

SARP Research Proceedings

Software developments in the SARP project: a guide to applications and tools

J.J.M. Riethoven, H.F.M. ten Berge & H. Drenth (Editors)

Bibliotheek AB-DLO
Toerensloeg 65
Postbus 14
6700 AA Wageningen

SARP Research Proceedings - September 1995

**DLO Research Institute for Agrobiological Sciences, Wageningen
WUR-Department of Theoretical Production Ecology, Wageningen
International Rice Research Institute, Los Baños
National Agricultural Research Systems of Asian countries**

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Software

Software developments in the SARP project: a guide to applications and tools / J.J.M. Riethoven, H.F.M. ten Berge & H. Drenth (eds.). - Wageningen [etc]: DLO Research Institute for Agrobiological Sciences and Soil Fertility [etc.] - Ill. - (SARP research proceedings)

Met lit. opg.

ISBN 90-73384-37-0

NUGI 835

Trefw.: decision support systems / rijstbouw.

Cover design: Ernst van Cleef, DUOTONE, Wageningen

Printing: Grafisch Service Centrum Van Gils B.V., Wageningen

Preface

This volume presents two software packages (FSU and MANAGE-N) that were developed within the context of the SARP network, to facilitate the application of crop growth models in rice research. The packages are 'user-interfaces' that provide support in the handling of model inputs and outputs, and in performing model runs for specific, routine-type analyses. By offering these functions, user-interfaces open the world of modelling to a wider audience, much beyond the small group of scientists involved in the actual development of models. The tools thus become available for use by agronomists, breeders, economists, etc., who no longer need to have all the basic skills of modelling. In those cases, however, full model documentation remains essential to clarify the model's founding assumptions and limitations to users.

The emergence of FSU and MANAGE-N is part of a chain. During the early years of SARP, PCSMP (PC version of the Continuous Systems Modelling Program) was the modelling language wherein the first simulation models (MACROS) were developed. In recent years, PCSMP was replaced by two alternatives, each for its own usergroup: FST (FORTRAN Simulation Translator) for a quick and easy start in modelling; and FSE (FORTRAN Simulation Environment) as a general structure for writing models, offering a number of pre-programmed facilities. FST is a precursor to FSE: it translates a compact simulation model into full FORTRAN programs. Advanced programmers skip the first (FST) step and develop their models straight away in FSE, or elaborate FST-generated FORTRAN programs. The more recent SARP rice models are written in FSE: ORYZA_1, ORYZA_0, ORYZA-modules, and ORYZA_W, and can be executed under user-interfaces.

One of the very reasons for switching from PCSMP to FST/FSE was the projected introduction of user-interfaces for the execution of routine tasks. Unlike models expressed in 'true' simulation languages, basic programming codes (e.g., in FORTRAN) can be run as part of larger 'envelope' structures. The combined use of the FST/FSE tandem with a user-interface (also called 'shell') offers the compact programming facilities of 'true' simulation languages, yet enables the easy execution of standard tasks such as regional (multilocation, multi-year) studies, uncertainty and sensitivity studies, optimisation studies, all of which require repeated running of models with a large number of input combinations. The interface thus takes care of communications between the user, the model, the data sets, and the FSE utility routines.

Part I of this volume describes FSU (FSE User Interface), at earlier stages named the 'SARP-shell'. The general layout of the package is explained, and the tasks performed and facilities offered. FSU is written as a generic tool: any FSE-structured model can be successfully executed in this environment. FSU supports the user in preparing and

arranging inputs, performing different types of model studies, and presenting and evaluating output results.

The package includes the module RIGAUS (Random Input Generator for the Analysis of Uncertainty in Simulation) that can be used to investigate the effects of distribution in model parameter values on simulated output. This facility has proven useful in regional studies for dealing with uncertainty and spatial variability, and in plant ideotype design for finding optimum plant characteristics.

Part II describes the MANAGE-N user shell for designing optimum N management strategies in rice, based on the ORYZA_0 model. MANAGE-N facilitates the translation of quantitative soil, weather and genotype information into customised fertiliser management recommendations.

Both the MANAGE-N and the FSU package will be available for the Microsoft® Windows® operating system within the near future. A summary description of support tools associated with FSU and MANAGE-N concludes the volume in Part III. The appendices present details of input formats, technical references, and selected source files.

Wageningen
November, 1995

The editors

Acknowledgements

Research and development of new products are never the work of a single person. Many people have made contributions in one way or another to the software packages that are presented in this volume of the SARP Research Proceedings series.

For FSU (formerly known as the SARP-shell), we would like to acknowledge the individual researchers in the SARP project for useful suggestions, ideas and support. The theme co-ordinators of SARP (P.K. Aggarwal, B.A.M. Bouman, A. Elings, F.P. Lansigan, R.B. Matthews, E.G. Rubia, T.M. Thiyagarajan, and M.C.S. Wopereis) played an active role in the development and current configuration of FSU, ever introducing new features that could be added and were “of utmost importance for my theme”. B.A.M. Bouman and M.J.W. Jansen supplied the RIGAUS tool that is incorporated within FSU. The support of D.W.G. van Kraalingen is much appreciated: he often had to make modifications to the FSE structure in order to enhance the communication between FSU and the models that are programmed in FSE. We hope that it also improved FSE.

The MANAGE-N package was developed under the SARP-project as decision support tool with the aim of optimising fertiliser N recommendations. Many researchers have contributed useful ideas and suggestions, beginning with the very first version of ORYZA_0. Their help and support are much appreciated. We would like to especially thank the following persons for testing MANAGE-N during its earlier days, for the discovery of ‘undocumented features’, and for the excellent suggestions for improvements: Qinghua Shi, T.M. Thiyagarajan, and R. Sivasamy.

We would like to thank the management of IRRI and of all the national institutes participating in SARP for the organisation and hosting of workshops where FSU and MANAGE-N were presented. Much feedback on both products has been received during these workshops. Furthermore, the assistance of the SARP secretariat at IRRI (Lyn Malabanan, Susan Telosa and Perly Villamayor) and the SARP secretariat at AB-DLO (Lettie Berben-Jorritsma, Trix Claassen and Els Nijhuis) is much appreciated.

Microsoft, FoxPro, Excel and MS-DOS are registered trademarks of Microsoft Corporation in the United States of America and other countries.

IBM is a registered trademark of International Business Machines Corporation.

Epson is a registered trademark of Seiko Epson Corporation, Inc.

Hercules is a registered trademark of Hercules Computer Technology.

DeskJet, HP and LaserJet are registered trademarks of Hewlett-Packard Company.

Intel is a registered trademark of Intel Corporation.

Contents

PART I -- FSU

1	Introduction to FSU	1
1.1	General description of FSU	1
1.2	Using this guide	2
1.2.1	Audience	2
1.2.2	How to use this guide	2
1.2.3	What you should read	4
1.3	Conventions	4
1.3.1	Menu bar	5
1.3.2	Menu pads	5
1.3.3	Menu popups	6
1.3.4	Menu options	6
1.3.5	Windows	7
1.4	Requirements	9
1.5	Quick installation	9
1.5.1	Making a safety backup	9
1.5.2	Actual installation	9
1.6	Getting help	10
2	Examples of FSU use	13
2.1	Your first steps	13
2.1.1	Model selection	14
2.1.2	Selecting input	15
2.1.3	Running the model	16
2.1.4	Plotting the model output	17
2.1.5	Leaving the shell	18
2.2	Changing input files	18
2.3	Single runs	20
2.4	Multiple runs	22
2.4.1	Comparison runs	22
2.4.2	Forcing function runs	23
2.4.3	Parameter modification runs	25
2.4.4	Zonation runs	27
2.4.5	Monte Carlo runs	29
3	FSU reference guide	31
3.1	Main Menu	31
3.1.1	File	31

3.1.2 Applications	34
3.1.3 Help	35
3.2 Model/Experiment Studies Menu	37
3.2.1 System	37
3.2.2 Model	40
3.2.3 Input	42
3.2.4 Run	49
3.2.5 Output	50
3.3 Regional Studies Menu	53
3.3.1 System	54
3.3.2 Model	56
3.3.3 Input	58
3.3.4 Run	65
3.3.5 Output	65
3.4 The Item Selector	69
3.5 The File Selector	71
3.6 The plotting utility TTSELECT	71

PART II -- MANAGE-N

4	Introduction to MANAGE-N	75
4.1	General description of MANAGE-N	75
4.2	Using this guide	76
4.2.1	Audience	76
4.2.2	How to use this guide	76
4.2.3	What you should read	78
4.3	Conventions	79
4.3.1	Menu bar	80
4.3.2	Menu pads	80
4.3.3	Menu popups	81
4.3.4	Menu options	81
4.3.5	Windows	82
4.4	Requirements	84
4.5	Quick installation	84
4.5.1	Making a safety backup	84
4.5.2	Actual installation	84
4.6	Getting Help	85
5	The theory of optimising the N application pattern by MANAGE-N	87
5.1	ORYZA_0: the core of MANAGE-N	87
5.1.1	N supply	87
5.1.2	N demand, uptake capacity, and actual crop N uptake	88

5.1.3 N allocation	88
5.1.4 N utilisation by the crop	88
5.1.5 Grain yield	88
5.1.6 Root-shoot ratio	89
5.1.7 Parameter input	89
5.1.8 Versions of ORYZA_0 used in MANAGE-N	89
5.2 Parameter derivation	89
5.3 The three steps toward fertiliser recommendation	90
5.4 Computing the FSV values ('Fit to Match')	90
5.4.1 Input requirements for 'Fit to Match'	91
5.4.2 Representative FSV values	91
5.5 Single run	92
5.5.1 Outputs	92
5.5.2 Input requirements for 'Single Run'	92
5.5.3 About A, B, C, M, and FERTMX	92
5.5.4 Optimisation of N application	93
5.5.5 Input requirements for 'Optimise N-curve'	94
5.5.6 Outputs	94
5.5.7 The numerical optimisation procedure	94
5.6 The final step: split evaluation	97
6 Optimising the N application patterns by MANAGE-N, an example	99
6.1 Prepare required inputs	99
6.2 Compute the FSV values	100
6.3 Examine the results by single run	106
6.4 Optimise the N applications	107
6.5 Evaluate split-dressings	113
7 MANAGE-N Reference guide	117
7.1 Main menu	117
7.1.1 System	118
7.1.2 Method	118
7.1.3 Input	120
7.1.4 Run	130
7.1.5 Output	133
7.1.6 Help	138
7.2 The Item-Selector	139
7.3 The plotting utility TTSELECT	141

PART III -- OTHER TOOLS

8 The FORTRAN Simulation Environment (FSE)	145
--	-----

8.1	Introduction	145
8.2	Integration and time loop	145
8.3	Initialisation of the states and parameters from external data files	147
8.4	The input data files: reading data	148
8.5	The CONTROL.DAT file	149
8.6	The timer file	151
8.7	The model data files	153
8.8	Weather data	153
8.9	Implementing reruns	155
8.10	Output of simulation results	156
8.11	Operation of the model	157
8.12	Errors and warnings from the FSE program	157
9	The FORTRAN Simulation Translator (FST), a simulation language ^{*)}	159
9.1	Introduction	159
9.2	The structure of the model	159
9.3	FST example program simulating logistic growth	160
9.4	Comment lines and FST statements	163
9.5	Rules for FST keywords, variable names, and values	163
9.6	Definition of input values of the model	163
9.7	Operation hierarchy in expressions and FST and FORTRAN functions	165
9.8	FST keywords for output	167
9.9	FST run control keywords	168
9.10	Weather data in FST programs	168
9.11	Rerun facility, the END keyword	169
9.12	FORTRAN subroutines with FST, the STOP keyword	170
10	RIGAUS	173
10.1	Statistical distributions	173
10.1.1	Uniform distribution	173
10.1.2	Beta distribution	174
10.1.3	Normal distribution	175
10.1.4	Seed	179
10.2	Measured data	179
10.3	Running RIGAUS	179
10.3.1	Program input	180
10.3.2	Program output	181
10.3.3	Error and warning messages	182
	References	185

APPENDICES

Appendix 1	Technical reference of FSU	191
Appendix 2	Structuring your model for FSU	195
Appendix 3	Problems with FSU	207
Appendix 4	Standard input files provided with FSU	209
Appendix 5	Problems that may occur with MANAGE-N or FSU	219
Appendix 6	Technical reference of MANAGE-N	223
Appendix 7	Problems with MANAGE-N	227
Appendix 8	Sample input files used by MANAGE-N	229
Appendix 9	Acronyms used in ORYZA_0	237
Appendix 10	Required input for the MANAGE-N system	241
Appendix 11	MANAGE-N parameter derivation and EXCEL® sheet	245
Appendix 12	Description of MANAGE-N input per <i>method</i>	257
Appendix 13	ORYZA_0 differences	277
Appendix 14	FST source of the ORYZA_0 model	285
Appendix 15	Examples of input and output of the RIGAUS program	299

Tables

1.1	Pointers to text you should read about FSU	4
1.2	Software and hardware requirements for FSU	8
4.1	Pointers to text about MANAGE-N	79
4.2	Software and hardware requirements of MANAGE-N.	83
6.1	Results of the 'Fit to Match' runs for the Jinhua 1993 data set	103
9.1	The order of hierarchy of arithmetic operators in FST	165
9.2	FST and FORTRAN intrinsic function lists	166
9.3	Weather variables that can be used in FST	169
10.1	Beta distribution: the relation between shape parameters and mean and variance	175
A9.1	Acronyms used in all ORYZA_0 sources under MANAGE-N	237
A10.1	Acronyms of required MANAGE-N input, arranged by method.	241
A10.2	Acronyms of required input variables and parameters for MANAGE-N	243
A12.1	Required input for the 'Fit to Match' <i>method</i> in MANAGE-N	260
A12.3	Required input for the 'Single Run' <i>method</i> in MANAGE-N	262
A12.4	Required input for the 'Optimise N-curve' <i>method</i> in MANAGE-N	265
A12.5	Required input for the 'Split Evaluation' <i>method</i> in MANAGE-N	267

Figures

1.1	The structure of FSU	1
2.1	An example of a time-coursed graph of multiple state variables	17
2.2	Results of changes in the value of the PRDEL parameter	21
2.3	Simulated and measured time-coursed development stage	23
2.4	Two methods to view effects of DVS on other state variables	24
2.5	Effect of development rate changes on biomass and grain yield	26
2.6	Cumulative frequency plot of WRR (weight of the rough rice)	28
2.7	Examples of changes in STTIME and WCFC on grain yield WSO	30
3.1	The 'File' menu popup and its contents	32
3.2	The 'Debug' window displays the computer configuration	34
3.3	The 'Applications' menu popup	34
3.4	An example of the 'Help Index' screen of FSU	36
3.5	An example of a help topic with cross-references	36
3.6	The 'About FSU' window	37
3.8	The 'Simulation Controls' window	39
3.9	An example of the actual model options that can be selected by the user	41
3.10	An example of some main sections of the model options	41
3.11	The 'Set Definition' window	43
3.12	An example of the status screen, showing the active settings	43
3.13	An overview of the 'Change Parameters' window	45
3.14	The 'Parameter List' window	46
3.15	An overview of the simulate/forcing window	47
3.16	The 'View Tables' menu option shows the contents of the output files	50
3.17	An example of a graph created by TTSELECT	52
3.18	An overview of the 'System' menu pad in the 'Regional Studies' menu	53

3.19	The 'Simulation Controls' window in the 'Regional Studies' menu	54
3.20	An example of the actual model options that can be selected by the user	56
3.21	An example of some main sections of the model options	56
3.22	The 'Set Definition' window in the 'Regional Studies' menu	58
3.23	The 'Enter STTIME' window, a part of the 'Set Definition' window	59
3.24	An example of the status screen, showing the active settings	61
3.25	The Monte Carlo rerun generation screen	62
3.26	The 'View Tables' menu option shows the contents of the output files	65
3.27	The 'Graph Options' screen of the 'Regional Studies' menu	66
3.28	An example of the output from the 'Graph Options' window (multiple runs)	68
3.29	The Item-Selector as used in this example to select weather files	69
3.30	An example of the File Selector	70
3.31	An example of the menu system in the TTSELECT utility	71
4.1	MANAGE-N: a user-interface to facilitate running the ORYZA_0 model	76
5.1	Examples of APCUM curves that can be generated during optimisation.	93
5.2	Speed of convergence toward an optimum parameter set	96
6.1	Changing the value of DATFSV to get a better fit	102
6.2	Results of the 'Fit to Match' method on the seven treatments of the Jinhua experiment	104
6.3	Effect of changes in the shape of the cumulative fertiliser N application curve	108
6.4	Maximum attainable simulated grain yield for a number of fertiliser N input levels	110
6.5	Recommended timings of the relative N application for cultivar Xieyou 10	111
6.6	Optimised cumulative fertiliser N application curves with phenology indicators	112
6.7	Additional income graphs for different costs of 1 kg fertiliser N	113

7.1	The 'System' menu pad and the accompanying menu popup	118
7.2	The 'Method' menu pad and its contents	119
7.3	The 'Input' menu pad and the contents of the menu popup	121
7.4	The first screen of the 'Edit A Crop/Soil File' menu option	121
7.5	The second screen of the 'Edit A Crop/Soil File' menu option	122
7.6	The last screen of the 'Edit A Crop/Soil File' menu option	123
7.7	The 'Edit Datafile Comment Block' window	124
7.8	The 'Edit Timer File' window enables editing of the start date of the simulation	125
7.9	The 'Select Input Files' window	125
7.10	The 'Set Optimisation Parameters' window	126
7.11	The 'Phenology Dates' window	128
7.12	The 'Set Split Evaluation Parameters' window	129
7.13	The 'Start Run' menu option, displayed just prior to a single run	131
7.14	The 'Select Best Matchfactor' window	131
7.15	An example of the best fit graph for FSV1 calculation	133
7.16	The contents of the 'Output' menu popup	134
7.17	The 'Store Output' window	134
7.18	An example of the 'View Tables' menu option	135
7.19	The 'View Optimisation Graphs' window	136
7.20	The 'View Optimisation Results' window	137
7.21	An example of the 'Help Index' screen	138
7.22	An example of a help topic with cross-references to other topics	138
7.23	The 'About MANAGE-N' window	139
7.24	The Item-Selector, used in this example to select weather files	140
7.25	An example of the menu system in the TTSELECT utility	142

8.1	Overview of the generic structure of models that are programmed in FSE	146
8.2	The FSE integration and time loop	147
8.3	General FSE structure for incorporating several subprocesses into a single simulation model	150
10.1	An example of uniformly distributed random value generation	174
10.2	Examples of randomly generated data with beta distributions	177
10.3	Examples of randomly generated values using the normal distribution	178
A11.1	Derivation of the FNCLV parameter	246
A11.2	Derivation of the FNMAXT table	247
A11.3	Derivation of the SST native soil supply table	248
A11.4	Derivation of the nitrogen recovery table RECT	250
A11.5	Derivation of the MXUP1T values	251
A11.6	Derivation of the parameters that define the APCUM validity check	253
A11.7	Examples of input-derivation graphs created by the ORYZA_0.XLS EXCEL® sheet	254
A11.8	Examples of other graphs created by the ORYZA_0.XLS EXCEL® sheet	256
A12.1	An example of the required incident daily global radiation RDT	258

PART I

FSU

1 Introduction to FSU

J.J.M. Riethoven¹

¹ DLO Research Institute for Agrobiology and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

1.1 General description of FSU

FSU, an acronym for FSE User-interface, is an easy-to-use environment for agricultural -- such as crop growth and water balance -- simulation models that are programmed in FSE (FORTRAN Simulation Environment, see van Kraalingen, 1995). The FSU system takes care of communication with selected models. On the input side, FSU provides selection of models and input data files, the setting of model options (as defined by the chosen model), user-defined modification of model parameters, swapping between observed (forced) and simulated state-variables, and the generation of random values (using selected statistical distributions) for chosen parameters. After running a model, FSU provides the user with tabulated output of the model, time-coursed plots of variables, scatter plots, frequency plots, and cumulative frequency plots. Output can be stored for later processing or imported in other applications such as spreadsheets or a GIS (Geographical Information System).

Settings of input can be saved as scenarios and retrieved later on. FSU can be used for validation of (parts of) the models, detailed study of model behaviour, comparison of experimental versus model results, sensitivity/uncertainty and risk analysis, agroecological zonation, and crop rotations.

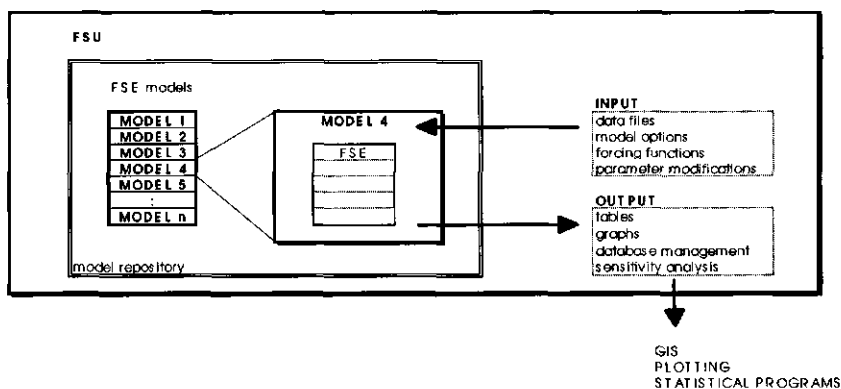


Figure 1.1 The structure of FSU. Models programmed in FSE can be added by the user to the model repository of FSU. Results can be exported and used by other packages.

FSU is developed (Figure 1.1) such that the user can easily add any models and data files to the simulation environment. The only condition is that models should comply with standard FSE input and output specifications.

The part of FSU that is dedicated to the extensive analyses of model behaviour is named 'Model/Experiment Studies'. A chosen model can be run a single time, or several times with slight modifications in chosen parameters or input-tables. Also, state-variables can be either simulated or forced (using observed data as forcing function).

The 'Regional Studies' part of FSU focuses on zonation studies and uncertainty/risk analysis. For this purpose, a chosen model can be run for a large number of inputs: locations, years, cultivars, soil types (and more, depending on the model) to quantify for example site-to-site and year-to-year variation. Furthermore, the embedding of the nifty tool RIGAUSS (Bouman & Jansen, 1993) allows easy access to uncertainty and risk analyses based on Monte Carlo simulation with the chosen model.

A scientific subroutine library is envisaged, consisting of modules each of which describes a process. Various alternative modules can be developed to describe a given process (e.g. potential transpiration); observed time series of relevant variables can replace such modules. Users can then build their models on the basis of well defined and documented elements, depending on their level of observations.

For the database component, the viewpoint is taken that the basic data (input and output) should be stored and made accessible in plain ASCII format. Database facilities of FSU are used, however, to manipulate and prepare input sets and to store and analyse outputs.

1.2 Using this guide

The next paragraphs summarise how you can best use this guide. They describe the manual's intended audience, the audience's prerequisite knowledge, and the manual's organisation and its conventions.

1.2.1 Audience

This manual assumes that you have at least a basic knowledge of crop growth models or simulation in general, since almost all the examples and references refer to crops (and rice in particular). Furthermore, some general computer skills are needed: you should be familiar with keywords like 'keyboard', 'mouse', 'window', and 'disk-drive'.

1.2.2 How to use this guide

An overview of the manual's organisation and content is given in this section. The manual's Chapters are described, followed by a section on how to use each Chapter.

Chapter 1 “Introduction to FSU”

gives some background information on FSU in general and gives a brief overview of the capabilities of FSU. Also explains conventions used in this manual and describes the kind of control structures (like windows, menu bars) that appear on the screen. The third part of this Chapter lists the requirements needed for the use of FSU. The last parts of this Chapter describe the quick installation procedure of FSU and give instructions on how to get help from the authors.

Chapter 2 “Examples of FSU use”

provides a series of examples of case studies for different applications like agro-ecological zonation, model-experiment comparison, sensitivity analysis, etc. The user will be taken step by step through the process of selecting a model to work with, using the model-options screen to activate different parts of the model, creating sets of input files, running the model, and examining the model output in various ways.

Chapter 3 “FSU reference guide”

gives a detailed overview of all available applications, tools, and options in the FSU environment. Menu options and windows are listed in a systematic way, top-to bottom-level.

Appendix 1 “Technical reference of FSU”

contains information on a full-fledged installation of FSU, how to add new models and data files, and how to enable the printing of graphs to printers.

Appendix 2 “Structuring your model for FSU”

contains information on how to prepare models for best interfacing with FSU. This Appendix is vital when the user decides to add new models to FSU.

Appendices 3 “Problems with FSU” and 5 “Problems that may occur with MANAGE-N or FSU”

give information on causes of common problems and how to resolve them. Appendix 5 lists common problems that can arise due to computer systems that are not properly set up or problems specific to the Microsoft® FoxPro® environment in which FSU is programmed. Appendix 3 lists problems specific to the FSU system.

Appendix 4 “Standard input files provided with FSU”

lists all important (data) files of FSU. Contains examples of data files for models when needed to visualise special options that the shell can use.

1.2.3 What you should read

The following table points you to specific Chapters in this manual for information on particular topics. For further documentation and literature, refer to 'References' on page 185.

Table 1.1 Pointers to text you should read about FSU.

For information on	You should read
FSU in general	Chapter 1
installation of FSU	Chapter 1, paragraph 1.5 (and maybe Appendix 1)
using the menu system	Chapter 1, paragraph 1.3
contacting the author of the shell	Chapter 1, paragraph 1.6
in-depth overview of all the options and features	Chapter 3
using FSU as a research/educational tool (examples)	Chapter 2
solving common problems	Appendices 5 and 3
adding models or new data files to the system	Appendix 1
creating new models for use within FSU	Chapter 1 and Appendices 5, 3 and 4

1.3 Conventions

When the manual mentions a 'key', it means a key that you can find on the keyboard of your computer. A 'button' means a button in a window, like the [OK] or [Cancel] button. In this manual, keys and buttons will be enclosed in square brackets []. The text between those brackets should be taken as a whole, not as separate keys. For example, [Backspace] means press the key (or button) defined as backspace, and not the word 'backspace'. Sometimes a combination of two keys is used, as in [Ctrl+X]. This means press the [Ctrl] key and keep it pressed while you press the [X] key.

The names of menus and menu options are enclosed between '....', like for example 'Change Parameters'. Menus and menu options are explained in paragraphs 1.3.1 to 1.3.5.

A window is a square box on the screen that contains options which you can activate or edit (see also paragraph 1.3.5).

Names of parameters, variables, programs (except FSU), data files, etc. are printed in COURIER script to make distinction with normal text easier. The names FSU and 'shell' or 'simulation environment' are interchangeable.

In this manual, you will also encounter words like 'selecting', 'choosing', 'activating', etc. The following section defines those actions.

'Selecting' a menu option is done by moving the so-called 'highlighted' bar on top of a menu option or an item. 'Selecting' is done only by using the [Arrow] keys. When you are sure a certain item or menu option should be activated, then you press [Enter] or the [Spacebar]. 'Activating' an option means that an action will be carried out. This is the same as 'choosing'. An action can be something like running the model or marking an item (for example, an input file) for later use.

'Pressing' and 'pushing' a button means move the highlighting bar on top of the button and press [Enter] or [Spacebar].

FSU is heavily based on menus and windows to provide a user-friendly interface between the models and you. If you have no previous experience with menus, please read the next paragraphs very thoroughly. It should give you a quick understanding on the use of menus, windows, and all kinds of buttons. Operating the menu is easiest if you use a mouse. If you are not the lucky owner of such a comfortable device, some functions related to moving and re-sizing windows are not available.

The FSU menu system consists of the following parts: menu bar, menu pads, menu popups and menu options. The menu system provides a quick and easy-to-use interface with the simulation models that are installed in the shell. Each part of the menu system is described below.

Accessing the menu is described for keyboard only, the mouse variant is a simple point and click.

1.3.1 Menu bar

The menu bar is located along the top of the screen. The menu bar displays menu pads with names; these pads give access to menu popups. The content of the menu bar changes as you access different parts of FSU.

1.3.2 Menu pads

Menu pads appear on the menu bar and display the names of the menu popups. Sometimes, certain menu pads appear dimmed ('ghosted') and cannot be chosen. These menu pads are disabled.

To access the menu bar, press the [Alt] key. One of the menu pads will appear highlighted¹ because it is selected. Press the [Right Arrow] and [Left Arrow] keys to move from one menu pad to another, or use the [Tab] or the hot key. The hot key is a highlighted character and is usually the first letter in the menu pad name.

1.3.3 Menu popups

Menu pads control menu popups. Menu popups are lists of related options. When you choose an option from a menu popup, you are telling FSU what action to take. 'Choose' means to activate a selection (highlighted option) by pressing the [Spacebar] or [Enter]. To display a menu popup, press [Alt] and then press the hot key in the menu pad name. If you are already positioned at the menu bar, you can display a popup by pressing the hot key in the menu pad name. You can also press the [Left Arrow] and [Right Arrow] keys until the menu pad is selected, then press the [Spacebar] to display the menu popup.

Once a menu popup is displayed, you will usually choose an available option, as described in the next section. If you wish to deactivate a menu popup without choosing an option, press [Escape].

You cannot display a menu popup of which the menu pad has been disabled.

1.3.4 Menu options

Menu popups contain options. The options on each menu popup are logically related to the menu pad name. On a single menu popup, options may be further grouped to indicate that they produce similar outcomes. These groups are separated by divider lines.

Some menu options have a [Alt] key shortcut listed next to them on the popup. You can use a [Alt] key combination to choose the menu option without displaying the menu popup. For example, exiting FSU can be done by pressing [Alt+Q]. Sometimes, a menu option appears dimmed and cannot be chosen. This menu option is disabled. Before you choose a menu option, use one of the methods discussed in the previous section to display the menu popup. Once the popup appears, choose an option in one of the following ways:

- Press the hot key for the option
- Use the [Up Arrow] and [Down Arrow] keys to select the desired option, then press [Enter] or double-press the [Spacebar].

When you choose a menu option, an action occurs. A window may open or close, a switch may be set, a dialog may appear or an external command may be executed.

¹ Highlighted text appears in contrast to its surroundings on the screen (lighter or darker) so that it stands out from the surroundings. Highlighting something typically indicates that it is selected or about to be chosen.

1.3.5 Windows

FSU uses windows to communicate with you when extra information is required to perform an action. A window is a rectangular box, possibly containing various items like text fields, buttons, or lists. These items will be explained in detail below. As with the menu pads, you jump from item to item within a window by pressing [Tab] or [Shift+Tab]. Currently selected items will be highlighted. The following items can be found in a window:

Push Button

A push button is defined on the screen by angle brackets <...>. If you press this button, an action will occur immediately. The kind of action is defined by the text placed inside the push button. For example, when you press an [OK] push button, you agree with a certain kind of action (described somewhere else in the window) or you think you have filled in all inputs correctly. This may sound confusing to you right now, but when you read the examples, this will be clear.

You select a push button by jumping to it ([Tab] key). When the push button highlights, press [Spacebar] or [Enter] to activate it. Owners of a mouse activate a push button by positioning the mouse pointer at the push button and then clicking once.

A special push button is the confirmation/continuation button. It is defined by the double angle brackets <<...>>. Press [Ctrl+Enter] to activate it. There is no need to highlight it first. This button will immediately continue with an action. Another special push button is in effect: all windows have a default 'cancel' or 'no' button. This button cancels all actions or choices that you have made in that particular window and restores your latest settings (if any). All windows include this button (it is not especially marked, the [Cancel] button is normally used for this purpose) , press [Escape] to activate it.

Check Box

A pair of square brackets on the screen defines a check box. This type of box toggles an option or setting. If the box has an X in it, the option is turned on, otherwise it is turned off. Press [Tab] to move to the box (it highlights when selected) and press [Spacebar] or [Enter] to toggle.

Radio Button

This type of button is defined by a pair of parentheses followed by a text. These buttons are used when you select an option that is mutually exclusive with other options. Press [Tab] to move to the button and press [Spacebar] or [Enter] to activate it. A bullet • appears between the two parentheses. If another option (within that group of radio buttons) was selected, the bullet disappears from that button.

Input Text Field

For some actions, you must supply FSU with text or numbers. This is done by a text field. It shows as a bar within the window, and may or may not contain text or numbers already. Move to the text field ([Tab] key) and when it highlights, type or edit the values contained within the text field.

List Box

A list is a box within the window that supplies you with a list of items. These may be files, names, parameters, etc. Depending on the type of window, you can perform different actions upon these items. Most windows, however, just expect you to select one of the given items. Press [Tab] until you arrive at the list. Then press the [Up Arrow] or [Down Arrow] keys to scroll through the list. The [End] and [Home] key move to the end and the beginning of the list, respectively. Note that if the number of items is larger than what can appear simultaneously in the list, a so-called scroll bar appears at the right side of the list. This bar can be used only by mouse users. Other users can scroll through the list by using the cursor keys. Select an item in the list by pressing [Spacebar] twice, or [Enter], or double-click with the left mouse button.

Table 1.2 Software and hardware requirements for FSU.

Component	Requirement
Computer	8088/8086 microprocessor (CPU) or higher. A mathematical co-processor (FPU) is recommended.
Operating system	MS-DOS [®] version 3.3 or later. Version 6.0 or later is recommended. To check your version, type VER at the command prompt.
Memory	512 Kb of free RAM memory before starting FSU. You can check this by typing MEM or CHKDSK . The result should be 524288 bytes or more of conventional memory free.
Disk space	3 Mb of disk space to install FSU, another 3 Mb of free disk space to work ² with the shell.
Screen	CGA screen adapter or higher (EGA, VGA, SVGA). Hercules is not recommended.

² FSU (in fact, the models themselves) uses a lot of disk space when performing many reruns. When using the 'Monte Carlo' option in the shell, up to 999 runs can be executed. Each run needs space to store the results.

1.4 Requirements

Before you start to use FSU, or if you encounter any problems, make sure your computer system meets the minimum requirements as specified in Table 1.2.

1.5 Quick installation

This section deals with installing FSU. It is called 'Quick installation' because you do not have to configure many options. FSU will be installed on your system with a minimum of effort, using predefined settings. For the user who prefers more control over the installation (and the settings of the shell), please refer to Appendix 1 'Technical reference of FSU' on page 191.

An installation on an average PC typically takes about 5 minutes.

1.5.1 Making a safety backup

Before you proceed with the installation of FSU, make a backup of your original FSU disks. Please refer to the manuals accompanying your computer on how to do this. Ask a local computer wizard if you are unsure.

Store the original disks at a safe place and use your newly made backup disks to install the FSU system.

