUS

Random Input Generator for the Analysis of Uncertainty in Simulation

B.A.M. Bouman & M.J.W. Jansen (GLW-DLO)

cabo-dlo

292553

Simulation Reports CABO-TT, nr. 34
December 1993

Simulation Reports CABO-TT

Simulation Reports CABO-TT is a series giving supplementary information on agricultural simulation models that have been published elsewhere. Knowledge of those publications will generally be necessary in order to be able to study this material.

Simulation Reports CABO-TT describe improvements of simulation models, new applications or translations of the programs into other computer languages. Manuscripts or suggestions should be submitted to: H. van Keulen (CABO) or J. Goudriaan (TPE).

Simulation Reports CABO-TT are issued by CABO and TPE and they are available on request. Announcements of new reports will be issued regularly. Addresses of those who are interested in the announcements will be put on a mailing list on request.

The DLO Centre for Agrobiological Research (CABO-DLO) falls under the Agricultural Research Department (DLO) of the Dutch Ministry of Agriculture, Nature Management and Fisheries.

The aim of DLO is to generate knowledge and develop expertise for implementing the agricultural policies of the Dutch government, for strengthening the agricultural industry, for planning and management of rural areas and for the protection of the environment. At CABO-DLO experiments and computer models are used in fundamental and strategic research on plants. The results are used to:
- achieve optimal and sustainable plant production systems;
- find new agricultural products and improve product quality;
- enhance nature and environmental quality in the countryside.

Address (from 1 November 1993):
AB-DLO
P.O. Box 14
6700 AA Wageningen
The Netherlands

tel. 31.8370.75700
fax. 31.8370.23110
e-mail postkamer@ab.agro.nl

GLW-DLO
Postbus 10
6700 AC Wageningen
The Netherlands

tel: 31.8370.74676
fax: 31.8370.11524
e-mail postkamer@glw.agro.nl

Table of Contents

Abstract

This report contains a description and users guide of the program RIGAUS (Random Input Generator for Uncertainty Analysis in Simulation) for the random generation of parameter values from uniform, beta and normal statistical distributions and from measurement series. The generated parameter values can be used for Monte Carlo (MC) analysis with simulation models. RIGAUS was especially designed for MC analysis with crop growth and water balance simulation models under the FORTRAN Simulation Environment (FSE). However, the output data can be used with any other program, and in other simulation environments as well. RIGAUS has special provisions for the generation of random parameter values for the soil water balance model SAHEL.

RIGAUS is written in the programming language FORTRAN, and was developed on a 486 IBM compatible PC. A full listing of the FORTRAN source code, and of the input and output files is given in the Appendices.

1 Introduction

In statistical evaluation and uncertainty and/or variation analysis of simulation models, Monte Carlo (MC) simulation is a useful technique (Hazelhof et al., 1990; Kros et al., 1990; Bouman, 1993a, 1993b; Rossing et al., 1993) . In applying this technique, a simulation model is run a large number of times using random values for specified input parameters and/or variables. These random values are drawn from statistical distributions or from measurement series. The program described in this report, RIGAUS, allows the user to draw random values from uniform, beta and normal statistical distributions, and from measured data sets for a number of variables at the same time. With one exception (see below), values for different parameters/variables are drawn independently, i.e. without taking correlation between parameters/variables into account. RIGAUS is specifically designed for MC simulation with crop growth and soil water balance models under the FORTRAN Simulation Environment (FSE), as developed by van Kraalingen at CABO-DLO and TPE-WAU (van Kraalingen, 1989). However, the output generated by RIGAUS can also be used with other simulation programs, and under other environments. RIGAUS has special provisions for drawing random input data for the soil water balance model SAHEL(van Keulen, 1975; van Keulen & Wolf, 1986; Penning de Vries & van Laar, 1982; Penning de Vries et al.,1989). The correlation between the 'water content' variables, i.e. water content at saturation (WCST), water content at field capacity (WCFC) and water content at wilting point (WCWP) can optionally be taken into account. Relationships between these variables have been derived from empirical data and are used in RIGAUS to generate values for WCST and WCWP from randomly drawn values of WCFC. The generated values of WCST, WCFC and WCWP are automatically assigned to all three soil layers defined in SAHEL. Also, the initial water content, expressed as fraction of WCFC (FWCLI) is automatically assigned to all three soil layers.

In Chapter 2 of this report, the statistical distributions Uniform, Beta and Normal are explained. Chapter 3 deals with the drawing from measured data. In Chapter 4, the special provisions for the soil water balance SAHEL are described. Chapter 5 explains the use of RIGAUS, and describes the input and output files. The appendices comprise a complete listing of the FORTRAN source of RIGAUS, the input file RIGAUS.IN and the output files RERUNS.DAT and COLUMN.DAT.

To obtain the RIGAUS program (source code and all necessary object libraries), write to:

B.A.M. Bouman
Research Institute for Agrobiology and Soil Fertility (AB-DLO)
P.O. Box 14,
6700 AA Wageningen,
The Netherlands.

2 Statistical distributions

In RIGAUS, values can randomly be generated from uniform, beta or normal statistical distributions. In the current version, a maximum of 10 uniform distributions, 10 beta distributions and 10 normal distributions are available simultaneously (hence in total 30 distributions). Random values can be generated up to a maximum of 2000 per distribution.

2.1 Uniform distribution

Random values for a uniform distribution are generated using the function RUNI. The algorithm in RUNI originates from L'Ecuyer (1986) as implemented in Bratley et al. (1983) and Press et al. (1992). The values generated by RUNI are restricted between 0 and 1, but are rescaled in RIGAUS between upper and lower boundaries as specified by the user. An example of the frequency distribution of 2000 randomly generated values from a uniform distribution is given in Fig. 1.

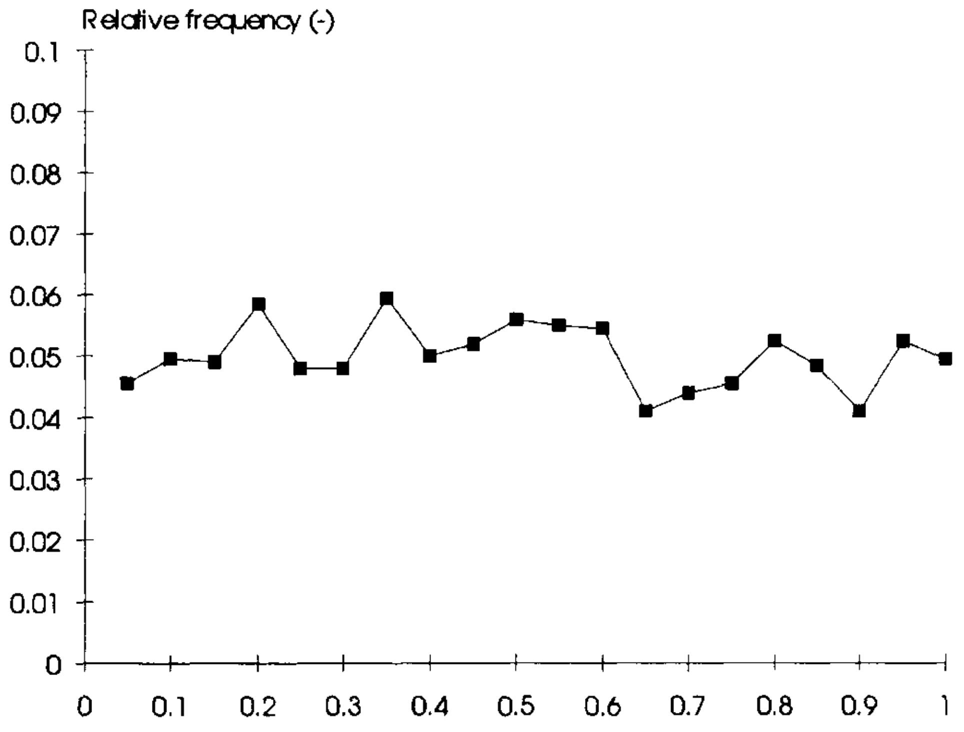


Figure 1 Relative frequency distribution (fraction) of randomly drawn values from a uniform distribution, using RIGAUS. N=2000.

The input that has to be supplied by the user for drawing from a uniform distribution is (per parameter/variable):

- Name of variable(s) for which random values have to be chosen (VUNI)
- Upper limit (UNIUP)
- Lower limit (UNILO)

2.2 Beta distribution

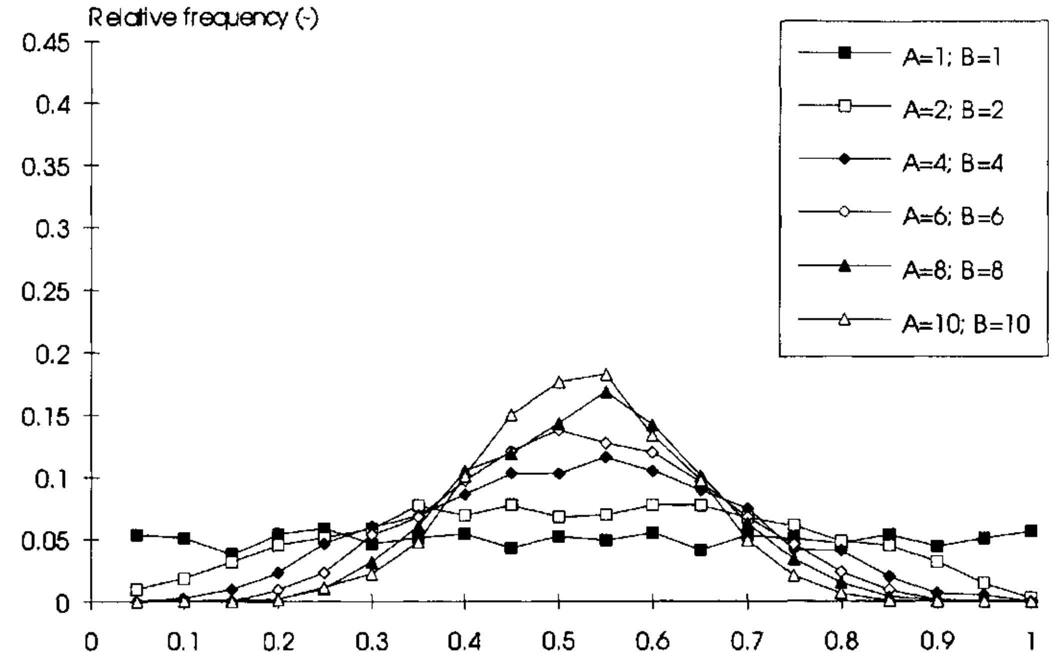
Random values for a beta distribution are generated using the function RBET. This random generator is fully based on the function BETACH (Bratley et al., 1983). A beta distribution is characterised by two 'shape' parameters, A and B, that define the shape of the distribution, e.g. 'bell' shaped, 'triangular' or 'skewed'. The examples given in Fig. 2 are distributions of 2000 randomly generated values using RIGAUS with different A and B values. The mean of the distribution is A/(A+B) and the variance is AB/[(A+B+1)(A+B)(A+B)], as illustrated in Table 1. As with the uniform distribution, the values generated by RBET are restricted between 0 and 1, but are rescaled in RIGAUS between upper and lower boundaries as specified by the user.

Table 1 Mean μ (upper number, bold) and variance σ^2 (lower number) of the beta distribution between 0-1 as function of the shape parameters A and B.