1.5.2 Actual installation

Follow these steps to perform a first-time and quick installation on your PC system³:

1. Be sure you have plenty of space on your hard disk (about 3 Mb), FSU extracts with example models, data- and weather files.
2. Disk 1 (of 2) contains a file called `INSTALL1.EXE`. This file creates its own directory named FSU in which it places all necessary files and subdirectories. To install, move to the directory in which you want the new directory FSU to be created. For example, to install the shell in directory `C:\FSU`, enter the commands:

```
C:
CD \
A: INSTALL1.EXE
```

³ If by any chance unexpected problems arise, or you cannot get FSU to work, please refer to Appendix 1, 3, and 5 for solving this.

To install FSU in directory D: \SARP\FSU, enter:

```
D:
CD SARP
A: INSTALL1.EXE
```

Of course, it is assumed that directory SARP exists.

3. After INSTALL1 . EXE finishes, put disk 2 (of 2) in the drive and enter the command

```
A: INSTALL2 . EXE
```

After INSTALL2 . EXE finishes, a directory named FSU has been created. This directory is the main directory for the shell.

4. You have now completed the installation. Change to the FSU directory and execute FSU . BAT to start the shell:

```
CD FSU
FSU.BAT
```

1.6 Getting help

In case of unexpected problems, you can get support for FSU by contacting the author. Before you take this step, read through Appendices 1, 5 and 3 and ask a local computer expert to look into the problems. If the trouble persists, contact the author of FSU at this address:

J.J.M. Riethoven c/o H.F.M. ten Berge & D.W.G. van Kraalingen
DLO Research Institute for Agrobiological Sciences (AB-DLO)
P.O. Box 14
6700 AA Wageningen
The Netherlands

Fax: +31.317.423110
Telex: 75209 abw nl
Phone: +31.317.475962
E-mail: riethoven@ab.dlo.nl
kraalingen@ab.dlo.nl

The author requires specific information before your question can be handled correctly. Include the following information in your correspondence, or have this information ready if you call:

Version number of FSU

The version number of FSU is displayed in the first window that appears on the screen each time you start FSU.

Complete problem description

Include all error messages and symptoms of the problem. If you can consistently reproduce the problem, please provide the exact commands entered and data files/models that differ from the standard distribution set.

Debugging information

FSU has a special window that is accessible through the main 'File' menu. It lists a series of settings and gives general information on the kind of system that you are using. Please provide all information listed in this window exactly.

2 Examples of FSU use

J.J.M. Riethoven¹

¹ DLO Research Institute for Agrobiological and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

In this Chapter, a number of examples are presented to get you acquainted with using FSU (also referred to as the shell). It is a good exercise to execute the examples yourself. The assumption is made that you are already familiar with the menu system of the shell and the conventions used to name menu options and buttons. If not, refer to paragraph 1.3 on page 4 and on. In case that FSU has not yet been installed, do that now (see paragraph 1.5 on page 9).

You start with the easiest mode of using the shell: running a model just once and inspecting the output. When you have gained experience doing this, you continue with multiple runs, comparison runs, the forcing function system, sensitivity analysis, and Monte Carlo runs. In the beginning of this Chapter, every step is described in detail; after the first exercises you are expected to continue without this detailed help. When you need additional help with a certain menu option or window, look it up in Chapter 3, from page 31 onwards. All the exercises will be made with the standard model that is distributed with FSU: ORYZA_WZ 2.1. This model is a slight modification of the ORYZA_W model, a rice growth model for irrigated and water-limited conditions, and is described in Bouman & Wopereis (1994a).

2.1 Your first steps

Before you continue, start the shell. To do this, move to the directory in which FSU has been installed. In our example, it is C:\USR\FSU. To move to that directory, enter the following commands:

```
C:
CD \USR\FSU
```

Then start the simulation environment by entering:

```
FSU.BAT
```

After a few seconds, a window pops up with some basic information on the FSU system. Move to the [OK] button and activate it (press [Enter] or click with the mouse). The 'About FSU' window now closes⁴ and you enter the shell at the main menu.

2.1.1 Model selection

Move to the 'Applications' menu pad in the main menu and activate it. This menu pad is the main entrance to the applications in the shell. Select 'Model/Exp. Studies'. You are presented with a new menu bar. Note that some of the menu pads are dimmed (disabled). The first thing that is required from the user in the shell is selecting a model to work with. Only then are other choices made available. To select the ORYZA_WZ model, move to the 'Model' menu pad and choose the 'Select Model' option. A window appears. Take a good look at this window. It is called the Item-Selector and plays a central role in selecting certain items that FSU should process. In this case, the item is a model. As you can see, one item appears within the list box now: only the ORYZA_WZ model is included in the standard distribution of FSU.

Press [Tab] until you are within the list box. Then press [Enter] to choose the 'ORYZA_WZ.EXE' model (or double-click the mouse). A square-root sign $\sqrt{}$ appears in front of the model, signalling that this item has been selected. Now take a look at the other parts of the selector window. You note an [OK], a [Cancel], and an [Info] button. Press the [Info] button. A window appears, with information about the model (press [Ctrl+F10] to enlarge the window to full-screen size). Close the window by pressing [Escape]. Then move to the [OK] button and activate it. You have selected the ORYZA_WZ model to work with and you return to the 'Model' menu pad.

Some models have special options built-in. These options are submodels that can be turned on or off (for example, a water- or nitrogen balance), or these options might be alternative ways to calculate certain physiological processes. The ORYZA_WZ model offers you the choice of three different water balances: irrigated lowland, rainfed lowland, and rainfed upland. You find the options listed in the 'Select Model options' window, that opens when you choose the 'Select Model options' menu pad. A production environment of your preference can be turned on by moving to it and pressing [Spacebar] or [Enter], or you can point-and-click with the mouse to activate it. A bullet between two parentheses (•) designates that your choice is in effect.

For this particular example, select 'Irrigated Lowland'.

Now return to the 'Model/Exp. Studies' menu (press [OK]).

⁴ If, after the 'About FSU' window, a message appears asking whether you want to retrieve an existing project file, answer [No].

2.1.2 Selecting input

Now that you have chosen a model and a production environment, next you have to select the input files that the model needs. For the ORYZA_WZ model, one crop file, one soil file for either the lowland or upland situation, one weather file, and a so-called timer file are required. The first three files contain data about the specific crop (variety), soil type, and weather parameters. The last file contains some parameters that will influence various kinds of processes during the simulation. For now, it is not important for you to know exactly what these processes are.

FSU stores combinations of input files and model options in so-called 'sets'. A set is a special file that you will use to select a complete configuration of input. Selecting such a set enables you to activate previous settings of model input and options in one step.

To select input files and store them in a 'set', follow the next steps carefully. Move to the 'Input' menu pad and choose 'Create New Set(s) Of Input files'. A screen opens with several buttons and text fields. The following buttons are important right now: [CROP], [SOIL], [WEATHER], and [TIMER]. Cycle through the buttons by pressing [Tab] and take good notice of the highlighted areas. Then move to the [CROP] button and press [Spacebar] or click with your mouse. This will cause the Item-Selector to appear. Press [Tab] until the highlighted text is located within the list of crop files. Then use the [Up Arrow] and [Down Arrow] keys to move to the file ORYZA_WZ.DAT. This is a sample crop data file of *Oryza sativa* cv. IR72, with parameters derived from the IRRI/APPA 1992 dry season experiment at 225 kg N. Select it by pressing [Enter]. Note that a square-root sign $\sqrt{}$ appears if you successfully select the file; if not, try again.

View the contents of the crop file by pressing the [Info] button. A window appears with the contents of the file; press [Ctrl+F10] to enlarge the window to its maximum size. Use the [Arrow] keys to move in the window and to view the complete data. Press [Escape] once to return. Now move to the [OK] button and press it. The Item-Selector closes and you return to the previous window, where you see the name ORYZA_WZ.DAT appearing behind the text 'Current crop file'.

Follow the same procedure to select the file PUDS05.DAT at the [SOIL] button⁵, and the file TORYZ_WZ.DAT at the [TIMER] button. Lastly, you select a weather file. Press button [WEATHER] and move to the file PHIL1.992 and choose it. Remember to watch for the square-root sign. Return to the [OK] button and press it. The text field at the right of 'Current weather file' shows you 'PHIL1.992'. The model will run for year 1992 with weather data collected at the wet-site station 1 (Los Baños) in the Philippines.

⁵ It is important that you do not select soil files other than the PUDxxx.DAT. The PUDxxx.DAT files are for puddled conditions (e.g. irrigated lowland); other soil data files are for non puddled (rainfed upland) conditions.

The last thing that needs to be done is saving this particular set of input files. To do this, move to the [Save Set] button and press it. Enter the text 'MYFIRST' in the text field and press [OK] to really save it as 'MYFIRST.SET'. Because the distribution of FSU already includes a set called 'MYFIRST', a message pops up warning you of the existence of another file called 'MYFIRST.SET'. Answer 'yes' (i.e., press button [Yes]) to overwrite this file with your new set of input files. When you have saved your set, press the [Exit] button to return to the 'Model/Exp. Studies' menu. You have now successfully created a set of input files that can be used to run the model.

The set that you created is automatically selected when you return to the 'Model/Exp. Studies' menu. However, for practice, you will explicitly select the set that you created. To do this, choose 'Select Existing Set(s) Of Input files' from the 'Input' menu pad. A warning message will inform you that a set is already active and will ask you to confirm your choice of selecting a new set. Answer [Yes]. The Item-Selector appears again. Jump to the list and select your set, 'MYFIRST.SET'. Note that the square-root sign appears, indicating that you made a successful choice. If you like, move to the [Info] button and press it. The contents of the set will appear. Press [Escape] once to close the window. Then choose the [OK] button. As you return to the 'Model/Exp. Studies' menu, the status text on the screen is updated. It reflects the current active choices. You should be able to see the name of the model ('ORYZA_WZ.EXE'), the set you selected ('MYFIRST.SET'), what kind of model run is chosen, and the forcing functions (if any) that are active. If no name appears after the text 'Selected set:', you were not successful in activating the set⁶.

2.1.3 Running the model

Now it is time to actually run the model. Move to the 'Run' menu pad and choose option 'Run Model'. The shell asks you if you 'really want to do this'. Answer [Yes]. FSU sets up the ORYZA_WZ model to run with the data files you have chosen. After the model finishes running, you return to the simulation environment at the 'Model/Exp. Studies' menu, and the 'Output' menu pad is automatically selected.

If, by chance, the model does not run and gives an error 'Variable name not in data file', then you have not followed the instructions to the letter: you selected an incompatible production environment and soil type. Remember that lowland production environments need puddled soils as input, the rainfed upland situation does not. Go back and follow the instructions to the letter.

⁶ If so, return to the 'Select Existing Set(s) Of Input Files' menu option and try again.

2.1.4 Plotting the model output

Choose the 'View Graphs' menu option. You now enter a simple plotting program that lets you plot the simulated state variables against each other, for example TIME versus WRR (weight of rough rice) or DVS (development stage) versus LAI (leaf area index).

First of all, you will note the list of available variables that can be plotted. The state variables that are available depend on the model. Let us assume that you would like to see LAI as it is related to DVS.

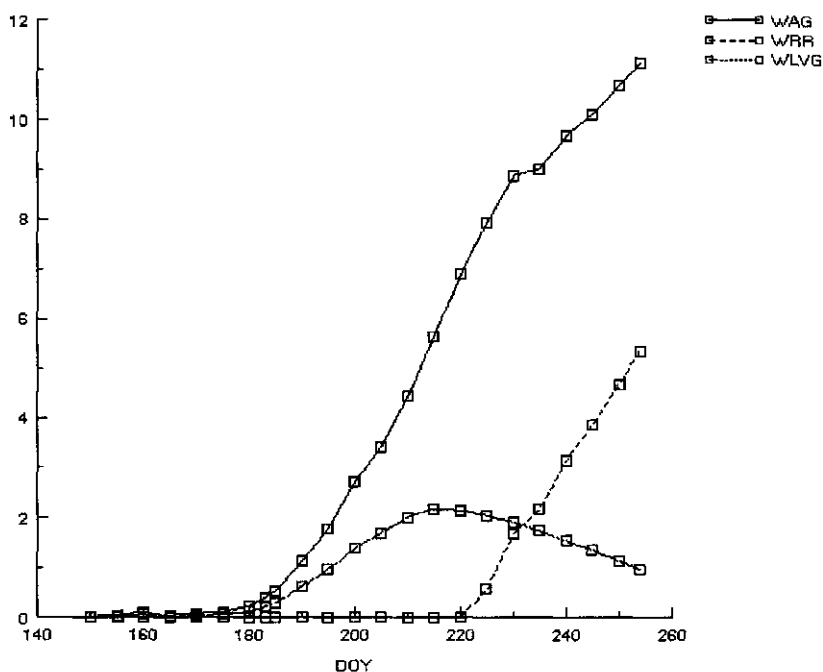


Figure 2.1 An example of a time-coursed graph of multiple state variables. The weights on the Y-axis are expressed in tons ha⁻¹.

Enter

DVS, LAI

when variable names are asked. Press [Enter] and the graph appears. Press [Enter] again to leave the graph. Ignore the next three questions about printing and keep pressing [Enter] until you are again presented with the list of variable names.

Now, first delete the old text and then type

DOY, WAG, WRR, WLVG

to view weight above ground (WAG), weight of rough rice (WRR), and weight of green leaves (WLVG) in the same graph as a function of the day of year DOY (see Figure 2.1). Press [Enter] to view the graph. The weight above ground should reach about 11000 kg. Press [Escape] until you are back at the 'Output' menu popup.

2.1.5 Leaving the shell

You now have had a short introduction to the simulation environment. As mentioned before, this little demonstration is only meant to get you to know the menu system and to realise the ease with which you can run a model and inspect the simulation results. This leaves us to quitting the simulation environment. Move to the 'System' menu pad, and choose 'Quit Shell'. A window appears, asking you if you really want to quit. Answer [Yes]. Next, FSU asks if you want the present settings to be stored. Answer [No]. FSU should return you now to MS-DOS®, at the place where you have started the simulation environment. In our example, at C:\USR\FSU.

2.2 Changing input files

Before you continue with the exercise of running models, you will practise a feature of FSU that can be used to modify data files. Often it is necessary to alter the content of a data file, e.g. when recent research has come up with other values of certain crop parameters or if you need a more precise output, for instance, output every day instead of output once a week (simulation time). In this example, you will practice the edit feature of the shell by creating and then modifying timer files: you will create one timer file for a dry season (sowing date in January) and another for a wet season (sowing date in May) for the IRRI site (Los Baños, Philippines). First of all, change to the directory in which the timer files are placed (you are assumed to be at the MS-DOS®-prompt, see previous exercise):

CD C:\USR\FSU\DATA\INPUT\TIMER

Now enter the following commands:

COPY TORYZ_WZ.DAT JANUAR.DAT
COPY TORYZ_WZ.DAT MAY.DAT

These commands copy the original contents of the timer file to two new files. You will edit these new files later on in this exercise. Now return first to the main shell directory:

CD C:\USR\FSU

And restart FSU:

FSU.BAT

Do not retrieve the old project file. Go to the 'File' menu pad and activate the 'Edit a file' menu option⁷. The File-Selector window (see also paragraph 3.5 on page 69) appears. It shows you the directories or folders that are placed within the main shell directory. Activate the [DATA] directory (move the highlighted bar over the [DATA] item and then press [Enter]. Mouse users can double-click the [DATA] item to do the same. See also paragraph 3.5). Two new directories appear: [SETS] and [INPUT]. Under the [INPUT] directory, you can find the various types of input files that can be used with a model. Activate the [INPUT] directory. Of the next list, select the [TIMER] directory. Now you will get a list of three data files: the original `TORYZ_WZ.DAT` file (do not modify it) and the `JANUAR.DAT` and the `MAY.DAT` files that you just created. Move the bar over the `JANUAR.DAT` file and press [Enter]. A new window will appear with the contents of the data file. Enlarge the window to its full size by pressing [Ctrl+F10]. Change the value of the parameter `STTIME` from 150. to 4. Make sure that you put a decimal dot behind the 4; after editing it should look exactly like this:

STTIME = 4. ! Start time (day) of simulation (sowing)

To store your changes, press [Ctrl+W] or use the mouse to click the little rectangular gadget at the left top of the window. Take a look at the `MAY.DAT` file also, the value of the starting time (=sowing date) should equal 150. You now have created two timer files that can be used to quickly select a model run for a dry season (`JANUAR.DAT`) or a wet season (`MAY.DAT`) at the Los Baños (Philippines) site. This editing procedure can be used for every normal ASCII (text) file that the shell uses (so, also for crop and soil files).

For example, activate again the 'Edit a file' menu option. Go to the [TIMER] directory as you have done in the above exercise. When you see the three timer files listed, move to the [All Files] check button and press it. A cross appears between the square brackets [X], indicating that all files will be listed. You notice an additional file called `RUNCTRL.DTA`. This file contains some default simulation control settings that are used for each model run. Open this file (move the highlighting bar over it and press [Enter]). Enlarge the window by pressing [Ctrl+F10]. One of the parameters listed can be of importance to you. It is the `PRDEL` parameter, which denotes the step size between consecutive outputs.

⁷ In the application menus, the 'Edit a file' menu option can be found in the 'System' menu pad.

Normally, this parameter is set to 5., which means that every 5 simulation days, output (a data point) is written to file. Sometimes, you might want a more precise output: change it to 1. then. With zonation studies, you are not interested in intermediate output, and you can set it to a very large value to speed up computation. For now, leave the value as it is (to 5.). Press [Escape] to leave the window (if you made any changes, answer [Yes] to the question 'discard the changes?').

2.3 Single runs

When you run a model for precisely one environment, year, variety, treatment, etc., then this is considered to be a single run. The model will simulate only one very specific environment, and it will then return to FSU. Examples of these runs are given below.

In the main menu of the shell, activate the 'Model/Exp. Studies' menu. Note that the 'Input', 'Run' and 'Output' menu pads are dimmed. They cannot be selected, simply because you first need to select a model before any of these options have meaning. So, move to the 'Model' menu pad and press 'Select Model'. Choose the `ORYZA_WZ.EXE` model. Now activate the 'Select Model Options' menu option to specify the production environment. Three production environments are listed: irrigated lowland, rainfed lowland, and rainfed upland. Move to the 'Rainfed Upland (non puddled)' option and select it (press [Spacebar], see also paragraph 'Select Model Options' on page 55). Press the [OK] button afterward to activate this setting. You have now successfully selected both a model and the options (in this case, production environment) that belong to the model.

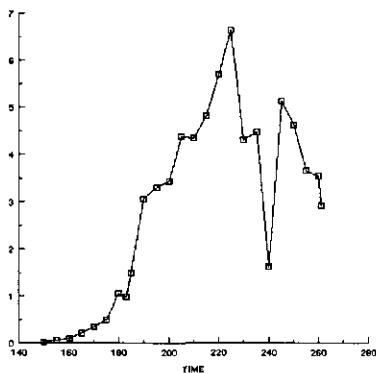
The next step is to create a set of input files. You will do this for both dry and wet seasons in the Philippines. Move to the 'Input' menu pad and select the 'Create Set(s) of Input files' menu option. Like you did in paragraph 2.1.2 (page 15), you will select input files and create a set. Select `ORYZA_WZ.DAT` for the crop file, `LOAM.DAT` for the soil file, `PHIL1.992` for the weather file and `JANUAR.DAT` for the timer file. Now save this set as 'RUDDRY' (rainfed upland, dry season). Move back to the [TIMER] button, press it and select the 'MAY.DAT' file. Save this collection of input files as 'RUWET'. Now press [Escape] to return to the 'Input' menu pad. Select the set 'RUWET.SET' (go to 'Select Existing Set(s) of Input files' and activate it, choose 'RUWET.SET' and press [OK]).

You are now ready to run the model for rainfed upland conditions on a loamy soil in the wet season of year 1992 at Los Baños, Philippines. Run the model (move to the 'Run' menu pad, activate 'Run model', and answer [Yes] to the question 'Do you really want to run the model?'). FSU starts the model for you, and it will take a few seconds to a minute (depending on your computer) before the model finishes. You will return to the shell in the 'Output' menu.

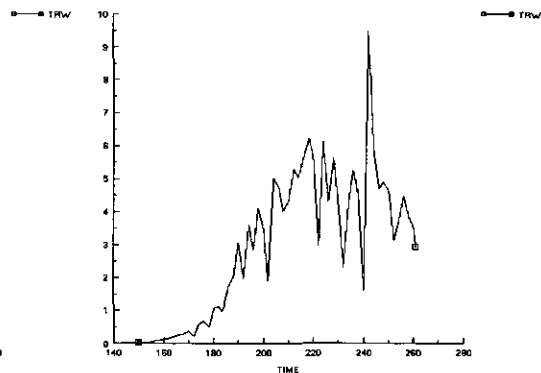
There are several options open to you now. You can store the simulation results, view the results in tabular format, or plot the output. Do not store the output right now but proceed to 'View Tables' and activate it. You are presented with the results of the simulation in tabular format. The first column is always the **TIME**, the other columns are output

specifically defined in the model itself. Note that every 5 days, there is a row of results. This is exactly what the PRDEL parameter (see end of paragraph 2.2) is about.

Press [Escape] to go back to the menu. Now activate the 'View Graphs' option. Make some graphs as you did in the first exercise, e.g. DOY versus WRR, WAG, WST, and WLVG; DVS versus LAI, DOY versus RAINCU (cumulative rainfall). Take good note of the number of data points that are displayed in the graph of TIME versus TRW (actual transpiration rate). Suppose that you want to monitor the fluctuations in TRW at the end of the season in more detail. You can facilitate that by changing the PRDEL parameter in the RUNCTRL.DTA file to a lower value, say 2. To do so, leave the graph utility by pressing [Escape]. Now move to the 'System' menu pad and activate the menu option 'Edit a file'. Go to the [TIMER] directory (see previous exercises), enable [X] [All Files] and open the RUNCTRL.DTA file. Enlarge the window (press [Ctrl+F10]) and change the value of PRDEL from 5. to 2. (remember to enter the decimal dot .). Save this change by pressing [Ctrl+W] and run the model again. When the model finishes, plot again TIME versus TRW. The graph shows not all the data points because there are too many, but the graph is being plotted more fluently (now data is known for every 2 days, see Figure 2.2). Leave the graph utility and check out the tables (activate the 'View Tables' option) and examine the time interval. Set the value of PRDEL back to 5. ('Edit a file' menu option). That concludes the example about single runs.



(2.2a)



(2.2b)

Figure 2.2 Results of changes in the value of the PRDEL parameter. Both plots show the same data; Figure 2.2a however is created using an output time-step of 5 days. That large a step cannot show all the short-time fluctuations that occur in the actual transpiration TRW. The second Figure 2.2b represents those fluctuations better, at an output time-step of 2 days. Note the difference in scale of the Y-axis.

2.4 Multiple runs

Beside single runs, there is also the option to perform multiple runs of the same model. These runs use the so-called rerun facility of the models; there is no need to go into detail about this right now (see Chapter 3 and Chapter 8). The multiple runs are most often used to run the model for different sites, years, crops, soils, pest/diseases and to compare these different runs in a convenient way with each other. Also, the model can be run for a range of values for a certain model parameter to examine the output (manual model calibration, sensitivity analysis) or to perform so-called Monte Carlo runs that can be helpful in doing, for example, a risk analysis.

The next section will deal with comparison runs, forcing function runs, zonation runs, parameter modification runs, and will give a very brief introduction to Monte Carlo runs.

2.4.1 Comparison runs

These types of multiple runs are used when you want to compare the simulation results of different model options (e.g., the production environments in ORYZA_WZ) or when the input files differ completely (different treatments for example). Generally, comparison runs are facilitated by choosing more than one set in the 'Select Existing Set(s) of Input files' menu option. Comparison runs can only be done within the 'Model/Exp. Studies' application menu. You will execute a very simple example: the comparison of the simulation results of two production environments in the wet season. First go to the 'Model/Exp. Studies' menu. Then select the 'ORYZA_WZ' model again. Set the production environment to 'Irrigated Lowland' via the 'Set Model Options' menu option. Now create a set with the following data files: ORYZA_WZ.DAT as a crop file, PHIL1.992 as a weather file, PUDS05.DAT as the soil file, and MAY.DAT as the timer file. Save this set as 'IRLOWWET'.

Select both the 'IRLOWWET.SET' and the 'RUWET.SET' sets of input files. Run the model. You will notice that the model notifies you of the existence of one so-called rerun set. In this case, the rerun set is the rainfed upland production environment and contains the required switches to select a different water balance in the model. After the model finishes, activate the 'View Tables' menu option and examine the results. You note that for both runs the information is stored. To make life easier, comparison of the simulation results also can take place in the 'View Graphs' menu option. Activate it. Besides a list of variable names, the plotting utility also displays the numbers of the runs: 0 and 1. Run 0 is always the default run, run 1 and further are runs in which something has changed with regard to input. In this case, a model switch has been changed. Plot DOY versus WRR (weight of rough rice). On the question 'All runs?' press [Enter] because you want both runs plotted. Answer the next question 'Only end of run values?' with 'N' (press [Enter]), because you like to compare the state variable WRR through time. The first run (run number 0, irrigated lowland) gives slightly better yield (5.4 tons) and is harvested about 10

days earlier than the second run (run number 1, rainfed upland) which yields about 5.0 tons of rough rice. Plot some other variables also. Answer the question 'All runs?' with 'N' (no) and specify a run number to view only one of the model runs. After you finish, leave the graph program by pressing [Escape].

2.4.2 Forcing function runs

If a model supports it, special forcing function runs can be made. With these runs, state variables can transparently either be simulated or measured values can be used as forcing function. The ORYZA_WZ model distributed with FSU has been especially prepared to allow DVS (development stage) as a forcing function. In the crop data file ORYZA_WZ.DAT, hypothetical measured data for DVS has been inserted. These

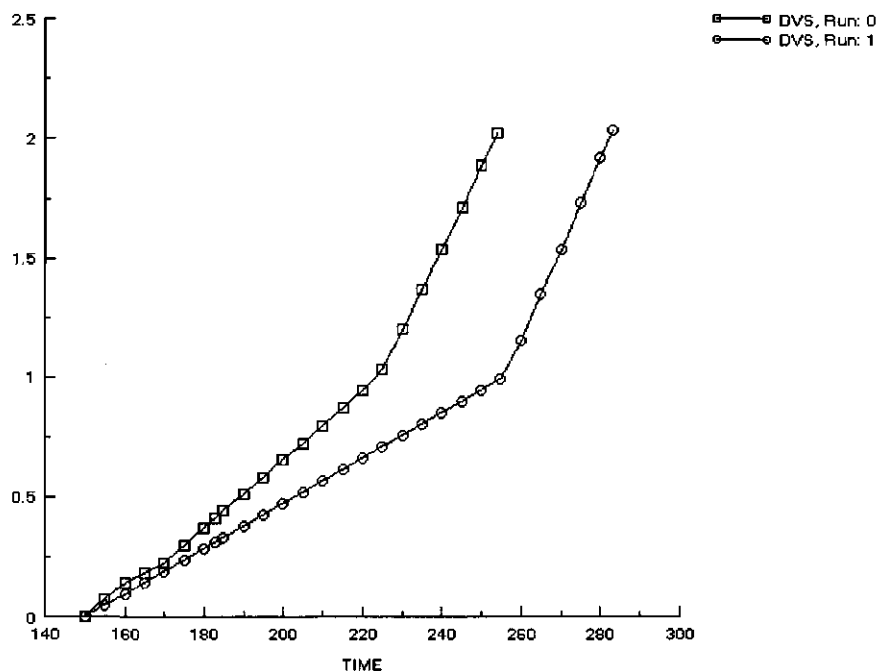


Figure 2.3 Simulated and measured time-coursed development stage. In run 0, the DVS was simulated; in run 1, fabricated measured DVS data were used as forcing function.

'observed' DVS data are purely meant as a demonstration of the forcing function system and should not be used for serious simulation.

In the 'Model/Exp. Studies' menu, select the set 'MYFIRST.SET' again. Now activate the menu option 'Switch Simulated/Forced'. FSU will search for observed data in all the input

files. It will find only one variable though: DVS. In the 'Change Forcing Functions' window that opens, move with [Tab] to the [Method] popup (the rectangular box that reads 'Simulate'). Press [Spacebar] when you highlight it, and then move with the [Arrow] keys to 'Force' and press [Enter]. The [Method] popup now displays 'Force'. Moreover, the extension after the state variable DVS in the left-side box indicates ('FRC') that the forcing function is in effect. Go to the [OK] button and press it to return to the 'Model/Exp. Studies' menu. Run the model. The model runs twice: once in its normal state (i.e. simulate DVS) and once using the observed values of DVS in the calculations. Since DVS has a profound effect on many processes, you will see a noticeable effect on the simulation results. Enter the 'View Graphs' menu option. Plot TIME against DVS to check out the differences in a quick glance (Figure 2.3).

You can verify that DVS has really been forced by plotting it together with DVS_OBS. For run 1, the lines of DVS and DVS_OBS will coincide.

The effects of a longer vegetative and a shorter reproductive phase can be viewed when you plot, for example, WSO (weight of the storage organs) or LAI (leaf area index) versus TIME. You can also plot them against DVS when you are not interested in differences in timing but only in absolute differences at the various values of the development stage (see Figure 2.4).

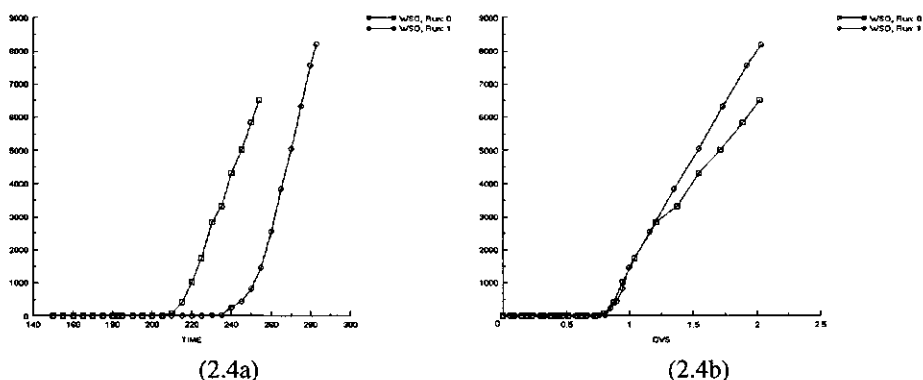


Figure 2.4 Two methods to view effects of DVS on other state variables. In the above graphs, the weight of the storage organs is first plotted against TIME (Figure 2.4a) to make timing differences noticeable. Figure 2.4b shows the absolute differences in WSO at the same development stage. Simulated DVS was used in run 0; fabricated measured values of DVS were used as a forcing function in run 1.

2.4.3 Parameter modification runs

Modifications of parameters are often used to perform simple sensitivity analyses or to compare the effect that changed values of the parameters have on the simulation results. Three examples are presented in this paragraph, ranging from a simple change in one parameter to a systematic modification and combination of three parameters over a certain range of values.

Starting with the first example, go to the 'Model/Exp. Studies' application menu. Select the ORYZA_WZ model and choose the 'RUWET.SET' set that you created earlier in these exercises. Note that the 'Change Parameters' menu option is now enabled. Select this option. After a few moments, you are presented with a list of all available input parameters that you can alter. Select only the parameter RGRL (relative growth rate of the leaves) and press [OK]. A new window opens in which you can alter the values of the parameter. You must change the value in the text field in the centre of the window, just right of the text 'Enter value'. Change the value into 0.7000E-02, move to the button [Store this run], and press it. Press [Definitions Ready] to signal that you are through with changing parameter values. An overview window appears, showing you the additional runs you defined. Press [OK].

Now run the model. The model will run twice, once for the default value of parameter RGRL (as defined in the input files) and once with a value of 0.07 for RGRL. View the tabular output and then proceed to the graphs. Make plots of DOY versus LAI (leaf area index), DOY versus WLVG (weight of the green leaves), and DOY versus WST (weight of the stem) and other plots that you are interested in.

Return to the 'Model/Exp. Studies' menu and select again the set 'RUWET.SET'. Selecting this set will clear the parameter modifications you just made⁸. Activate 'Change Parameters' again and select STTIME (starting time = sowing date) as the parameter to change. In the 'Change Parameters' window check the [X] [Range] button (move to it and press [Spacebar]). Enter 180 as upper limit and 140 as lower limit. Enter 5 as step size. Press [Store This Run] and then [Definitions Ready]. Press [OK] in the overview window. Now run the model. The shell will prepare things in such a way that the model runs with values of STTIME ranging from 140 to 180 with step size 5. In total, 10 runs will be made, of which 9 reruns. Remember that when you use the 'Change Parameters' menu option, the first run will always be the standard/control run (with the values of the parameter as defined in the input file). In this case, the very first run (run number 0 in the plotting utility) will be done with STTIME = 150. View the graphs, for example DOY versus WRR for all runs except run 0.

⁸ It is possible to clear parameter modification without selecting the same set again. Activating 'Change Parameters' for the second time gives you the option to select new parameters. See also Chapter 3, paragraph 'Change Parameters' on page 44.

The last example performs a simple sensitivity analysis. Assuming that the development rates for the different growth phases are not correlated, you will inspect the impact of changes in the four development rates on the simulation results. Select again the same set 'RUWET.SET' and then activate the 'Change Parameters' menu option. Choose the following parameters: DVRI (development rate during photoperiod-sensitive phase), DVRJ (development rate during juvenile phase), DVRI (development rate during panicle development) and DVRR (development rate during reproductive phase) and press [OK]. For the parameters DVRI, DVRJ and DVRI, go to the 'Range' button and press the [Spacebar]. Enter the following values: 0.8E-3 (upper limit), 0.7E-3 (lower limit), and 0.5E-4 (step size) for the parameters DVRI and DVRJ. For the DVRI, parameter enter 0.2E-2 (upper limit), 0.1E-2 (lower limit), and 0.5E-3 (step size).

The above procedure instructs FSU to prepare a so-called reruns file that contains all combinations of parameter values. The model will systematically run with as input the four-dimensional⁹ parameter space that is the result of all the above combinations. In this case, 81 reruns will be made. Run the model now, and take a short break while your computer does the work. When the model finishes, enter the 'View Graphs' option and plot TIME versus WSO and TIME versus WAG to inspect sensitivity of grain yield and biomass to a variety of development rates (see Figure 2.5).

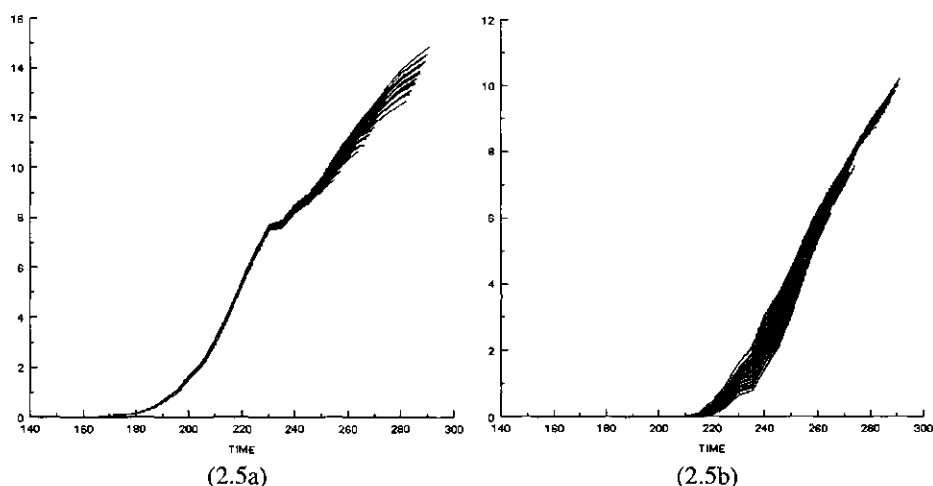


Figure 2.5 Effect of development rate changes on biomass and grain yield; Figure 2.5a and 2.5b, respectively. The changes in timing and final yield are the result of systematic changes in values of four parameters (DVRJ, DVRI, DVRR, and DVRI) related to development rates. Values on the Y-axis are tons ha⁻¹.

⁹ Three parameters generate a three-dimensional parameter space (three axes, e.g. x, y and z). You can picture this as a cube. With a little imagination, a four-dimensional parameter space can be envisioned as a vector (row) of the earlier mentioned cubes.

2.4.4 Zonation runs

Go to the main menu via the 'System' menu pad and enter the 'Regional Studies' application menu. Select the ORYZA_WZ model and set the production environment to 'rainfed upland'. Now proceed to the 'Create New Set of Input files' menu option. You would like to know the effect of soil type on the yield. Choose ORYZA_WZ.DAT for the crop file, PHIL1.992 as the weather file, and MAY.DAT for the timer file. Now select all the soil files, except for the PUDSxx.DAT soil files. Eighteen soil files should be selected at this point. Press the [OK] button; the 'Current Soil file' text field shows you '*MULTIPLE*' to indicate that more than one soil file has been chosen. Save this set as 'MURSOIL.SET', and select this set as input.

Run the model. The model will execute 18 runs, of which 17 reruns. After the model finishes, view the tabular simulation results 'View Tables'. At the 'Regional Studies' menu, this option does not show you the time-coursed data for the state variables in the RESULTS.OUT file. However, in the OP.DAT file (press [Q] or [Ctrl+PageDown] to skip to the next file in the viewer), the end-of-season results are presented. The rows with data are in the same order as the soil files listed in the 'Set Definition' window: starting with CLOAM.DAT and ending with SLOAM.DAT. Examine the values of WAG (weight aboveground), WRR (weight of rough rice) and GRDUR (growth duration) and comment on the differences with regard to the soil types. Leave the viewer by pressing [Alt+X] or [Escape]. Take a look at the plotting facilities in this application menu. Activate 'View Graphs'. A window opens with many options. In the 'Regional Studies' menu, the plotting facilities are focused on end-of-season values of the state variables and not on the time-sequenced data. Therefore, you will notice that three graph types are available (left top): scatter plots, frequency plots, and cumulative frequency plots.

Note that you only ran the model for eighteen different soil types, so only 18 data points per state variable are present. Normally, you would make more runs and hence get more detailed and significant graphs.

Move to the button of the independent variable and press the [Spacebar]. Now move the highlighting bar on top of WRR and press [Enter]. Press the [View Graph] button. After a few seconds, you are presented with a frequency plot of the weight of the rough rice. Press [Enter] to leave to graph and enable (●) Scatter plot. Select TRWCU (cumulative actual transpiration) as independent variable and choose WRR as dependent variable. View the graph. And, last but not least, enable the (●) Cumulative Frequency Plot and select RAINCU (cumulative rainfall) as independent variable. View the graph. Try some more combinations yourself.

As a next example, leave the 'Graph Options' window and move to the 'Create New Set of Input files' again. Choose the ORYZA_WZ.DAT as crop file, LOAM.DAT as soil file, MAY.DAT as timer file and select PHIL1.979 until PHIL1.992 for weather files. Save this combination of input files under 'RUMYEAR' and select this set. Now run the model with it. Enter again the 'View Graphs' menu option and make a number of graphs.

Examine the differences between the years as related to the total amount of rain and evaporation during the growing season.

Another feature of the 'Regional Studies' menu is to make zonation runs with combinations of two different inputs. For example, all combinations of a number of weather files (years) and soil files, or all combinations of weather files and starting dates. That last example will be worked out here. Go to the 'Select Model Options' menu option and make sure that the 'Rainfed Upland' production environment is selected. Then create a set with the following specifications: ORYZA_WZ.DAT as the crop file, LCLAY.DAT as a soil file, choose all available Philippine weather files and MAY.DAT as the timer file. Now enter a range of sowing dates (press the [STTIME] button):

140-160

in the box at the left side of the window. This will generate starting dates 140 to 160. Go to the [OK] button and press [Enter]. The number of starting dates should indicate 21. Save this set under the name of 'MURW-TIM' and select it.

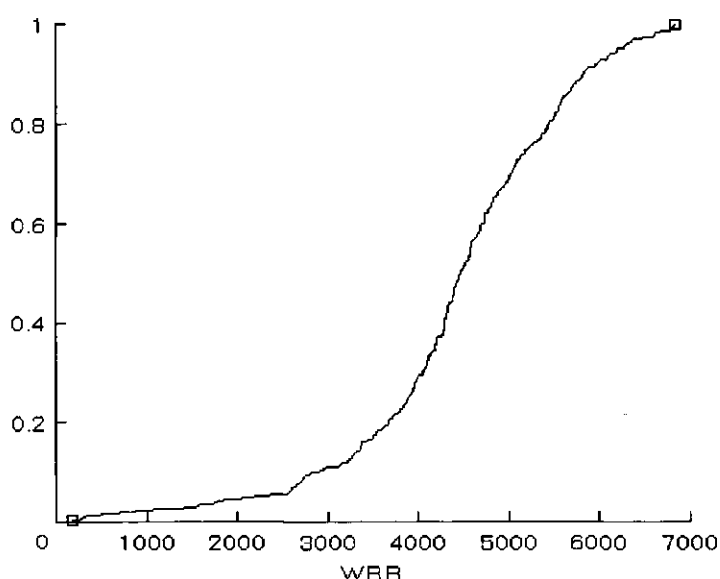


Figure 2.6 Cumulative frequency plot of WRR (weight of the rough rice). This plot shows the results of simulation runs using all simulated WRR values from all possible combinations of year (1979 to 1992) and sowing date (140 to 170).

When you run the model, all combinations of starting dates and years will be used to simulate crop growth: a little more than 400 model runs will be executed. If you have a

slow computer, you might consider skipping this exercise. Activate the 'View Graphs' option to inspect the results. Make a cumulative frequency plot (Figure 2.6) of the weight of the rough rice (WRR). Return to the 'Regional Studies' menu. This concludes the exercises on zonation runs.

2.4.5 Monte Carlo runs

These multiple runs can be used to run the model a large number of times, with changing parameter values drawn in a pseudo-random manner from a specific distribution. The main purpose of these runs is to perform risk analysis, uncertainty analysis, or just to make sure that the model performs accurately when all kinds of combinations of parameter values are used as input to the model. In the following example, you will examine the combined effects of slight changes in sowing date and water content at field capacity on the yield of a rice crop at a rainfed upland production environment.

Go to the 'Regional Studies' application menu and select the model ORYZA_WZ with the rainfed upland production environment. Select the 'RUWET.SET' set of input files. In the 'Input' menu popup, activate the menu option 'Generate Monte Carlo rerun file'. A large window appears. At the left side, you will notice a list with parameters that are available for selection. Move the highlighting bar to the STTIME (sowing date) parameter. Press [Tab] to jump to the [Method] list button, press [Spacebar], and select 'Beta distribution'. Note that the 'upper limit', 'lower limit', 'A' and 'B' input fields become accessible. Enter 170 for value of 'upper limit', 140 for 'lower limit', 5 for 'A' and 5 for 'B'.

Move back to the parameter list and position the highlighting bar on top of WCFC (water content at field capacity). Press [Tab] to move to the [Method] list button and select 'Uniform distribution'. Enter 0.6 for upper limit and 0.4 for lower limit. Press the [Generate] button. This activates the RIGAS program and 100 value combinations of STTIME and WCFC are independently created according to the distributions and limits chosen. You can view the results by tabular output [View Tables] or by graph [View Graph]. Make frequency graphs of both STTIME and WCFC to inspect the distribution of values¹⁰. You can [Generate] the values anew when the distribution is not good enough. However, since 100 values were generated, the distributions should be balanced enough. Leave the graphs and set the number of draws to 20. [Generate] the values again. Now press to [OK, use it] button to actually use the generated values for reruns. You will be returned to the 'Regional Studies' menu. Run the model. A total of 20 reruns will be made; on a slow computer this may take a while.

¹⁰ The plotting utility uses per default 5 frequency classes; the resolution of the results in the graph may not be high enough for some cases. To increase resolution, disable the [Autoscale] check button, enter a higher number of classes, enable [Autoscale] again, and then plot.

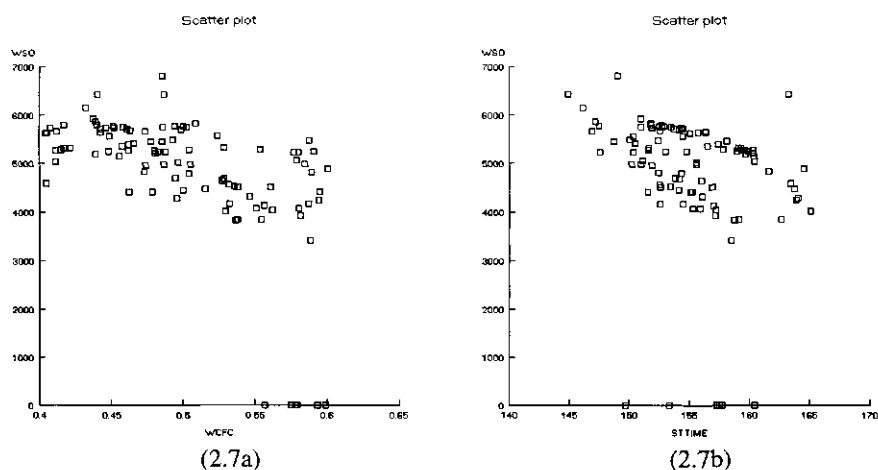


Figure 2.7 Examples of changes in STTIME and WCFC on grain yield WSO. Grain yield is plotted in kg ha^{-1} . The data points lying on the X-axis (zero grain yield) are those cases that have been struck by severe drought (> 25 days), at which the crop is assumed to die.

When the model finishes, view the graphs. Plot cumulative frequency plots of WRR, WAG, and WST. Use the scatter plots of the parameters STTIME and WCFC against the state variables to examine the effects of changes in the parameters more closely (Figure 2.7).

Warning: a normal distribution (not used in this example) 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 10.1.2).

3 FSU reference guide

J.J.M. Riethoven¹

¹ DLO Research Institute for Agrobiological and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

An in-depth description of each menu pad, menu option, and window that is part of FSU is presented to the user in this Chapter. The information is purely meant as a reference, only to be looked into when specific questions arise regarding a certain option in FSU, or when the user seeks a more detailed understanding of the system. Because this Chapter directly aims at the user, the style adopted here is writing 'active' in the second person (using 'you' instead of 'the user'). This makes reading the current Chapter more easy.

A top-down structure is used, meaning that first a brief description is given of each menu pad, then each menu option is discussed, and within each menu option, the input or action window is described (if applicable). Some options are present in more than one menu. Rather than to insert references to other parts of the manual, these options are described fully whenever they are available in the menus. Furthermore, at the end of this Chapter a description can be found of one of the most important windows: the item-selector, as well as a short introduction on how to use the TTSELECT utility for creating graphs.

Please note that the actual Figures¹¹ of screens and menus might differ slightly depending on the exact version of FSU on the distribution disks. All snap-shots of screens and menus have been made of FSU version 2.0. Note that 'FSU', 'shell' and 'simulation environment' refer to the same entity in this Chapter.

3.1 Main Menu

The very first menu that you will be shown is, obviously, called the main menu. This menu offers some general services and an entrance to the application menus.

3.1.1 File

At this menu pad (see Figure 3.1), you will find some options that are classified as file handling. Also, the 'Quit Shell' option is located within this menu pad. This version of FSU only has a rudimentary set of file-handling options built in, the listed project commands only refer to one (special) file.

¹¹ Since the Figures of menu options and windows in this Chapter are printed in two colours only, information has been lost regarding the differences in colour or intensity of several items (e.g., 'dimmed' or disabled items). On screen they display much better; the Figures should be taken as a reference only and not as an accurate copy of the screen.

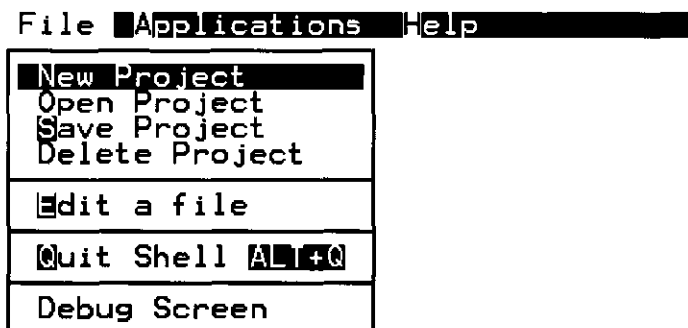


Figure 3.1 The 'File' menu popup and its contents.

The shell features a special file that allows you to save all current active settings (e.g., model name, what modules, names of the input files, etc.). This so-called project file can be used to restore the settings at a later time. For example, saving the settings at the end of the day will allow you to restore them the next day, to continue working without need to select anew.

There is only one limitation at this time: only one project file can be created.

New Project

The 'New Project' menu option will reset all previously made selections (if any). So, after execution of this command, no model or input file is selected, type of production level will be reset to its default value and all (result) files that are located in the working area will be erased (gone forever). You have to take caution using this option, if you have any unsaved result files in the working area¹². See also menu options 'Save Results' and 'Restore Results'.

Open Project

You can load the project file (previously saved settings) when you activate this option. Normally, FSU detects the presence of a project file during startup, and you will be asked to either load it or ignore it. If you accidentally choose or press the wrong button, you can use this option to reload the previously saved settings as selections. Remember that you can only restore settings if you have either explicitly saved them using the menu option

¹² FSU uses a special working area to store its intermediate files: the files that are needed to control the correct execution of the models and the result files. No data files (for example, crop or weather files) are stored in this area, so no accidental deletion of precious (input) data will occur.

'Save Project' or if you answered [Yes] when the shell asks whether you want to save the settings (prior to exiting the simulation environment).

The 'Open Project' menu option is disabled if the project file `PROJECT.PRJ` does not exist.

Save Project

This menu option allows you to save a complete session, including all settings (model, modules, input files etc.) and the menu status. It enables you to make a safety backup as your work progresses and to save vital information at the end of a workday to continue next day.

In this version of FSU, only one project file exists, called `PROJECT.PRJ`.

Delete Project

If you want to delete the project file, use this menu option. It will not affect selections and settings currently active in the simulation environment, it only removes the file. In case the project file `PROJECT.PRJ` does not exist, this menu option is disabled.

Edit A File

It can be quite handy if there is an option from within the shell to do some quick modification of input files. This menu option will let you do exactly that. When you activate 'Edit A File', a window opens that bears some similarity to the Item Selector and enables you to choose a file for editing (see paragraph about 'The File Selector' for detailed information on the workings of this window). After you choose a file, the 'File Selector' window disappears and a new window opens. The latter presents you with the contents of the chosen input file, and you can make any change you like. Note that this is a very crude way of editing files, so do not expect much options like 'cutting' and 'pasting' text. All text-based files (plain ASCII) can be edited with this option; you are not limited to editing files that are part of the FSU environment or its models.

Pressing [Escape] will close the window without saving changes; [Ctrl+W] saves the changes first and then closes the window.

Quit Shell

Activating this option obviously lets you leave the simulation environment. Before it does, however, you will be asked to confirm your action. If you still want to quit, and if any changes have been made during your session, the shell asks you an additional question: do you want to save the changes that you have made to a project file, or not? See also 'File' and 'Save Project' menu options.

The 'Quit Shell' menu option has a hot key assigned to it: pressing [Alt+Q] also leaves the shell.

Debug Screen

To display a window with information that gives a general idea of the configuration of your computer system, press the 'Debug Screen' menu option. Whenever you have to consult with the author of FSU concerning a bug or an 'undocumented feature', please activate this window and provide the information listed on this window to the author together with a complete problem description (see also paragraph 1.6, 'Getting help', on page 10).

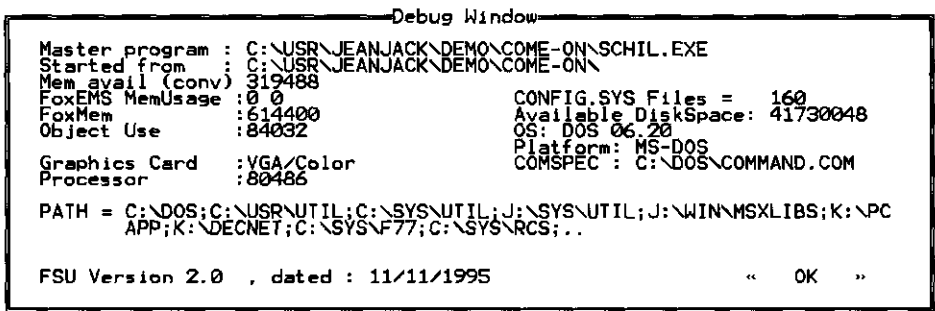


Figure 3.2 The 'Debug' window displays the computer configuration.

3.1.2 Applications

This menu pad (see Figure 3.3) is the main entrance to the separate studies that can be done with the crop models. Only two options are implemented at this moment: model/experiment studies and regional (zoning) studies.

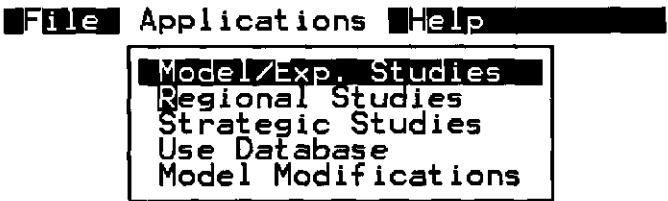


Figure 3.3 The 'Applications' menu popup.

Model/Exp. Studies

Activating this menu option will lead you to the menu that can, in general, be used to compare model output and field (experimental) data. This includes using experimental data as forcing functions, incorporation of several modules into one model (potential productions, N limitations, photosynthesis, water balance), plotting of model data versus experimental data, reruns with same input files but with user-defined modifications in selected parameters.

Regional Studies

The 'Regional Studies' menu will let you use a model to do research in (for example) expected yield for a whole range of varying conditions, like soil types, weather, crop and varieties, and management options. Select a maximum of two types of conditions to vary and let FSU generate the reruns file for you. After running the model to simulate all these conditions, inspect and analyse the output by (cumulative) frequency or scatter plots. Another feature is to use this menu to perform sensitivity, uncertainty, or risk analysis via the embedded support of the RIGAUS program (see Chapter 10).

The generated result tables can be loaded into your favourite spreadsheet or statistical package for further number crunching.

3.1.3 Help

If you like more information about FSU, or if anything is unclear to you, select this menu pad. The 'Help' menu pad gives you access to the 'Help Index' and the 'About FSU' menu options.

Help Index

Pressing this menu option enables you to review all the topics that are available in the on-line help file. Topics range from general notes about the user-interface to detailed information on menu options and windows. The help topics cover roughly the same information as given in this Chapter, only in less detail. You can scroll through the help index (Figure 3.4) by using the [Arrow] keys. When you encounter the topic you have searched for, press [Enter] or double-click with the mouse or [Spacebar] to read the help text. Take note of the first part in the help index: it contains the copyright notice as well as a brief overview of any changes that might have been made to FSU or the documentation after publishing this booklet.

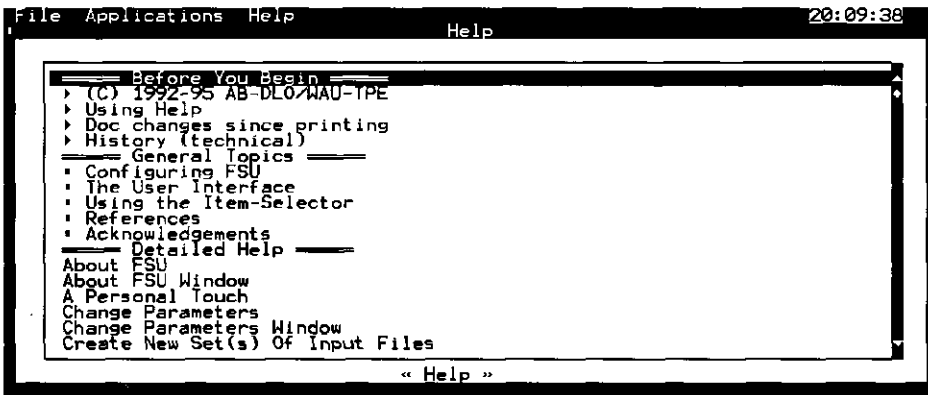


Figure 3.4 An example of the 'Help Index' screen of FSU.

Select [Topics] to return to the list of help topics, use [Next] and [Previous] to walk forward and backward to the help texts (see Figure 3.5). Some help topics are cross-referenced. This means that a similar or a related help topic is present in the list of help topics. The [See also] button can be used to display a short list of the related topics. Choose one of these topics and you are presented with the corresponding help text.

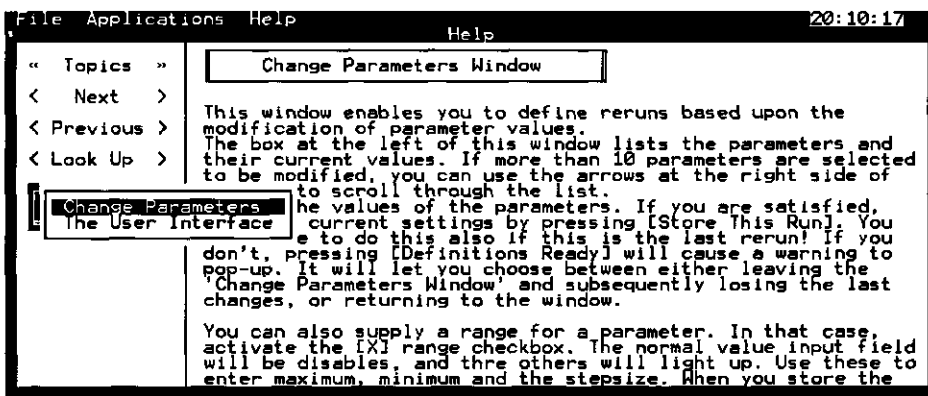


Figure 3.5 An example of a help topic with cross-references.

You can close the 'Help' window by pressing [Escape] once. Mouse users can 'Zoom' or 'Resize' the window. See Chapter 1, paragraph 1.3.5, on how to do this. Keyboard users can maximise the 'Help' window by pressing the [Ctrl+F10] key combination.

About FSU

This menu option will show you a window (Figure 3.6) with some basic information on FSU and the primary contact persons. Note that a larger list of contact persons can be found in the 'Help Index', in one of the very first topics (copyright notice).

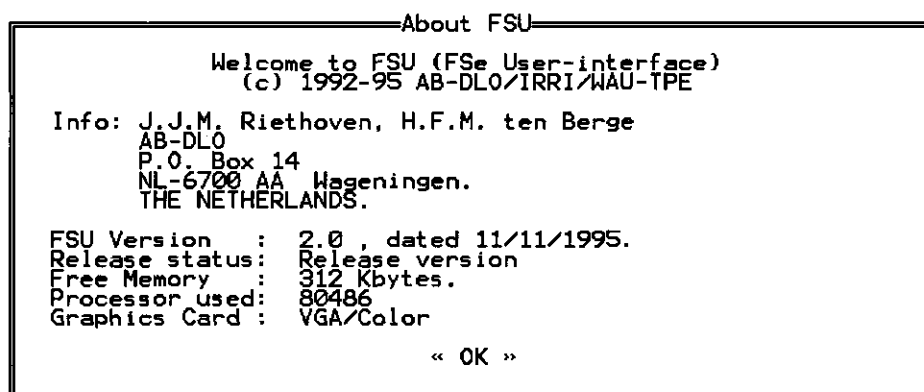


Figure 3.6 The 'About FSU' window.

3.2 Model/Experiment Studies Menu

To compare model output and field (experimental) data, the 'Model/Exp. Studies' menu should be used. Besides comparing model output with field data, this menu also offers the functionality of using experimental data as forcing functions, incorporation of several modules into one model (potential production, N limitation, alternative photosynthesis routines, water balances), plotting of model data versus experimental data, reruns with same input files but with user-defined modifications in selected parameters, and so on. As you will see when you enter this menu, a status text is displayed on the screen. It shows your current active selections, like for example the name of the model you chose, which input files are selected, and what type of run is about to be executed. At all times, this status menu reflects the current selections.

3.2.1 System

Four general-purpose functions are located at this menu pad: 'Main Menu', 'Quit Shell', 'Simulation Controls', and 'Edit A File'.

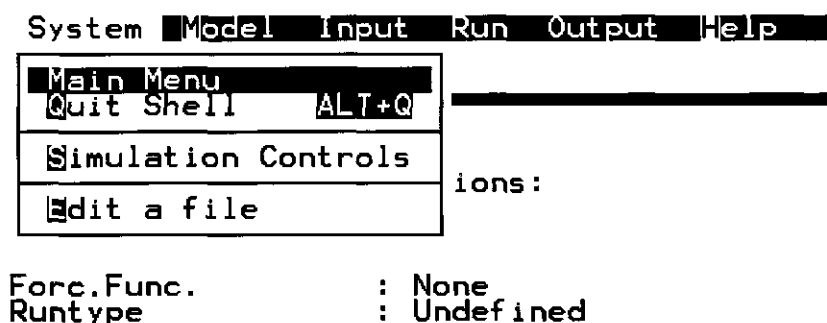


Figure 3.7 An overview of the 'System' menu pad.

Main Menu

The 'Main Menu' option returns you to the main menu. All selections and options that you have selected will remain active. So, if you decide to return to the 'Model/Experiment Studies' or change to 'Regional Studies', you can continue working with the same settings.

Quit Shell

Activating this option obviously lets you leave the simulation environment. Before it does, however, you will be asked to confirm your action. If you still want to quit, and if any changes have been made during your session, the shell asks you whether you want to save the changes you have made. See also the 'Save Project' menu option.

This menu option has a hot key assigned to it: pressing [Alt+Q] also activates the 'Quit Menu' option.

Simulation Controls

FSU uses some built-in defaults in the way it controls the behaviour of the actual simulation of models. Often, these defaults are related to the type or amount of output that is produced by a model.

The 'Simulation Controls' window (see Figure 3.8) enables you to override these defaults. Options are grouped by application. For the 'Model/Experiment' menu, there is only one setting that is of interest: whether or not the menu model options override the set-defined model options. Normally, when you select a set of input files, the exact values of the model options are taken from the set. When you run the model, those model options are passed to the model. It is possible to override the model options for a single set: after selecting the set of input files you must enter the 'Model Options' menu option and change

the model options before running the model. When running the model, the just changed menu model options will be passed to the model instead of the model options stored in the set. This procedure works only when one set has been selected. In the case that more than one set of input files has been selected, per default only the very first set in the sequence can be changed where model options are concerned. If you want the same model options for all selected sets of input files, make sure that in the 'Simulation Controls' window, the first option is checked. When this 'Menu model options override set-defined options' button is activated, the actual menu model options are taken and passed to the model instead of the options that are stored in each set of input files.

See also the 'Select Model Options' and 'Select Existing Set(s) Of Input files' menu options.

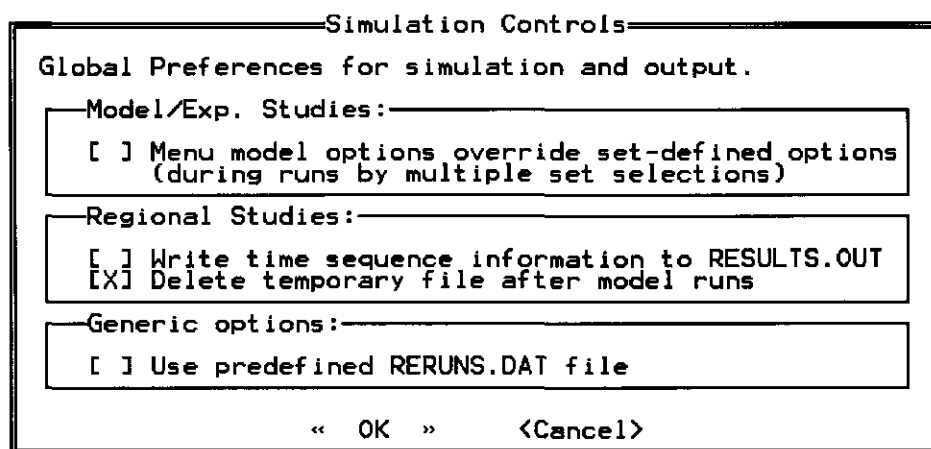


Figure 3.8 The 'Simulation Controls' window.

Another option that can be interesting in the 'Model/Experiment Studies' menu is the 'Use predefined RERUNS.DAT file'. When this option is checked, the shell will use a user-defined file that specifies the number of reruns that should be made with a model and how each run differs from the other runs. This is especially useful when you have a number of data sets that you wish to present at a demonstration, or when you always want a model to run using specific sets of changes in certain parameters. After you check this option, go to the 'Create New Set(s) of Input Files' menu option: the 'Set Definition' window includes an additional button now, with which you can select a pre-defined reruns file.

Refer to Chapter 8 for a discourse on how the FSE system (in which the models are programmed) uses the RERUNS . DAT file, and to Appendix 2 on how to set up FSU to use these predefined rerun files.

Edit A File

It can be quite handy if there is an option from within the shell to do some quick modifications of input files. The 'Edit A File' menu option will let you do exactly that. When you activate 'Edit A File', a window opens that bears some similarity to the Item Selector and enables you to choose a file for editing (see paragraph 3.5 about 'Selector' for detailed information on the workings of this window). After you choose a file, the 'File Selector' window disappears and a new window opens. The latter presents you with the contents of the chosen input file, and you can make any change you like. Note that this is a very crude way of editing files, so do not expect any options like 'cutting' and 'pasting' text.

All text-based (plain ASCII) files can be edited with this option; you are not limited to editing files that are part of the FSU environment or its models.

Pressing [Escape] will close the window without saving changes, pressing [Ctrl+W] will save the changes first and then will close the window.

3.2.2 Model

Menu pad 'Model' lets you select a model and choose several options with regard to the model (if implemented) that can be used when you start a model run. Options are part of the model proper, and not of the shell. See also Appendix 2 for detailed information on adding and configuring models.

Select Model

The 'Select Model' menu option presents you with the Item-Selector (see paragraph 3.4) and lets you choose from the models listed. When available, you can get information about a model by moving to the desired model, then [Tab] to the [Info] button and press it. A window should open with some information about the model. Depending on the author of the text, this may range from very basic information to long background stories about the history and complete context of the model.

To select a model, move the highlighting bar over the name of the model, and press [Enter] or double-click the mouse to select. Press the [OK] button to activate your selection.

Select Model Options

Some models incorporate different production environments or alternative formulas for a certain process. These can be set in this menu option. Note that this option is only available when you have previously selected a model. After activating this option, a window will pop up and present you with a list of available options. This list may look like Figure 3.9, or it may have another structure and show you first some main groups in which you can activate certain settings (Figure 3.10).

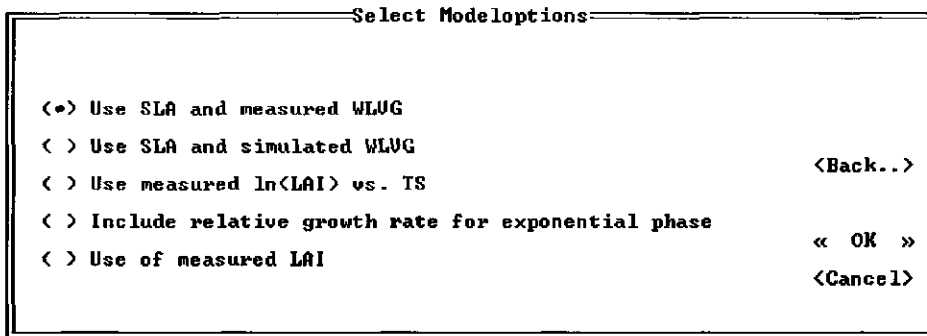


Figure 3.9 An example of the actual model options that can be selected by the user. This particular screen is configured by the ORYZA_N model (Drenth et al., 1995), which is not part of the standard distribution of FSU.

In the above Figure 3.9, you move to the setting of your choice and press [Space] to activate it. [Back..] lets you go one level up to the previous screen. If the [More..] button is selectable, the option you selected has more settings available. Press [More..] to show those settings.

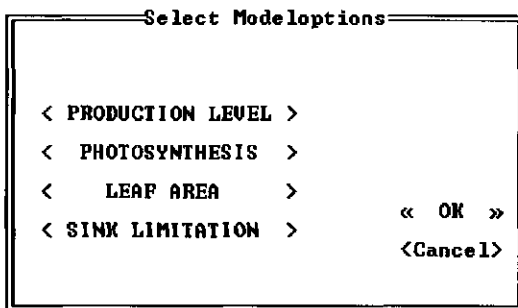


Figure 3.10 An example of some main sections of the model options. This particular screen is configured by the ORYZA_N model (Drenth et al., 1995), which is not part of the standard distribution of FSU.

In the second type of screen (mostly used in more complex models), you move to the push button of your interest (for example [Production Level]) and press it. You will be presented with a new screen with model options, all related to the topic of the button you just pressed. When you have made all the necessary modifications, press [OK] to accept the changes you made or press [Cancel] to undo.

Note that the manner in which model options are presented to you is the responsibility of the model developer. A model has to be properly documented and configured to be installed in the shell. See Appendix 2 for further information.