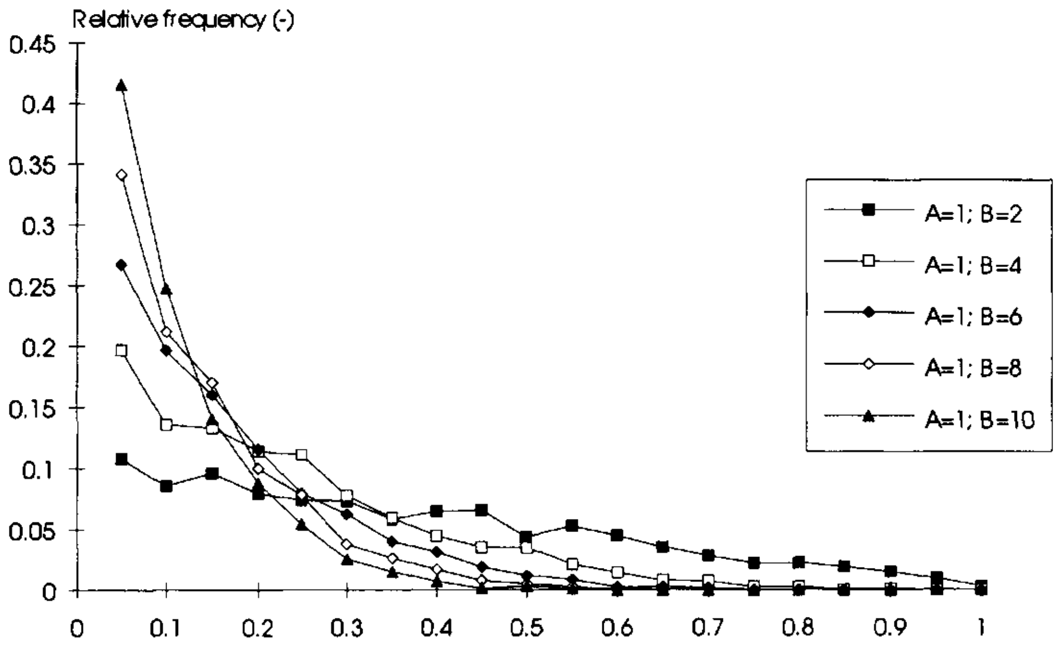
A \ B	1	2	4	6	8	10
1	0.500 0.083	0.333 0.056	0.200 0.027	0.143 0.015	0.111 0.010	0.091 0.007
2	0.667 0.056	0.500 0.050	0.333 0.032	0.250 0.021	0.200 0.015	0.167 0.011
4	0.800 0.027	0.667 0.032	0.500 0.028	0.400 0.022	0.333 0.017	0.286 0.014
6	0.857 0.015	0.750 0.021	0.600 0.022	0.500 0.019	0.429 0.016	0.375 0.014
8	0.889 0.010	0.800 0.015	0.667 0.017	0.571 0.016	0.500 0.015	0.444 0.013
10	0.909 0.007	0.833 0.011	0.714 0.014	0.625 0.014	0.556 0.013	0.500 0.012

The input t`hat has to be supplied by the user for drawing from a beta distribution is (per parameter/variable):

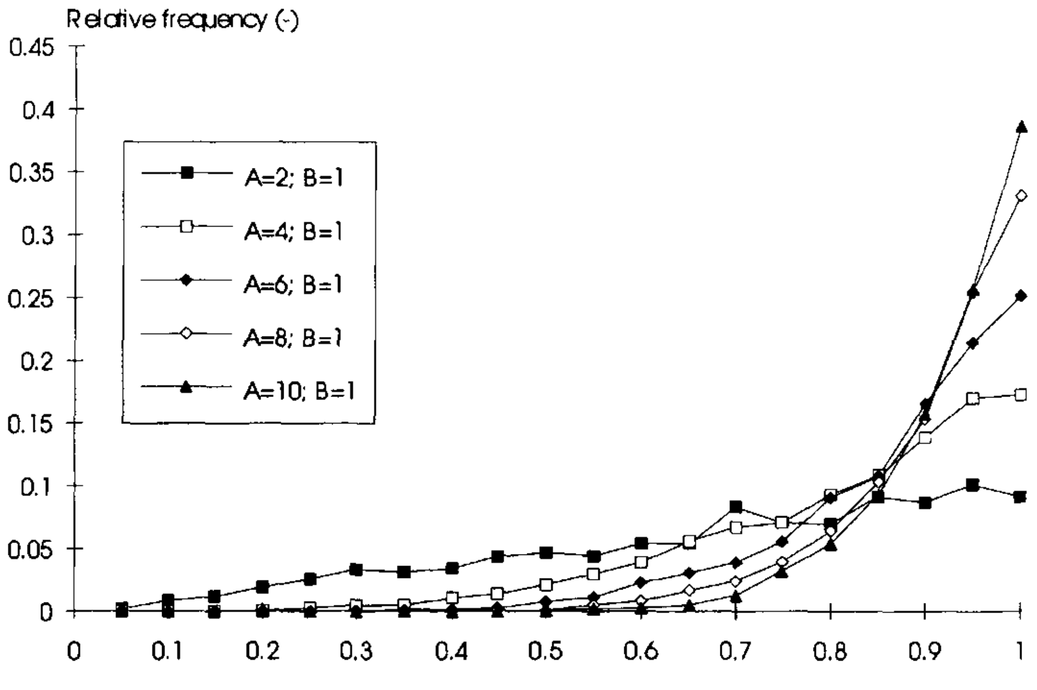
- Name of variable(s) for which random values have to be chosen (VBETA)
- Shape parameter A (ABETA)
- Shape parameter B (BBETA)
- Upper limit (BETAUP)
- Lower limit (BETALO)


Relative frequency (-)
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
A=1; B=1
A=2; B=2
A=4; B=4
A=6; B=6
A=8; B=8
A=10; B=10

(a)


Relative frequency (-)
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
A=1; B=2
A=1; B=4
A=1; B=6
A=1; B=8
A=1; B=10

(b)



(c)

Figure 2 Relative frequency distribution (fraction) of randomly drawn values from a beta distribution, using RIGAUS. N=2000. Different combinations of the A and B parameters are used in Figs. 2a, 2b and 2c.

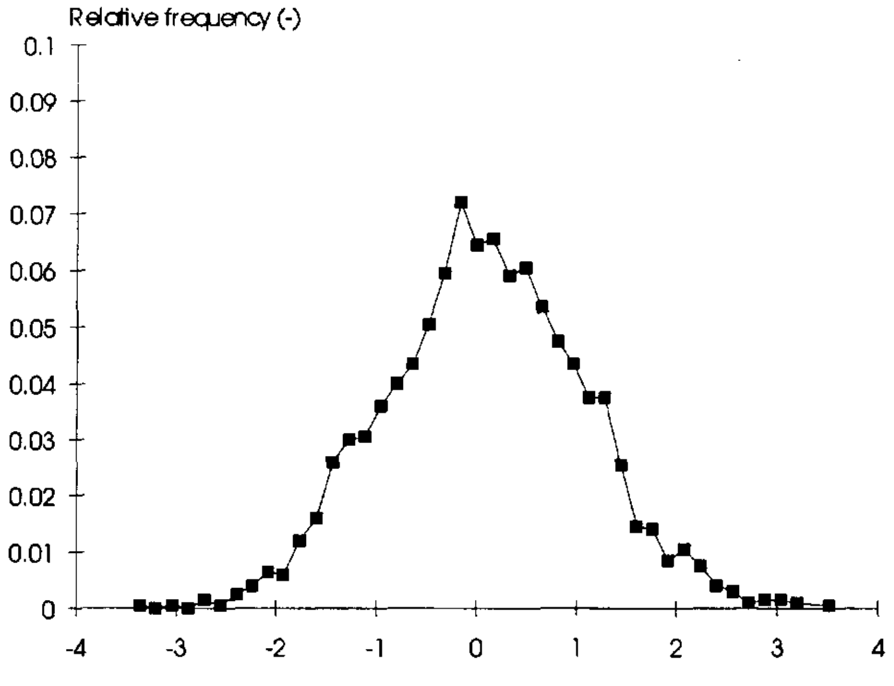
2.3 Normal distribution

Random values for a normal distribution are generated using the function RGAU. This random generator is based on the Box-Muller method (Box & Muller, 1958). The normal distribution generated by RGAU has a mean of 0 and a variance of 1, but in RIGAUS, the mean and variance of the distribution can be set by the user. Examples of normal distributions with different means m and variances σ2 , as generated by RIGAUS, are given in Fig. 3. Note, that on average, 95 % of the values of a normal distribution lie between m- 2σ2 and μ+ 2σ2.

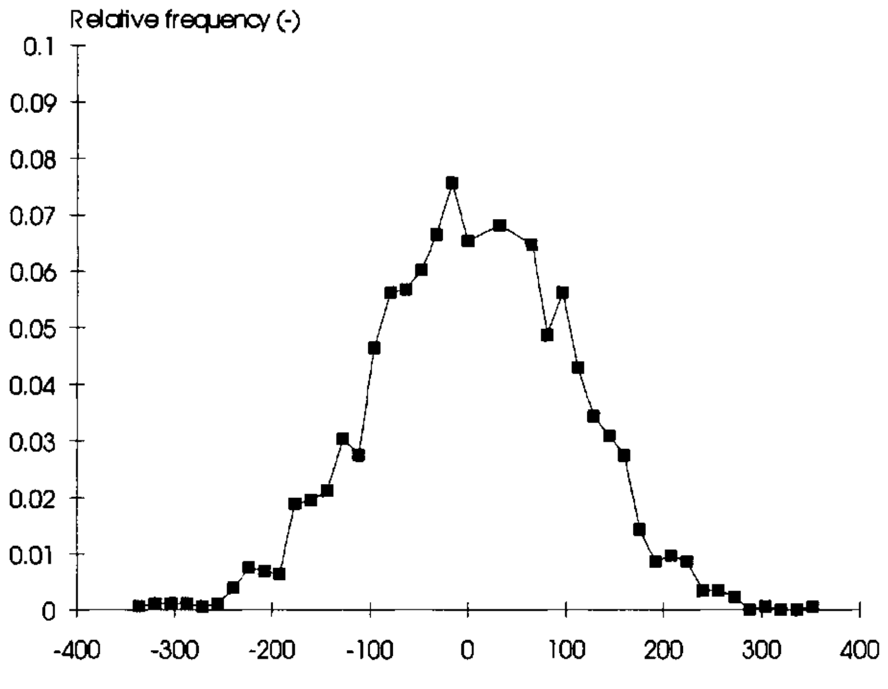
Warning: a normal distribution is not bound by pre-set minimum and maximum values. If values from a normal-type distribution have to be contained between fixed boundaries (as is often the case for model parameter values), a beta distribution with equal A and B values can be used (see Paragraph 2.2).

The input that has to be supplied by the user for drawing from a normal distribution is (per parameter/variable):

- Name of variable(s) for which random values have to be chosen (VNORM)
- Mean μ of the distribution (MEANU)
- Variance σ^2 of the distribution (VARU)



(a)



(b)

Figure 3 Relative frequency distribution (fraction) of randomly drawn values from a normal distribution, using RIGAUS. N=2000. In Fig. 3a, the variance (VARU) of the distribution was 1, in Fig. 3b, it was 100. The mean of the distribution (MEANU) was 0.

2.4 Seed

The seed of a random generator controls the starting point of the generator and determines the reproducibility of the generated values. In RIGAUS, the seed is called ISEED and is used by the function RUNI for uniform distributions. Because RUNI is also called by the functions RGAU and RBET, the same ISEED 'controls' the generation of normal and beta distributions respectively.

The value for ISEED is read from the input file RIGAUS.IN (see Paragraph 5.1). When the supplied ISEED is 0, an integer function TSEED is called in RUNI to generate a seed value. TSEED produces a seed in the range 1-86412 based on the system (computer) time in seconds from midnight. This generated seed value is written to the output file RERUNS.DAT (see Paragraph 5.1). Each time RIGAUS is run with ISEED = 0 in the input file, a new seed is generated and subsequent runs of RIGAUS produce different output. When the results of RIGAUS should be reproducible, any value not equal to 0 can be given for ISEED in the input file RIGAUS.IN. Each run with RIGAUS that uses the same ISEED value produces the same results.

3 Measured data

Random variables are uniformly drawn from a series of measured data using the RUNI function. In the current version of RIGAUS, random values can be drawn simultaneously and independently from five measurement series (five parameters/variables). Measured values can be randomly drawn simultaneously and independently with draws from the statistical distributions. The maximum number of random values is 2000 per parameter/variable.

The input that has to be supplied by the user for drawing from measured data is (per parameter/variable):

- Name of measured variable(s) for which random values have to be chosen (NMVAR)
- Measured data

4 Special provisions for the water balance 'SAHEL'

RIGAUS was especially designed for MC analyses with crop growth and water balance models as developed at CABO-DLO and TPE-WAU. One of the most generally used soil water balance models is SAHEL (Soils in semi-Arid Habitats that Easily Leach; van Keulen, 1975; van Keulen & Wolf, 1986; Penning de Vries & van Laar, 1982; Penning de Vries et al., 1989)). RIGAUS has special provisions for SAHEL on three points:

1. Important input data for this model are three characteristic points on the pF curve: water contents at saturation (WCST), at field capacity (WCFC) and at wilting point (WCWP). In sensitivity and MC analyses, these three parameter values may be varied to study the effect on crop growth. However, these three parameter are correlated, and these correlations should be taken into account when drawing random values. In RIGAUS, empirical relations between WCST-WCFC-WCWP are included and can optionally be used.