3.2.3 Input

After you select a model, the 'Input' menu pad becomes available to you. Use the 'Input' menu pad to create sets of input files, to select these sets or to create a reruns file based on parameter modifications. A set is a small file used by FSU to keep track of the combination of input files and model options that you wish to run the model with. When given a descriptive name, a set can also be used to quickly retrieve specialised input.

The steps to take in selecting your input are to first create a set with the 'Create New Set Of Input Files' menu option, and then to select that set using menu option 'Select Existing Set Of Input Files'.

Create New Set(s) Of Input Files

After you select a model and its options, this is the second step you should take if you do not have a prefabricated set of input files. The 'Create New Set(s) Of Input Files' enables you to select the various input files and store that specific combination of input files for later use. Together with the combination of the input files, the current settings of the model options are also stored.

The types of input files that can be selected are model-dependent. Some models only need weather and crop data, others also need soil data or cultivar data of a second crop. The 'Set Definition' window only lists those types of input files that are relevant to the current model. Figure 3.11 displays a typical example of the 'Set Definition' window.

The push buttons on the left-hand side of the 'Set Definition' window consist of the [Experimental Info] button and a number of buttons named after the type of required input files. In the above example these types of input files are [CROP], [SOIL], [WEATHER], and [TIMER]. Pressing the buttons related to input files gives access to the Item-Selector and lets you choose the files that belong to that specific type of input file.

The [Experimental Info] button presents you with a list of files. These files can be used by researchers to describe experiments or treatments in detail. Selecting one of these files will open a window with the contents of this file. The window can be moved or re-sized (for mouse users) in such a way that both the content of the window and the 'Set Definition' window are shown on the screen, making it easy to use the information within the first window to choose input files accordingly in the latter.

Set Definition			
<Experimental Info>		Last set saved	: IRL0W-88.SET
<	TIMER	>	Current timer file : TORYZ_WZ.DAT
<	SOIL	>	Current soil file : PUDS10.DAT
<	CROP	>	Current crop file : ORYZA_WZ.DAT
<	WEATHER	>	Current weather file : PHIL2.988
<p style="text-align: center;">« Save Set » < Exit ></p>			

Figure 3.11 The 'Set Definition' window.

As soon as you have chosen a file in one of the input file-related buttons, the actual file will be shown at the right side of the window. Press the [Save Set] button to store the set. You will be asked to enter a name for this set. Use an easy to recognise name, preferably related to the selected conditions, cultivar or site. Press the [OK] button to save. If a set already exists with the same name, you are presented with a choice: either overwrite (press [Yes]) or choose another name (press [No]). Note that in a set, besides the input files needed to run the model, special options related to the model are also stored automatically. These options are chosen via the 'Select Model Options' window. If you want to make sure that the model runs with the preferred options, then you have to select the model options first, and then create the set.

After you have saved a set, you can either define a next set or press both [Exit] or [Cancel] to leave the 'Set Definition' window and to return to the 'Input' menu pad.

System Model Input Run Output Help	
Menu MODEL/EXP. STUDIES	
Currently active selections:	
Model name	: ORYZA_WZ.EXE
Forc.Func.	: None
Runtype	: Single
Selected set	: IRL0W-88.SET
Input files in set	: TORYZ_WZ.DAT, PUDS10.DAT, ORYZA_WZ.DAT, PHIL2.988

Figure 3.12 An example of the status screen, showing the active settings.

Select Existing Set(s) Of Input Files

When you activate this menu option, the Item-Selector window appears. You now have the opportunity to select the set (of input files) you want. If you do not remember exactly which set to use, press the [Info] button to examine the contents of a set (see also paragraph 3.4).

If you already made a previous selection of a set, and then press the menu option 'Select Existing Set(s) Of Input Files' again, a window appears. This window warns you that selecting a new set will undo your previous selection. You can press [No] to exit or [Yes] to continue. By selecting a new set, you will see that (part of) the content of the set will be shown on the screen (see Figure 3.12) and also the type of model run.

You can select more than one set in the 'Model/Exp. Studies' menu. After a model run, you can compare the results of the different scenarios defined in the sets by the 'View Graphs' menu option.

Change Parameters

The analysis of the important processes in a model can be greatly aided by being able to compare two or more runs with different values for major parameters. This menu option lets you select a few parameters that you want to modify, and then offers you the opportunity to actually change those values. By running the model, first the standard values of the parameters (as they are defined in the input files) are used to simulate, and then every next model run will use the modified parameter values inserted.

For example, if you would like to know the effect of a variable X on the model output, you select this variable and then enter a wide range of values for this parameter. Each new value symbolises a new model run. After executing the model, you can compare output, either graphically or by tabular means.

When you press the 'Change Parameters' menu option for the first time since selecting a set, the shell first converts the data files to another format for quick access. The screen reads 'Converting data files....' and shows you the name of the data file currently being converted.

After this process completes, there are two possible pathways. In the first case, you did not make a selection of parameters to modify yet. You will be presented with a list of parameters from which you can choose. In the second case, you have already made a selection of parameters and entered some modifications of them, and you have three choices (a window pops up to let you confirm one of the choices) to continue. You can choose a complete new set of parameters, thereby deleting any parameter value modifications that you made earlier. Press the [Parameters] button to do so. The second choice is to use the current selection of parameters but erase the modifications of the parameter values themselves. Press the [Values] button to start defining the values for the first rerun. The last choice is to both use the current parameters and to add to the existing list of parameter modifications. To make it so, press the [Append] button.

After selecting the parameters that you like to modify (the first case mentioned or the first choice of the second case), press [OK]. A new window opens (see Figure 3.13) in which you can change parameter values and save these. Note that modifications of the parameters are not written to the original input file but are kept secure by FSU. Just prior to the model run, the shell will extract the parameter values and write these to the reruns file. So the reruns file always contains the changed values, and the first model run is the standard run using the original data files.

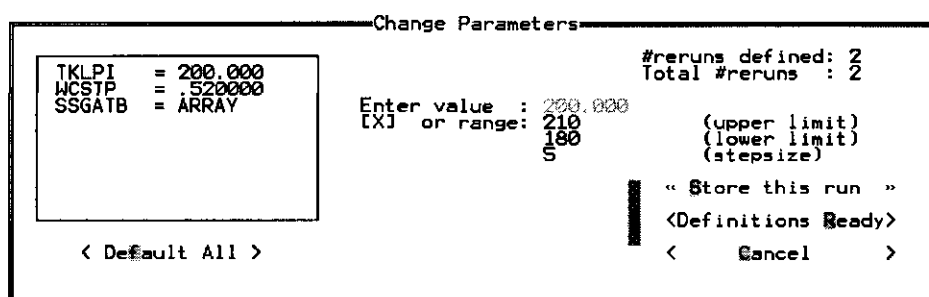


Figure 3.13 An overview of the 'Change Parameters' window.

The left side of the window shows a box with parameter names and actual values. Use the cursor keys to select the parameter for which you like to change values (move the highlighting bar on top of the parameter). Move to the values field by pressing [Tab]. You can change the value now. If the parameter you select is in fact a table instead of a single value, you will not see the actual value in the box at the left side of the window. The phrase 'ARRAY' will be displayed, and you can only view the contents by moving the highlighting bar on top of the parameter. The large box at the mid-bottom of the window can be used to edit the values of the table.

The push button [Default All] that is placed right underneath the parameter list box resets all parameter values to their default (i.e., as they are listed in the data files). This can be handy when you make systematic changes to a large number of parameters. At the right bottom part of the window three push buttons are placed. You use [Store This Run] to store the parameter values currently listed. The number at the top right of the window ('#runs') should increase, as well as 'total #runs'. The first represents the number of reruns defined in this session, and the second the total number of reruns. Most of the time, these numbers equal each other, but whenever you [Append] to an already existing parameter modifications list, these numbers may differ. Press [Definitions Ready] when you finish making changes. If any changes have been made since the time you last saved (since pressing [Store This Run]), you will be warned and given the opportunity to return to the 'Parameter Modifications' window. A new window (see Figure 3.14) will open and list the modified parameter values and the rerun number to which they belong. If one of the runs is

not to your liking, you can delete it by moving over one of the parameters and pressing [Delete Run]. [Cancel] will discard any changes made to the parameters during that particular session (you begin a new session when you press [Append]; see above). Changes made in previous 'Change Parameter' sessions will remain intact. Press [OK] to accept the reruns as they are listed.

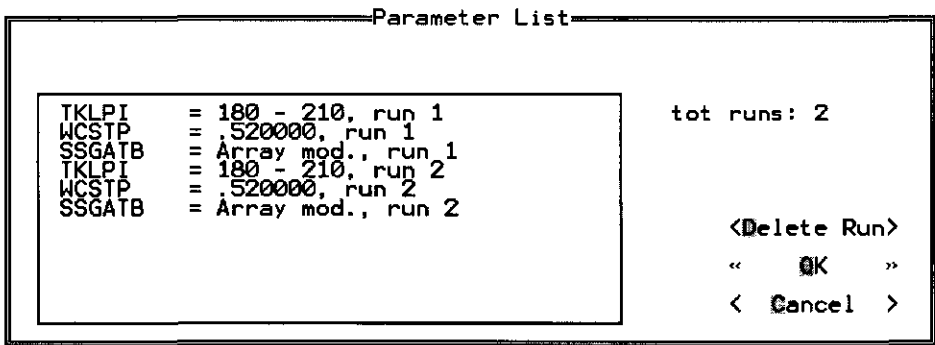


Figure 3.14 The 'Parameter List' window. This window enables you to remove incorrect reruns.

The 'Change Parameters' window is also capable of automatically generating a whole sequence of values for a parameter (except for the 'table' or 'array' kind of parameters and variables). Skip to the 'range' check box and press [Spacebar]. The box is now checked (see Figure 3.13) and three new fields are enabled: 'upper limit', 'lower limit', and 'step size'. Enter these values and press [Store This Run]. If you entered an incorrect value (e.g., upper limit is less than the lower limit or the step size is zero or too small), a warning will be displayed. In that case, the run will not be stored: you must correct the values first. For example, entering 0.80 as upper limit, 0.30 as lower limit, and 0.1 as step size will generate the following sequence when you run the model: 0.30, 0.40, 0.50, 0.60, 0.70 and 0.80. The upper and lower limit will always be included in the generated range; the maximum number of steps equals 20.

You can activate more than one parameter with ranges. In that case, all combinations of values are generated, thus creating a systematic overview of the parameter space. For two parameters, the parameter space can be visualised as a plane; for three, a cubic; for four, a plane of cubics; and so on.

An example for two imaginary parameters, A and B: A has upper limit 10, lower limit 2, and step size 4; B has upper limit 5, lower limit 2, and step size 3. The generated reruns (the combinations of the values of these parameters) file looks like this:

```
A = 2;  B = 2
A = 6;  B = 2
A = 10; B = 2
```


A = 2; B = 5
A = 6; B = 5
A = 10; B = 5

This feature of being able to make these kinds of combinations is a powerful tool for systematically examining the scope of model output with regard to fluctuations in parameter values.

Note that the 'Change Parameters' menu option is dimmed (disabled) whenever reruns (other than reruns by parameter modification) are active.

Switch Simulated/Forced

This option is a very good tool for validating certain parts of a model. Basically, it boils down to the question of whether to simulate a state variable or to use observed data during (parts of) the simulation. However, the selected model needs to be prepared to use this feature (see Appendix 2).

When you press the 'Switch Simulate/Force' menu option for the first time since selecting a set, FSU first converts the data files to another format for quick access. The screen reads 'Converting data files....' and shows you the name of the data file currently being converted. After this process is completed, a new window opens (see Figure 3.15).

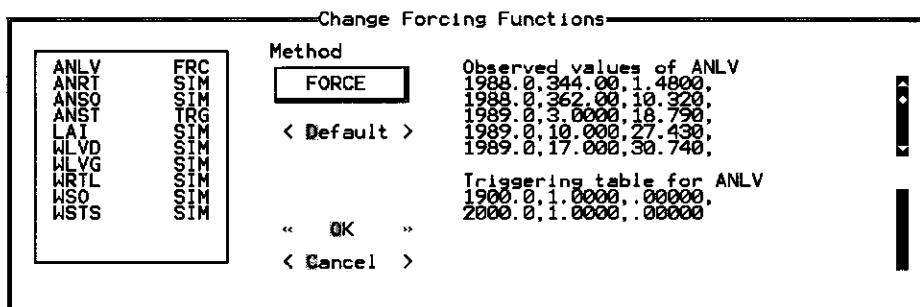


Figure 3.15 An overview of the simulate/forcing window. This particular screen shows available state-variables as they are defined in the ORYZA_N (Drenth et al., 1995) model, which is not part of the standard distribution of FSU.

In the above window, the leftmost box presents you with a list of state variables of which you can choose the simulation mode. This mode can be set to simulate ('SIM'), to force ('FRC'), or to trigger ('TRG'). The first mode is the default: the state variable will be simulated when you run the model. The second method will not simulate the state variable, but will use the observed data instead to calculate the value of the state variable at a given time. Between data points, a linear interpolation will be performed to estimate the value.

The last method is a mix between the two: it enables you to force or simulate state variables for time ranges that you defined. This procedure will be explained later in greater detail.

As you can see in the window, a popup button is placed between the boxes. It reads either 'SIMULATE', 'FORCE', or 'TRIGGER'. You can use this button to change the simulation mode of the current selected state variable (the one with the triangular sign in front of it and highlighted by the selection bar). For example, in the above window you could move to the state variable ANST, then press [Tab] to jump to the 'Method' popup, press [Space] and activate the setting of your preference. The list on the left side is automatically updated so it always displays the current settings.

At the right side of the window, two boxes are displayed. The top box lists the current observed data that are available for the state-variable, the second box lists triggering information (see later). The first two columns in both boxes are of the same type: year separated with a comma by day of year (Julian date). The third column shows the observed value or the triggering method (in case of the triggering information). You can edit the values that are listed. Modified values are used when you run the model. However, these modified values are not written to the original input files, but are kept by the shell. Whenever you enter incorrect information in one of the edit boxes, an alert window will present itself with information on the error. You have to correct the error before you are able to leave the edit box.

The [Default] push button will restore the original values into both observed data and triggering information edit boxes for the currently selected state variable. Press [OK] to use the settings for the listed state variables, press [Cancel] to discard any changes you made.

When you press the 'Run Model' menu option, FSU will create a rerun file with the modified simulation modes (and observed data) of the state variables. Thus a model will run twice: once using the original data and simulating the state variables and a second time with (possibly changed observed data and) the new simulation mode. The results of the simulation can then be compared using tabular or graphical output.

The triggering information needs some more explanation. There are three triggering modes: simulate (value = 0), force (value = 1), and reset (value = 2). The first two modes are in effect until another simulation mode is encountered in the triggering list. The third mode (reset) will realign the value of the state variable with the observed data and then continue with simulation. For example, let us say that the triggering table for a state variable WLVG looks like this:

1988., 344.,	0.,
1989., 16.,	1.,
1989., 20.,	0.,
1989., 32.,	2.,
1989., 60.,	1.

If you set the state variable to triggering, then the following will occur during the simulation of WLVG. At the start of the second simulation run, WLVG will be simulated normally. When time reaches day 16 of year 1989, the value of WLVG will represent the observed values from there on until we reach day 20, at which normal simulation of WLVG will proceed. At day 32, the value of WLVG will be reset to the corresponding value of WLVG in the observed data table and then normal simulation will continue. And at the last part of the simulation, the observed values of WLVG are again used in the model. The reset option can be quite handy if you like to realign the model to a state variable of central importance each time that observation has been made.

Read also Appendix 2, the paragraph about 'OBSSYS: simulate/force system' for more information on setting up this simulating/forcing system in existing models and how to organise your data files accordingly.

3.2.4 Run

The 'Run' menu pad has only one menu option, 'Run model', and can only be activated when both a model and a set have been chosen.

Run Model

When you choose this option, a window appears and asks if you are sure that you want to run the model. Press [Yes] if you are, otherwise press [No]. If you press [Yes], FSU checks if the necessary input files are available and updates or generates other required input files for the model. Read Appendix 4 to learn more about this.

On successful completion of the above procedure, the model will be executed.

3.2.5 Output

One of the most important menu pads is the 'Output' menu pad. This pad enables you to take a peek at the output data the model generated and makes it possible to show some basic types of graphs.

Store Output!

After you have run the model and are satisfied with the data it produced, you might want to store the output of the model. This enables you to use external packages like statistical programs or professional plotting packages for further data analysis and it keeps the data safe. Especially the last, since FSU will clean up the working area each time you leave the shell or choose another set or model. Cleaning will irrevocably erase the output data which a model has produced, except if you have saved it previously. Input data are not affected.

To save data, activate menu option 'Store Output!' and then type a name under which you like to keep your data safe. The name will be used to create a directory in the save path (see Appendix 4) and to copy the data to this directory. If such a directory already exists, and data are already stored into this directory, the shell informs you about this and asks you to either overwrite the data already stored or to [Cancel] and try another name. For your information, a maximum of five files will be displayed that are at risk of being overwritten.

Restore Output!

Sometimes you wish to re-examine some output of a specific model run. Choose the menu option 'Restore Output!'. An item-selector pops up and shows you a list of names. These are the names under which you saved the output, with an extra extension of '. SAV'. So if you saved under the name 'IRLOWLAN', choose 'IRLOWLAN.SAV' to restore your data. If you have never stored any data, FSU informs you of your mistake. Also, if data files are still located in the working area, the shell displays a warning message and asks you if these files can be overwritten ([Yes]) or not ([Cancel]).

After FSU restores the output files, you can re-examine them by the 'View Tables' and 'Graphs' menu options.

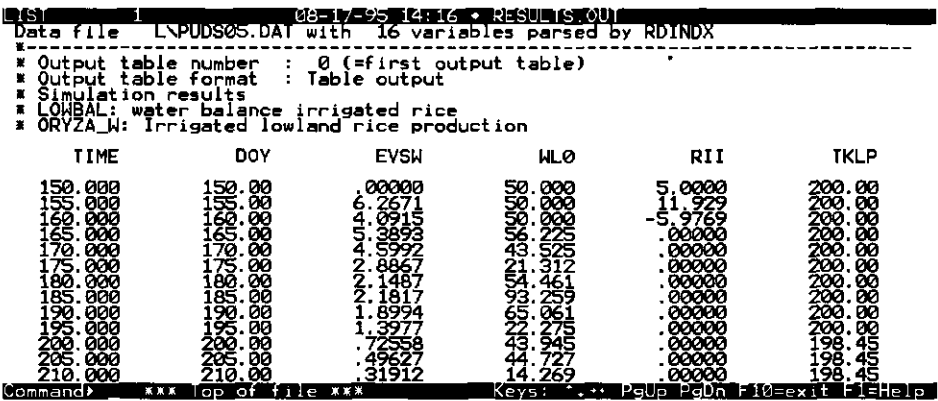


Figure 3.16 The 'View Tables' menu option shows the contents of the output files. In the above example, the primary simulation results are shown (file RESULTS .OUT).

View Tables

After a successful model run, several files are present in the working area that are part of the model output. These files are RESULTS .OUT, RESULTS .LOG, WEATHER .LOG, and OP .DAT (optional). The first file is always generated, but only after a single run will this file be of interest to you. It lists the output variables for each time step. The RESULTS .LOG file contains messages from the rest of the model (e.g., an overview of warnings that have occurred, and in case of multiple runs it also logs the replacements of

parameter values). The headers of the weather files are listed in the `WEATHER.LOG` file, together with warnings and errors from the weather system (only if applicable). The last output file will only be present if a model has been set up to facilitate the output-processing routines that are generally affiliated with the 'Regional Studies' menu. It contains the end-of-season values for some important state variables. Please read Appendix 2, the paragraph about 'OPSYS: end-of-season values system' for more information on this subject.

When you activate menu option 'View Tables' a viewer¹³ will be started. This viewer shows you the contents of each output file. An example of the output can be seen in Figure 3.16.

View Graphs

If you wish to display the results of a model run in a graph, choose the 'View Graphs' menu option. FSU uses the `TTSELECT` plotting utility to create graphs. For detailed information on the `TTSELECT` utility, refer to paragraph 3.6 on page 70.

State variables can be plotted against time (Figure 3.17) or against each other to examine possible correlations. More than one state variable at a time can be plotted against the independent variable, so comparisons between variables are easy to display. In the case of multiple runs (for example, due to parameter modifications), the results of every run can be plotted at the same time to reflect the effects of different input on the model output.

On most systems it is not possible to print a graph directly from within FSU. Refer to Appendix 1 on how to set up your system to facilitate printing of graphs (hard copies).

When you use 'Retrieve Output!' and then 'View Graphs', you might run into minor problems. FSU does not store the settings when you use 'Store Output!': it only saves the output files. Hence, FSU cannot know what type of run you performed to get that output, and so it also does not know what kind of graphs should be activated. You might, for example, run into messages from FSU saying 'Did you really run the model?' because it cannot find the binary output file `RES.BIN` (in case you retrieved results from a 'Monte Carlo' rerun). The trick to circumvent this is based on reconstruction of the conditions prior to the model run that created the output. For a multiple run based on 'Monte Carlo' distributions that means selecting a model (any model), a single run set, entering the 'Generate Monte Carlo Rerun File' menu option, selecting a dummy variable, press [Generate] and [OK, Use it], and finally activating 'View Graphs'. This will show you again the specialised graph window for multiple runs.

¹³ FSU uses an external viewer for this purpose, the `List 7.1a` program by Vernon D. Buerg. At the top of the screen, you note the filename. Use the [Arrow] keys to scroll through the file. [Ctrl+PageUp] and [Ctrl+PageDown] list the previous and next file, respectively. [Alt+X] or [F10] leaves the viewer.

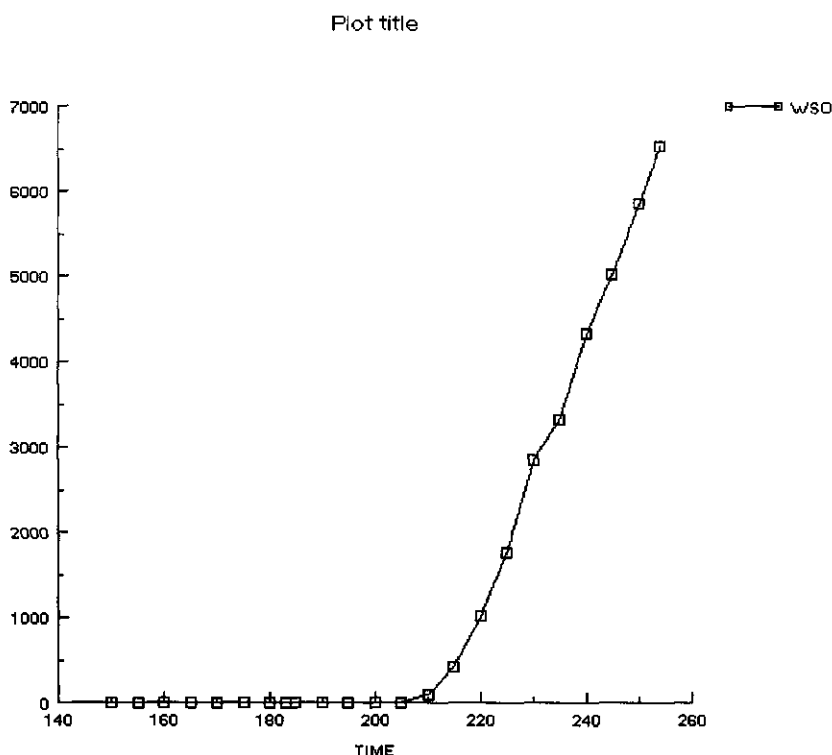


Figure 3.17 An example of a graph created by TTSELECT.

3.3 Regional Studies Menu

The 'Regional Studies' menu will let you use a model to do research on (for example) expected yield for a whole range of varying conditions, like soil types, weather, varieties, and management options. Select a maximum of two types of conditions to vary, and let FSU generate the reruns file for you. After running the model to simulate all these conditions, inspect and analyse the output by (cumulative) frequency or scatter plots. Another feature is the use of this menu to perform sensitivity, uncertainty, or risk analysis via the embedded support of the RIGAUS program (see Chapter 10). The generated result tables can be loaded into your favourite spreadsheet or statistical package for further number crunching.

When you enter this menu, a status text will be shown on screen. It displays your current active selections (e.g., the name of the model you chose, what production level is active, which input files are selected, etc.). At all times, this status reflects the current selections.

3.3.1 System

Four general purpose functions are located at this menu pad: 'Main Menu', 'Quit Shell', 'Simulation Controls', and 'Edit A File' (see Figure 3.18).

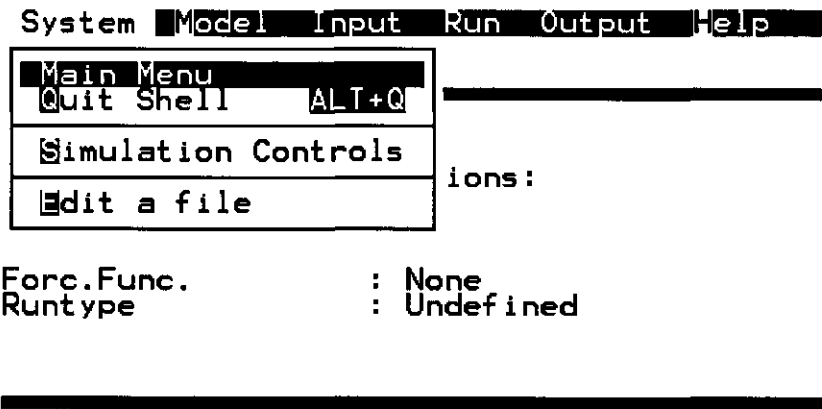


Figure 3.18 An overview of the 'System' menu pad in the 'Regional Studies' menu.

Main Menu

This menu option returns you to the main menu. All selections and options that you have selected will remain active. So, if you decide to return to the 'Regional Studies' or change to 'Model/Exp. Studies', you can continue working with the same settings.

Quit Shell

Activating this option obviously lets you leave the simulation environment. Before it does, however, you will be asked to confirm your action. If you still want to quit, and if any changes have been made during your session, FSU asks you whether you want to save the changes you have made. See also the 'Save Project' menu option. This menu option has a hot key assigned to it: pressing [Alt+Q] also activates the 'Quit Menu' option.

Simulation Controls

FSU uses some built-in defaults in the way it controls the behaviour of the actual simulation of models. Often these defaults are related to the type or amount of output that is produced by a model.

The 'Simulation Controls' window (see Figure 3.19) enables you to override these defaults. Options are grouped by application. For the 'Regional Studies' menu, there are two options of immediate interest. The first option is related to whether time-sequenced output should be produced or not. Usually, only end-of-season values of the state variables

are of interest during the 'Regional Studies' type of model runs. Per default, the time-sequenced output is not enabled, thus reducing the amount of disk-space required for the model runs. Sometimes, however, it is necessary to enable time-sequenced output. You can make it so by activating the first option ('Write time sequence information to RESULTS.OUT') in the second box. The second option in that box ('Erase temporary binary file') can be disabled if you want to be able to plot time-sequenced data using the TTSELECT plotting utility. Per default, the binary file will be erased, so only plots based on end-of-season values can be viewed.

Simulation Controls

Global Preferences for simulation and output.

Model/Exp. Studies:

- ☐ Menu model options override set-defined options (during runs by multiple set selections)

Regional Studies:

- ☐ Write time sequence information to RESULTS.OUT
- ☒ Delete temporary file after model runs

Generic options:

- ☐ Use predefined RERUNS.DAT file

« OK » <Cancel>

Figure 3.19 The 'Simulation Controls' window in the 'Regional Studies' menu.

Another option that can be interesting in the 'Regional Studies' menu is the 'Use predefined RERUNS.DAT file'. When this option is checked, FSU will use a user-defined file that specifies the number of reruns that should be made with a model and how each run differs from the other runs. This is especially useful when you have a number of data sets that you wish to present at a demonstration, or when you always want a model to run using specific sets of changes in certain parameters. After you check this option, go to the 'Create New Set(s) of Input Files' menu option: the 'Set Definition' window includes an additional button now, with which you can select a predefined reruns file.

Refer to Chapter 8 for a discourse on how the FSE system (in which the models are programmed) uses the RERUNS.DAT file and to Appendix 2 on how to set up the shell to use these predefined rerun files.

Edit A File

It can be quite handy if there is an option from within the shell to do some quick modifications of input files. The 'Edit A File' menu option will let you do exactly that. When you activate 'Edit A File', a window opens that bears some similarity to the Item Selector and enables you to choose a file for editing (see paragraph 3.5 about 'The File Selector' for detailed information on the workings of this window).

After you choose a file, the 'File Selector' window disappears and a new window opens. The latter presents you with the contents of the chosen input file, and you can make any change you like. Note that this is a very crude way of editing files, so do not expect any options like 'cutting' and 'pasting' text. All text-based (plain ASCII) files can be edited with this option; you are not limited to editing files that are part of the FSU environment or its models. Pressing [Escape] closes the window without saving changes; pressing [Ctrl+W] saves the changes first and then closes the window.

3.3.2 Model

Menu pad 'Model' lets you select a model and choose several options with regard to the model (if implemented) that can be used when you start a model run. Options are part of the model proper and not of FSU. See also Appendix 2 for detailed information on adding and configuring models.

Select Model

The 'Select Model' menu option presents you with the item-selector (see paragraph 3.4) and lets you choose between the models listed. When available, you can get information about a model by moving to the desired model, then [Tab] to the [Info] button and press it. A window should open with some information about the model. Depending on the author of the text, this may range from rudimentary information to long background stories about the history and complete context of the model.

To select a model, move the highlighting bar over the name of the model and press [Enter] or double-click the mouse to select. Press the [OK] button to activate your selection.

Select Model Options

Some models incorporate different production environments or alternative formulas for a certain process. These can be set in this menu option. Note that this option is only available when you have previously selected a model. After activating this option, a window will pop up and present you with a list of available options. This list might look like Figure 3.20, or it might have another structure and show you first some main groups in which you can activate certain settings (Figure 3.21).

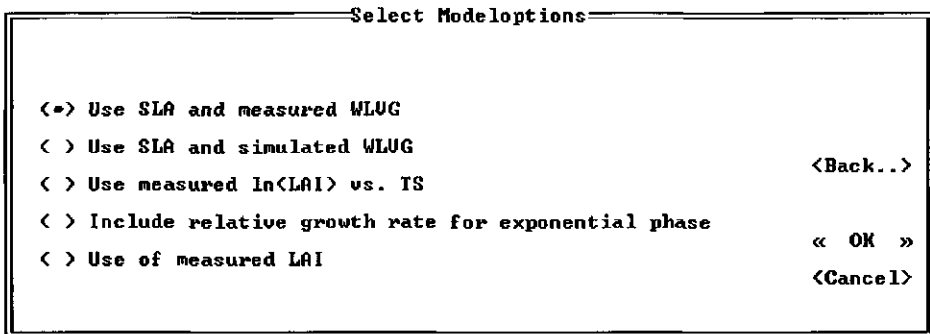


Figure 3.20 An example of the actual model options that can be selected by the user. This particular screen is configured by the ORYZA_N model (Drenth et al., 1995), which is not part of the standard distribution of FSU.

In the above Figure 3.20, you move to the setting of your choice and press [Space] to activate it. [Back..] lets you go one level up to the previous screen. If the [More..] button is selectable, the option you selected has more settings available. Press [More..] to show those settings.

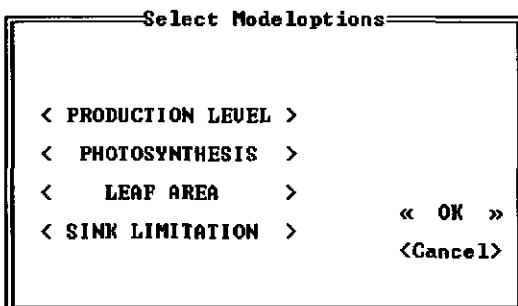


Figure 3.21 An example of some main sections of the model options. This particular screen is configured by the ORYZA_N model (Drenth et al., 1995), which is not part of the standard distribution of FSU.

In the second type of screen (mostly used in more complex models), you move to the push button of your interest (for example [Production Level]) and press it. You will be presented with a new screen with model options, all related to the topic of the button you just pressed. When you have made all the modifications necessary, press [OK] to accept the changes you made or press [Cancel] to undo.

Note that the manner in which model options are presented to you is the responsibility of the model developer. A model has to be properly documented and configured to be installed in the shell. See Appendix 2 for further information.

3.3.3 Input

After you select a model, the 'Input' menu pad becomes available to you. Use the 'Input' menu pad to create sets of input files, to select these sets, or to create a reruns file based on parameter modifications. A set is a small file used by FSU to keep track of the combination of input files and model options that you wish to run the model with. When given a descriptive name, a set can also be used to quickly retrieve specialised input. The steps to take in selecting your input are to first create a set with the 'Create New Set Of Input files' menu option and then to select that set using menu option 'Select Existing Set Of Input Files'.

Create New Set(s) Of Input Files

After you select a model and its options, this is the second step you should take if you do not have yet a prefabricated set of input files. The 'Create New Set(s) Of Input Files' enables you to select the various input files and store that specific combination of input files for later use. Together with the combination of the input files, the current settings of the model options are also stored. The types of input files that can be selected are model-dependent. Some models only need weather and crop data, others also require soil data or cultivar data of a second crop. The 'Set Definition' window only lists those types of input files that are relevant to the current model. Figure 3.22 displays a typical example of the 'Set Definition' window.

At this point you need to know the difference between a single-run set and a multiple-run set. The first type of set has for each file type (e.g., crop file, soil file, weather file, and timer file) only one selected file. The names of the files are displayed at the right side of the window. This type of set requires no entry for the starting date (the [STTIME] button, see later). By running the model with this set, you typically get only a one-year run. Note that the status text on your screen reflects the actual selections you have made. Also, when you create and select a single-run set, you typically get output that can be used to plot the variables against time. A multiple-run set can be used to simulate more than one specific environment or condition. If you select more than one file for a specific file type (for example, both PHIL4.983 and PHIL4.984 as weather files), a set automatically becomes a multiple-run set. When you select more than one file, the file name is not displayed but rather the word '*MULTIPLE*' can be seen. Furthermore, when you enter any starting times (the [STTIME] button), this also results in a multiple-run or rerun set. When a multiple-run finishes, you are able to create scatter plots, normal frequency and cumulative frequency plots of the end-of-year states of the variables.