2. The soil water content at the start of simulation (WCLI in SAHEL) also depends on the soil moisture characteristic of the soil. With variable values for e.g. WCFC, WCLI can not be a fixed value as is currently used in SAHEL. Therefore, it is suggested to calculate WCLI in SAHEL as a fraction FWCLI of WCFC (the same way as it was defined from WCWP in the 'original' version of SAHEL, van Keulen, 1975)

   WCLI = FWCLI * WCFC

   This way, values for WCST, WCFC and WCWP can be varied without running into problems with a fixed value for WCLI. The variable name FWCLI is automatically recognised in RIGAUS (optionally).

3. Three soil layers are distinguished in SAHEL, and for each layer the variable names WCST, WCFC, WCWP and WCLI have a suffix to identify the layer number (from top to bottom), i.e. WCST1, WCFC1, WCWP1, WCLI1, WCST2,.... WCLI3. In RIGAUS, random values for these variables can optionally be assigned to all three layers. The generated random values are, per variable, the same for all three layers.

The above three options can be implemented when drawing random variable values for SAHEL by setting the control switch ISWI in the input file: ISWI = 1: take into account; ISWI = 0: ignore. When ISWI = 1, the output of RIGAUS meets the input requirements of the version of SAHEL as used in the FORTRAN module L2SU of the MACROS series (van Kraalingen & Penning de Vries, 1990), with one exception: WCLI has to be initialised in the model as indicated at point 2 above (FWCLI1-3 has to be read from the SAHEL-input file instead of WCLI1-3).

4.1 Empirical relations

Measured values of WCST, WCFC and WCWP were used to investigate the correlations among these parameters (Fig. 4). The measurements refer to Dutch soils ranging from coarse sands to heavy clays and peat (Wösten et al., 1987). There was a close relationship between WCWP and WCFC, and between WCST and WCFC, regardless of soil type (except for peat in the WCWP-WCFC relationship). The following quadratic expressions were fitted through the data set:

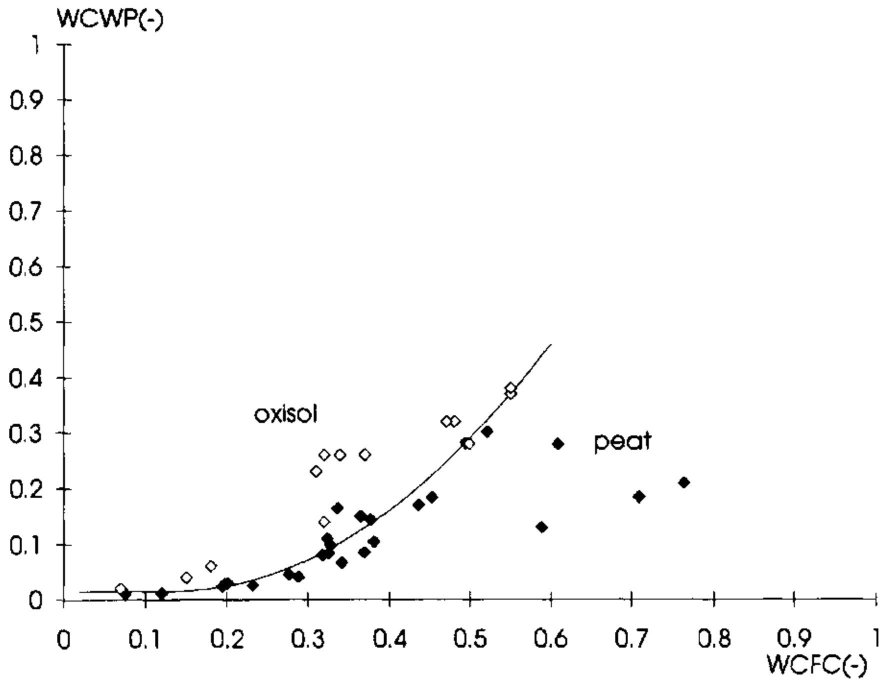
$$WCWP = 0.050 - 0.535*WCFC + 2.027*WCFC^2 \ (cm^3\ cm^{-3}) \qquad [1]$$
(with WCWP minimum = 0.015; see Fig. 4a)

$$WCST = 0.347 - 0.164*WCFC + 1.217*WCFC^2 \ (cm^3\ cm^{-3}) \qquad [2]$$

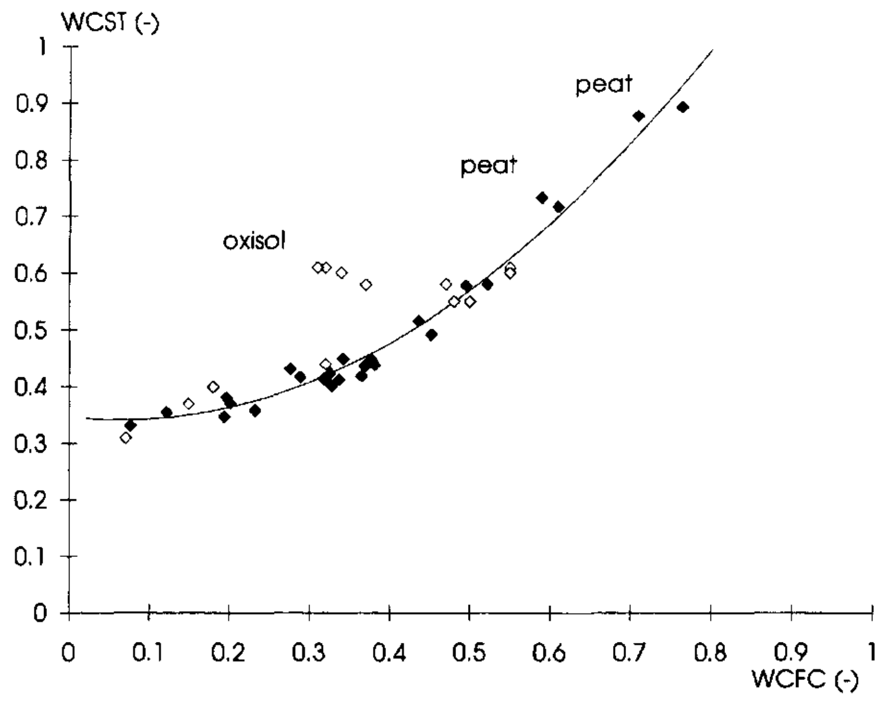
Some statistical information on the regression lines is given in Table 2.

A validation set of various soils in the tropics supported the above relationships, except for deeply weathered oxisols (Fig. 4). For all soil types, the water content at air-dryness, pF 7 (WCAD), was close to 0 and no relationship with the other water contents could be established.

The derived regression equations are only valid between the limits of 0.05 and 0.60 for WCFC. When the user specifies boundaries of WCFC outside these limits, RIGAUS is terminated and sends an error message to the screen (see Paragraph 5.3).



(a)



(b)

Figure 4 Measured values of WCWP versus WCFC (4a) and of WCST versus WCFC (4b). The black diamonds are data from Dutch soils, the white diamonds are data from tropical soils. The drawn lines are the fitted regressions.

Table 2 Statistical information on the regression lines (Equations 1 and 2) derived between WCWP and WCFC and between WCST and WCFC (in cm^3 cm^{-3}).

		A	B	C
1. WCWP = A + B*WCFC + C*WCFC^2	Value	0.050	-0.535	2.027
	Sigma	0.0286	0.1910	0.2930
	T-value	1.87	-2.80	6.91
2. WCST = A + B*WCFC + C*WCFC^2	Value	0.347	-0.164	1.217
	Sigma	0.0182	0.0982	0.1210
	T-value	19.07	-1.66	10.04

		1		2
Number of data (N)	=	30 (without data peat soils)	=	34
Variance accounted for	=	93 %	=	97 %
Mean square residual σ^2	=	0.000993	=	0.000647
Validity limits	:	0.05 < WCFC < 0.60 (cm^3 cm^{-3})	:	0.05 < WCFC < 0.60 (cm^3 cm^{-3})

4.2 Random drawing of WCST, WCFC and WCWP

When the switch ISWI is set to 0, the above relations are ignored in RIGAUS and random values for WCST, WCFC and WCWP can independently be drawn from any of the statistical distributions or from measured data series. Because three soil layers are distinguished in SAHEL, values have to be generated for each of the three layers separately, i.e. WCST1, WCST2, WCST3, WCFC1, WCFC2,...., WCWP3.

When the switch ISWI is set to 1, the WCST-WCFC and WCWP-WCFC relations are included. The user has to specify a statistical distribution for the parameter WCFC, either uniform, beta or normal (**Warning:** random drawing from measured data is not possible in this situation). RIGAUS automatically recognises the variable name WCFC and uses equations 1 and 2 to calculate a corresponding value for WCST and WCWP from each randomly drawn value for WCFC. Variation around these regression lines (Fig. 4) is accounted for by adding a randomly drawn value from a normal distribution with the root mean square residual of the regression lines as standard deviation: in equation 1, $\sigma = 0.032$, in equation 2, $\sigma = 0.025$. An example of 500 generated values of WCST, WCFC and WCWP is given in Fig. 5 where WCFC was drawn from a uniform distribution between 0.05 and 0.60 $cm^3\ cm^{-3}$. The random data accurately reproduced the variation around the regression lines (Fig. 6).

The random values generated for WCST, WCFC and WCWP are assigned to all three soil layers distinguished in SAHEL:

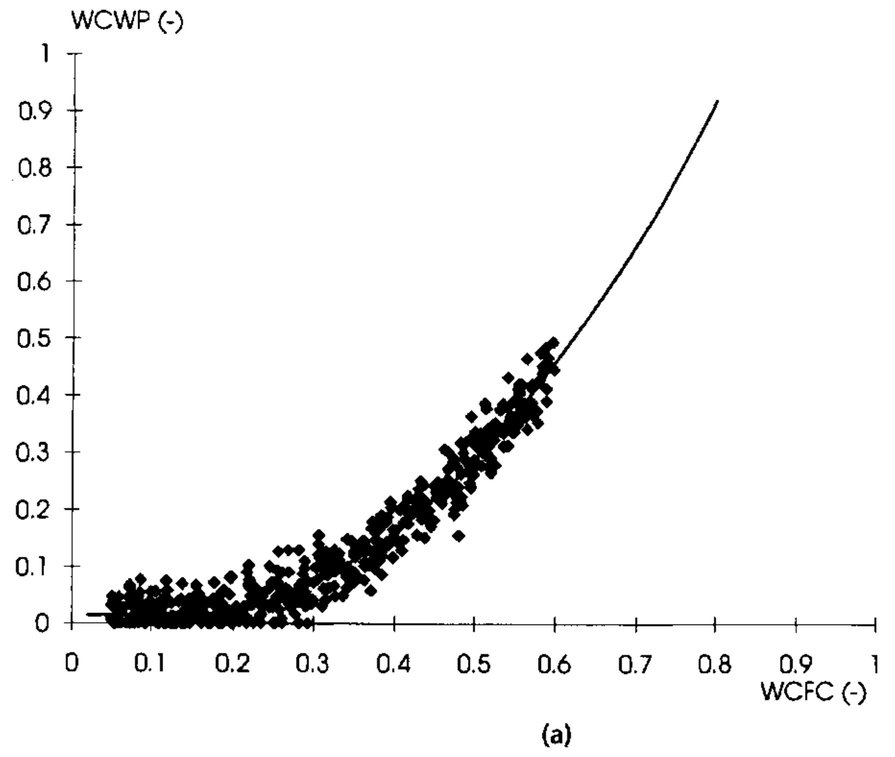
WCST1=WCST2=WCST3, WCFC1=WCFC2=WCFC3 and WCWP1=WCWP2=WCWP3.

In the output file RERUNS.DAT, the generated random values are defined with the above suffixes; in the output file COLUMN.DAT, without (see Paragraph 5.2).

4.3 Random drawing of FWCLI

When the switch ISWI is set to 0, FWCLI is not recognised by RIGAUS as a special variable and is treated as any other variable. Values can be generated from any of the three statistical distributions or from measured values. Because three soil layers are distinguished in SAHEL, values have to be generated for each of the three layers separately, i.e. FWCLI1, FWCLI2 and FWCLI3.

When the switch ISWI is set to 1, FWCLI is automatically recognised by RIGAUS, and randomly generated values from any of the three statistical distribution types are assigned to all three soil layers: FWCLI1=FWCLI2=FWCLI3. In the output file RERUNS.DAT, the generated random values are defined with the above suffixes; in the output file COLUMN.DAT, without (i.e. FWCLI) (see Paragraph 5.2).



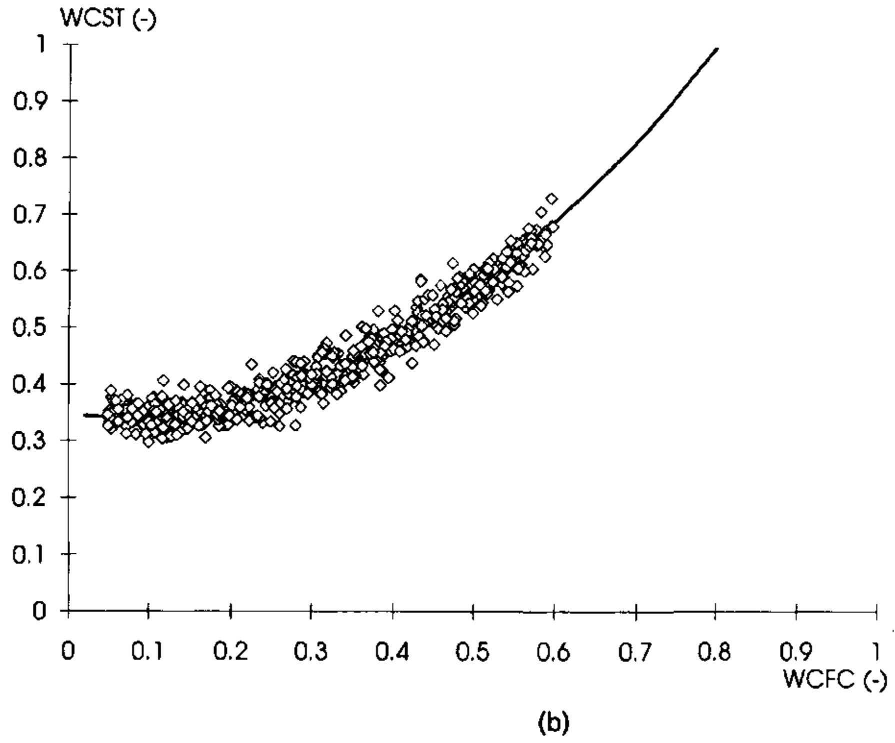
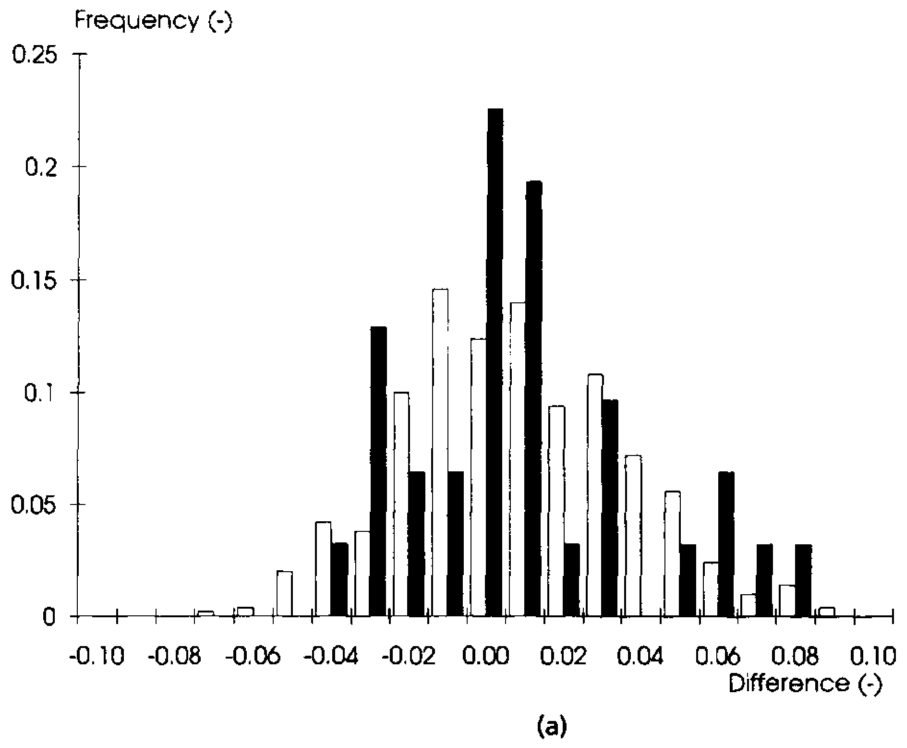


Figure 5 500 randomly drawn variables for WCFC (uniform distribution) and WCWP (5a) and WCST (5b), using RIGAUS. The drawn lines are the regression lines.



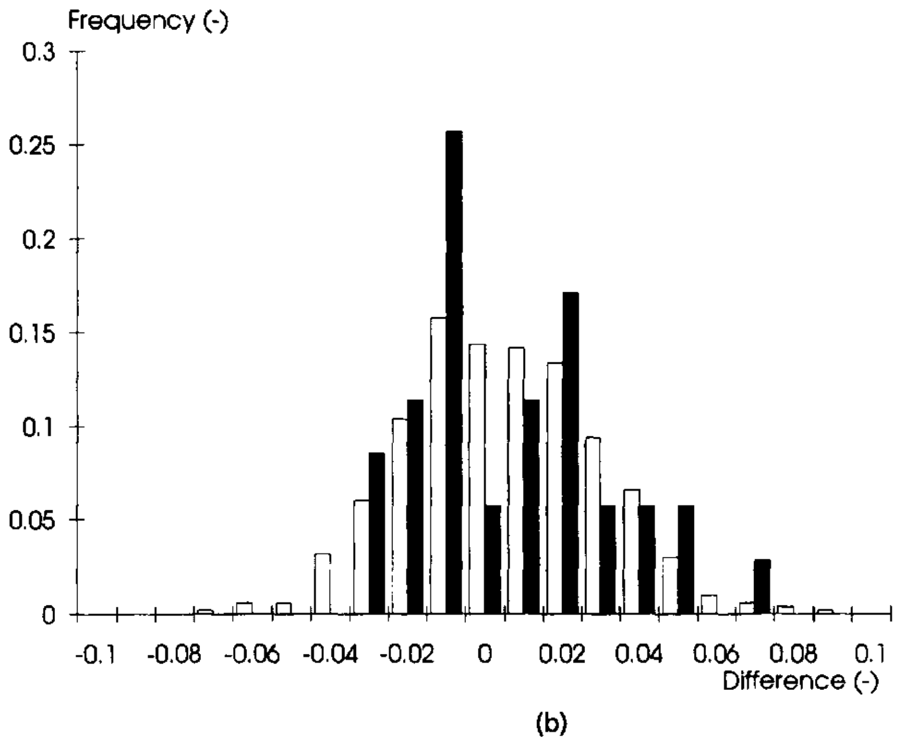


Figure 6 Frequency distribution of the difference between measured values and the regression lines (black bars), and between randomly drawn values and the regression lines (white bars) for WCWP (6a) and WCST (6b).

5 Running RIGAUS

RIGAUS (developed on an 486 IBM compatible PC) is written in the programming language FORTRAN77. A full listing of the source code is given in Appendix 1. Subroutines and functions are called from the CABO/TPE library TTUTIL (Rappoldt & van Kraalingen, 1990), which should be linked when the user changes the source code. The function TSEED uses compiler-specific subroutines. The current RIGAUS program uses the Microsoft compiler, but a provision for the use of a VAX compiler is included in the source code.

The maximum number of draws that can be made, of variables that can be selected from each statistical distribution type, and of measured data per variable are set in the main program:
JMNP = maximum number of draws (=2000)
IMNP = maximum number of variables per statistical distribution type (=10)
KMNP = maximum number of measured data per variable (=500)
The current values (behind brackets) can be reset by the user in accordance with the capacity of the computer on which RIGAUS is run (recompiling and linking using TTUTIL is then needed).

The maximum number of variables that can simultaneously be selected from measured data series is (currently) five and can not be modified by the user without making major changes in the source code of the program.

One input file is needed, RIGAUS.IN, and two output files are generated, RERUNS.DAT and COLUMN.DAT. Examples of these files are given in Appendix 2.

5.1 Program input

The number of random draws, the type of statistical distributions and the measured data series to choose from are specified in the input file RIGAUS.IN. The format of the required input is 'real' (R), i.e. with decimal point, 'integer' (I), i.e. without decimal point, and 'character' (C). The following data have to be supplied.

<u>General</u>

First the switch defining the mode of the program, i.e. whether to include or ignore the special provisions for the water balance model SAHEL, should be set (Chapter 4).
ISWI = 0: special provisions are ignored (I)
ISWI = 1: special provisions are included (I)

The number of random draws should be set.
TND = ... (maximum is 2000) (I)

The seed should be supplied.
ISEED = 0: a seed between 1-86412 will be generated RIGAUS itself
ISEED = 'any integer value': the supplied value is used as seed.

UNIFORM distributions

NDU = ...	Number of variables (maximum = 10) (I)
VUNI = '........', '.........', ...	List of variable names (max. = 10) (C)
UNILO = ... , ... , ... ,	Lower boundary of variable values, in the order of the variables specified above (max. = 10) (R)
UNIUP = ... , ... , ... ,	Upper boundary of variable values, in the order of the variables specified above (max. = 10) (R)

BETA distributions

NDB = ...	Number of variables (maximum = 10) (I)
VBETA = '........', '.........', ...	List of variable names (max. = 10) (C)
ABETA = ... , ... , ... ,	A-value for beta distribution, in the order of the variables specified above (max. = 10) (R)
BBETA = ... , ... , ... ,	B-value for beta distribution, in the order of the variables specified above (max. = 10) (R)
BETALO = ... , ... , ... ,	Lower boundary of variable values, in the order of the variables specified above (max. = 10) (R)
BETAUP = ... , ... , ... ,	Upper boundary of variable values, in the order of the variables specified above (max. = 10) (R)

NORMAL distributions

NDN = ...	Number of variables (maximum = 10) (I)
VNORM = '........', '.........', ...	List of variable names (max. = 10) (C)
MEANU = ... , ... , ...	Mean of the normal distribution, in the order of the variables specified above (max. = 10) (R)
VARU = ... , ... , ...	Variance of the normal distribution, in the order of the variables specified above (max. = 10) (R)

MEASURED data

NDN = ...	Number of variables (maximum = 5) (I)
VNORM = '........', '.........', ...	List of variable names (max. = 5) (C)
MDATA1-5 = ... ' ...... ' ....'	Measured data first to fifth variable (max. = 500) (R)

5.2 Program output

Two output files are generated by RIGAUS: RERUNS.DAT and COLUMN.DAT.

RERUNS.DAT

This file has been formatted to serve as a reruns file in the FSE system. Appendix 3 illustrates the output generated using the input file RIGAUS.IN given in Appendix 2. When the special provisions for the soil water balance model SAHEL are included in the random drawing (ISWI = 1), values for WCST, WCFC, WCWP and FWCLI are generated for all three soil layers (as distinguished in SAHEL) each time the variable names 'WCFC' and 'FWCLI' are encountered

in RIGAUS.IN. All variable values drawn simultaneously, that should serve as one rerun set for the model are separated with the comment line '* This is rerun set x', with 1 < x < 2000.
The seed value (ISEED) is given in the first line of the file for reproducibility of the generated distributions.

All output data (random values) are declared REAL, and formatted in exponential notation E10.3

COLUMN.DAT

In this file, the randomly drawn values are listed in columns per parameter/variable, as illustrated in Appendix 4 that was generated using the input file RIGAUS.IN given in Appendix 2. This file can be used in programs such as GENSTAT or EXCEL for checking and evaluating the data, e.g. to check the generated distributions or the boundary values. When the special provisions for the soil water balance SAHEL have been included in the random drawing (ISWI = 1), values for WCST, WCFC, WCWP and FWCLI are given without suffixes each time the variable names 'WCFC' and 'FWCLI' are encountered in RIGAUS.IN as to avoid redundancy (COLUMN.DAT only serves to check and evaluate the generated results).

All output data (random values) are declared REAL, and formatted in exponential notation E10.3

5.3 Error and warning messages

A number of consistency checks on the input data are incorporated in RIGAUS. When inconsistencies are detected, either fatal error messages are given and the program is aborted, or warning messages are given when the program can still be successfully completed.

What does RIGAUS check automatically?