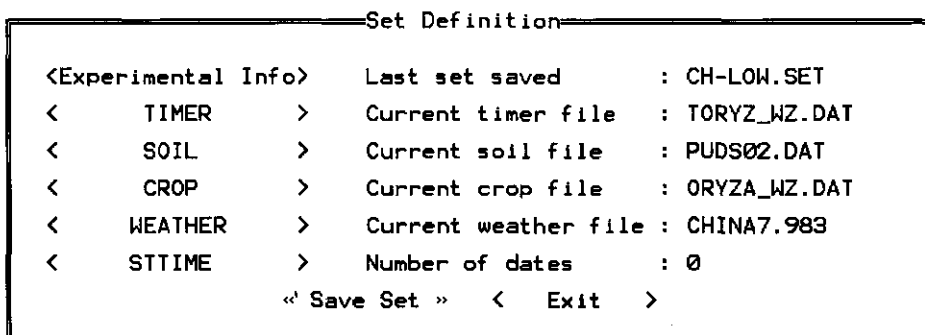


Figure 3.22 The 'Set Definition' window in the 'Regional Studies' menu.

The push buttons on the left-hand side of the 'Set Definition' window consist of the [Experimental Info] button and a number of buttons named after the type of required input files. In the above example, these types of input files are [CROP], [SOIL], [WEATHER], and [TIMER]. Pressing the buttons related to input files accesses the Item-Selector and lets you choose the files that belong to that specific type of input file.

The [Experimental Info] button presents you with a list of files. These files can be used by researchers to describe experiments or treatments in detail. Selecting one of these files will open a window with the contents of this file. The window can be moved or re-sized (for mouse users) in such a way that both the content of the window and the 'Set Definition' window are shown on the screen, making it easy to use the information within the first window to choose input files accordingly in the latter.

When you have chosen a file in one of the input file-related buttons, the actual file will be shown at the right side of the window. If you make a multiple selection in one of these buttons, the word '*MULTIPLE*' is displayed.

The 'Set Definition' window in the 'Regional Studies' menu always has an additional button: [STTIME]. This button can be used to enter a number of starting times of the simulation (in most models, starting time equals sowing date or transplanting date; check the documentation of the model, though). If you make a single-run set, there is no need to push this button.

As long as you make only one selection for each file type and do not enter a starting date [STTIME], you create a single-run set when you press the [Save Set] button. You will be asked to enter a name for this set. Use an easy to recognise name, preferably related to the selected conditions, cultivar or site. Press the [OK] button to save. If a set already exists with the same name, you are presented with a choice: either overwrite (press [Yes]) or choose another name (press [No]). If you select more than one file at a file button (for example, the [Weather] button), the sets you are going to save will be multiple-run sets. This is also the case for the [STTIME] button. The starting date is always extracted from

the timer file (chosen under [Time]), any dates entered with the [STTIME] button will generate additional model runs with those starting dates.

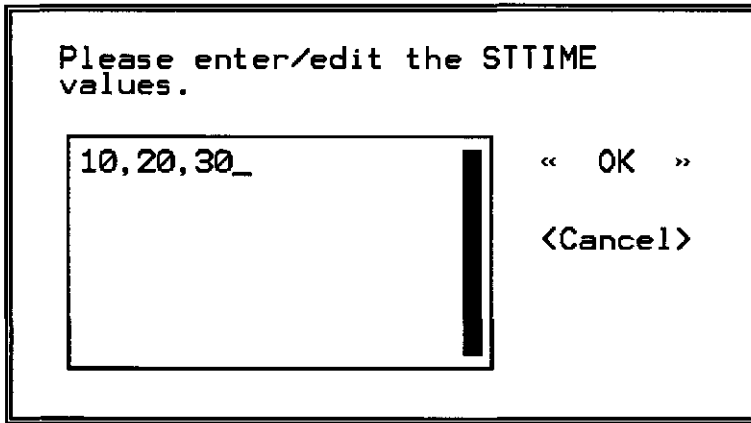


Figure 3.23 The 'Enter STTIME' window, a part of the 'Set Definition' window.

When you press [STTIME], you are presented with a new window (Figure 3.23). Jump to the rectangular box and enter your dates in the following format:

<date1>, <date2>, <date3>

For example:

4,5,9,10

This will generate additional runs with the Julian starting dates 4, 5, 9 and 10. It is also possible to enter ranges:

<first date> - <last date>

For example:

4 - 20

All Julian starting dates 4, 5, 6, 7 until 20 will be used. An experienced user can also mix single dates and ranges. For example, you wish to run the model with sowing dates 4 until 9, 11, and 15 until 20. To make it so, enter:

4 - 9, 11, 15 - 20

Choose the [OK] button to activate the starting dates you just entered, or press the [Cancel] button to return to the 'Set definition' window without activating any changes you might have made. At the right side of the [STTIME] button, you notice a text field which designates the number of starting dates you have entered. Any number higher than zero means that a reruns set will be created when you save. The number should be zero if you do not want reruns.

Up to two input types¹⁴ (a file or the starting date) can be designated as being '*MULTIPLE*' at the same time. In that case, the shell will prepare a reruns file in such a way that all combinations of the files selected will be executed by the model. The first input type listed in the 'Set Definition' screen is taken as the 'master': it is nicely sorted within the reruns file. For each value listed of the first input type, the second input type will change quickly from the first to the last value. For example, suppose that for the weather file PHIL1.984 and PHIL1.986 have been selected (the first input type), and for STTIME, the values 40 and 80 have been entered. The shell will set up the model in such a way that the following combinations will be executed:

```
PHIL1.984,  STTIME = 40
PHIL1.984,  STTIME = 80
PHIL1.986,  STTIME = 40
PHIL1.986,  STTIME = 80
```

The number of different files (or values in case of starting time) that can be selected is not limited to only two. The only restriction is a memory limitation set by the model itself. More about this limitation can be found in Chapter 8. Normally, you will not encounter this limitation.

Note that in a set, besides the input files needed to run the model, special options related to the model are also stored automatically. These options are chosen via the 'Select Model Options' window. If you want to make sure that the model runs with the preferred options, then you have to select the model options first, and then create the set. After you have saved a set, you can either define a next set or press both [Exit] or [Cancel] to leave the 'Set Definition' window and to return to the 'Input' menu pad.

Select Existing Set(s) Of Input Files

When you activate this menu option, the Item-Selector window appears. You now have the opportunity to select the set (of input files) you want. If you do not remember exactly which set to use, press the [Info] button to examine the contents of a set (see also the paragraph 3.4 on the Item-Selector).

¹⁴ FSU does not allow you to select more than two input types. If you try to do so anyway, FSU notifies you of your mistake and will not add the new input type to the list of combinations.

```

System Model Input Run Output Help
-----
Menu REGIONAL STUDIES
Currently active selections:
Model name      : ORYZA_WZ.EXE
Forc.Func.      : None
Runtype         : Multiple by Weather and Starting Time
Selected set    : MURW_TIM.SET
Input files in set : MAY.DAT, LCLAY.DAT, ORYZA_WZ.DAT, PHIL1.979 (first run)

```

Figure 3.24 An example of the status screen, showing the active settings.

If you already made a previous selection of a set, and you then press the menu option 'Select Existing Set(s) Of Input Files' again, a window appears. This window warns you that selecting a new set will undo your previous selection. You can press [No] to exit or [Yes] to continue. By selecting a new set, you will see that (part of) the content of the set will be shown on the screen (see Figure 3.24), and also the type of model run.

You cannot select more than one set in 'Regional Studies'. Instead, you can select a so-called multiple-run set.

Generate Monte Carlo Rerun File

Sometimes you might want to analyse the sensitivity of a model for changes in one or more parameters, or you might take it a step further, being interested in uncertainty analysis of the yield of a crop given special environmental conditions. This menu option enables you to do so. It enhances the input of a model in such a way that runs can be automatically made with a wide range of parameter-value combinations¹⁵.

The procedure to do so is as follows: make a decision on which parameters to vary. For each parameter, gather information about the biologically relevant range in values. Choose a statistical distribution from which new values will be generated or enter actual values from which a random selection will be made. Enter the number of model runs you like to do and generate the parameter values. Check the distribution of the values by table or by graph. If these values are suitable, run the model. Afterwards, examine the output carefully by graph. A complete example of this procedure can be found in Chapter 2. This paragraph, however, will give you an in-depth description of the generation of a Monte Carlo rerun file. Figure 3.25 gives you an idea of how the screen may look like once you have made some selections.

¹⁵ To use the 'Generate Monte Carlo Rerun File' menu option, you need to set up a model for this particular feature. Not all the parameters are of interest, and sometimes certain modules within the model use parameters other than what other modules use. Please refer to the 'The model definition file' paragraph on page 200 for detailed information.

Use/Generate Monte-Carlo drawings			
<div> <div>SPSOIL</div> <div>STTIME</div> <div>DTRP</div> <div>NH</div> <div>NPLH</div> <div>NPLSB</div> <div>NPLDS</div> <div>WLOXI</div> <div>RIPUD</div> <div>DVSIE</div> </div> <div> <div>NORM</div> <div>UNIF</div> <div>BETA</div> </div>		<div>Select method:</div> <div>Normal distr</div> <div>[X] Correlation</div> <div>Nr. draws: 100</div> <div>ISEED : 0</div>	<div>Enter distribution parameters:</div> <div>Upper limit: 1.0</div> <div>Lower limit: 0.0</div> <div>a: 1.0 Mean : 40</div> <div>b: 1.0 Variance: 8</div> <div>Measured values: 0.</div>
<div> <div>< Generate.. ></div> <div>< View Table ></div> <div>< View Graph ></div> <div>< OK, use it ></div> <div>< Cancel ></div> </div>			

Figure 3.25 The Monte Carlo rerun generation screen.

The left-side box lists the available model parameters and displays information on the distribution chosen. In Figure 3.25, for example, the model parameter STTIME is associated with a normal distribution, meaning that values for the model parameter will be drawn accordingly (mean value 40, variance 8). On the right side of the window, the values for the distribution parameters are displayed. The values that are accessible (in Figure 3.25 mean value and variance) depend upon the distribution chosen. In the model parameter list box on the left side of the window you can see immediately for which model parameter the distribution values are displayed. In the case of Figure 3.25, it is STTIME: the selection bar is positioned on STTIME.

As you move around in the list box with the cursor keys, the accessible fields and the values change: they will always reflect the current selected model parameter. In the case that no distribution has been chosen, no fields are accessible on the right side. You can include a model parameter for value generation by selecting a distribution for it. Move to the model parameter of your choice (place the selection bar on top of it), then press [Tab] to move to the [Method] list button. Press [Enter] to display the list of available distributions. Activate one of the distributions and press [Enter] again. An abbreviation of the selected distribution now appears at the right side of the model parameter and one or more fields in the right side of the window become accessible. Enter the distribution parameters in these fields. If you enter invalid data, the shell displays a warning and you must correct the error before you can leave that particular field. One of the warnings you might encounter is when you enter a lower value in the 'Upper Limit' field than the one that is specified in the 'Lower Limit' field.

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 10.1.2).

Except for the uniform, beta and normal distribution there, also exists a special 'distribution': measured values. Whenever you have values for parameters that were collected at or derived from experiments, you can use them to generate new values. Enter the values, separated by a comma, in the 'Measured Values' box. You can specify up to 500 values in this box. You can have a maximum number of 10 model parameters chosen for each distribution, except for the measured values 'distribution' which can be assigned to a maximum of 5 model parameters. These limits are defined by the RIGAUS program (Bouman & Jansen, 1993) which is used to generate the rerun values of the model parameters. After you set all the model parameters and their distributions, move to the 'Nr. Draws' field to enter the number of reruns (model parameter values) that you want to generate. By default, the number of 'draws' from the distributions is set to 100. The 'ISeed' field is the seed for the random generator of the RIGAUS program. If you set it to 0 (zero), it will use the internal time of the computer as seed. Set it to another value if you want reproducible drawings. Move to the [Generate] button and press it to start the RIGAUS program: it only takes a couple of seconds to finish. In the case that something went wrong, FSU warns you with a beep and a message: you should check the error message of the RIGAUS program that is displayed on the background screen (or press [View Table] to see the errorlog).

After a successful generation, you can press [View Table] or [View Graph] to inspect the results. [View Table] starts the viewer and presents you with the contents of the RERUNS.DAT file (the actual file that the model will use to run), with a columnar file that lists all model parameters and their values per run and a file that lists warning messages of RIGAUS (if any). The distribution of the generated values can be shown graphically by choosing [View Graph]. See also paragraph 'View Graphs' on page 66 for detailed information on the graphs screen. The [View Graph] button enables you to plot frequency, cumulative frequency, and scatter plots of the model parameters. When the distribution of model parameter values does not meet your requirements, choose new distribution types or alter the distribution parameters and generate the rerun file again. Inspect it, and when you are 'happy' with the generated distributions press [OK, use it] to store it for use with the model. You return to the 'Regional Studies' menu and you will notice that the status reflects the changes you have just made: it should display '(Monte Carlo runs)' as the type of model run. When you run the model, it will use the generated reruns file. After the successful completion of the model runs, you can plot the results with 'View Graphs' in the 'Output' menu. For example, you can create a scatter plot of the values of a certain parameter against a state variable to (in a global way) inspect the relationship between modification in parameter values and the yielded model output.

Pressing [Cancel] in the 'Use/Generate Monte Carlo drawings' window causes the shell to omit all changes made in this session of the rerun file definition and to force a normal 'Single run'. If you still want to do a Monte Carlo rerun, then you have to return to this menu and press [Generate] to use the settings that you defined before the previous session. In this way, you will not unintentionally run the model with a rerun file that is out of date. There is only one option in the 'Use/Generate Monte Carlo drawings' window that has not

yet been described: the [Correlation] check box. This is a very special option that has its use only for a specific model in non puddled soils. For other models, this option is not functional. Please refer to Bouman & Jansen (1993) for an in-depth description of this option.

3.3.4 Run

The 'Run' menu pad has only one menu option, 'Run model', and can only be activated when both a model and a set have been chosen.

Run Model

When you choose this option, a window appears and asks if you are sure that you want to run the model. Press [Yes] if you are; otherwise press [No]. If you press [Yes], the shell checks if the necessary input files are available and updates or generates other required input files for the model. Please read Appendix 4 to learn more about this. On successful completion of the above procedure, the model will be executed.

3.3.5 Output

One of the most important menu pads is the 'Output' menu pad. This pad enables you to take a peek at the output data which the model generated and makes it possible to show some basic types of graphs.

Store Output!

After you have run the model and are satisfied with the data it produced, you may want to store the output of the model. This enables you to use external packages like statistical programs or professional plotting packages in the further analysis of your data. Moreover, it keeps data safe. FSU will clean up the working area each time you leave the shell or choose another set or model. Cleaning will irrevocably erase the output data produced by a model, except if you have saved it previously. Input data are not affected.

To save data, activate menu option 'Store Output!' and then type a name under which you like to keep your data safe. The name will be used to create a directory in the save path (see Appendix 4) and to copy the data to this directory. If such a directory already exists and if data are already stored into this directory, the shell informs you about this and asks you to either overwrite the data already stored or to [Cancel] and try another name. For your information, a maximum of five files will be displayed that are at risk of being overwritten.

Restore Output!

Sometimes you wish to re-examine some output of a specific model run. Choose the menu option 'Restore Output!'. An item-selector pops up and shows you a list of names. These are the names under which you saved the output, with an extra extension of '. SAV'. So if you saved under the name 'IRLOWLAN', choose 'IRLOWLAN.SAV' to restore your data. If you have never stored any data, FSU informs you of your mistake. Also, if data files are still located in the working area, the shell displays a warning message and asks you if these files can be overwritten ([Yes]) or not ([Cancel]).

After FSU restores the output files, you can re-examine them by the 'View Tables' and 'View Graphs' menu options.

08-17-95 18:07 • OP.DAT						
RUNNUM	EVSACU	RYTCU	RAINC	SPCU	GRDUR	HRR
1	134.11	600.00	671.00	521.33	118.00	8964.6
2	134.11	600.00	671.00	521.33	118.00	8964.6
3	182.35	650.00	689.00	512.04	116.00	8530.2
4	171.63	600.00	758.50	520.51	117.00	8335.2
5	143.35	550.00	672.20	513.07	118.00	8775.3
6	148.94	600.00	669.00	524.23	118.00	8616.1
7	169.67	700.00	740.10	532.93	119.00	8721.1
8	171.63	600.00	758.50	520.51	117.00	8335.2
9	153.63	550.00	690.40	510.45	118.00	8420.0
10	169.11	550.00	777.20	518.00	117.00	8185.2
11	149.61	650.00	709.00	528.57	119.00	8922.5
12	139.40	600.00	684.00	522.60	118.00	8836.3
13	169.80	600.00	714.50	520.95	117.00	7989.3
14	130.90	650.00	645.00	521.08	119.00	9169.7
15	143.35	550.00	672.20	513.07	118.00	8775.3
16	168.37	600.00	725.20	512.79	117.00	8023.7
17	128.55	650.00	654.40	524.17	119.00	9180.2
18	175.14	600.00	730.00	513.60	117.00	8658.8
19	149.61	650.00	709.00	528.57	119.00	8922.5
20	171.63	600.00	758.50	520.51	117.00	8335.2
21	166.25	550.00	717.00	507.20	116.00	9665.4
22	130.91	600.00	677.10	519.32	118.00	8951.4
Command: *** top of file *** Keys: PgUp PgDn F10=exit F1=Help						

Figure 3.26 The 'View Tables' menu option shows the contents of the output files. In the above example, the end-of-season results are shown (file OP.DAT).

View Tables

After a successful model run, several files are present in the working area that are part of the model output. These files are RESULTS.OUT, RESULTS.LOG, WEATHER.LOG, and OP.DAT (optional). The first file is always generated, but only after a single run will this file be of interest to you. It lists the output variables for each time step. The RESULTS.LOG file contains messages from the rest of the model (e.g., an overview of warnings that have occurred; in the case of multiple runs, it also logs the replacements of parameter values). The headers of the weather files are listed in the WEATHER.LOG file, together with warnings and errors from the weather system (only if applicable). The last output file will only be present if a model has been set up to facilitate the output-processing routines that are generally affiliated with the 'Regional Studies' menu. It contains the end-of-season values for some important state variables. Please read

Appendix 2, the paragraph about 'OPSYS: end-of-season values system' for more information on this subject.

When you activate menu option 'View Tables', a viewer¹⁶ will be started. This viewer shows you the contents of each output file. An example of the output can be seen in Figure 3.26.

Graph Options																			
Select graph type:		Independent variable	Dependent variable																
<input type="radio"/> Scatter plot <input type="radio"/> Frequency plot <input type="radio"/> Cumulative frequency plot		STTIME	WRR																
<input type="checkbox"/> Autoscale		<table border="1"> <thead> <tr> <th colspan="2">Enter X-range</th> <th colspan="2">(and Y-range)</th> </tr> </thead> <tbody> <tr> <td>Min X :</td> <td>0</td> <td>Min Y :</td> <td>0</td> </tr> <tr> <td>Max X :</td> <td>15000</td> <td>Max Y :</td> <td>15000</td> </tr> <tr> <td>Step X :</td> <td>3000.0</td> <td>Step Y :</td> <td>3000.00</td> </tr> </tbody> </table>		Enter X-range		(and Y-range)		Min X :	0	Min Y :	0	Max X :	15000	Max Y :	15000	Step X :	3000.0	Step Y :	3000.00
Enter X-range		(and Y-range)																	
Min X :	0	Min Y :	0																
Max X :	15000	Max Y :	15000																
Step X :	3000.0	Step Y :	3000.00																
«View Graph»		< Cancel >																	

Figure 3.27 The 'Graph Options' screen of the 'Regional Studies' menu.

View Graphs

You may want to display the results of a model run in a graph. To do this, choose the 'View Graphs' menu option. FSU uses the TTSELECT plotting utility to create graphs. For detailed information on the TTSELECT utility, please refer to paragraph 3.6 on page 70. Depending on the type of model run, you will be either guided to the TTSELECT plotting utility (single run) or be presented with a graph options screen. In the first case, you can plot simulated output variables against time or against each other to examine possible correlations. More than one variable at a time can be plotted against the independent variable, so comparisons between variables are easy to display. After a multiple run, the 'View Graphs' menu option presents you with the graph options. (Figure 3.27). You can plot the end-of-year values of the output variables and, in case of Monte Carlo reruns, also the modified model parameters for each run¹⁷. The 'Graph Options'

¹⁶ FSU uses an external viewer for this purpose, the List 7.1a program by Vernon D. Bueg. At the top of the screen you note the file name. Use the [Arrow] keys to scroll through the file. [Ctrl+PageUp] and [Ctrl+PageDown] list the previous and next file, respectively. [Alt+X] or [F10] leaves the viewer.

¹⁷ In the event of normal multiple runs defined in the 'Set Definition' window, only the end-of-season output state variables are listed for each run (including the default = first run). After a Monte Carlo rerun, using the file created in the 'Generate Monte Carlo Rerun File' menu option, both end-of-season state variables are listed as well as the parameters that were used to do the reruns. The very first run (the default run) is

screen lets you choose the type of graph, the name of the variables to be plotted, and some scaling options.

To begin with the latter: each graph that you plot will be auto-scaled by default. This means that you do not have to worry about the minimum and maximum value and the step size. However, sometimes you wish only to see part of the graph or to specify more classes in case of a frequency plot. To do so, disable the auto-scale feature by pressing the [Autoscale] button and enter the desired values in the scaling fields. [Scatter plot] enables you to plot one variable against another. Note that the second list (dependent variables) can only be accessed if scatter plot is selected. [Normal Frequency] creates a frequency distribution plot of the end-of-year values (or parameters used in Monte Carlo reruns). If you do not use the [Autoscale] option, then you might want to specify the number of classes before you decide to plot. The default number of classes is set to five, which does not give a high resolution in general.

Furthermore, take a look at the minimum and maximum X (and Y, if scatter plot is selected) values. As with the number of classes, you might consider modifying them into more appropriate values. Note that on changing either maximum or minimum, the step size also changes automatically if scatter plot or cumulative frequency has been chosen. The step size determines the amount of space between the so-called 'ticks' (little vertical lines crossing the X-axis). If you like to modify the step size, first make sure that both minimum and maximum are set to the correct values. This is because the shell will update the step size again if you change either maximum or minimum after you modified the step size. To display the graph, press the [View Graph] button (see example graph, Figure 3.28). Return to FSU by pressing [Escape] once while the graph is being displayed.

It is not possible, on most systems, to print a graph directly from within FSU. Refer to Appendix 1 on how to set up your system to facilitate printing of graphs (hard copies).

When you use 'Retrieve Output!' and then 'View Graphs' you may run into minor problems. FSU does not store the settings when you use 'Store Output!': it only saves the output files. Hence, FSU cannot know what type of run you performed to get that output, and so it also does not know what kind of graphs should be activated. You may, for example, run into messages from FSU saying 'Did you really run the model?' because it cannot find the binary output file RES.BIN (in case you retrieved results from a 'Monte Carlo' rerun). You can circumvent this by reconstructing the conditions that existed prior to the model run that created the output. For a multiple run based on 'Monte Carlo' distributions that means selecting a model (any model), a single run set, entering the 'Generate Monte Carlo Rerun File' menu option, selecting a dummy variable, press [Generate] and [OK, Use it], and finally activating 'View Graphs'. This will show you again the specialised graph window for multiple runs.

not listed since it does not contain changed values for the parameters. Besides, when doing sensitivity analysis, you do not want to be bothered with 'noise' from predefined parameter values from a data file.

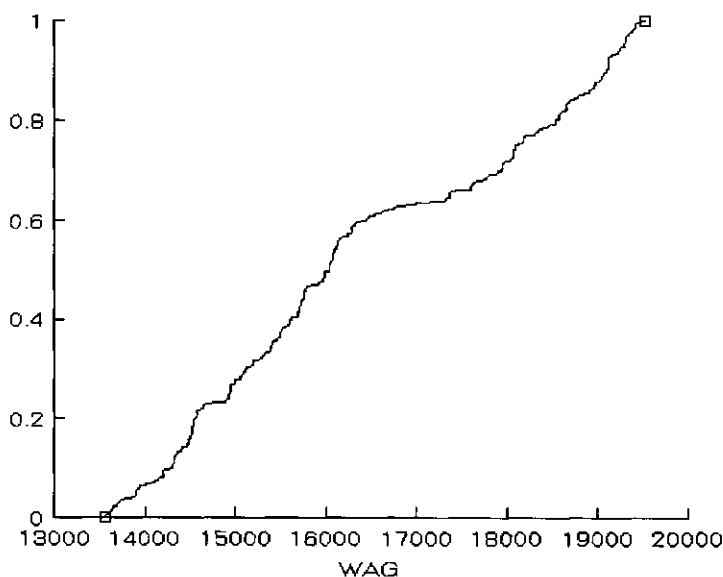


Figure 3.28 An example of the output from the 'Graph Options' window (multiple runs). This specific Figure shows a cumulative frequency plot of the above ground weight of a rice crop, created by Monte Carlo runs on the sowing date.

3.4 The Item Selector

The screen called the Item-Selector is a window of paramount importance (see Figure 3.29). It enables you to make a selection out of a given list of items. Some examples of items are models, names of input files, parameters, or directories. Furthermore, some items offer additionally the opportunity to get more information about that particular item.

The rectangular box displays the list of possible selections. If more items are present than can fit in the box in one time, small arrows appear at the right side of the box. This indicates that you can 'walk' (the correct term is 'scroll') through the list by using the arrow keys, the [PageUp] and [PageDown] keys, or by using a mouse (click on the arrows). As you walk through the list, the highlighting bar moves on top of each current item. When this bar is positioned on top of an item and the [Enter] key is pressed, that item is chosen. A square-root sign $\sqrt{}$ appears at the left side of the item. This indicates that you have successfully made that item a part of the selection. If more than one choice is possible, you can move to the next item and press [Enter] again. FSU displays a warning

message when you try to choose more items than are allowed (a window appears in the upper right corner of the screen). Most uses of the Item-Selector only allow one item to be chosen. When the selection has been made to satisfaction, the [Tab] key should be pressed until it highlights the [OK] button. Press this button to activate the selection. [Cancel] will not activate any selections that might have been made.

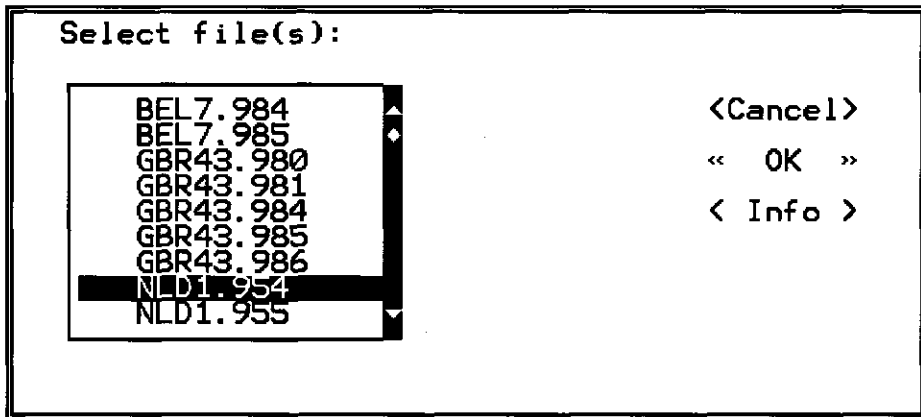


Figure 3.29 The Item-Selector as used in this example to select weather files.

When information is available for a certain type of item, the [Info] button will be enabled (i.e., not dimmed). To get the information on a particular item, move the highlighting bar on top of it and then [Tab] to the [Info] button and press it. If the information could be successfully retrieved, a window opens which contains the requested information. When more information is available than what the window can display at one time, small arrows appear in the right border of the window. Move through the information by pressing the arrow keys, or use a mouse. The information window can be exited by pressing the [Escape] key.

3.5 The File Selector

The File Selector is exclusively used by the 'Edit A File' menu option. The File Selector enables you to pick (any) file for editing. Using this selector is much the same as using the item selector. Figure 3.30 presents an overview of this window.

The left box lists the available files and directories. Items between square brackets [] are directories (folders). Move the highlighting bar over one of these directories and double-

press [Space], press [Enter] once or double-click mouse to enter that directory: a new list should appear on your screen. Press the [..] to go one directory up (the parent directory). Every item that is not enclosed between square brackets represents a file. Move the highlighting bar on top of the file of your preference and press [Enter] to open that file (or move to the [Open] button and press it). When you open a file, a new window appears with the contents of that file. You can use [Ctrl+F10] to maximise that window. [Ctrl+W] stores any modifications you have made and closes the window. If you decide to discard any changes, just press [Escape]. At the bottom of the File-Selector, a check button named 'All files' is placed. When you activate this button, all files in the current directory are shown instead of the default files (*.DAT, *.TXT, and *.SET).

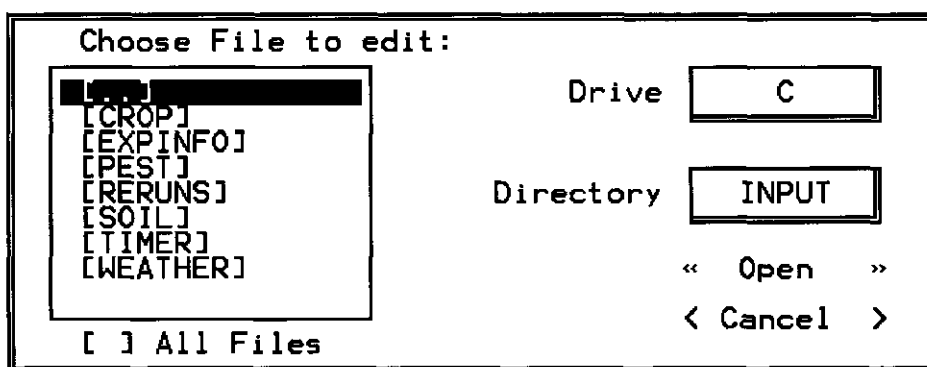


Figure 3.30 An example of the File Selector.

3.6 The plotting utility TTSELECT

The TTSELECT plotting utility (Kraalingen van, 1991b) is a program that enables you to display graphs using data that the model has stored in a binary output file. This utility is invoked by the shell to display custom-made graphs (like for example, cumulative frequency graphs or scatter plots) or user-defined runs.

In the last case, TTSELECT reads all the data in the binary output file and presents you with a list of the variables that have been found (see Figure 3.31). Then, if applicable, it also presents you with a list of the run numbers that have been made. The next thing TTSELECT will do is to ask you to enter the names of the variables you want to plot. The first variable that you enter is the independent variable, the next variable is the dependent. You can also enter more than one dependent variable. Variables should be separated by a comma (,) or a blank. If you like a custom title, enter it between quotation marks. For example:

TIME, WCR, WCR_OBS, "Simulated & Observed crop weight"

(Only if applicable: after you press [Enter], TTSELECT will ask you if you like to plot all runs, or a selection. You enter the selection as a list of run numbers, separated by blanks.). Then, TTSELECT asks you whether you wish to display only the end-of-run values. If you answer 'N' No, then you get time-coursed graphs. Time-coursed graphs are the default. After this, the graph will be shown. Press [Enter] to leave the graph. TTSELECT will ask you three questions about printing or storing graphs. The first question will ask you if you like the graph to be saved in a TTPLOT (the native format of TTSELECT) file, the second question asks you whether you wish to save it in a screendump file, and the last question asks you if you like this plot to be printed on a HP DeskJet or LaserJet printer. The first two questions are not of interest to you, the third may be if you have a DeskJet or LaserJet printer connected to your computer. You can press three times [Enter] to skip all the above three questions and return to the TTSELECT main menu. Again you will be able to enter your choice of variables to plot.

You can exit TTSELECT by pressing [Escape] until you are back in the menu of FSU.

TTSELECT version 2.21, October 1994 ...

The following variable names have been found:

```
=====
TIME      FERTMX      STPDAT      DOY      YEAR      DAT
DATEH     ROM        NL          WLV_OBS  FNLV_OBS  WCR
P         EPSIL       FSV1       FSV2     FSV       RNEFF
LNUC      GRUC        GCR        WCR_OBS  WCR_OBS   RDDSUM
RSR       RELTME      FITWCR     X         APSLOP    RECOV
NAVAIL    APCUM      SOLSUP     CUMNSS   SWIEXP    MAXUP0
MAXUP1    MAXUP2     FNMAX      MAXUP3   MAXUP4    MAXUP5
M12345    DEMAND     NUPT       ANCR      FNACT     ANSOCH
ANLVCH    ANLV        WRR        HI        WCRFF     NAPPLD
ACTREC    NHRVST
```

Supply two or more variable names separated by a comma or a space.
First variable name will be used as X-value, type <ESC> to exit.

Which variable names :DAT, WCR, WCR_OBS "Simulated and observed crop weight"

Figure 3.31 An example of the menu system in the TTSELECT utility.

PART II

MANAGE-N

4 Introduction to MANAGE-N

J.J.M. Riethoven¹, H.F.M. ten Berge¹ & H. Drenth^{1,2}

¹ DLO Research Institute for Agrobiological Sciences (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

² Department of Theoretical Production Ecology, Wageningen Agricultural University, P.O. Box 430, 6700 AK Wageningen, The Netherlands

4.1 General description of MANAGE-N

MANAGE-N is an advisory tool for rice researchers and extension workers. The overall purpose of MANAGE-N is to provide a sound agronomic basis for designing optimum nitrogen (N) fertiliser management strategies for specific combinations of cultivar, site, season, and soil type.

When supplied with the proper parameter values, MANAGE-N generates information on

- the highest attainable yield (versus fertiliser N level)
- the highest attainable additional income from applied N fertiliser (versus fertiliser N level)
- the optimal time path for N application at each N input level
- yield response to specific split N application schemes
- a site-and-cultivar specific calibration factor (FSV) expressing productivity of light and nitrogen
- patterns of N uptake, biomass accumulation, and other intermediary variables

MANAGE-N requires crop and soil characteristics (parameters) and weather data (daily radiation) as inputs. The core of the package is the dynamic simulation model ORYZA_0 (ten Berge et al., 1994c). The model describes those aspects of rice physiology that have a direct relevance to fertiliser management: N uptake, N partitioning, and N utilisation by the crop. ORYZA_0 is an explanatory (process-based) model, yet uses plant and soil characteristics that are measurable at most experiment stations.

The ORYZA_0 model is completely embedded in MANAGE-N (Figure 4.1), which is custom-made to provide easy communication with the model. The procedure of selecting input files, editing input data files, and viewing outputs (tabular and graphical) is completely supported by MANAGE-N in a user-friendly way.

Numerical optimisation of nitrogen application patterns that result in maximum grain yield is fully automated: several user-defined N input levels can be optimised in one go. Based on the outcome of these optimisations, MANAGE-N calculates grain yield response to N

input level, displays the optimum timing of fertiliser N applications, and determines economic optimum based on additional income of the individual farmer.