Input data are checked on the maximum numbers allowed and on consistency. RIGAUS is aborted and fatal error messages are given when:

- The total number of draws (TND) exceeds 2000
- The number of variables for uniform (NDU), beta (NDB) or normal (NDN) distributions exceeds 10
- The number of data for the statistical distributions is inconsistent (e.g. the number of UNIUP values is not the same as that of UNILO values)
- The number of data or the number of variable names for the statistical distributions exceeds 10 (e.g. the number of UNIUP values or VUNI names exceeds 10)
- The number of data or the number of variable names for the statistical distributions is smaller than the number of variables given for random drawing (e.g. the number of UNIUP values is smaller than NDU)
- Supplied values of upper boundaries are lower than supplied values of lower boundaries (e.g. UNIUP < UNILO)
- The number of measured variables (NMV) exceeds 5
- The number of variable names for drawing from measured data is smaller than the given number of measured variables (NMV)
- The number of measured data exceeds 500

Error and warning messages can also be generated by the TTUTIL subroutines that are used in RIGAUS (Rappoldt & van Kraalingen, 1989). E.g. the program is aborted and an error message is given by the 'read' routines when:

- Format of supplied input does not match the defined format (e.g. 'integer' is given when 'real' should be given, or vice versa).

Informative warnings are also given when some inconsistencies are detected but when RIGAUS can still be successfully completed:

- The number of data or the number of variable names for the statistical distributions exceeds the number of variables given for random drawing (e.g. the number of UNIUP values exceeds NDU)

When the special provisions for the soil water balance SAHEL are included in the random drawing (ISWI = 1), checks are carried out on the boundary values of WCST, WCFC, WCWP and FWCLI, and on consistencies among the generated values for these variables. RIGAUS is aborted and fatal error messages are given when:

- Boundary values supplied for WCFC are outside the validity range of the derived relationships with WCST and WCWP, i.e. smaller than 0.05 or larger than 0.60 in all statistical distributions (e.g. UNIUP > 0.60)
- Randomly generated values of WCFC are outside the validity range of the derived relationships with WCST and WCWP, i.e. WCFC smaller than 0.05 or larger than 0.60, in the normal distribution.
- Randomly generated values of FWCLI are smaller than 0 or larger than 1 in the normal distribution.

In RIGAUS, all randomly drawn values for WCST, WCFC and FWCLI are restricted between 0.001 and 0.999.

When the special provisions for the soil water balance SAHEL are ignored (ISWI=0), no consistency checks on the values of WCST, WCFC, WCWP and FWCLI are carried out. Also, no consistency checks are carried out when these variables are randomly drawn from measured values.

What does the user have to check ?

- The format in which the input data are given should match the required format.

- The user carefully has to check the (input) boundary values for drawing from the uniform and beta statistical distributions. The same applies to the measured input data. When random draws are made from a normal distribution, there are in principle no limits on the range of possible values. Therefore, the results (randomly drawn values) have to be carefully checked whether no unrealistic values have been generated.

- It is advisable to check the generated distributions (for shape and minimum and maximum values) of the randomly drawn parameter/variable values before actually using these data for Monte Carlo simulation. Checks can simply be made by graphically plotting the generated values.

- The standard format of the randomly generated parameter/variable values is REAL with exponential notation E10.3. When this format is not compatible with the desired format of a model parameter/variable (e.g. INTEGER data are needed), either the output format in RIGAUS may be adapted, or the format in the simulation model should be converted (e.g. INT and NINT functions to convert REAL data into INTEGER data). Variables in a crop growth simulation model that are often selected for MC analysis are the so-called 'TIMER' variables, e.g. STTIME (start time). In FSE, these variables are REAL, but have no decimal places. Values generated by RIGAUS with three decimal places (E10.3 format) can be truncated to zero decimal places in the simulation model by using the AINT and ANINT functions (see textbooks on programming in standard FORTRAN77, e.g. Balfour & Marwick, 1989)

References

Balfour, A. & D.H. Marwick, 1989. Programming in standard FORTRAN77. Heinemann Educational Books, Oxford. 388 pp.

Bouman, B.A.M. 1993a. Uncertainty in soil and management parameters in crop growth modelling on regional scale. In: Bouman, B.A.M., H.H. van Laar & Wang Zhaoqian (Eds), 1993, Agro-ecology of rice-based cropping systems. SARP Research Proceedings, CABO.DLO, Wageningen, The Netherlands, 58-72.

Bouman, B.A.M. 1993b. A framework to deal with uncertainty in soil and management parameters in crop yield simulation; a case study for rice. In press by Agricultural Systems.

Box, G.E.P. & M.E. Muller, 1958. A note on the generation of random normal deviates. Annal of Mathematical Statistics 29: 610-611.

Bratley, P., B.L. Fox & L.E. Schrage, 1993. A guide to simulation. Springer-Verlag, New York. 397 pp.

Hazelhof, L., A.P.J. de Roo & G.B. Heuvelink, 1990. The use of Monte Carlo simulations to estimate the effects of spatial variability of infiltration on the output of a distributed runoff and erosion model. In: Harts, J., H.F.L. Ottens & H.J. Scholten (Eds), EGIS'90: First European Conference on Geographic Information Systems, Amsterdam, The Netherlands, April 10-13. EGIS Foundation, Utrecht (Faculty of Geographical Services), 442-452.

Keulen, H. van, 1975. Simulation of water use and herbage growth in arid regions. Simulation Monographs, Pudoc, Wageningen, The Netherlands, 176 pp.

Keulen, H. van, & J. Wolf, 1986. Modelling of agricultural production: weather, soils and crops. Simulation Monographs, Pudoc, Wageningen, 479 pp.

Kraalingen, D.W.G. van, & F.W.T. Penning de Vries, 1990. The FORTRAN version of CSMP MACROS (Modules for Annual CROp Simulation). Simulation Report CABO-TT nr. 21. CABO.DLO Wageningen, 145 pp.

Kraalingen, D.W.G. van, 1991. The FSE system for crop simulation. Simulation Reports CABO-TT nr. 23, CABO.DLO, Wageningen, 77 pp.

Kros, J., P.H.M. Janssen, W. de Vries & C.I. Bak, 1990. Het gebruik van onzekerheidsanalyse bij modelberekeningen; een toepassing op het regionale bodemverzuringsmodel RESAM. Rapport 65, Staring Centrum, Wageningen, The Netherlands, 127 pp.

L'Ecuyer, P., 1986. Efficient and portable combined pseudo-random number generators. Commun. ACM.

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

Penning de Vries, F.W.T., D.M. Jansen, H.F.M. ten Berge & A. Bakema, 1989. Simulation of ecophysiological processes of growth in several annual crops. Simulation Monographs, Pudoc, Wageningen, 271 pp.

Press, W.H.B., B.P. Flannery, S.A. Teukolsky & W.T. Vetterling, 1992. Numerical Recipes, the art of scientific computing, Cambridge University Press.

Rappoldt, C. & D.W.G. van Kraalingen, 1989. FORTRAN utility library TTUTIL. Internal report 18, Department of Theoretical Production Ecology, Wageningen Agricultural University, Wageningen, The Netherlands, 61 pp.

Rossing, W.A.H., R.A. Daamen & M.J.W. Jansen, 1993. Uncertainty analysis applied to supervised control of aphids and brown rust in winter wheat. 1. Quantification of uncertainty in cost-benefit calculations. Accepted by Agricultural Systems.

Wösten, J.H.M., M.H. Bannink and J. Beuving, 1987. Waterretentie- en doorlatendheidskarakteristieken van boven- en ondergronden in Nederland: de Staringreeks, ICW report 18, ICW, Wageningen, The Netherlands (in Dutch), 75 pp.