Another module offers opportunity to assess effects of discrete split schemes on crop performance. These discrete splits can be derived from (in the user's mind) optimised continuous N applications.

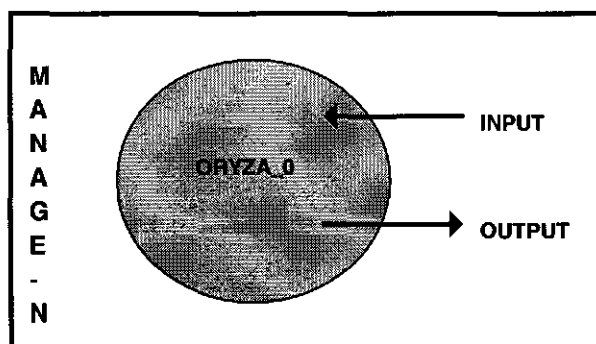


Figure 4.1 MANAGE-N: a user-interface to facilitate running the ORYZA_0 model for different purposes.

4.2 Using this guide

This section describes how to best use this guide. It outlines the manual's intended audience, the audience's prerequisite knowledge, and the manual's organisation and its conventions.

4.2.1 Audience

MANAGE-N and its documentation were written with rice researchers and extension functionaries in mind. This manual assumes that you have at least a basic knowledge of crop growth processes and agronomic principles. Furthermore, some elementary background on computer use is highly recommended: you should be familiar with keywords like 'keyboard', 'mouse', 'window', and 'disk-drive'.

4.2.2 How to use this guide

This paragraph provides a summary of the manual's organisation and content. The manual's Chapters are described, followed by a section on how to use each Chapter.

Chapter 4 'Introduction to MANAGE-N'

gives some background information on MANAGE-N in general and gives a brief overview of the capabilities of MANAGE-N. Also explains conventions used in this manual and describes the kind of control structures (like windows, menu bars) that appear on the screen. The third part of this Chapter lists the requirements needed for using MANAGE-N. The description of the quick installation procedure of MANAGE-N and how to get help from the authors form the last parts of this Chapter.

Chapter 5 'The theory of optimising the N application pattern by MANAGE-N'

provides background information "along the track" of the N optimisation procedure. Explains how MANAGE-N uses the ORYZA_0 model to accomplish optimised N applications and describes in summary how the ORYZA_0 model itself is structured. Parameter derivation is discussed: how to get the parameters needed for the ORYZA_0 model. Computation of the FSV factor is explained and so is the precise procedure followed when optimising N application curves.

Chapter 6 'Optimising the N application patterns by MANAGE-N, an example'

guides the user step by step through the whole procedure of generating a practical advice on the timing and size of N dressings for increased final yield. This Chapter will start with the creation of data files per treatment and use of these data files to compute the site/variety match factor FSV. It then continues with storing a mean FSV value in a so-called generic data file. That data file is used to let MANAGE-N optimise the N application scheme for several N input levels. The user will be guided through the split evaluation section: breaking down the recommended continuous N application curve into discrete splits and evaluating what effect split sizes and timing have on the final yield. The best and most feasible split-dressing is then taken as an advised application scheme for the given site under study.

Chapter 7 'MANAGE-N Reference guide'

lists all the menus, menu options, and windows in a systematic way and provides detailed information on the functions of these items.

Appendix 5 'Problems that may occur with MANAGE-N or FSU' and Appendix 7 'Problems with MANAGE-N'

give information on causes of common problems and how to resolve them. Appendix 5 lists common problems that can arise due to computer systems that are not properly set up, or problems specific to the Microsoft® FoxPro® environment in which MANAGE-N is programmed. Appendix 7 lists problems specific to the MANAGE-N system.

Appendix 8 'Sample input files used by MANAGE-N'

lists all important (data)files of MANAGE-N. Contains examples of input files for the ORYZA_0 model and its derivatives.

Appendix 9 'Acronyms used in ORYZA_0'

lists all acronyms that are used in the ORYZA_0 model. That includes acronyms that are not usually found in the output files generated by the model.

Appendix 10 'Required input for the MANAGE-N system'

cross-references the required input (parameters) per *method*. The second part of this Appendix lists the required parameters together with a short description.

Appendix 11 'MANAGE-N parameter derivation and EXCEL® sheet'

provides information on how to derive parameters that are required for MANAGE-N.

Appendix 12 'Description of MANAGE-N input per method'

describes required input in more detail as the previous Appendices. Lists shortly, together with description, per *method* the parameters that are necessary.

Appendix 13 'ORYZA_0 differences'

summarises differences and modifications that have been made to the ORYZA_0 model since Drenth et al. (1994) was published. Most of these modifications were made because of feedback we got from fellow researchers in the SARP project. Furthermore, adaption had to be made to integrate ORYZA_0 into the MANAGE-N system.

Appendix 14 'FST source of the ORYZA_0 model'

lists the FST source code of the ORYZA_0 model and explains where the FST source code can be found on the distribution disks of MANAGE-N.

4.2.3 What you should read

The following table points to specific Chapters in this manual for information on particular topics. For further documentation and literature, refer to 'References' on page 185.

Table 4.1 Pointers to text about MANAGE-N.

For information on	You should read
MANAGE-N in general	Chapter 4
Installation of MANAGE-N	Chapter 4, paragraph 4.5 (and maybe Appendix 6, 5, and 7)
Using the menu system	Chapter 4, paragraph 4.3
Contacting the author of the shell	Chapter 4, paragraph 4.6
The theory behind N optimisation	Chapter 5
Using MANAGE-N as a research tool (examples)	Chapter 6
In-depth overview of all the options and features	Chapter 7
Solving common problems	Appendices 5 and 7

4.3 Conventions

When the manual mentions a 'key', it means a key that you can find on the keyboard of your computer. A 'button' means a button in a window, like the [OK] or [Cancel] button. Keys and buttons will be enclosed in square brackets [] in this manual. The text between those brackets should be taken as a whole, not as separate keys. For example, [Backspace] means: press the key (or button) defined as backspace and not the word 'backspace'. Sometimes a combination of two keys is used, like for example in [Ctrl+X]. This means press the [Ctrl] key and keep it pressed while you press the [X] key.

The names of menus and menu options are enclosed between '....', like for example, 'Edit A Crop/Soil File'. Menus and menu options are explained in paragraphs 4.3.1 to 4.3.5.

When the manual mentions *method* (written just like this), it aims at one of the four steps that are taken toward generating an optimised N-dressing: 'Fit to Match', 'Single Run', 'Optimise N-curve', and 'Split Evaluation'. All these *methods* are listed as menu options in the 'Method' menu pad.

Names of parameters, variables, files and programs are written in COURIER script to distinguish them from the normal text.

A window is a square box on the screen which contains options that you can activate or edit (see also paragraph 4.3.5).

In this manual you will also encounter words like 'selecting', 'choosing', 'activating', etc. The following section defines those actions.

‘Selecting’ a menu option is done by moving the so-called ‘highlighted’ bar on top of a menu option or an item. ‘Selecting’ is done only by using the [Arrow] keys. When you are sure a certain item or menu option should be activated, then you press [Enter] or the [Spacebar]. ‘Activating’ an option means that an action will be carried out. This is the same as ‘choosing’. An action can be something like running the model or marking (=selecting) an item (for example, an input file) for later use.

‘Pressing’ and ‘pushing’ a button means move the highlighting bar on top of the button and press [Enter] or [Spacebar].

The MANAGE-N tool is heavily based on menus and windows to provide a user-friendly interface between you and the system. If you have no previous experience with menus, please read the next paragraphs thoroughly. It should give a quick understanding on the use of menus, windows and all kinds of buttons. Operating the menu is easiest if you use a mouse. If you are not the lucky owner of such a comfortable device, some functions related to moving and resizing windows are not available.

The MANAGE-N menu system consists of the following parts: menu bar, menu pads, menu popups, and menu options. The menu system provides a quick and easy-to-use interface with the crop model ORYZA_0. Each part of the menu system is described below.

Accessing the menu is described for keyboard only, the mouse variant is a simple point and click.

4.3.1 Menu bar

The menu bar is located along the top of the screen. The menu bar displays menu pads with names; these pads give access to menu popups.

4.3.2 Menu pads

Menu pads appear on the menu bar and display the names of the menu popups. Sometimes, certain menu pads appear dimmed (‘ghosted’) and cannot be chosen. These menu pads are disabled.

To access the menu bar, press the [Alt] key. One of the menu pads will appear highlighted¹⁸ because it is selected. Press the [Right Arrow] and [Left Arrow] keys to move from menu pad to menu pad. Or use the so-called hot key of a menu pad. The hot

¹⁸ Highlighted text appears in contrast to its surroundings on the screen (lighter or darker) so that it stands out from the surroundings. Highlighting something typically indicates that it is selected or about to be chosen.

key is a highlighted character and is usually the first letter in the menu pad name. The [Tab] key can be used to jump from menu pad to menu pad.

4.3.3 Menu popups

Menu pads control menu popups. Menu popups are lists of related options. When you choose an option from a menu popup, you are telling MANAGE-N what action to take. 'Choose' means to activate a selection (highlighted option) by pressing the [Spacebar] or [Enter].

To display a menu popup, press [Alt] and then press the hot key in the menu pad name. If you are already positioned at the menu bar, you can display a menu popup by pressing the hot key in the menu pad name. You can also press the [Left Arrow] and [Right Arrow] keys until the menu pad is selected, then press the [Spacebar] to display the menu popup. Once a menu popup is displayed, you will usually choose an available option, as described in the next section. If you wish to deactivate a menu popup without choosing an option, press [Escape].

Of course, you cannot display a menu popup of which the menu pad has been disabled.

4.3.4 Menu options

Menu popups contain options. The options on each menu popup are logically related to the menu pad name. On a single menu popup, options may be further grouped to indicate that they produce similar outcomes. These groups are separated by divider lines.

Some menu options have a [Alt] key shortcut listed next to them on the popup. You can use a [Alt] key combination to choose the menu option without displaying the menu popup. For example, exiting MANAGE-N can be done by pressing [Alt+Q].

Sometimes, a menu option appears dimmed and cannot be chosen. This menu option is disabled.

Before you choose a menu option, use one of the methods discussed in the previous section to display the menu popup. Once the popup appears, choose an option in one of the following ways:

- Press the hot key for the option.
- Use the [Up Arrow] and [Down Arrow] keys to select the desired option, then press [Enter] or double-press the [Spacebar].

When you choose a menu option, an action occurs. A window may open or close, a switch may be set, a dialog may appear, or an external command may be executed.

4.3.5 Windows

The MANAGE-N tool uses windows to communicate with you when extra information is required to perform an action. A window is a rectangular box, possibly containing various items like text fields, buttons, or lists. These items will be explained in detail below. As with the menu pads, you jump from item to item within a window by pressing [Tab] or [Shift+Tab]. Currently selected items will be highlighted. The following items can be found in a window:

Push Button

A push button is defined on the screen by angle brackets <...>. If you press this button, an action will occur immediately. What kind of action this is, is defined by the text placed inside the push button. For example, when you press an [OK] push button, you agree with a certain kind of action (described somewhere else in the window) or you think you have filled in all inputs correctly. This may sound confusing right now, but when you read the examples, this will be clear.

You select a push button by jumping to it ([Tab] key). When the push button highlights, press [Spacebar] or [Enter] to activate it. Owners of a mouse activate a push button by positioning the mouse pointer at the push button and then clicking once.

A special push button is the confirmation/continuation button. It is defined by the double angle brackets <<...>>. Press [Ctrl+Enter] to activate it. There is no need to highlight it first. This button will immediately continue with an action. Another special push button is in effect: all windows have a default 'cancel' or 'no' button. This button cancels all actions or choices that you have made in that particular window and restores your latest settings (if any). All windows include this button (it is not especially marked, the [Cancel] button is normally used for this purpose) , press [Escape] to activate it.

Check Box

A pair of square brackets on the screen defines a check box. This type of box toggles an option or setting. If the box has a X in it, the option is turned on, otherwise it is turned off. Press [Tab] to move to the box (it highlights when selected) and press [Spacebar] or [Enter] to toggle.

Radio Button

This type of button is defined by a pair of parentheses followed by a text. These buttons are used when you select an option that is mutually exclusive with other options. Press [Tab] to move to the button and press [Spacebar] or [Enter] to activate it. A bullet • appears between the two parentheses. If another option (within that group of radio buttons) was selected, the bullet disappears from that button.

Input Text Field

For some actions you must supply the MANAGE-N tool with text or numbers. This is done by a text field. It shows as a bar within the window and may or may not contain text or numbers already. Move to the text field ([Tab] key) and when it highlights, type or edit the values contained within the text field.

List Box

A list is a box within the window that supplies you with a list of items. These may be files, names, parameters, etc. Depending on the type of window, you can perform different actions upon these items. Most windows, however, just expect you to select one of the given items. Press [Tab] until you arrive at the list. Then press the [Up-] or [Down Arrow] keys to scroll through the list. The [End] and [Home] key, respectively, move to the end and the beginning of the list. Note that if the number of items is larger than what can appear simultaneously in the list, a so-called scroll bar appears at the right side of the list. This bar can be used only by mouse owners. Other users can scroll though the list by using the cursor keys.

Select an item in the list by pressing [Spacebar] twice, or [Enter], or double-click with the left mouse button.

Table 4.2 Software and hardware requirements of MANAGE-N.

Component	Requirement
Computer	8088/8086 micro-processor (CPU) or higher. A mathematical co-processor (FPU) is recommended.
Operating system	MS-DOS® version 3.3 or later. Version 6.0 or later is recommended. To check your version, type VER at the command prompt.
Memory	512 Kb of free RAM memory before starting MANAGE-N. You can check this by typing MEM or CHKDSK . The result should be 524288 bytes or more of conventional memory free.
Disk space	3 Mb of disk space to install MANAGE-N, another 3 Mb of free disk space to work ¹⁹ with MANAGE-N.
Screen	CGA screen adapter or higher (EGA, VGA, SVGA). Hercules® is not recommended.

¹⁹ MANAGE-N uses a lot of disk space when doing reruns for the FSV calculation, about 2.5 Mb. Also, if you like to store results of simulation runs, bear in mind that for example a standard single run will take up between 30 Kb and 200 Kb, depending on the settings in of RUNCTRL.DTA. See also Appendix 6 'Technical reference of MANAGE-N'.

4.4 Requirements

Before you start to use MANAGE-N, or if you encounter any problems, make sure your computer system meets the minimum requirements listed in Table 4.2.

4.5 Quick installation

This section deals with installing the MANAGE-N tool. It is called 'Quick installation' because you do not have to configure many options. MANAGE-N will be installed on your system with a minimum of effort, using predefined settings. For the user who prefers more control over the installation (and the settings of MANAGE-N), please refer to Appendix 6 'Technical reference of MANAGE-N' on page 223.

An installation on an average PC system typically takes about 5 minutes of your time.

4.5.1 Making a safety backup

Before you proceed with the installation of MANAGE-N, make a backup of your original MANAGE-N disks. Please refer to the manuals accompanying your computer on how to do this. Ask a local computer wizard if you are unsure.

Store the original disks at a safe place and use your newly made back-up disks to install MANAGE-N.

4.5.2 Actual installation

Follow these steps to perform a first-time and quick installation on your system²⁰:

1. Be sure you have plenty of space on your hard disk (about 3 Mb).
2. Disk 1 (of 2) contains a file called `INSTALL1.EXE`. This file creates its own directory named MANAGE-N in which it places all necessary files and subdirectories. To install, move to the directory in which you want the new directory MANAGE-N to be created. For example, to install MANAGE-N in the directory `C:\MANAGE-N` type:

```
C:
CD \
A: INSTALL1.EXE
```

And to install the MANAGE-N tool in the directory `D:\SARP\MANAGE-N`, enter:

²⁰ If, by any chance, unexpected problems arise or you cannot get MANAGE-N to work, please refer to Appendix 6, 5 and 7 for solving this.

```
D:
CD SARP
A:INSTALL1.EXE
```

Of course, it is assumed that the directory SARP exists.

3. After INSTALL1 . EXE finishes, put disk 2 (of 2) in the drive and enter the command

```
A:INSTALL2.EXE
```

After INSTALL2 . EXE finishes, a directory named MANAGE-N has been created. This directory is the main directory for the MANAGE-N tool.

4. You have now completed the installation. Change to the MANAGE-N directory and execute MANAGE-N . BAT to start the MANAGE-N tool:

```
CD MANAGE-N
MANAGE-N.BAT
```

4.6 Getting Help

In case of unexpected problems, you can get support for MANAGE-N by contacting the author. Before you take this step, read through Appendices 6, 5 and 7 and ask a local computer expert to look into the problems. If the trouble persists, contact the author of the MANAGE-N tool at this address:

J.J.M. Riethoven c/o H.F.M. ten Berge & D.W.G. van Kraalingen
DLO-Research Institute for Agrobiolology and Soil Fertility (AB-DLO)
P.O. Box 14
6700 AA Wageningen
THE NETHERLANDS

Fax: +31.317.423110
Telex: 75209 abw nl
Phone: +31.317.475962
E-mail: riethoven@ab.dlo.nl
kraalingen@ab.dlo.nl

Specific information is required before your question can be handled correctly. Include the following information in your correspondence or have this information ready if you call

Version number of MANAGE-N

The version number of MANAGE-N is displayed in the window that appears on the first screen each time you run MANAGE-N.

Complete problem description

Include all error messages and symptoms of the problem. If you can consistently reproduce the problem, please provide the exact commands entered and data files/models that differ from the standard distribution set.

5 The theory of optimising the N application pattern by MANAGE-N

H.F.M. ten Berge¹, J.J.M. Riethoven¹ & H. Drenth^{1,2}

¹ DLO Research Institute for Agrobiology and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

² Department of Theoretical Production Ecology, Wageningen Agricultural University, P.O. Box 430, 6700 AK Wageningen, The Netherlands

5.1 ORYZA_0: the core of MANAGE-N

The ORYZA_0 model (Figure 4.1) describes nitrogen (N) supply from soil and fertiliser, crop N demand and uptake capacity, actual crop N uptake, N allocation to crop organs, utilisation of leaf N to capture radiation and produce biomass, and the formation of final grain yield. All these processes are expressed with the help of empirical coefficients. These can be derived from field experiments (see Appendix 11). The model operates with one-day time steps and requires daily values of incident global radiation. Detailed information on the ORYZA_0 model is provided in ten Berge et al. (1994c), with the addition of Appendix 13. This paragraph briefly describes the main principles.

ORYZA_0 is an explanatory model: biomass and yield are calculated from underlying physiological and agronomic principles. It is therefore different from statistically based response functions or 'correlation models'. Explanatory models allow the exploration of crop behaviour under 'new' conditions.

5.1.1 N supply

Soil N supply (mineralisation rate) can be specified as a function of time but will often be taken as a constant. Nitrogen supply in the form of fertiliser is expressed as a continuous logistic function with 4 parameters (A, B, C, M). For evaluation of split N application schemes, this 4-parameter supply curve is replaced by a formulation of discrete split applications. Only a fraction of supplied fertiliser N can be absorbed by the crop. This maximum fraction is called the potential recovery. It is specified as an input table, as a function of time. The actual and simulated recovery may be well below the potential level, if limitations in the crop's absorption capacity do not allow potential recovery.

5.1.2 N demand, uptake capacity, and actual crop N uptake

Under ample supply, daily N uptake is limited by the crop's ability to absorb N. Limiting factors are the maximum crop N concentration versus time, the maximum daily uptake rate, the maximum ratio of daily uptake to daily growth, and a maximum relative uptake rate during early crop growth. Actual uptake of nitrogen is the minimum of supply and demand on a daily basis.

5.1.3 N allocation

A fixed fraction (FNCLV) of absorbed N is allocated to the leaves. This fraction is constant during the entire preflowering stage. From flowering onward, N is remobilised from leaves and other tissues (stems, roots) and translocated to the panicles. The panicle N demand is calculated from panicle growth rate and the cultivar-specific panicle N concentration. During this stage, translocation from leaves and stems is reduced if crop N uptake continues.

5.1.4 N utilisation by the crop

An exponential growth equation is used to calculate daily crop growth from the current total amount of N contained in the leaf canopy (per ha ground surface) and daily incident global radiation. Mathematically, no separation is made between interception of light and its subsequent utilisation. Two parameters used in this equation are the initial radiation use coefficient (2.5 g per MJ incident radiation) and the initial leaf nitrogen use coefficient (10 g biomass per d per g leaf N). The growth equation contains a calibration factor (FSV) to express site-season and cultivar-specific effects. This factor is to be determined from field observations, with the help of a special feature of MANAGE-N. It often ranges from 0.8 to 1.1, but may be much lower under poor growth conditions.

5.1.5 Grain yield

Grain yield is calculated from final biomass via a harvest index. The harvest index is entered as a function of postflowering cumulative radiation and crop biomass at flowering. By specifying a zero value to the HISLOP parameter, the value of the harvest index is fixed and is set equal to the HI5 parameter.

5.1.6 Root-shoot ratio

The model does not recognise distinct crop organs (leaves, stems, etc.). Only total crop biomass and final grain yield are distinguished. There are, however, occasions where tabulated input values of the root-shoot ratio (versus time) are invoked. This is the case when the user wishes to compare observed versus simulated crop biomass but lacks data of observed total crop biomass. Instead, only shoot biomass data are available. This is rather common because rice roots are not often sampled. A tabulated root-shoot ratio function is then used to convert simulated total crop biomass values to simulated shoot biomass. This variable can then be compared with measured shoot biomass. This procedure introduces inaccuracies. The use of total crop biomass data including roots is always preferable. A default table of root-shoot ratio is provided (Table A10.1 on page 241)

5.1.7 Parameter input

Three input files are used: a weather file, a crop/soil parameter file, and a timer file. The names of these three files are user-defined. Crop and soil parameters are read by ORYZA_0 from a crop/soil input file. Timer variables are read from the timer file. MANAGE-N provides a mouse-driven menu to inspect and edit the input files. The various *methods* of MANAGE-N make use of different versions of ORYZA_0, each with their specific input requirements. Only those parameters 'active' under a given *method* are visible and accessible for editing. The 'inactive' parameters are disabled (names are replaced by dashes '-----' and values cannot be edited).

5.1.8 Versions of ORYZA_0 used in MANAGE-N

The 'Method' menu pad of MANAGE-N offers four menu options. Each of these accesses a different version of ORYZA_0. The versions differ mainly in terms of input and output (for differences in input requirements, the reader is referred to Appendix 10 on page 241, Table A10.1). The *methods* 'Fit to match' and 'Single Run' use the same ORYZA_0 source code, with a switch (SWINLV) to bypass N uptake and allocation for 'Fit to Match'. 'Optimise N-curve' uses a special model version, adapted to allow multiple runs and to optimise a set of parameters. The *method* 'Split Evaluation' is to be used as a post-optimisation check and is based on an adapted version of ORYZA_0.

5.2 Parameter derivation

Parameters and functions in the crop/soil data file are site-cultivar-season specific. A range of treatments is required to obtain a full characterisation of the local situation, i.e. of the site-cultivar-season combination. (site=soil+weather). Where observed time series are

included in the crop/soil data file, such a file obviously becomes specific for the particular treatment from which the observations were derived. It is recommended to express this in the user-defined name assigned to the file.

To fully characterise the local conditions, experiments should include treatments without fertiliser N application and also treatments with ample N application. Appendix 11 elaborates on the derivation of parameters and functions. Several parameters represent physiological limitations (N demand parameters) or soil characteristics (N supply) and can be derived only from treatments designed to expose those limitations. All these limiting factors are, however, required to execute simulations, also for non-extreme conditions.

The crop-soil data file can be edited with the help of the 'Edit A Crop/Soil File' menu option.

5.3 The three steps toward fertiliser recommendation

The approach to site-tailored fertiliser management optimisation by MANAGE-N consists of three steps:

1. Parameterisation. This includes quantification of local efficiency by which leaf N and radiation are utilised (calibration).
2. Numerical optimisation of an ideal, continuous N application curve. This is repeated for each N input level.
3. For the chosen optimal N level: translation of the ideal curve into an actual split-dose application scheme (number, size, and timing of the respective doses) and evaluation of the proposed schemes.

Step 1 is performed by the *method* 'Fit to Match'. Step 2 is executed by the *method* 'Optimise N-curve'. Step 3 is done manually by the user but is based on the results obtained in step 2. The user-defined proposed split application scheme is then evaluated by the *method* 'Split Evaluation'. These three *methods* are offered as menu options under the menu pad 'Method' of MANAGE-N. There is one more *method*, named 'Single Run', which offers single simulation runs (as opposed to numerical optimisation runs) with ORYZA_0 for in-depth analysis of model behaviour.

5.4 Computing the FSV values ('Fit to Match')

This *method* is used to identify the value of FSV from field observations. FSV is a calibration factor expressing the efficiency by which the crop utilises radiation and leaf N to produce dry matter. Because this efficiency often changes at some point during the season, FSV is specified as two values: FSV1 before a critical date (DATFSV), and FSV2 afterwards. DATFSV often coincides with the first flowering (FF) stage, but it may also be earlier or later.

The values of FSV1 and FSV2 are determined by matching a simulated time curve of crop biomass with an observed curve. The corresponding observed time curves of leaf biomass and leaf N content are used as forcing functions in the simulation. Selection of the option 'Fit to Match' automatically sets the value SWINLV to =1, indicating that the amount of leaf N is not simulated but forced from observations. Hence, only a small part of ORYZA_0 is active under this option. (Inactive are the model sections describing supply, demand, uptake, distribution, and reallocation of N.)

MANAGE-N first seeks a best value of FSV1, by default first for the period up to first flowering (FF) stage. The best fit is found by minimising the sum of squared deviations between simulated and observed biomass. Since the numerically best fit is not always the curve judged best by the researcher, the FSV1 value may be adjusted 'manually' if so required. The date DATFSV may also be changed, if required. Once satisfactory values for FSV1 and DATFSV are found, these are fixed and the user proceeds with fitting the remaining part of the biomass curve to obtain FSV2. Again, final readjustment is possible. When a satisfactory value for FSV2 is found, MANAGE-N stores the value into the crop/soil data file, and thus completes the 'Fit to Match' run.

See also Appendices 10 and 12

5.4.1 Input requirements for 'Fit to Match'

All required inputs are shown on screen when MANAGE-N is used to edit a crop/soil file. The crop/soil data file should contain a consistent set of observations of crop biomass, leaf N content, and leaf biomass. These 3 variables are specified in the form of tabulated functions versus days after transplanting and should all refer, obviously, to the same growth curve (=treatment) within an experiment. Other required parameters are the date of first flowering and the date of harvest. The weather data file should contain the corresponding observed daily radiation data, and the timer file should be consistent with the experiment. See also Appendices 10 and 12.

5.4.2 Representative FSV values

FSV values often differ slightly between treatments. To obtain a good 'average' value of FSV1 and FSV2 to represent this site-season-cultivar combination under study, the 'Fit to Match' procedure is executed for a number of growth curves available from an experiment. After completion, the user determines (from the individual treatment values) overall values of FSV1 and FSV2. These overall values are used under the 'Optimise N-curve' *method* and are to be specified in a file with extension * .ALL. They are supposed to be valid over the whole range of N input levels.

5.5 Single run

The 'Single Run' option allows detailed inspection of crop behaviour based on ORYZA_0 under a given set of conditions (specified in the crop/soil data file). 'Single Run' is used to study crop responses to changes in parameters, to identify potential yield level or zero-N yield level, and to make an in-depth evaluation of crop growth under recommended N management as specified from the 'Optimise N-curve' *method*.

5.5.1 Outputs

Time courses of many model output variables can be graphed or studied from tables via the menu pad 'Output'.

5.5.2 Input requirements for 'Single Run'

For a 'Single Run' execution, all parameters listed in the crop/soil input file are used, with the following exceptions. Time tables of observations (WCR_OBS, WLV_OBS, FNLV_OBS) are not required, unless leaf nitrogen is used as forcing function (SWINLV = 1), or unless the user desires a comparison between simulated and observed values.

The root-shoot ratio table is needed only when root mass is not included (RTINCL = 0) in the tabulated observed crop biomass WCR_OBS. In that case, the root-shoot ratio table is used to convert simulated crop biomass into shoot biomass to allow for a comparison of simulated versus observed shoot biomass.

The crop/soil data files used in 'Single Run' will often refer to a real case (experiment). It is recommended to give sufficient experimental details in the text section of the file. A statement should always be included as to whether crop biomass includes roots or not. Depending on the issues analysed (e.g., in sensitivity analysis), parameters and functions may attain any (biologically plausible) value.

5.5.3 About A, B, C, M, and FERTMX

Note that 'Single Run' simulations are based on the particular fertiliser scheme defined in the crop/soil input file by parameters A, B, C, M (time course of fertiliser application), and FERTMX (total fertiliser N input). These will largely determine the crop response. To inspect the meaning of these parameters and their consequences, the output variable APCUM can be plotted versus TIME (fertiliser application curve) for different values of A, B, C, M, and FERTMX.

5.5.4 Optimisation of N application

The fertiliser application curve APCUM (cumulative fertiliser application versus time) is defined by the parameters A, B, C, M, and FERTMX. Together, these parameters determine the shape and scale of the curve. Under the 'Optimise N-curve' *method*, the ideal time course of fertiliser application is sought by numerically optimising the values of A, B, C and M for a given (user-defined) N input level FERTMX. This ideal curve is called the recommendation curve and, by definition, results in the highest attainable grain yield. Note that this is all based on the assumption that N can be supplied continuously.

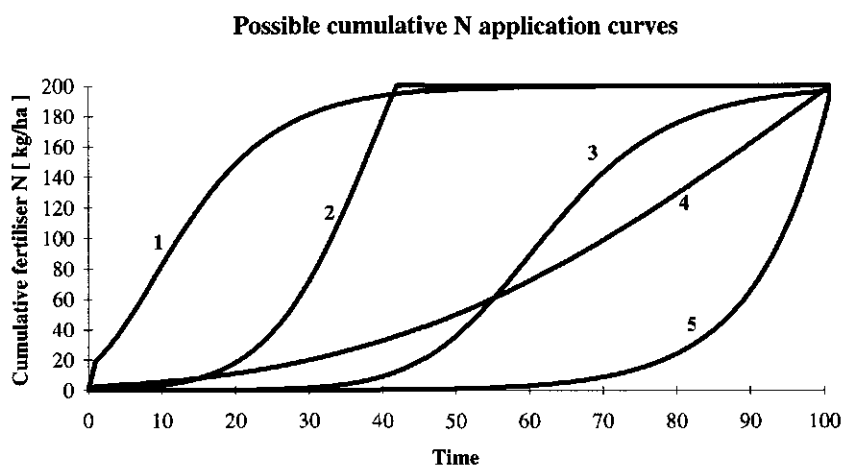


Figure 5.1 Examples of APCUM curves that can be generated during optimisation. The above curves were computed with $\text{FERTMX} \approx 200 \text{ (kg ha}^{-1}\text{)}$. The following curve defining parameter values were used: $A = 0.02$, $B = 0.11$, $C = 200$, $M = 9$ (line 1); $A = 0.5$, $B = 0.1$, $C = 400$, $M = 40$ (line 2); $A = 0.5$, $B = 0.1$, $C = 200$, $M = 60$ (line 3); $A = 0.01$, $B = 0.016$, $C = 610$, $M = 108$ (line 4) and $A = 10$, $B = 1$, $C = 500$, $M = 108$ (line 5).

The optimisation is repeated for each of the user-specified N input levels. An overview of the recommendation curves for the different FERTMX levels is produced in the form of a figure combining all the scaled APCUM curves. These show the time paths of relative application. They all reach the value of 1 when application of the full dose ($=\text{FERTMX}$) is completed.

'Optimise N-curve' also provides the curve of best attainable yield versus N input (N response curve). It is based on the same optimisation runs that identified the recommendation curves. This N response curve, when combined with prices of fertiliser and rice grain, is the basis for calculating the net cash income from investment in fertiliser.

The optimisation process requires many runs with the model ORYZA_0, and may thus be time-consuming.

5.5.5 Input requirements for ‘Optimise N-curve’

All required inputs are shown on screen when MANAGE-N is used to edit a crop/soil file. The ‘Optimise N-curve’ *method* uses a crop/soil file that is generic for a certain site-season-cultivar combination. Specific data for the several treatments within an experiment are not needed; instead average values for the experiment are used. Like in the other *methods*, some dates and initial weights are required, together with the mean values of FSV1 and FSV2. Several crop characteristics are also required, and time series related to nitrogen supply by the soil and nitrogen recovery. These parameters must be supplied in a data file with the extension *.ALL. See also Appendices 10 and 12.

5.5.6 Outputs

After optimisation, the parameters A, B, C, M, and FERTMX are stored in a special file for later inspection. Graphical outputs are:

- the individual recommendation curves for a specific N level (versus time),
- the scaled recommendation curves (presented jointly for all specified N input levels versus time)
- the N response curve (dry grain yield and biomass versus N input level)
- the net cash return curve (versus N input level)

5.5.7 The numerical optimisation procedure

The parameters A, B, C, and M are optimised using a controlled random search procedure for numerical optimisation (Price, 1979). A modified version of the FSEOPT program (Stol et al., 1992) is used to perform the optimisation of the parameters. The goal variable is grain yield, which is maximised.

For each N input level, the following procedure is followed. First, a number (INPS) of initial parameter sets are created. Independently of each other, INPS values are randomly drawn for the parameters A, B, C, and M. The ranges from which these values are drawn differ for the four parameters. Each combination of a random value for A, B, C and M is called a set. The number of initial sets equals INPS. The collection of INPS parameter sets is also called the ‘vase’.

Next, the ORYZA_0 model is run for each of the above sets, resulting in different grain yields WRR. The reciprocal value of the final grain yield (i.e., $1 / \text{WRR}$, from now on called WRR_{rec}) is stored with the corresponding set.