Appendix I: Listing of RIGAUS

```
*=============================================================*
* PROGRAM RIGAUS                                              *
* Authors: B.A.M. Bouman (AB-DLO) & M.J.W. Jansen (GLW-DLO)   *
* Date: November 1993                                         *
* Version: 1.0                                                *
* Purpose: program to draw at random (parameter) values from  *
*     statistical distributions and from measured data sets   *
*     for Monte Carlo simulation with simulation models.      *
*     The distributions are: UNIFORM, BETA and NORMAL.        *
*     Note: no more than 2000 drawings; no more than 10       *
*           variables per statistical distribution type,      *
*           no more than 5 measured variables, and no more    *
*           than 500 measured data per variable.              *
*     The correlation between soil moisture parameters WCFC,  *
*     WCWP and WCST (soil water balance SAHEL) is optionaly   *
*     included (switch ISWI).                                 *
*                                                             *
* PARAMETERS (Type: I=integer, R=real, C=character)           *
*            (Class: I=input, L=local, O=output)              *
* Name      type meaning                              class   *
* ----      ---- ------                               -----   *
* ISWI        I  Switch to take into account the correlation  *
*                between WCFC, WCST and WCWP             I    *
* NL          I  Number of soil layers                   L    *
* TND         I  Total number of drawings                I    *
* ISEED       I  Seed for random generators              I/O  *
* NDU         I  Number of variables for Uniform drawing I    *
* NDB         I  Number of variables for Beta drawing    I    *
* NDN         I  Number of variables for Normal drawing  I    *
* NMV         I  Number of measured variables            I    *
* VUNI(I)     C  Name of variable for Uniform drawing    I    *
* VBETA(I)    C  Name of variable for Beta drawing       I    *
* VNORM(I)    C  Name of variable for Normal drawing     I    *
* NMVAR(I)    R  Name of measured variable               I    *
* UNILO(I)    R  Lower boundary for Uniform distribution I    *
* UNIUP(I)    R  Upper boundary for Uniform distribution I    *
* ABETA(I)    R  Parameter A for Beta distributon        I    *
* BBETA(I)    R  Parameter B for Beta distribution       I    *
* BETALO(I)   R  Lower boundary for Beta distribution    I    *
* BETAUP(I)   R  Upper boundary for Beta distribution    I    *
* MEANU(I)    R  Mean value for Normal distribution      I    *
* VARU(I)     R  Variance for Normal distribution        I    *
* MDATA1-5(I) R  Measured data for variable 1, 2 or 3    I    *
* DRAWU(I,J)  R  Random value from Uniform distribution     O   *
* DRAWB(I,J)  R  Random value from Beta distribution        O   *
* DRAWN(I,J)  R  Random value from Normal distribution      O   *
* DDATA1-5(I) R  Random value from measured data            O   *
*                                                               *
* Subroutines and functions called:                             *
* - from TTUTIL: RDINIT,RDSINT,RDSREA,RDAREA,RDSCHA             *
*                                                               *
* - FUNCTION RUNI, RGAU, RBET                                   *
*                                                               *
* Inputfiles: DRAW.IN                                           *
* Outputfiles: RERUNS.DAT, COLUMN.DAT                           *
*                                                               *
* Error messages: fatal errors in consistency check input data *
*                 messages in consistency check input data      *
*============================================================= *

      PROGRAM RIGAUS

      IMPLICIT REAL(A-H,L-Z)
      IMPLICIT INTEGER(I-K)
      PARAMETER (IMNP=10)
      PARAMETER (JMNP=2000)
      PARAMETER (KMNP=500)
      PARAMETER (INMVAR=5)
      LOGICAL SWI

      CHARACTER*80 FILIN
      CHARACTER*6  VUNI, VBETA, VNORM, NMVAR
      CHARACTER*6 CWCFC, CWCST, CWCWP, CFWCI
      DIMENSION VUNI(IMNP), VBETA(IMNP), VNORM(IMNP), NMVAR(INMVAR)

      INTEGER TND, NDU, NDB, NDN, NMV, NL
      REAL UNILO(IMNP), UNIUP(IMNP), ABETA(IMNP), BBETA(IMNP)
      REAL VARU(IMNP), MEANU(IMNP), BETALO(IMNP), BETAUP(IMNP)
      REAL DRAWU(JMNP,IMNP), DRAWB(JMNP,IMNP), DRAWN(JMNP,IMNP)
      REAL DDATA(JMNP,INMVAR)
      REAL WCST(JMNP), WCWP(JMNP), WCFC(JMNP), FWCLI(JMNP)

      REAL MDATA1(KMNP), MDATA2(KMNP), MDATA3(KMNP)
      REAL MDATA4(KMNP), MDATA5(KMNP)
      INTEGER IDATA1, IDATA2, IDATA3, IDATA4, IDATA5

      IUNITD = 10
      IUNITO = 40
      FILIN = 'RIGAUS.IN'

      NL = 3

      CWCFC = 'WCFC'
      CWCST = 'WCST'
      CWCWP = 'WCWP'
      CFWCI = 'FWCLI'

*----------------------------------------------------------------*
*     Reading data from input file                               *
*----------------------------------------------------------------*
*-----open outputfile to write log messages to
      OPEN(40, FILE='RIGAUS.LOG', STATUS='UNKNOWN')

      CALL RDINIT (IUNITD, IUNITO, FILIN)

      CALL RDSINT ('ISWI', ISWI)
      CALL RDSINT ('TND', TND)
*     Check on number of drawings:
      IF (TND .GT. JMNP) THEN
        WRITE(*,*) 'ERROR: number of drawings', JMNP
        GO TO 100
      END IF

      CALL RDSINT ('ISEED', ISEED)
      CALL RDSINT ('NDU', NDU)
*     Check on number of UNIFORM variables:
      IF (NDU .GT. IMNP) THEN
        WRITE(*,*) 'ERROR: number of UNIFORM variables >', IMNP
        GO TO 100
      END IF

      CALL RDSINT ('NDB', NDB)
*     Check on number of BETA variables:
      IF (NDB .GT. IMNP) THEN
        WRITE(*,*) 'ERROR: number of BETA variables >', IMNP
        GO TO 100
      END IF

      CALL RDSINT ('NDN', NDN)
*     Check on number of NORMAL variables:
      IF (NDN .GT. IMNP) THEN
        WRITE(*,*) 'ERROR: number of NORMAL variables >', IMNP
        GO TO 100
      END IF

*-----Reading data for UNIFORM distribution
      CALL RDACHA ('VUNI', VUNI, IMNP, IVUNI)
      CALL RDAREA ('UNILO', UNILO, IMNP, IUNILO)
      CALL RDAREA ('UNIUP', UNIUP, IMNP, IUNIUP)
*     Check on consistency in supplied number of data values
      IF (IUNILO .NE. IUNIUP .OR. IUNILO .NE. IVUNI
     $    .OR. IUNIUP .NE. IVUNI) THEN
        WRITE(*,*) 'ERROR in data UNIFORM distribution'
```

```
          WRITE(*,*) 'inconsistency in number of data'
          GO TO 100
        END IF
        IF (IVUNI .LT. NDU) THEN
          WRITE(*,*) 'ERROR in data UNIFORM distribution'
          WRITE(*,*) 'number of supplied data < NDU'
          GO TO 100
        ELSE IF (IVUNI .GT. IMNP) THEN
          WRITE(*,*) 'Error in data UNIFORM distribution'
          WRITE(*,*) 'number of supplied data >', IMNP
          GO TO 100
         END IF
        IF (IVUNI .GT. NDU) THEN
          WRITE(*,*) 'Message: in data UNIFORM distribution'
          WRITE(*,*) 'number of supplied data > NDU'
        END IF
*       Check on upper and lower boundary values
        DO 25 J=1,NDU
        IF (UNILO(J) .GE. UNIUP(J)) THEN
          WRITE(*,*) 'ERROR in boundaries UNIFORM distribution:'
          WRITE(*,*) 'in variable no:',J
          GO TO 100
        END IF
25      CONTINUE

*-----Reading data for BETA distribution
        CALL RDACHA ('VBETA', VBETA, IMNP, IVBETA)
        CALL RDAREA ('ABETA', ABETA, IMNP, IABETA)
        CALL RDAREA ('BBETA', BBETA, IMNP, IBBETA)
        CALL RDAREA ('BETALO', BETALO, IMNP, IBETLO)
        CALL RDAREA ('BETAUP', BETAUP, IMNP, IBETUP)
*       Check on consistency in supplied number of data values
        IF (IBETLO .NE. IBETUP .OR. IABETA .NE. IBBETA) THEN
          WRITE(*,*) 'ERROR in data BETA distribution:'
          WRITE(*,*) 'inconsistency in number of data'
          GO TO 100
        END IF
        IF (IVBETA .NE. IABETA .OR. IVBETA .NE. IBETLO
       $     .OR. IABETA .NE. IBETLO) THEN
          WRITE(*,*) 'ERROR in data BETA distribution:'
          WRITE(*,*) 'inconsistency in number of data'
          GO TO 100
        END IF
        IF (IVBETA .LT. NDB) THEN
          WRITE(*,*) 'ERROR in data BETA distribution:'
          WRITE(*,*) 'number of supplied data < NDB'
          GO TO 100
        ELSE IF (IVBETA .GT. IMNP) THEN
          WRITE(*,*) 'Error in data BETA distribution'
          WRITE(*,*) 'number of supplied data >', IMNP
          GO TO 100
        END IF
        IF (IVBETA .GT. NDB) THEN
          WRITE(*,*) 'Message: in data BETA distribution:'
          WRITE(*,*) 'number of supplied data > NDB'
        END IF
*       Check on upper and lower boundary values
        DO 26 J=1,NDB
        IF (BETALO(J) .GE. BETAUP(J)) THEN
          WRITE(*,*) 'ERROR in boundaries BETA distribution:'
          WRITE(*,*) 'in variable no:',J
          GO TO 100
        END IF
26      CONTINUE

*-----Reading data for NORMAL distribution
        CALL RDACHA ('VNORM', VNORM, IMNP, IVNORM)
        CALL RDAREA ('MEANU', MEANU, IMNP, IMEANU)
        CALL RDAREA ('VARU', VARU, IMNP, IVARU)
*       Check on consistency in supplied number of data values
        IF (IVNORM .NE. IMEANU .OR. IVNORM .NE. IVARU
       $    .OR. IMEANU .NE. IVARU) THEN
          WRITE(*,*) 'ERROR in data NORMAL distribution'
          WRITE(*,*) 'inconsistency in number of data'
          GO TO 100
        END IF
        IF (IVNORM .LT. NDN) THEN
          WRITE(*,*) 'ERROR in data NORMAL distribution'
          WRITE(*,*) 'number of supplied data < NDU'
          GO TO 100
        ELSE IF (IVNORM .GT. IMNP) THEN
          WRITE(*,*) 'Error in data VNORM distribution'
          WRITE(*,*) 'number of supplied data >', IMNP
          GO TO 100
        END IF
        IF (IVNORM .GT. NDN) THEN
          WRITE(*,*) 'Message: in data NORMAL distribution'
          WRITE(*,*) 'number of supplied data > NDU'
        END IF

*-----Reading measured data
        CALL RDSINT ('NMV', NMV)
*       Check on number of MEASURED variables:
        IF (NMV .GT. INMVAR) THEN
          WRITE(*,*) 'ERROR: number of MEASURED variables >', INMVAR
          GO TO 100
        END IF
        CALL RDACHA ('NMVAR', NMVAR, IMNP, INVAR)
*       Consistency check on number of supplied variable names
        IF (INVAR .LT. NMV) THEN
        WRITE(*,*) 'ERROR in MEASURED data'
        WRITE(*,*) 'number of variable names < NMV'
        GO TO 100
      ELSE IF (INVAR .GT. NMV) THEN
        WRITE(*,*) 'Message: in MEASURED data'
        WRITE(*,*) 'number of variable names > NMV'
      END IF
      IF (NMV .GE. 1) THEN
        CALL RDAREA ('MDATA1', MDATA1, KMNP, IDATA1)
*       Check on maximum number of measured data
        IF (IDATA1 .GT. KMNP) THEN
          WRITE(*,*) 'ERROR in MEASURED data'
          WRITE(*,*) 'number of data 1st variable >', KMNP
          GO TO 100
        END IF
      END IF
      IF (NMV .GE. 2) THEN
      CALL RDAREA ('MDATA2', MDATA2, KMNP, IDATA2)
        IF (IDATA2 .GT. KMNP) THEN
          WRITE(*,*) 'ERROR in MEASURED data'
          WRITE(*,*) 'number of data 2nd variable >', KMNP
          GO TO 100
        END IF
      END IF
      IF (NMV .GE. 3) THEN
      CALL RDAREA ('MDATA3', MDATA3, KMNP, IDATA3)
        IF (IDATA3 .GT. KMNP) THEN
          WRITE(*,*) 'ERROR in MEASURED data'
          WRITE(*,*) 'number of data 3thd variable >', KMNP
          GO TO 100
        END IF
      END IF
      IF (NMV .GE. 4) THEN
      CALL RDAREA ('MDATA4', MDATA4, KMNP, IDATA4)
        IF (IDATA4 .GT. KMNP) THEN
          WRITE(*,*) 'ERROR in MEASURED data'
          WRITE(*,*) 'number of data 4th variable >', KMNP
          GO TO 100
        END IF
      END IF
      IF (NMV .GE. 5) THEN
      CALL RDAREA ('MDATA5', MDATA5, KMNP, IDATA5)
        IF (IDATA5 .GT. KMNP) THEN
          WRITE(*,*) 'ERROR in MEASURED data'
          WRITE(*,*) 'number of data 5th variable >', KMNP
          GO TO 100
        END IF
      END IF

      CLOSE (IUNITD, STATUS='DELETE')
```

```
*-----------------------------------------------------------------*
*     Opening output files                                        *
*-----------------------------------------------------------------*
*-----Open output file RERUNS.DAT
      OPEN(50,FILE='RERUNS.DAT',STATUS='UNKNOWN')
*-----Open output file COLUMN.DAT:
      OPEN(51,FILE='COLUMN.DAT',STATUS='UNKNOWN')



*-----------------------------------------------------------------*
*     Random drawing, plus writing to output file RERUNS.DAT      *
*     Note: if the switch ISWI is set to 1, then:                 *
*           - When the variable name WCFC is found, the water     *
*           content values at wilting point (WCWP) and at         *
*           saturation (WCST) are calculated and written for      *
*           the three soil layers of SAHEL                        *
*           - When the variable name FWCLI is found, values       *
*           written for all three soil layers of SAHEL            *
*-----------------------------------------------------------------*
      SWI = .FALSE.

*-----Run RUNI(ISEED) as dummy to get ISEED if supplied
*     ISEED in RIGAUS.IN = 0
      DUMMY = RUNI(ISEED)
      WRITE (50, '(A10,I7)') '* ISEED =', ISEED

      DO 50 I=1,TND

*-----Write info on run number to file RERUNS.DAT
      WRITE (50,'(A20,I5)') '* This is rerun set', I

*-------Drawing from uniform distribution
        DO 15 J=1,NDU
          DRAWU(I,J)=RUNI(ISEED)*(UNIUP(J)-UNILO(J)) + UNILO(J)
*         Optionaly: calculate correlated soil moisture contents and
*         fraction initial moisture content of all three layers
*         of the SAHEL water balance.
          IF (ISWI .EQ. 1) THEN
            IF (VUNI(J) .EQ. 'WCFC') THEN
*             Test on upper and lower boundaries WCFC  (limits of
*             derived relationship between the water content params)
              IF (UNIUP(J) .GT. 0.6) THEN
                WRITE(*,*) 'Error; upper boundary WCFC > 0.6'
                GO TO 100
               ELSE IF (UNILO(J) .LT. 0.05) THEN
                WRITE(*,*) 'Error; lower boundary WCFC < 0.05'
                GO TO 100
               END IF
              WCFC(I) = LIMIT(0.001, 0.999, DRAWU(I,J))
```

```
5             WCSTI  = 0.025*RGAU(ISEED) + (0.347-0.164*WCFC(I) +
     $                 1.217*WCFC(I)**2)
              WCST(I)= LIMIT(0.001, 0.999, WCSTI)
              IF (WCST(I) .LE. WCFC(I)) GO TO 5
6             WCWPI  = 0.032*RGAU(ISEED) + (MAX(0.015, 0.050-0.535 *
     $                 WCFC(I)+2.027*WCFC(I)**2))
              WCWP(I)= LIMIT(0.001, 0.999, WCWPI)
              IF (WCWP(I) .GE. WCFC(I)) GO TO 6
              DO 101, K=1,NL
                WRITE (CWCFC(5:5), '(I1)') K
                WRITE (CWCST(5:5), '(I1)') K
                WRITE (CWCWP(5:5), '(I1)') K
                WRITE(50,'(A6,A1,E10.3)') CWCFC, '=', WCFC(I)
                WRITE(50,'(A6,A1,E10.3)') CWCST, '=', WCST(I)
                WRITE(50,'(A6,A1,E10.3)') CWCWP, '=', WCWP(I)
101           CONTINUE
              SWI = .TRUE.
            ELSE IF (VUNI(J) .EQ. 'FWCLI') THEN
*             Test on limits of UNIUP and UNILO
              IF (UNIUP(J).GT.1.0 .OR. UNILO(J).LT.0.0) THEN
                WRITE(*,*) 'ERROR; limits FWCLI out of bounds:'
     $                     ,UNIUP(J), UNILO(J)
                GO TO 100
              END IF
              FWCLI(I) = DRAWU(I,J)
              DO 501, K=1,NL
              WRITE (CFWCI(6:6), '(I1)') K
              WRITE(50,'(A6,A1,E10.3)') CFWCI, '=', FWCLI(I)
501           CONTINUE
            ELSE
            WRITE(50,'(A6,A1,E10.3)') VUNI(J),'=',DRAWU(I,J)
            END IF
          ELSE
          WRITE(50,'(A6,A1,E10.3)') VUNI(J),'=',DRAWU(I,J)
          END IF
15      CONTINUE

*-------Drawing from beta distribution
        DO 20 J=1,NDB
          A = ABETA(J)
          B = BBETA(J)
          DRAWB(I,J)=RBET(A,B,ISEED)*(BETAUP(J)-BETALO(J))+BETALO(J)
*         Optionaly: calculate correlated soil moisture contents
          IF (ISWI .EQ. 1) THEN
            IF (VBETA(J) .EQ. 'WCFC') THEN
*             Test on upper and lower boundaries WCFC (limits of
*             derived relationship between the water content params)
              IF (BETAUP(J) .GT. 0.6) THEN
                WRITE(*,*) 'Error: upper boundary WCFC > 0.6'
                GO TO 100
```

```
               ELSE IF (BETALO(J) .LT. 0.05) THEN
                WRITE(*,*) 'Error; lower boundary WCFC < 0.05'
                GO TO 100
               END IF
              WCFC(I) = LIMIT (0.001, 0.999, DRAWB(I,J))
7             WCSTI  = 0.025*RGAU(ISEED) + (0.347-0.164*WCFC(I) +
     $                1.217*WCFC(I)**2)
              WCST(I)= LIMIT(0.001, 0.999, WCSTI)
              IF (WCST(I) .LE. WCFC(I)) GO TO 7
8             WCWPI  = 0.032*RGAU(ISEED) + (MAX(0.015, 0.050-0.535 *
     $                WCFC(I)+2.027*WCFC(I)**2))
              WCWP(I)= LIMIT(0.001, 0.999, WCWPI)
              IF (WCWP(I) .GE. WCFC(I)) GO TO 8
              DO 102, K=1,NL
                WRITE (CWCFC(5:5), '(I1)') K
                WRITE (CWCST(5:5), '(I1)') K
                WRITE (CWCWP(5:5), '(I1)') K
                WRITE(50,'(A6,A1,E10.3)') CWCFC, '=', WCFC(I)
                WRITE(50,'(A6,A1,E10.3)') CWCST, '=', WCST(I)
                WRITE(50,'(A6,A1,E10.3)') CWCWP, '=', WCWP(I)
102           CONTINUE
              SWI = .TRUE.
            ELSE IF (VBETA(J) .EQ. 'FWCLI') THEN
*             Test on limits of BETAUP and BETALO
              IF (BETAUP(J).GT.1.0 .OR. BETALO(J).LT.0.0) THEN
                WRITE(*,*) 'ERROR; limits FWCLI out of bounds:'
     $                    ,BETAUP(J), BETALO(J)
                GO TO 100
              END IF
              FWCLI(I) = DRAWB(I,J)
              DO 601, K=1,NL
              WRITE (CFWCI(6:6), '(I1)') K
              WRITE(50,'(A6,A1,E10.3)') CFWCI, '=', FWCLI(I)
601           CONTINUE
              ELSE
              WRITE(50,'(A6,A1,E10.3)') VBETA(J),'=',DRAWB(I,J)
            END IF
          ELSE
          WRITE(50,'(A6,A1,E10.3)') VBETA(J),'=',DRAWB(I,J)
          END IF
20      CONTINUE

*-------Drawing from normal distribution
        DO 30 J=1,NDN
          DRAWN(I,J)=RGAU(ISEED)*VARU(J) + MEANU(J)
*         Optionaly: calculate correlated soil moisture contents
          IF (ISWI .EQ. 1) THEN
            IF (VNORM(J) .EQ. 'WCFC') THEN
*             Test on upper and lower boundaries WCFC  (limits of
*             derived relationship between the water content params)
```

```
            IF (DRAWN(I,J) .GT. 0.6) THEN
              WRITE(*,*)'Error; upper boundary WCFC > 0.6'
              WRITE(*,*)'-> choose other mean/variance'
              WRITE(*,*)'-> choose other probability distribution'
              WRITE(*,*)'So far,',I,'random values have been drawn'
              GO TO 100
             ELSE IF (DRAWN(I,J) .LT. 0.05) THEN
              WRITE(*,*)'Error; lower boundary WCFC < 0.05'
              WRITE(*,*)'-> choose other mean/variance'
              WRITE(*,*)'-> choose other probability distribution'
              WRITE(*,*)'So far,',I,'random values have been drawn'
              GO TO 100
             END IF
            WCFC(I) = LIMIT(00.1, 0.999, DRAWN(I,J))
9           WCSTI  = 0.025*RGAU(ISEED) + (0.347-0.164*WCFC(I) +
     $               1.217*WCFC(I)**2)
            WCST(I)= LIMIT(0.001, 0.999, WCSTI)
            IF (WCST(I) .LE. WCFC(I)) GO TO 9
10           WCWPI  = 0.032*RGAU(ISEED) + (MAX(0.015, 0.050-0.535 *
     $               WCFC(I)+2.027*WCFC(I)**2))
            WCWP(I)= LIMIT(0.001, 0.999, WCWPI)
            IF (WCWP(I) .GE. WCFC(I)) GO TO 10
            DO 103, K=1,NL
              WRITE (CWCFC(5:5), '(I1)') K
              WRITE (CWCST(5:5), '(I1)') K
              WRITE (CWCWP(5:5), '(I1)') K
              WRITE(50,'(A6,A1,E10.3)') CWCFC, '=', WCFC(I)
              WRITE(50,'(A6,A1,E10.3)') CWCST, '=', WCST(I)
              WRITE(50,'(A6,A1,E10.3)') CWCWP, '=', WCWP(I)
103         CONTINUE
            SWI = .TRUE.
          ELSE IF (VNORM(J) .EQ. 'FWCLI') THEN
            IF (DRAWN(I,J).GT.1.0 .OR. DRAWN(I,J).LT.0.0) THEN
            WRITE(*,*) 'ERROR; random value FWCLI out of bounds:'
     $                  ,DRAWN(I,J)
            WRITE(*,*)'-> choose other mean/variance'
            WRITE(*,*)'-> choose other probability distribution'
            WRITE(*,*)'So far,',I,' random values have been drawn'
            GO TO 100
            END IF
            FWCLI(I) = DRAWN(I,J)
            DO 701, K=1,NL
            WRITE (CFWCI(6:6), '(I1)') K
            WRITE(50,'(A6,A1,E10.3)') CFWCI, '=', FWCLI(I)
701         CONTINUE
          ELSE
          WRITE(50,'(A6,A1,E10.3)') VNORM(J),'=',DRAWN(I,J)
          END IF
        ELSE
        WRITE(50,'(A6,A1,E10.3)') VNORM(J),'=',DRAWN(I,J)
```

```
          END IF
30      CONTINUE

*-------Drawing at random from measured data
        IF (NMV .GE. 1) THEN
201       ICOUNT = INT(RUNI(ISEED)*IDATA1 + 1.)
          IF (ICOUNT .GE. (IDATA1+1) .OR. ICOUNT .EQ. 0) GO TO 201
          DDATA(I,1) = MDATA1(ICOUNT)
          WRITE(50, '(A6,A1,E10.3)') NMVAR(1), '=', MDATA1(ICOUNT)
        END IF
        IF (NMV .GE. 2) THEN
202       ICOUNT = INT(RUNI(ISEED)*IDATA2 + 1.)
          IF (ICOUNT .GE. (IDATA2+1) .OR. ICOUNT .EQ. 0) GO TO 202
          DDATA(I,2) = MDATA2(ICOUNT)
          WRITE(50, '(A6,A1,E10.3)') NMVAR(2), '=', MDATA2(ICOUNT)
        END IF
        IF (NMV .GE. 3) THEN
203       ICOUNT = INT(RUNI(ISEED)*IDATA3 + 1.)
          IF (ICOUNT .GE. (IDATA3+1) .OR. ICOUNT .EQ. 0) GO TO 203
          DDATA(I,3) = MDATA3(ICOUNT)
          WRITE(50, '(A6,A1,E10.3)') NMVAR(3), '=', MDATA3(ICOUNT)
        END IF
        IF (NMV .GE. 4) THEN
204       ICOUNT = INT(RUNI(ISEED)*IDATA4 + 1.)
          IF (ICOUNT .GE. (IDATA4+1) .OR. ICOUNT .EQ. 0) GO TO 204
          DDATA(I,4) = MDATA4(ICOUNT)
          WRITE(50, '(A6,A1,E10.3)') NMVAR(4), '=', MDATA4(ICOUNT)
        END IF
        IF (NMV .GE. 5) THEN
205       ICOUNT = INT(RUNI(ISEED)*IDATA5 + 1.)
          IF (ICOUNT .GE. (IDATA5+1) .OR. ICOUNT .EQ. 0) GO TO 205
          DDATA(I,5) = MDATA5(ICOUNT)
          WRITE(50, '(A6,A1,E10.3)') NMVAR(5), '=', MDATA5(ICOUNT)
        END IF


        WRITE(50,'(A)')

*-------Writing column names to file COLUMN.DAT
        IF (I .EQ. 1) THEN
          IF (SWI) THEN
            WRITE (51,'(32A10)') (VUNI(J), J=1,NDU),
     $       (VBETA(J), J=1,NDB), (VNORM(J), J=1,NDN),
     $       'WCST', 'WCWP', (NMVAR(J), J=1,NMV)
          ELSE
            WRITE (51,'(32A10)') (VUNI(J), J=1,NDU),
     $       (VBETA(J), J=1,NDB), (VNORM(J), J=1,NDN),
     $       (NMVAR(J), J=1,NMV)
          END IF
        END IF
```

```
*-----------------------------------------------------------------*
*       Writing to output file COLUMN.DAT                         *
*-----------------------------------------------------------------*
        IF (SWI) THEN
          WRITE(51,'(32E10.3)') (DRAWU(I,J), J=1,NDU),
     $    (DRAWB(I,J), J=1,NDB), (DRAWN(I,J), J=1,NDN),
     $    WCST(I), WCWP(I), (DDATA(I,J), J=1,NMV)
        ELSE
          WRITE(51,'(32E10.3)') (DRAWU(I,J), J=1,NDU),
     $    (DRAWB(I,J), J=1,NDB), (DRAWN(I,J), J=1,NDN),
     $    (DDATA(I,J), J=1,NMV)
        END IF

50    CONTINUE

      CLOSE(40)
      CLOSE(50)
      CLOSE(51)

      STOP  'Program RIGAUS successfully finished'
100   STOP  'Program RIGAUS aborted; error status'

      END

**********************************************************************
*     FUNCTION RUNI
*     Uniform(0,1) random generator
*     RUNI - pseudo-random uniformly distributed variate           O
*     ISEED - integer seed                                        I/O
*
*     Modification of UNIFL() by Kees Rappoldt (October 1989)
*     Author: Michiel Jansen, november 1993
*     The modification consists of the addition of the
*     I/O argument ISEED, used to (re)initialize the generator
*
*     At first call with ISEED.NE.0 the absolute value of ISEED
*     is used to seed the generator, negative ISEED is made positive
*     At first call with ISEED.EQ.0, integer function TSEED is called
*     to produce a seed value, passed to ISEED, enabling
*     reproduction of the random sequence if necessary.
*
*     At later calls zero and positive values of ISEED do not
*     disrupt the random sequence, moreover ISEED remains unaltered
*     At later calls, negative values of ISEED, will reinitialize
*     the generator,seeded with -ISEED,which value is passed to ISEED
*
*     UNIFL() is equivalent to RUNI(ISEED) with ISEED.EQ.1122334455.
*
*     The algorithm originates from L'Ecuyer (1986).It is implemented
*     in Bratley et al.,1983,(UNIFL),and in Press et al., 1992 (RAN2)
*     RAN2 implements an additional shuffling,to enhance the
*     generator, shuffling is not done in RUNI() for compatibility
*     with UNIFL().
*
*     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 (...).
*     Press, W., et al. (1992), Numerical Recipes, second edition,
*         Cambridge University Press.
**********************************************************************

      REAL FUNCTION RUNI(ISEED)

*     formal parameter
      INTEGER ISEED
**    local variables
      INTEGER JX,K
      INTEGER TSEED
      DIMENSION JX(3)
      LOGICAL INIT
      SAVE
      DATA INIT/.FALSE./

      IF ((ISEED .LE. -2147483563) .OR. (ISEED .GE. 2147483563)) THEN
         CALL ERROR('RUNI', 'INVALID ISEED')
      END IF
      IF (.NOT.INIT) THEN
*        initialize generator
         IF (ISEED .EQ. 0) ISEED = TSEED()
         IF (ISEED .LT. 0) ISEED = -ISEED
         JX(2) = ISEED
         JX(3) = 1408222472
         INIT = .TRUE.
      END IF

      IF (ISEED .LT. 0) THEN
*        reinitialize generator
         ISEED = -ISEED
         JX(2) = ISEED
         JX(3) = 1408222472
      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)
      RUNI = K * 4.656613E-10
      RETURN
      END


**********************************************************************
*     FUNCTION TSEED
*     produces a seed in the range 1...86412
*     based on the system time (in seconds) from midnight
*     authors: Jacques Withagen and Michiel Jansen
*     date: november 1993
*     warning: time-calculation is compiler-dependent
*
*     VAX compiler calculation of time
*     TSEED = INT(SECNDS(0.))
*     end of VAX compiler specific part
*
*     Microsoft compiler calculation of time
**********************************************************************
      INTEGER FUNCTION TSEED()

      INTEGER TIM(4)
      CALL GETTIM(TIM(1), TIM(2), TIM(3), TIM(4))
      TSEED = 3600*TIM(1) + 60*TIM(2) + TIM(3)
*     end of Microsoft compiler specific part
*     prevent zero seed
      IF (TSEED .EQ. 0) TSEED = 86400
*     scramble (one-to-one, since 86413 is a prime)
      TSEED = MOD(241*MOD(239*TSEED, 86413), 86413)
      RETURN
      END


**********************************************************************
*     FUNCTION RGAU
*     Generates unit normal deviate by Box-Muller method
*     RGAU - pseudo-random standard normal deviate                 O
*     ISEED - integer seed                                       I/O
*     Modification of BOXMUL by Kees Rappoldt, October 1989
*     Author: Michiel Jansen, november 1993
*     The modification consists of the addition of the
*     I/O argument ISEED, used to (re)initialize the
*     basic generator RUNI(ISEED)
*
*     Subroutines and/or functions called:
*      - RUNI
*     Some remarks:
*     Trigonometric function calls could be obviated, as shown
*     for instance in GASDEV() of Press et al. (1992).
*     This should slightly speed up the generator.
*     Not done in RGAU for compatibility with BOXMUL.
*     BOXMUL() is equivalent to RGAU(ISEED) with ISEED.EQ.1122334455.
*
*     References:
*     Box, G.E.P., and M.E.Muller. (1958). A note on the
*         generation of random normal deviates.
*         Ann.Math.Stat. 29:610-611.
*     Bratley,P., B.L.Fox and L.E.Schrage. 1983. A guide to
*         simulation. Springer-Verlag New York Inc. 397 pp.
*     Press, W., et al. (1992), Numerical Recipes, second edition,
*         Cambridge University Press.
*********************************************************************
      REAL FUNCTION RGAU(ISEED)

*     formal parameters
      INTEGER ISEED
**    local variables + function called
      REAL ANGLE,PI,U2,VECTOR,X,Y,RUNI
      PARAMETER (PI=3.14159265)
      LOGICAL NEWSET
      SAVE
      DATA NEWSET/.FALSE./

      IF ((ISEED .LE. -2147483563) .OR. (ISEED .GE. 2147483563)) THEN
         CALL ERROR('RGAU', 'INVALID ISEED')
      END IF

      IF (.NOT.NEWSET) THEN
*        generate random radius vector length and angle
         U2     = RUNI(ISEED)
         VECTOR = SQRT (-2.0 * ALOG(U2))
         ANGLE  = 2.0 * PI * RUNI(ISEED)

*        find the two normal deviates
         X = COS (ANGLE) * VECTOR
         Y = SIN (ANGLE) * VECTOR

*        at first X is returned
         RGAU    = X
         NEWSET = .TRUE.

      ELSE
```

```
*        in previous call X was returned ; now Y
         RGAU   = Y
         NEWSET = .FALSE.
      END IF

      RETURN
      END


**********************************************************************
*     FUNCTION RBET
*     Beta random generator
*     RBET - random variable from beta(A,B) distribution  O
*     A - parameter .GT.0                                 I
*     B - parameter .GT.0                                 I
*     ISEED - integer seed                              I/O
*
*     Author:       Walter Rosssing
*     Date  :       August 1990
*     Modified by: Michiel Jansen
*     Date   :      November 1993
*
*     Purpose:
*     The beta distribution has two parameters, A and B.
*     Mean of the distribution is A/(A+B),
*     variance is AB/[(A+B+1)(A+B)**2].
*     This pseudo random generator is fully based on FUNCTION
*      BETACH in the second edition (1987) of:
*     Bratley, P., B.L. Fox and L.E. Schrage. 1983. A guide
*        to simulation. Springer Verlag. New York. 397 pp.
*     Previous values of A and B are saved in AA and BB
*     which are initialized at impossible values
*     If A and B are not equal to AA and BB
*     the working array CON is (re)initialized.
*     The program has been adapted for the use of the RUNI
*     function for the generation of uniformly distributed
*     variates.
*
*
*     Subroutines and functions called:
*     - from library TTUTIL: ERROR
*     - RUNI
**********************************************************************
      REAL FUNCTION RBET(A, B, ISEED)

*     formal parameters
      REAL A,B
      INTEGER ISEED
*     local variables and used function
      REAL AA, BB, CON(3),U1,U2,RUNI,V,W
      PARAMETER (LN4 = 1.3862944)
      SAVE
      DATA AA/-1./, BB/-1./

      IF ((A.LE.0) .OR. (B.LE.0)) THEN
        CALL ERROR('RBET', 'INVALID ARGUMENTS')
      END IF
      IF ((ISEED .LE. -2147483563) .OR. (ISEED .GE. 2147483563)) THEN
         CALL ERROR('RBET', 'INVALID ISEED')
      END IF

      IF ((A.NE.AA) .OR. (B.NE.BB)) THEN
*        (re)intialize
         AA = A
         BB = B
         CON(1) = AMIN1(A,B)
         IF (CON(1) .GT. 1.) THEN
             CON(1) = SQRT((A + B - 2.)/(2.*A*B - A - B))
         ELSE
             CON(1) = 1./CON(1)
         END IF
         CON(2) = A + B
         CON(3) = A + 1./CON(1)
      END IF
*     generation
10    U1 = RUNI(ISEED)
      U2 = RUNI(ISEED)
      V  = CON(1)*ALOG( U1/(1.-U1) )
      W  = A*EXP(V)
      IF (CON(2)*ALOG(CON(2)/(B+W)) + CON(3)*V - LN4
     $    .LT. ALOG(U1*U1*U2) ) GO TO 10
      RBET = W/(B+W)
      RETURN
100   END
```

Appendix II: RIGAUS.IN

```
******************************************************************
*  RIGAUS.IN: file contains input data for the program     *
*  RIGAUS to draw at random variables from statistical     *
*  and measured distributions (November-1993)              *
******************************************************************
* First, choose if the special provisions for the soil water
* balance model SAHEL have to be taken into account:
ISWI = 1   ! 0=do not take into acount; 1=take into acount

TND = 5  ! Total Number of Drawings (MAXIMUM = 2000)

ISEED = 27426  ! Seed for random drawing

********** UNIFORM ***************************
NDU = 2  ! Number of variables for drawing from UNIFORM
       ! distribution  (MAXIMUM = 10)

* Names of variables for UNIFORM distribution
VUNI = 'WCFC', 'FWCLI', 'RDT', 'SLA'

* Give lower and upper boundary, in sequence of the
* variables specified above.
UNILO = 0.10, 0.0, 5000.0, 0.15
UNIUP = 0.50, 1.0, 25000.0, 0.35

*********** BETA ****************************************
NDB = 0  ! Number of variables for drawing from BETA
       ! distribution (MAXIMUM = 10)

* Names of variables for BETA distribution
VBETA = 'BETA1', 'BETA2', 'BETA3', 'BETA4', 'BETA5',
      'BETA6'
* Give A and B parameters for BETA distribution, in sequence
* of the variables specified above.
ABETA = 1.0, 2.0, 4.0, 6.0, 8.0, 10.0
BBETA = 1.0, 1.0, 1.0, 1.0, 1.0, 1.0
* Give lower and upper boundary, in sequence of the
* variables specified above.
BETALO = 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
BETAUP = 1.0, 1.0, 1.0, 1.0, 1.0, 1.0

*********** NORMAL *****************************************
NDN = 2   ! Number of variables for drawing from NORMAL
```

```
        ! distribution (MAXIMUM = 10)

* Names of variables for NORMAL distribution
VNORM = 'STTIME', 'DTRP', 'NORM3', 'NORM4', 'NORM5'

* Give mean and variance for the NORMAL distribution,
* in sequence of the variables specified above.
MEANU =  150., 12.0, 0.0, 30.0, 60.0
VARU  =  10., 2.0, 100.0, 10.0, 10.0

************* MEASURED DATA ********************************
NMV = 0     ! Number of measured variables (MAXIMUM = 3)

* Names of measured variables
NMVAR = 'MVAR1', 'MVAR2', 'MVAR3'

* Give measured data of the above variables (MAX = 500)
MDATA1 = 1., 2., 3., 4., 5.
MDATA2 = 6., 7., 8., 9., 10.
MDATA3 = 10., 20., 30., 40., 50.
```