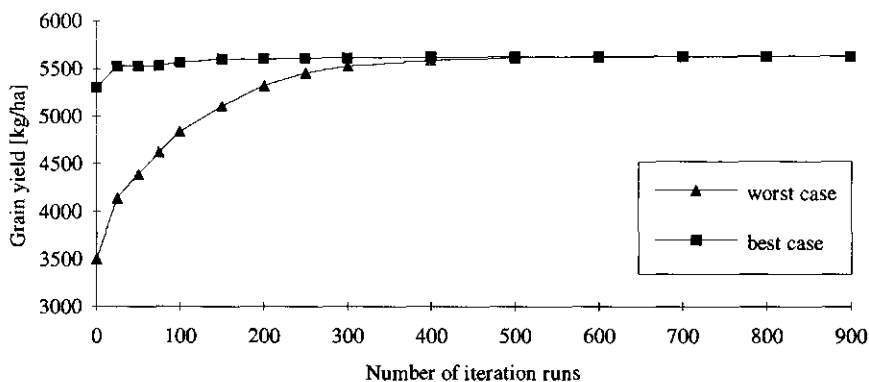
The optimisation continues with the so-called iteration runs: these will gradually search through the four-dimensional (A, B, C, and M) parameter space in a controlled way, until the optimisation is stopped by one of the finish conditions. One of the finish conditions is the maximum number of iteration runs INT: on reaching INT, the optimisation terminates. At the beginning of each iteration run, a five-point simplex in the four-dimensional parameter space (axes A, B, C, and M) is generated by random selection of points from the 'vase' and is used to spawn a new point as the mirror image of one of the simplex points. The model is run with this value, and again the reciprocal value of WRR is computed. If WRR_{rec} is smaller than the largest value of WRR_{rec} present in the 'vase', the newly computed parameter set replaces that worse-case set in the 'vase'. In other words, the parameter set that yields the lowest WRR is replaced by a better one. If the newly calculated WRR_{rec} was higher than the worst case in the 'vase', no set is replaced and the next iteration run is started.

Slowly the optimum parameter set is approached for which the highest grain yield is achieved. The user can define when a parameter set is 'good' enough, by specifying the tolerance level FTOL. When the relative difference between the best WRR_{rec} and the worst WRR_{rec} is smaller than FTOL, the optimisation is stopped.

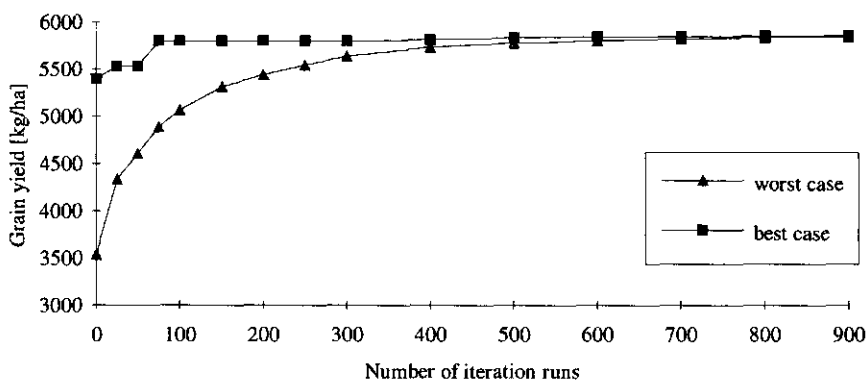
Because INPS and INT can seriously affect the quality of the results, the following rules-of-thumb should be taken into account:

- The initial number of parameter sets (INPS, the 'vase') should be at least 10 times the number of parameters for which the optimisation is performed. In the case of MANAGE-N, INPS is per default set to 40.
- The number of iteration runs (INT) should be at least 6 times the number of initial runs. In the case of MANAGE-N, 300 is used as a default.
- Small values of FTOL do grossly increase the number of iterations. A suitable value can be estimated from the required accuracy and the expected yield level. But do not set FTOL too small, or you might not achieve an optimum but are still 'on the slope toward the optimum'. MANAGE-N's default can be used in most cases.

To show the speed of convergence toward an optimum, we performed an optimisation in two different cases (Figure 5.2). The first case is for the N input level of 150 kg N ha⁻¹ (FERTMX=150), the second case for an input level of 300 kg N ha⁻¹ (FERTMX=300). For both cases, we plotted the lowest and highest WRR after an increasing number of iterations. As the graphs show, somewhere between the 300 and 400 optimisation runs for the FERTMX=150 case and at about 600 iteration runs for 300 kg N ha⁻¹, the best and worst parameter sets in the 'vase' result in roughly the same grain yield. The convergence would proceed to infinity if no stop condition were specified.



(5.2a)



(5.2b)

Figure 5.2 Speed of convergence toward an optimum parameter set. Figure 5.2a shows the case of a total N input level of 150 kg N ha⁻¹, Figure 5.2b the case of 300 kg N ha⁻¹ for a specific location, season, and cultivar. ‘Best case’ and ‘worst case’ refer to the highest and the lowest grain yield in the ‘vase’, respectively. Note that, even as the lines converge, the absolute grain yield is still increasing, although slowly. During none of the above runs was the optimisation terminated due to reaching the FTOL stop condition.

Naturally, the speed of convergence depends also on other parameters that are used by the model (local situation of site, variety, and season) so these graphs are examples only. Generally, more iteration runs are required when the N input level is higher. Therefore, for high N input levels (> 200 kg), the user is advised to set the number of iteration runs to 600 or more.

5.6 The final step: split evaluation

The *method* 'Split Evaluation' is used after the optimised N application curve has been identified. Since in farming practice, N cannot be supplied following a continuous curve, a split application scheme has to be proposed. The recommendation curve is then typically used as a basis. The user has to decide how many splits are to be given. Obviously, the total amount of N in all splits should be equal to the FERTMX value to which the recommendation curve refers. When the number of splits has been assessed, the size and timing of the split doses should be such that the resulting cumulative application resembles the recommendation curve. (This is done in the user's mind.) Then the 'Split Evaluation' *method* predicts crop performance under the proposed split scheme. Yields will usually be slightly lower than those calculated earlier under continuous application.

Split evaluation is an important step, especially for poor soils where it may be essential to meet early crop N demand, although small in absolute terms, to avoid a setback later on. A split application scheme which would ignore this early demand could then be less than the continuous recommendation curve from which it was derived and which it resembles.

MANAGE-N uses a special version of the ORYZA_0 model to simulate the effects of discrete split-dressings. When a N dose is applied, nitrogen is transferred to the fertiliser nitrogen pool N_{pool} in the soil. The size of N_{pool} decreases every day due to uptake by the crop and loss of fertiliser by other biotic and abiotic factors. The nitrogen that is available to the crop on time t , N_a , is calculated by equations 5.1 and 5.2:

$$K_1 = \frac{\rho_N}{1 - \rho_N} \times K_2 \quad (5.1)$$

$$N_a = S_N + (K_1 \times N_{pool}) \quad (5.2)$$

Where ρ_N : best attainable N recovery fraction at time t (as used in *method* 'Optimise N-curve')
 K_2 : proportionality factor relating N pool size to loss at time t
 K_1 : proportionality factor relating N pool size to potential uptake at time t
 S_N : native soil N supply at time t

In-season changes in recovery ρ_N , K_2 , and native soil N supply S_N are possible. Hence, daily values for these parameters are used, based on tabulated $\rho_N(t)$, $K_2(t)$ and $S_N(t)$.

The amount of nitrogen available to the crop consists of both native soil N and fertiliser N that is taken from the fertiliser N pool in the soil. Actual nitrogen uptake N_{upt} is determined both by crop N demand and N_a (ten Berge et al., 1994c).

The change in fertiliser N pool is described in equations 5.3 and 5.4:

$$N_{loss} = K_2 \times N_{pool} \quad (5.3)$$

$$\frac{dN_{pool}}{dt} = APSLOP - (N_{upt} - S_N) - N_{loss} \quad (5.4)$$

On a day when fertiliser N is applied, APSLOP is set equal to the size of the N dose ($\text{kg N ha}^{-1} \text{ d}^{-1}$). Every other day, APSLOP equals zero.

Appendix 13 lists the exact FSE (FORTRAN) source code of the ORYZA_0 section adapted to perform the N budget calculations under split-dose schemes. The FST source code of the split-dose fertiliser N application version of ORYZA_0 is listed in the second section of Appendix 14.

6 Optimising the N application patterns by MANAGE-N, an example

J.J.M. Riethoven¹, H.F.M. ten Berge¹, H. Drenth^{1,2}

¹ DLO Research Institute for Agrobiological Sciences (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

² Department of Theoretical Production Ecology, Wageningen Agricultural University, P.O. Box 430, 6700 AK Wageningen, The Netherlands

In this Chapter, the theory of optimising fertiliser nitrogen application in irrigated rice is brought into practice step by step. The examples presented in this Chapter are based on existing data from a field experiment performed in 1993 in the Jinhua county in the south of the Zhejiang Province, China, with cultivar Xieyou 10. Detailed information about this data set can be found in Zhiming et al. (1995). There are small differences with the results of the optimisations presented in Zhiming et al. because a later version of MANAGE-N is used here. Although the 1993 Jinhua data set is not distributed with the MANAGE-N system, it still shows a clear example of the several steps involved in computing optimised fertiliser N dressings. The various steps of the procedure are also called *methods*.

For brevity, an 'active' style of writing is used in the following paragraphs. There is no reference to 'the user'; he or she is addressed in the second person to facilitate both reading and hands-on work. Instructions and explanations alternate in this Chapter. The graphs are all actual snapshots of graphs that MANAGE-N displays, except where indicated otherwise. Analogous to the steps described in this Chapter, you can generate fertiliser recommendations based upon your own data.

The assumption is made that MANAGE-N is already installed (if not, refer to paragraph 4.5) and that you are already accustomed to the MANAGE-N menu system (see paragraph 4.3).

6.1 Prepare required inputs

Before you can start, it is imperative that you start MANAGE-N and prepare the inputs required for the different steps (see also Chapter 5). To kick off, go to the directory in which you installed MANAGE-N. If, for example, you installed MANAGE-N in the C:\SARP\MANAGE-N directory, enter these commands:

```
C:
CD \SARP\MANAGE-N
```

Now start the MANAGE-N tool:

MANAGE-N .BAT

and close the 'About MANAGE-N' window that appears (press the [OK] button).

Next, prepare all the inputs for MANAGE-N based on a proper field experiment conducted at the site for which the recommendation is sought. Although every *method* has other data requirements (see Appendix 10), we assume that all data are necessary from the start. Some data are collected easily: sowing/transplanting date, date of final harvest, biomass (time course), fraction of nitrogen in the leaves (time course), date of first flowering, and so on. Other data need to be derived (Appendix 11) or default values (supplied in MANAGE-N) should be used when derivation is not feasible.

For the purpose of entering data, special input windows are available in MANAGE-N. These windows list the parameters and observations required for a *method* (selected from the menu pad 'Method'), together with the description of the parameters (click name and view bottom of screen). Parameters not required for a *method* (for example, *SPLIT* for the 'Fit to Match' *method*), are not accessible for editing. Edit an existing crop/soil file (use the 'Edit A Crop/Soil File' menu option under the 'Input' menu pad), modify the already present values into your own values, and save the crop/soil file under another name that you choose. This, preferably, should be a descriptive name, in which you express the year, experiment, and treatment. Every treatment requires a separate crop/soil file.

Detailed information on the 'Edit A Crop/Soil File' window can be found in Chapter 7; the inputs required under each *method* are documented in Appendices 10, 11 and 12.

For now, we assume that you were successful in creating your own input files. In the next paragraphs, input files describing Treatment 0 (T0, 0 kg N ha⁻¹) to Treatment 6 (T6, 300 kg N ha⁻¹) are used.

6.2 Compute the FSV values

The 'Fit to Match' *method* is used to identify the values of FSV from field observations (please refer to paragraph 5.4 on page 90 for the theoretical background. The identification of the FSV values is required before you can continue with any of the other *methods*.

In some cases, FSV may change during the season. It is then useful to calculate two values for FSV: one before (FSV1) and one after (FSV2) the date where the change occurs (DATFSV, in days after transplanting). This change in FSV value often coincides with the date of first flowering (DATFF, in days after transplanting).

In this paragraph we will run MANAGE-N to calculate the values of FSV1, FSV2, and DATFSV for seven treatments of the same experiment (see also paragraph 5.4). MANAGE-N compares observed (WCR_OBS) and simulated (WCR) weight of the crop for different values of FSV and will present you the value that offers the best fit between

observed and simulated data. You are free, of course, to ignore the advised best fit and to choose a value yourself.

Let us start the procedure. Go to the 'Method' menu pad and activate the 'Fit to Match' menu option. The status screen now displays that you have selected the 'Fit to Match' *method*, and that leaf nitrogen will be forced (i.e., observed values will be used to rule out any external factors related to nitrogen uptake). Now go to the 'Input' menu pad and activate the 'Select Input Files' menu option. A new window opens that facilitates a quick and easy selection of input files. Remember that the files with the example data mentioned in this text are not distributed with the MANAGE-N disks: you have to select or prepare your own. We need three files: a crop/soil file, a weather file, and a timer file. Select them now.

When you return to the main menu (press [OK]), you notice that the status screen displays all the input files you selected. Now activate the 'Edit A Crop/Soil File' menu option. This option will show you the contents of the crop/soil file. The parameters and observations are grouped into three windows (use [Next Window] and [Previous Window] to cycle through the separate windows). Check if every parameter field is filled with the proper value. Only those parameters that are required with the 'Fit to Match' *method* are listed; other parameter names are hidden by a dashed line '-----'. If a required parameter has no value or an incorrect value, insert your own value. Do not forget to supply the observed time series of biomass, leaf N and leaf weight. Store changes by pressing the [Save] button, then press [Exit] to return to the main menu.

You are now ready to commence the first calibration of the FSV factor. Go to the 'Run' menu pad and activate 'Start Run'. MANAGE-N now starts the ORYZA_0 simulation model: it will perform 151 runs (one for each value of FSV1; from 0.00 to 1.50 with increments of 0.01). On a slow computer, this takes a while; on an average IBM™ 80486 computer, this procedure is finished within a minute or two.

As soon as the simulation runs end, MANAGE-N calculates the Goodness-of-Fit (GOF) between observed and simulated weight of the crop for each FSV1 value until the date of first flowering. The GOF is an ordinary sum of squares (of the differences at observation dates). The value of FSV1 that gave the lowest GOF (i.e., the best fit) is presented by MANAGE-N.

The 'Select Best Matchfactor' window (see Figure 7.14 and explanation, page 131) offers inspection of the resulting fit. The period over which the FSV1 factor is calculated is displayed on screen, as well as the value of FSV1 itself. You can press [View Best Fit Graph] to get a graph of the simulated and observed weight of the crop for the best FSV1 value (see Figure 6.1). The dashed vertical line at time DATFSV indicates the shift from FSV1 to FSV2: the determination of FSV1 is based on the interval from the start up to the dashed line.

It is often possible to move the switching point (DATFSV) back or forward in time. To do this, determine to which TIME you want to move the dashed line, then enter this TIME in

the ‘Until TIME =’ text field (to the right of the [Recalc Factor] button). After this, it is necessary to calculate the GOF anew. Press the [Recalc Factor] button to accomplish this. In the example presented here, we changed TIME from 246 to 255.

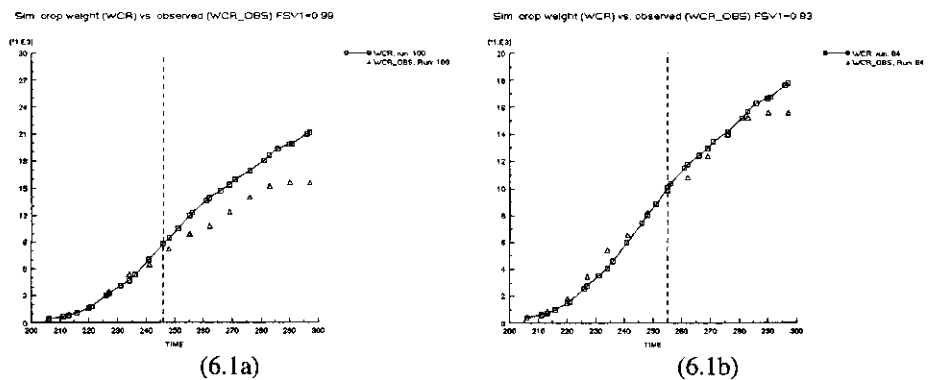


Figure 6.1 Changing the value of DATFSV to get a better fit. In Figure 6.1a, FSV1 has been calculated until DATFSV=40 (TIME=246). However, on analysing the graph, it seemed possible to move DATFSV to DATFSV=49 (TIME=255) to get a better fit for FSV1 (Figure 6.1b). Whether this improves this improves the postflowering fit (with FSV2) is not shown here. Note that the scale of the Y-axis is different for the two Figures.

If you are not satisfied with the FSV1 value that is presented by MANAGE-N, you can alter the FSV1 value manually (in the first box of the window, to the right of the button [Graph of run:]) and inspect the effects the modification has on the fit between observed and simulated weight of the crop (press [Graph of run:]). A lower FSV1 value lowers the slope of the simulated biomass line; a higher value increases the slope. A manually altered FSV1 value also needs to be entered in the field to the right of the [Accept Factor] button. Per default, MANAGE-N will use the value of FSV1 that was advised by MANAGE-N when you press [Accept Factor]. Only if you change the value of the ‘FSV1 =’ text field would MANAGE-N use your newly supplied value.

Although the FSV1 and FSV2 values presented by MANAGE-N are based on an objective and comparable calculation (minimum sum of squares), there may be reasons to adopt your own subjective values but this is not recommended as standard practice.

When the [Accept Factor] button is pressed, MANAGE-N inserts the values for FSV1 and DATFSV in the crop/soil file and then continues with the determination of FSV2. This will also take 151 runs. The same ‘Select Best Match Factor’ window is displayed to you after the simulation runs end. Follow the same procedure as described above for the FSV1 factor to select the value for FSV2. The only difference is that you cannot change the TIME: FSV2 is determined over the period from (the already fixed) DATFSV to the end of the simulation (final harvest, DATH).

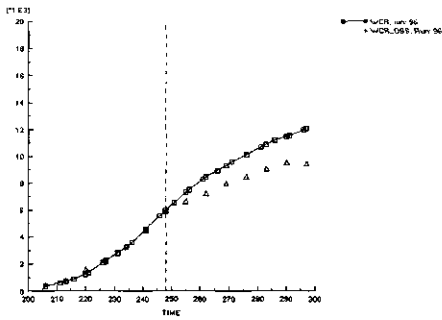
In the example presented in this Chapter, we calculated the FSV and DATFSV values for all growth curves (i.e. all the seven treatments T0 to T6). The resulting values of DATFSV, FSV1 and FSV2 are presented in Table 6.1. Snapshots of the graphs used in determining FSV are presented in Figure 6.2.

The next step, normally, is the optimisation of N application. You may, however, want to do some ‘Single Run’ simulations - with and without forced leaf N - to study model behaviour. That is what we will do next.

Table 6.1 Results of the ‘Fit to Match’ runs for the Jinhua 1993 data set. The accompanying graphs are displayed in Figure 6.2.

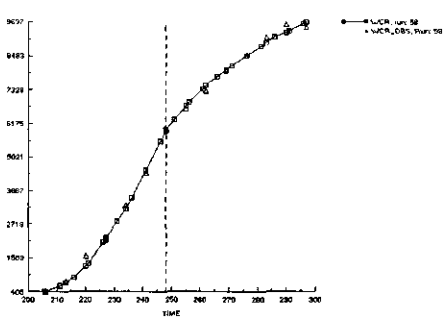
Treatment	N applied (kg ha ⁻¹)	DATFSV	FSV1	FSV2
T0	0	42	0.95	0.57
T1	150	45	0.92	0.58
T2	180	49	0.85	0.61
T3	210	49	0.83	0.67
T4	240	42	0.85	0.66
T5	270	49	0.84	0.66
T6	300	35	1.04	0.60

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV1=0.06



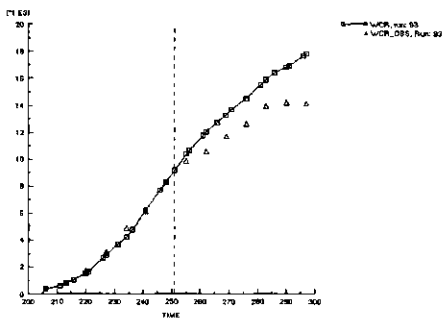
(6.2a)

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV2=0.57



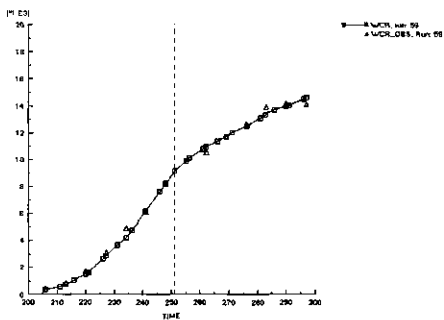
(6.2b)

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV1=0.92



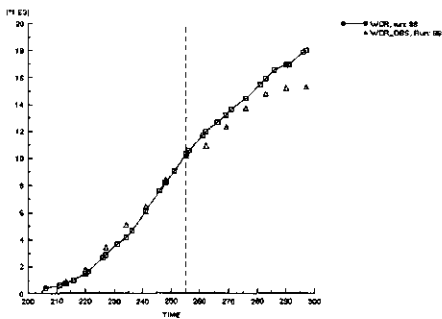
(6.2c)

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV2=0.58



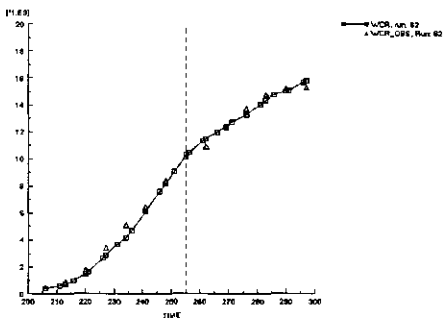
(6.2d)

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV1=0.85



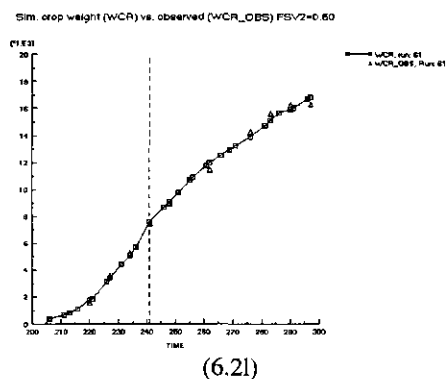
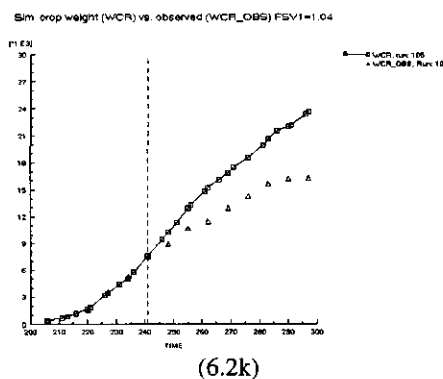
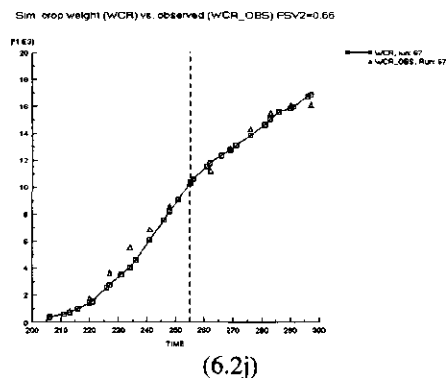
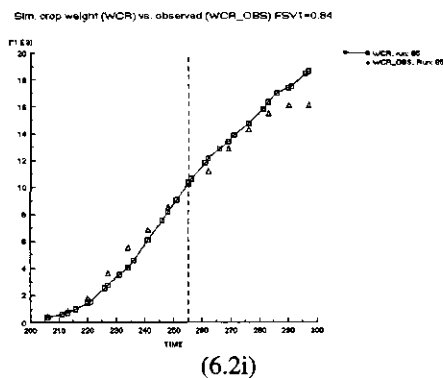
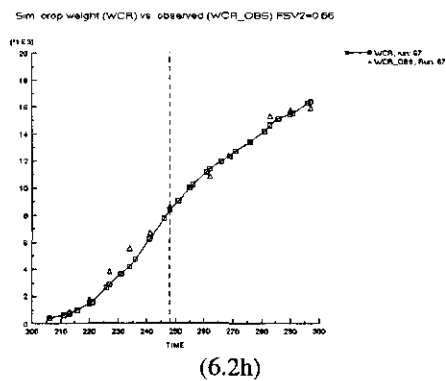
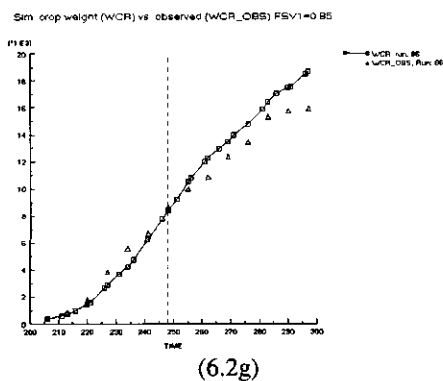
(6.2e)

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV2=0.51



(6.2f)

Figure 6.2 Results of the 'Fit to Match' method on the seven treatments of the Jinhua experiment. The left column always shows the results of the FSV1 fit, the right column displays the complete fit. The dashed vertical line indicates DATFSV, the date on which FSV1 switches to FSV2.



Triangular markers indicate observed values; the solid line is the simulated growth curve. The following treatments are displayed: T0 (a + b), T1 (c + d), T2 (e + f), T4 (g + h), T5 (i + j) and T6 (k + l). Part of treatment T3 has already been displayed in Figure 6.1. See Table 6.1 for details on FSV and DATFSV values.

6.3 Examine the results by single run

The 'Single Run' *method* is primarily used to study model behaviour under different conditions (as specified in the crop/soil file), to identify potential yield levels or zero N yield level, or to evaluate results obtained from the N optimisation runs. Here we use it to illustrate yield response to different shapes of the cumulative N application curve. The input requirements for this *method* are listed in Appendix 10.

A 'Single Run' can be executed in two modes: either simulate leaf nitrogen (and nitrogen uptake) or use observed values for leaf nitrogen (similar to the 'Fit to Match' model runs). For now, you will use observed leaf nitrogen to inspect the growth curve (WCR) you obtained from the 'Fit to Match' *method*.

Go to the 'Method' menu pad and choose the 'Single Run' menu option. Now go to 'Input' and activate 'Select Input Files'. Choose the crop/soil file that you used to determine FSV values during 'Fit to Match' and select the corresponding weather and timer files. Look at the status screen: it shows that leaf nitrogen will be simulated. Press the 'Toggle SWINLV' menu option in the 'Input' menu pad. The status screen should now display that leaf nitrogen is forced (i.e. observed values are used). Now take a look at your crop/soil file: activate the 'Edit A Crop/Soil File' menu option.

The 'Single Run' requires a lot more data than the 'Fit to Match' *method* did. Cycle through the windows (press the buttons [Next Window] and [Previous Window]) and supply all required data. The only parameter that you have to ignore is STPDAT. Leave it empty. [Save] the data when you are ready. Then [Exit] back to the main menu and activate the menu option 'Start Run'. The ORYZA_0 model is started and a lot of informative messages appear on the screen. This is normal. Press, when applicable, [Enter] at the end of the simulation to get back to the main menu.

If you use the 'View Graphs' menu option, you are able to plot all simulation results. Plotting both simulated (WCR) and observed (WCR_OBS) weight of the crop versus TIME shows you exactly the same graph as in 'Fit to Match' (except for the dashed line indicating DATFSV). A few other variables are also present but not as many as in the next example with 'Single Run'. The advantage of the above procedure is that you can print your graphs to printers (for example network printers) other than your local printer, which is not possible during the 'Fit to Match' *method* itself. At that time, you are limited to a local printer.

Next, you examine the effect of the shape of the continuous nitrogen application scheme on the yield (see Chapter 5 for detailed information on the parameters that determine the cumulative continuous nitrogen application curve). In this case, the model must simulate leaf nitrogen and uptake. Activate the 'Toggle SWINLV' menu option again and check the status screen: it should indicate that leaf nitrogen will be simulated.

Go to the 'Edit A Crop/Soil File' menu option again and supply the following values for these parameters²¹:

²¹ For extended analyses based upon parameter changes, FSU (see Part I of this book) is more suitable.

A = 1. B = 0.1
C = 300. M = 30. FERTMX = 210.

[Save] these values and run the model again. When you enter the 'View Graphs' menu option this time, you will notice that quite some more parameters and variables are listed. These deal all with nitrogen fertiliser application, demand, and uptake.

Make graphs of both the total crop biomass (WCR), yield (WRR), and the cumulative fertiliser nitrogen application (APCUM). Print these graphs if you can (see Appendix 6), or store the basic data when you are back in the 'Output' menu pad: 'Store Single Run Results'. If you use the last option, a separate directory will be created in the `MANAGE-N\SAVE` directory, in which all the output files are stored. You can use these output files to import them into, for example, a spreadsheet or a plotting program. Now change the values for A, B, C, and M to the following:

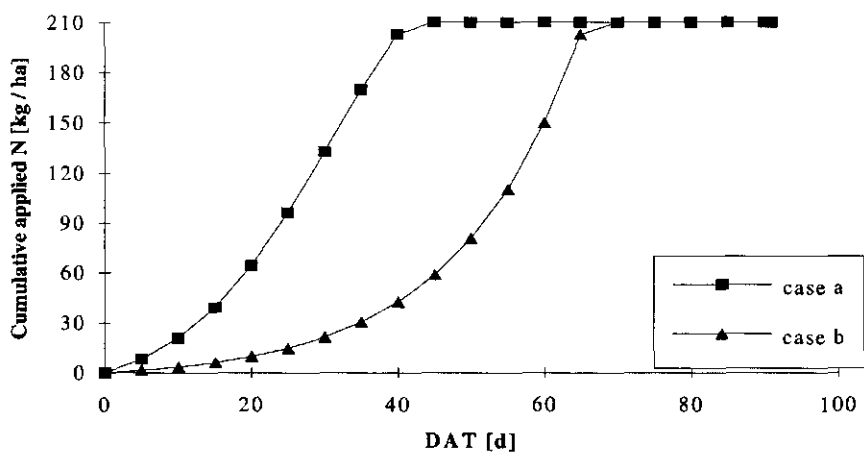
A = 5. B = 0.3
C = 400. M = 70.

and run the model again. Plot the results again and compare the graphs with the previous ones of APCUM (Figure 6.3), biomass and yield. The resulting graphs demonstrate the importance of proper timing of fertiliser application. For further exercise, you may repeat the above with different values for FSV1 and FSV2. Or you may change the potential recovery table RECT or the native soil N supply table SST to evaluate the effects of soil properties on growth and yield.

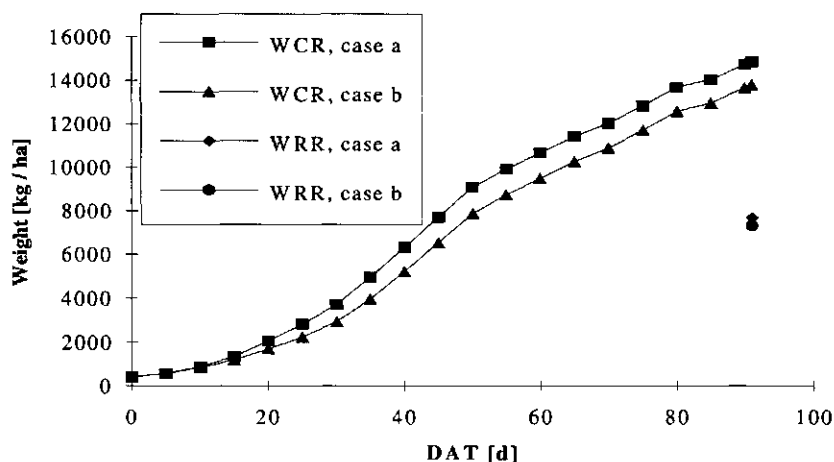
6.4 Optimise the N applications

The 'Optimise N-curve' *method* is used to determine the shape of the N application curve that corresponds with the maximum attainable weight of the rough rice (WRR) for a given fertiliser N input level. The shape of the cumulative fertiliser N application curve (APCUM) is defined by the parameters A, B, C, M and the total fertiliser N input level FERTMX (see paragraph 5.5.4 for more information).

The optimisation process is 'automated': you can supply a range of fertiliser N input levels for which N application should be optimised. `MANAGE-N` optimises the N application curves for all these input levels, one after the other and then uses the results (the best values of A, B, C and M) to run the `ORYZA_0` model again. These last runs supply detailed information that is used to draw graphs.



(6.3a)



(6.3b)

Figure 6.3 Effect of changes in the shape of the cumulative fertiliser N application curve. The top curve (Figure 6.3a) shows two different application curves: case (a) was computed using $A=1.$, $B=0.1$, $C=300.$ and $M=30.$; case (b) used $A=5.$, $B=0.3$, $C=400.$ and $M=70.$ Both curves used a maximum fertiliser N application of $FERTMX=210 \text{ kg ha}^{-1}$. Figure 6.3b shows the effects of the two application curves on total crop biomass (WCR) and dry grain yield (WRR) for both cases.

Optimisation is generally performed in absence of prior knowledge regarding the 'best' N level. Hence, we seek a value of FSV that represents the site, season and cultivar rather

than a particular N management scheme. We looked at the results of the 'Fit to Match' runs (Table 6.1) and decided to use the mean values of DATFSV (=44), FSV1 (=0.90), and FSV2 (=0.62) for our optimisation presented in this paragraph. These values have to be inserted into an overall 'site-season-cultivar specific' file. Such files should have extension '*.ALL', to indicate that they are not 'treatment specific'. We assume here that such a file does not yet exist. Now, first activate the 'Optimise N-curve' menu option. Then go to the 'Edit A Crop/Soil File' menu option and edit one of the crop/soil files that you used earlier, in order to make the values represent the site-season-cultivar combination. (That original file was very specific for a treatment). The first thing now is to store the file under another name: go to the 'Crop/Soil file' text field (the left-bottom corner of the 'Edit Data File' window) and change the name to a descriptive name with extension '*.ALL'. In this example, we changed the name from 'ZZT3.DAT' to 'ZZ1993.ALL'. Now [Save] this file. Only then do you change the values of DATFSV, FSV1, and FSV2 into the generic (mean) values that you have computed for this site. Also change the TREATM variable to indicate that this file is the generic crop/soil file for the experiment and not the treatment. In our example, we modified the text to 'Jinhua 1993, generic optimisation file'. Now [Save] the crop/soil file again and press [Exit]. Go to 'Select Input Files' and choose the corresponding weather and timer file (the crop/soil was already selected upon leaving the 'Edit A Crop/Soil File' menu option).

Two additional menu options will be available at this time in the 'Input' menu popup: 'Set Optimisation Parameters' and 'Set Phenology Dates'. The first menu option is required for an optimisation. Activate this 'Set Optimisation Parameters' menu option. A new window opens that allows you to supply and modify various parameters that are related to the numerical optimisation procedure.