Note: this file would give the following 'warning' messages:

```
   Message: in data UNIFORM distribution
   number of supplied data > NDU
   Message: in data BETA distribution
   number of supplied data > NDB
   Message: in data NORMAL distribution
   number of supplied data > NDN
   Message: in MEASURED data
   number of variable names > NMV
   Program RIGAUS successfully completed
```

Appendix III: RERUNS.DAT

```
* ISEED = 27426

 * This is rerun set    1
WCFC1   =  .155E+00
WCST1   =  .285E+00
WCWP1   =  .654E-02
WCFC2   =  .155E+00
WCST2   =  .285E+00
WCWP2   =  .654E-02
WCFC3   =  .155E+00
WCST3   =  .285E+00
WCWP3   =  .654E-02
FWCLI1  =  .880E+00
FWCLI2  =  .880E+00
FWCLI3  =  .880E+00
STTIME  =  .164E+03
DTRP    =  .125E+02

 * This is rerun set    2
WCFC1   =  .211E+00
WCST1   =  .350E+00
WCWP1   =  .272E-01
WCFC2   =  .211E+00
WCST2   =  .350E+00
WCWP2   =  .272E-01
WCFC3   =  .211E+00
WCST3   =  .350E+00
WCWP3   =  .272E-01
FWCLI1  =  .368E-01
FWCLI2  =  .368E-01
FWCLI3  =  .368E-01
STTIME  =  .136E+03
DTRP    =  .132E+02

 * This is rerun set    3
WCFC1   =  .408E+00
WCST1   =  .531E+00
WCWP1   =  .184E+00
WCFC2   =  .408E+00
WCST2   =  .531E+00
WCWP2   =  .184E+00
WCFC3   =  .408E+00
WCST3   =  .531E+00
WCWP3   =  .184E+00
FWCLI1  =  .595E+00
FWCLI2  =  .595E+00
FWCLI3  =  .595E+00
STTIME  =  .156E+03
DTRP    =  .117E+02

 * This is rerun set    4
WCFC1   =  .326E+00
WCST1   =  .428E+00
WCWP1   =  .766E-01
WCFC2   =  .326E+00
WCST2   =  .428E+00
WCWP2   =  .766E-01
WCFC3   =  .326E+00
WCST3   =  .428E+00
WCWP3   =  .766E-01
FWCLI1  =  .461E+00
FWCLI2  =  .461E+00
FWCLI3  =  .461E+00
STTIME  =  .147E+03
DTRP    =  .114E+02

 * This is rerun set    5
WCFC1   =  .194E+00
WCST1   =  .353E+00
WCWP1   =  .203E-01
WCFC2   =  .194E+00
WCST2   =  .353E+00
WCWP2   =  .203E-01
WCFC3   =  .194E+00
WCST3   =  .353E+00
WCWP3   =  .203E-01
FWCLI1  =  .694E+00
FWCLI2  =  .694E+00
FWCLI3  =  .159E+03
DTRP    =  .714E+01
```

Appendix IV:

COLUMN.DAT

```
  WCFC     FWCLI    STTIME   DTRP      WCST     WCWP
.155E+00 .880E+00 .164E+03 .125E+02 .285E+00 .654E-02
.211E+00 .368E-01 .136E+03 .132E+02 .350E+00 .272E-01
.408E+00 .595E+00 .156E+03 .117E+02 .531E+00 .184E+00
.326E+00 .461E+00 .147E+03 .114E+02 .428E+00 .766E-01
.194E+00 .694E+00 .159E+03 .714E+01 .353E+00 .203E-01
```