In the example presented here, we increased the number of iteration runs from the default of 300 to 950 (see paragraph 5.5.4 on page 93 for a detailed information on these topics). Next, we specified total fertiliser N input levels in the 'N-levels to optimise' list box: 0, 50, 80, 100, 150, 180, 200, 210, 240, 270, and 300. We removed the 400 kg input level. Then, we supplied a descriptive name for the oncoming optimisation in the 'Storage name' text field: 'JIN1993'. Press [OK]. All results of the optimisation will be stored in a directory called `MANAGE-N\SAVE\JIN1993.OPT` and can be retrieved for processing after the optimisation is finished.

An optional set of data that you can supply are the dates of several phenological stages (e.g., panicle initiation, first flowering). If you enter these dates (in days after transplanting) via the 'Set Phenology Dates' menu option, the phenology stages will be indicated in the cumulative fertiliser N application graphs. This is a handy option that can be used to check in one glance how timing of N application should relate to different growth phases of the crop. In the example presented in this Chapter, we entered the dates²² for panicle initiation (PI), first flowering (FF), and maturity (M).

²² Note that this information is only for output presentation purposes. It does in no way affect the simulations nor the outcome of the optimisation process.

Activate the 'Start Run' menu option to start the optimisation process. The very first model runs are performed for the 0 kg ha⁻¹ input level. Of course, no optimisation of nitrogen application is possible for this level: the results will only be used later as a reference. The next runs are the actual optimisation runs. When optimisation for a specific N input level is finished, all the results are stored in a separate directory for later reference. MANAGE-N continues until the N application of the last N input level has been optimised.

Note that the running itself on a IBMTM 80486 machine will take 3 to 6 hours (because 10 N levels are optimised with high precision, i.e., INT=900).

The next step is calculating the detailed dynamics of several state-variables throughout the growing season for every optimised N application. The resulting values are used for graphs, and the maximum attainable yield (weight of the rough rice, WRR) is also displayed in the 'Optimisation Results' window that appears when the abovementioned post-optimisation runs are completed.

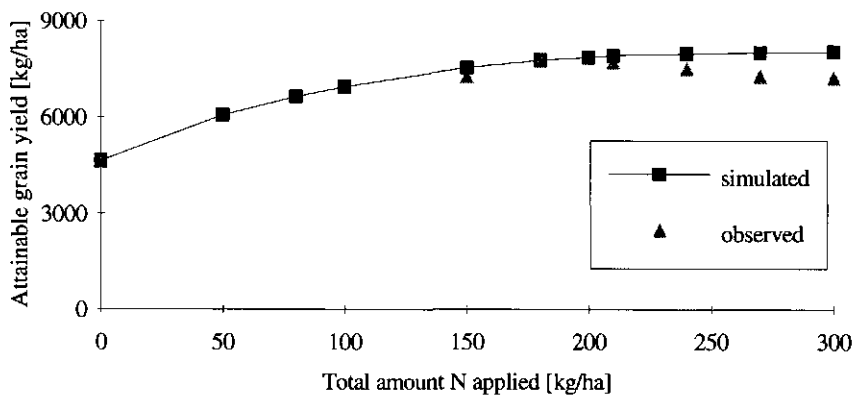


Figure 6.4 Maximum attainable simulated grain yield for a number of fertiliser N input levels. The simulated grain yields are obtained by using the optimised nitrogen applications for the individual N input levels, computed for cultivar Xieyou 10 at Jinhua, late season 1993. (This graph is not a snapshot of the MANAGE-N screen: observed values are not included in the MANAGE-N graph).

The 'Optimisation Results' window shows for every N input level the best values for the A, B, C and M parameters together with the corresponding WRR. After viewing the results, [Exit] the window and proceed to the 'View Graphs' menu option. A window opens that gives you instant access to the most interesting graphs that summarise the results obtained from the N optimisation.

The first two graphs are grain yield (WRR) and total crop biomass (WCR) response to the selected N input levels (press [Yield-response] and [Biomass-response] respectively). Figure 6.4 displays the yield-response that was achieved (simulated) in the Jinhua 1993 case, together with the observed values. There seems to be a potential for improving N management, especially at the higher N input levels. Of course, this theoretical analysis is only the first step: specific experiments should be performed to confirm these results.

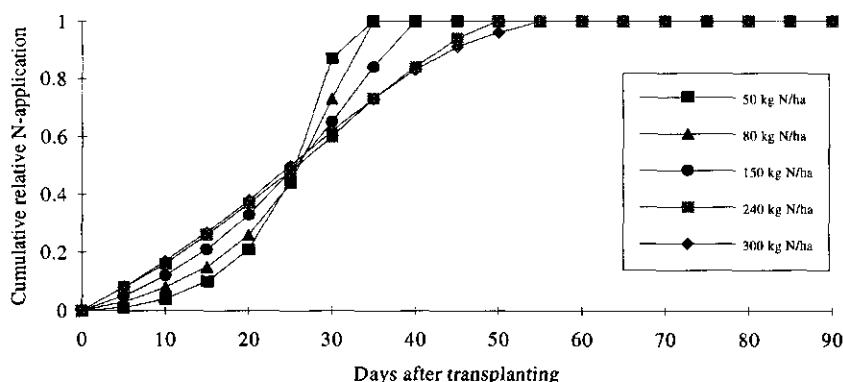


Figure 6.5 Recommended timings of the relative N application for cultivar Xieyou 10. Not all N levels that have been optimised in this Chapter are shown, only a selected few are displayed to show differences in timing. This graph is not a snapshot from the MANAGE-N system; because there were so many N levels for which we optimised the graph in MANAGE-N was to crowded for printing here. Instead, we imported the APCUM graphs of 5 selected N-levels in a spreadsheet to create the above graph.

The actual timings of the N applications can be viewed in the APCUM graphs. There are two options here: you can view a [Scaled APCUM] graph, which is excellent for comparing timings between N input levels. Or you can view [APCUM per N-level]. An example of the first type of graph is presented in Figure 6.5. APCUM curves for only a few N input levels were selected and composed in Figure 6.5.

Two individual APCUM graphs are shown in Figure 6.6 together with the earlier mentioned phenology indicators. These APCUM graphs are a basis for developing discrete N application schemes for evaluation by the 'Split Evaluation' method.

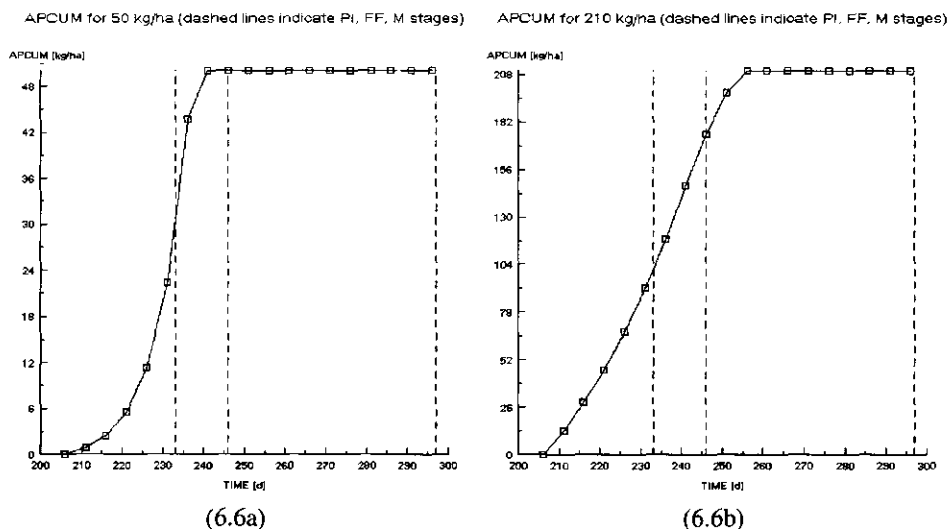


Figure 6.6 Optimised cumulative fertiliser N application curves with phenology indicators. The above graphs have been computed by MANAGE-N for the 50 kg N ha⁻¹ level (FERTMX=50., Figure 6.6a) and the 210 kg N ha⁻¹ level (FERTMX=210., Figure 6.6b). Both graphs used the data from the Jinhua 1993 experiment. The dashed lines indicate panicle initiation (PI), first flowering (FF) and maturity (final harvest, M) respectively. Note the difference in timing of the N application.

A very simple 'economic analysis' can be performed by using the right-hand part of the 'View Optimisation Graphs' window. When you supply values for the cost of 1 kg fertiliser nitrogen and the price that 1 kg of rice fetches on the local market, the [Additional Income] button displays a graph (Figure 6.7) that indicates the additional income a farmer should expect at each of the fertiliser N input levels. The additional income is defined as the total net income at a certain fertiliser N input level minus the income at the zero fertiliser N input level. You can supply the cost of nitrogen as well as the price of rice in local currency. The additional income graph shows the economic optimum N input level for the farmer. Of course, this does not necessarily correspond with the maximum attainable yield. When fertiliser nitrogen is cheaper, the economic optimum moves towards the higher N input levels. A change in cost of nitrogen and price of rice is not only reflected in the shape of the economic optimum curve, but also in the absolute amount of additional income.

Since MANAGE-N computes dry grain yield, a factor is present to correct for the moisture level of the grains. Valid values for this factor range from 0 to 25 % moisture content (dry weight basis).

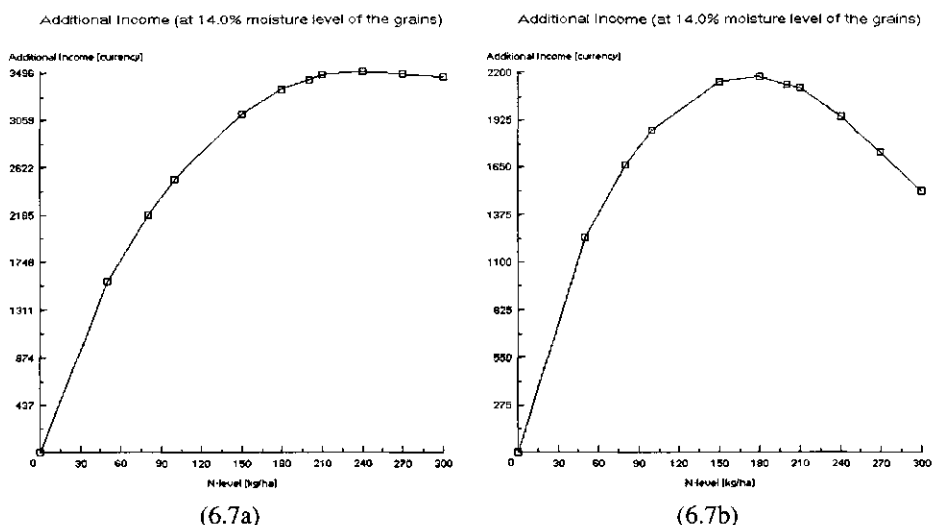


Figure 6.7 Additional income graphs for different costs of 1 kg fertiliser N. In Figure 6.7a, the cost of 1 kg fertiliser nitrogen is 1.5 times the price that 1 kg of rice fetches on the local market; in Figure 6.7b, the cost is 8.0 times the price of 1 kg rice. Note that between these scenarios, the optimum amount of fertiliser to be used by the farmer changes. The absolute amount of additional income changes too.

6.5 Evaluate split-dressings

The fourth *method* is available to evaluate discrete split-dressings of nitrogen. This evaluation process is often the last step in the procedure of optimising N recommendations, but the evaluation of split-dressings can also be an independent activity (i.e. without prior numerical optimisation). In both cases a generic crop/soil file is required for assessment of crop performance under different split schemes: the mean FSV values for a specific combination of site, variety and season are necessary. In short: before using the 'Split Evaluation' *method*, it is essential to go through the exercise of calculating FSV1, FSV2 and DATFSV for the various treatments, select mean values for these and store them in a generic crop/soil file (with extension '*.ALL', see previous paragraph). Also the other parameters should, obviously, represent the site-season-cultivar combination under study. More information on the theory behind the 'Split Evaluation' *method* can be found in paragraph 5.6 on page 97.

Without further ado, let us continue with the example. Go to the 'Method' menu pad and activate the 'Split Evaluation' menu option. Next, select the input files. In the example presented here, we use the same crop/soil file as we did when the 'Optimise N-curve' *method* was executed: 'ZZ1993.DAT'. The corresponding weather ('JINHUA1.993') and timer file ('T_JIN93.DAT') are also selected. As soon as you return to the main menu, the menu option 'Set Split Evaluation Parameters' has become accessible. Activate

it. A new window opens, which enables you to enter both timing and size of the nitrogen splits. Both timing and size can be based on output graphs from the previous exercise. These cumulative fertiliser N application curves for selected N input levels can be displayed now for your convenience as reference.

Besides the SPLIT and SPLDAT tables, there is another parameter that we have not encountered in the earlier *methods*. That is the parameter K2, tabulated versus time in the input function K2T. This parameter (see paragraph 5.6) is necessary to express the rate of N loss from the N pool stored in the soil. For $K2 = 0.10$, the pool size drops by 10% every day as a result of the losses (plus by some more, due to uptake). We recommend that K2 be fixed in time, unless there is strong evidence that reality is different. So for the time being, fill out the K2T table such that K2 obtained by linear interpolation from this table is constant. Now try different values: 0.1, 0.15, 0.20. It becomes clear that the value of K2 has an impact on final yield, which is sensitive to changes in K2. In this example we will use the default value of 0.1 for the whole simulation period. These values are automatically supplied by MANAGE-N.

Let us retrieve the results of our previous N optimisation so we can use the optimised N application curve as reference. Go to the [Choose Optimisation] button in the 'Split Evaluation' window. Press the button, then select the name of the optimisation case you wish to be retrieved. In the example presented in this Chapter, we used 'JIN1993' as the storage name of the optimisation, so the name to look for is 'JIN1993.OPT'. When the optimisation case is properly selected (watch for the square-root $\sqrt{}$ sign), press the [OK] button. All the results and graphs are now retrieved by MANAGE-N. The [Optimisation Info] button becomes available, and pressing it shows you the complete optimisation output with regard to total N input levels, values of A, B, C and M and the corresponding grain yield (WRR).

The 'N-level' and [Graph] button work together to display the cumulative fertiliser N application curve (APCUM) for a specific N input level. Of course, an optimisation must have been performed for that level. In the example presented here, we will look at different N split schemes for the 210 kg N ha⁻¹ input level. Therefore, go to the 'N-level' field, enter '210' and press the [Graph] button²³. When we look at the APCUM graph (see Figure 6.6) of 210 kg N ha⁻¹, we can think of several ways to break down the continuous N application curve into discrete splits. Not only timing itself, but also the number of discrete splits is important. One point should be stressed here: as long as production is N-limited, it is virtually impossible to get as high a grain yield via split application as under the (hypothetical) continuous application because of smaller losses in the latter case. Knowing this, we will try combinations of split numbers and timing of the splits, but all based on the APCUM curve. Also, take your expert knowledge into account.

²³ If you want to look at another N input level, or if you did not optimise for 210 kg ha⁻¹, choose another value.

For the first try, we decide upon two splits: one before panicle initiation (PI), and one between PI and the day when the APCUM curve reaches its maximum: 100 kg N at 13 days after transplanting and 110 kg at 39 after transplanting (DAT). Enter these values in the proper boxes and then press [Use Data] to instruct MANAGE-N to store these values.

[Exit] the 'Split Evaluation' window and run the model. When you activate 'View Tables', you notice that other state-variables are present, all relating to discrete splits and to the nitrogen pool in the soil. You will also notice that the grain yield (7125 kg ha^{-1}) is not as high as the grain yield obtained at the continuous nitrogen application (7906 kg ha^{-1}). Examining the periods over which the nitrogen demand of the crop is higher than the actual uptake can help us decide how to time the splits better. Use the [View Graphs] menu options to draw graphs of DEMAND (N demand of the crop) and NUPT (actual nitrogen uptake by the crop) versus TIME.

In this case, it is better to increase the number of splits. After a few tries we arrive at the following doses and timing: 50 kg N (8 DAT), 60 kg N (15 DAT) and 100 kg N (30 DAT). The final simulated grain yield with this combination is 7489 kg ha^{-1} (for comparison, the highest observed yield in this experiment was 7690 kg ha^{-1}).

'Split Evaluation' is a good tool to quickly assess the effects that split schemes have on final yield. The uncertainty on the K_2 factor, however, makes that predictions of that absolute N uptake and absolute yield level are of less significance than the relative effects of split scheme variations. Further research to expose the ranges of K_2 is highly recommended.

7 MANAGE-N Reference guide

J.J.M. Riethoven¹

¹ DLO Research Institute for Agrobiological Sciences (AB-DLO), P.O. Box 14,
6700 AA Wageningen, The Netherlands

In this Chapter, the user is presented with an in-depth description of each menu pad, menu option, and window that is part of the MANAGE-N tool. The information in this section is purely meant as a reference, only to be looked into when specific questions arise regarding a certain option in MANAGE-N, or when the user seeks a more detailed understanding of the system. Because this Chapter directly aims at the user, the style adopted here is 'active' writing in the second person (using 'you' instead of 'the user'). This makes reading easier. A top-down structure is used meaning that first, a brief description is given of each menu pad, then each menu option is discussed, and within each menu option the input or action window is described (if applicable).

Furthermore, at the end of this Chapter, a description can be found of one of the most important windows: the item-selector, as well as a short introduction on how to use the TTSELECT utility for creating graphs. Please note that the actual Figures²⁴ of screens and menus might differ slightly depending on the exact version of MANAGE-N on the distribution disks. All snapshots of screens and menus have been made of MANAGE-N version 1.0.

7.1 Main menu

After starting MANAGE-N, the menu bar that appears at the top of the screen is called the main menu. This name seems to indicate that more menus are present. However, this is not the case. The main menu is just the top-level of all menu options and screens, and exiting a window is defined as returning to the main menu.

Menu pads are located on the main menu bar; each menu pad groups menu options in the menu popup that are closely related (see also paragraph 4.3). The menu is organised in such a manner that options are only available when needed, or when certain prerequisite conditions have been met. For example, the 'Run' menu pad cannot be accessed before you have chosen both a *method* and input files (in some cases, also parameters have to be defined).

²⁴ Since the Figures of menu options and windows in this Chapter are printed in two colours only, information has been lost regarding the differences in colour or intensity of several items (e.g., 'dimmed' or disabled items). On screen they display much better; the Figures should be taken as a reference only and not as an accurate copy of the screen.

7.1.1 System

The 'System' menu pad gives you access to one of the smallest menu popups: the only menu option that is located there is the 'Quit MANAGE-N' menu option. The reason for making a separate menu pad out of this option is to prevent you from accidentally pressing the 'Quit MANAGE-N' option.

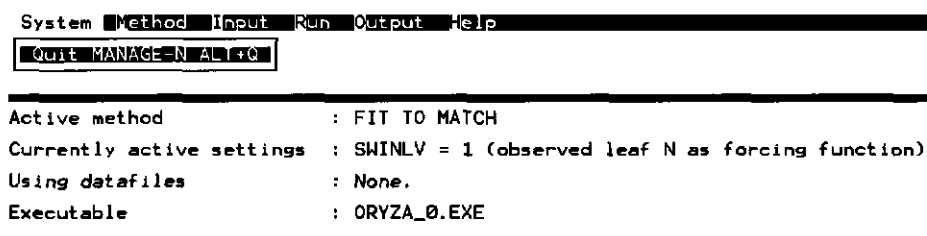


Figure 7.1 The 'System' menu pad and the accompanying menu popup.

Quit MANAGE-N

Activating this option lets you exit the MANAGE-N tool. Before exiting, you must confirm that action. This menu option has a hot key assigned to it: pressing [Alt+Q] will exit the MANAGE-N tool anywhere in the system.

When you leave MANAGE-N, it cleans the work area. The work area is used as a temporary repository of input and result files. If results of the 'Single Run' and 'Split Evaluation' *methods* should be stored for later reference, you must do so before leaving MANAGE-N. Refer to paragraph 7.1.5 for further instructions.

7.1.2 Method

The 'Method' menu pad lists the available *methods* or tasks that can be performed by MANAGE-N. By activating a *method*, you determine further options offered and inputs required by MANAGE-N. The *methods* are listed in a logical order: you would go through these *methods* successively when performing a full analysis to generate a well-founded fertiliser management recommendation.

Each *method* uses its own version of the ORYZA_0 model. These are pre-set by MANAGE-N when a *method* is chosen; it cannot be changed.

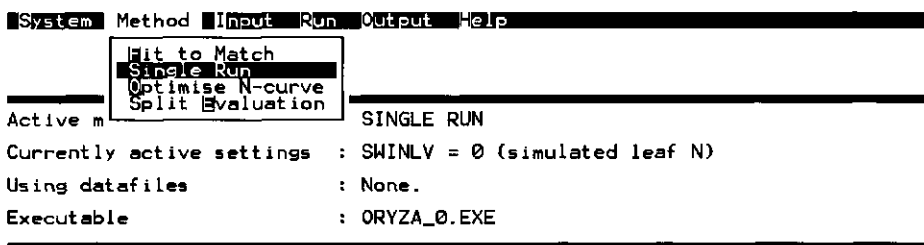


Figure 7.2 The 'Method' menu pad and its contents.

Fit to Match

By this *method*, you compute the value of FSV via a comparison between field observations and simulated growth curves (i.e., biomass versus time). FSV is a calibration factor expressing the efficiency by which the crop utilises radiation and leaf N to produce dry matter. FSV values and the accompanying DATFSV date are used to characterise a particular combination of site, season, and cultivar. A representative FSV value is usually composed from the FSV values obtained by analysing (fitting) various treatments available from a case.

After activating this menu option, the MANAGE-N tool will set the switch value of SWINLV to =1, indicating that the model (ORYZA_0.EXE) will impose measured leaf nitrogen as forcing function instead of simulating leaf nitrogen from underlying principles. You should check whether all input requirements are met. Use the 'Edit A Crop/Soil File' menu option (see paragraph 7.1.3) to view all parameters and measured values and, in case they are incorrect or not present, edit them.

Please refer to paragraph 5.4 (page 90) and paragraph 6.2 (page 100) to get detailed information about the entire process of computing FSV.

Single Run

This menu option activates the 'Single Run' *method*. With the 'Single Run' *method*, you can experiment freely with input of the model. It offers a wide range of outputs from ORYZA_0 and is meant for detailed study of crop responses. 'Single Run' can be used at any time. For details see paragraph 5.5 (page 92). Graphs can be viewed of every variable. ORYZA_0 can be run using simulated leaf nitrogen or with forced (observed) leaf nitrogen (see the 'Toggle SWINLV' menu option in paragraph 7.1.3).

Optimise N-curve

Choosing this *method* instructs MANAGE-N to use the special optimisation derivative of the ORYZA_0 model (ORY0OPT.EXE). 'Optimise N-curve' will eventually maximise

final yield for each N level that you select. Maximising yield is performed by changing the shape of the APCUM (cumulative N application) curve until optimum values for the curve-defining parameters A, B, C, and M have been found. This procedure is explained in detail in Chapter 5.

Before the actual optimisation can commence, you must define some parameters that influence the optimisation procedure, as well as enter a descriptive name for the optimisation. The name will be used to store the optimisation results. Defining these parameters is done through the 'Set Optimisation Parameters' menu option (page 126).

Split Evaluation

After N optimisation finishes, you can further explore how continuous N application (as is used in the N-optimisation procedure) can be transformed into discrete split-dressing and what the effect will be on final yield. This is done by the 'Split Evaluation' *method*. It is recommended to base split evaluation on a case study of N optimisation, but the 'Split Evaluation' *method* can also be used directly. A special derivative of the ORYZA_0 (ORY0SPL.EXE) model is used to compute the effects of discrete split-dressings on final yield. Detailed information on the processes underlying nitrogen dynamics in case of discrete split-dressings can be found in paragraph 5.6 on page 97. Enter your values for the size and timing of splits in the 'Set Split Evaluation Parameters' menu option (page 129).

7.1.3 Input

Whenever you must select or define data for the model, the 'Input' menu pad is where to look. It offers user-friendly editing of data files, easy selection of data files, and the effortless definition of parameters required under the active *methods*. All the menu options except the 'Toggle SWINLV' option give access to windows that provide a wealth on information about the required inputs.

It is essential to use the 'Input' menu pad before any *method* can be used (i.e., before any model can be run). You should, however, first select a *method* via the 'Methods' menu pad. When you use the 'Edit A Crop/Soil File' menu option, only those parameters are required that correspond with the active *method*.

If your input files need no editing, you can proceed to 'Select Input Files' on page 125.

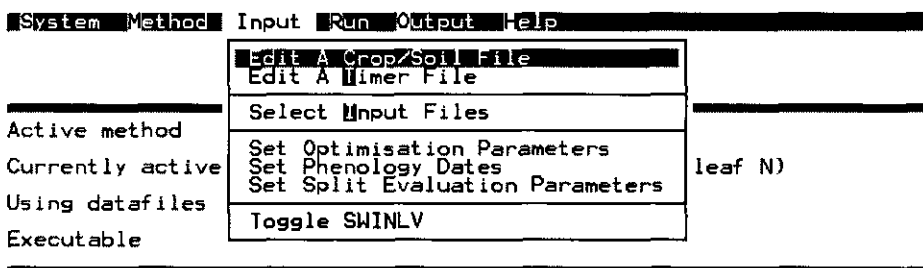


Figure 7.3 The 'Input' menu pad and the contents of the menu popup.

Edit A Crop/Soil File

This menu option offers an easy interface to entering and changing data in the crop/soil file. When a crop/soil file is already active (i.e., already chosen with the 'Select Input Files' menu option), then the 'Edit Datafile' window opens at once. If no crop/soil file has yet been selected, MANAGE-N first asks you which file should be edited. It then parses the file and extracts all data that are vital to ORYZA_0.

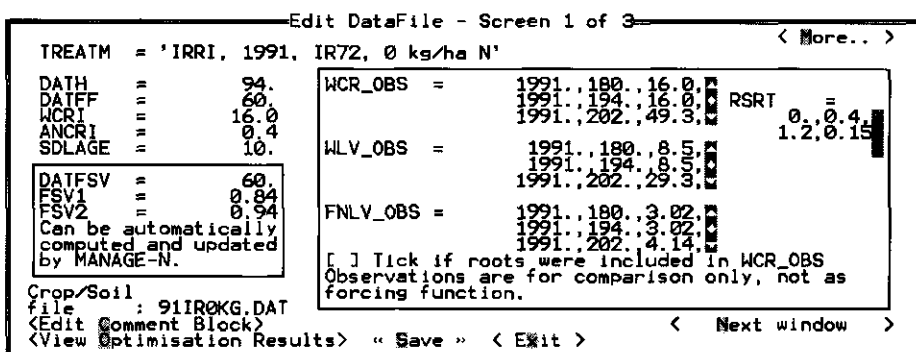


Figure 7.4 The first screen of the 'Edit A Crop/Soil File' menu option. This screen lists parameters, variables, and functions that are treatment-specific. The time-series of observations refer to a particular case, i.e. a treatment within an experiment, and should be mutually consistent.

The parameters and variables are presented in three windows. With the two push buttons at the bottom right side named [Next Window] and [Previous Window], you can switch to these windows. All parameter names are displayed, with next to them the current value(s) they have in the crop/soil file. If no value is displayed, then no value was present for that particular parameter in the data file. The parameters are displayed on a 'need-to-know'

basis. Whenever a parameter is not needed for calculation in the current *method*, the name of the parameter is replaced by dashes '-----'. The value cannot be altered.

For example, in the 'Single Run' *method*, the *SPLIT* parameter is not required, because 'Single Run' simulations use a continuous application curve expressed in A, B, C, M, and FERTMX. The *SPLIT* parameter (located in the third window, see Figure 7.6) is replaced by dashes.

For each parameter and variable, a short description is available. Moving to the value field of a parameter or variable will display this description at the bottom of the screen. This description is also stored in the data file when the [Save] button is pressed, thus making documenting data sets very user-friendly. The description cannot be edited by you.

The parameters and variables are grouped in a logical manner. All observations or parameters derived from treatment-specific observations are listed in the first window. This includes time-series of observations like crop weight (WCR_OBS); these should be entered as an array with columns that denote the year, day of year, and the actual observed value.

Parameters that are supposed to characterise the variety-site-season under study are listed in the second window. The third window holds all parameters that define the continuous fertiliser application curve, as well as the discrete split dressings; this also includes parameters needed for the estimation of the amounts of phosphorus and potassium required (these are optional).

Edit DataFile - Screen 2 of 3

Crop and soil parameters apply to all treatments in an experiment (experiment = a specific combination of cultivar, site and season). < More.. >

Crop parameters	Soil parameters
FNCLV = 0.50 FNMAXT = 0., 0.03, 40., 0.017, 60., 0.015, 80. RUR = 0.2 NUPCO = 0.03 ANLVMX = 100. FNSO = 0.01	HIS = 0.551 HISLOP = 0.00767 HI will be computed SST = 0., 0.6, 800., 0.6 RECT = 0., 0.0, 60., 0.8, 65., 0.4, 800. MXUP1T = 0., 5, 800., 5. ----- = 0., 0.1, 999., 0.1

Crop/Soil file: 91IR0KG.DAT

<View Optimisation Results> " Save " < Exit > { Next window } { Previous window }

Figure 7.5 The second screen of the 'Edit A Crop/Soil File' menu option. It lists parameters that characterise the site, season and cultivar, i.e., are not dependent on N management (treatment).

When entering and editing data in these windows, MANAGE-N will perform a simple check on the correctness of the values. This check is not on biologically plausible values of the parameters but on correctness of syntax (e.g., check if input is really a numerical value, check if an array consists of the right number of elements, and so on). MANAGE-N will

warn when an incorrect response is detected; and it will then return to the current parameter.

During the entering and changing of parameter values, you can also take a quick look at the results of previous optimisations: that is, you can retrieve information on the optimised N levels (FERTMX) and the corresponding A, B, C, M, and dry grain yield (WRR). This can come in handy when you want to analyse model behaviour based on the optimum fertiliser application curve. The push button [View Optimisation Results] at the bottom left side of the screen gives the opportunity to recollect these parameter values.

Default values are automatically assigned for certain parameters that have no value yet in the crop/soil file. The [More..] button at the right top part of every screen displays, besides other relevant information, also the default values for a selection of parameters.

Edit DataFile - Screen 3 of 3

< More.. >

Parameters defining continuous fertiliser application curve and discrete split dressings. The parameters A, B, C, and M can be optimised (for maximum grain yield) for given FERTMX by MANAGE-N, but values may also be user-defined. Values of split-dressings can only be user-defined.

<pre> ----- = 35. ----- = 99999999. ----- = 0. ----- = (leave empty if not used) Crop/Soil file: 911R0KG.DAT </pre>	<pre> FERTMX = 50.0000 ----- = 1 ----- = 0.1 ----- = 300. ----- = 30. KNRAT = PNRAT = </pre>	<pre> Split dressings: ----- = ----- = </pre>
---	---	---

< View Optimisation Results > " Save " < Exit > < Previous window >

Figure 7.6 The last screen of the 'Edit A Crop/Soil File' menu option. The parameters listed in this screen define fertiliser application regimes.

Besides parameter values, you can also edit a special comment that will be placed at the top of the data file when it is stored. The [Edit Comment Block] push button in the first window (see Figure 7.4) guides you to a special window in which this comment can be changed. The comment is best used as a detailed documentation of the treatment and experiment, or as a place to put in any special remarks concerning observed values or parameter derivations (methods of measurement, used algorithms, who did the measuring, maybe literature from which values were taken, etc.). This way parameters can be traced back to original observations.

MANAGE-N will put asterisks '*' in front of every line that you enter. Pressing the [Exit & Save] push button will store any changes that have been made and will return you to the first 'Edit DataFile' window. [Cancel] negates any changes while returning to the 'Edit Datafile' window.

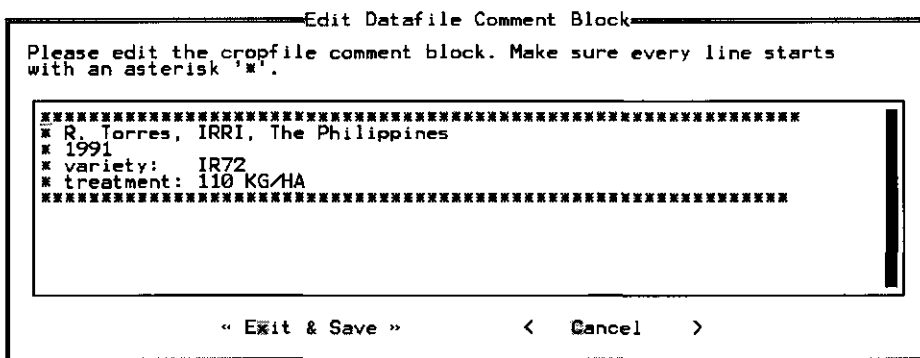


Figure 7.7 The 'Edit Datafile Comment Block' window. It enables easy editing of vital information.

The changed data can be saved by pressing the [Save] push button. When data must be saved under another name, you can change the name of the data file in the 'Crop/Soil File' field at the left side of the window. If a file already exists with the same name, you will be asked to confirm the save action.

Note that the 'Optimise N-curve' and the 'Split Evaluation' *methods* use values for crop and soil parameters that are supposed to be general for the site-season-cultivar combination, and thus do not refer to a specific case (=treatment within an experiment). This is in contrast to the 'Single Run' and 'Fit to Match' *methods*. As a reminder of this difference, the crop/soil file must be saved using an extension of *.ALL for the 'Optimise N-curve' and 'Split Evaluation' *methods*.

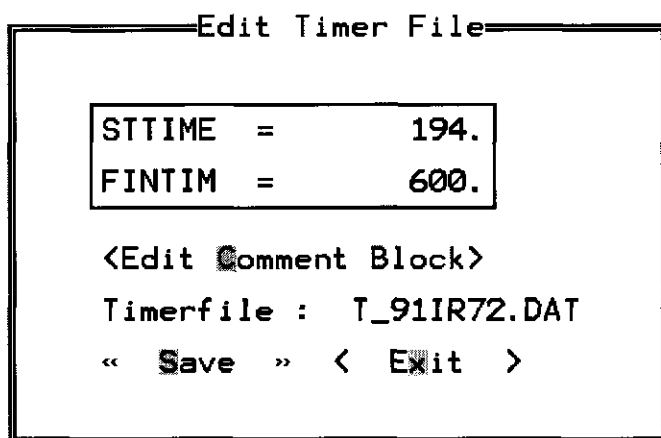
Edit A Timer File

Press this menu option if you want to edit a timer file. If no timer file has yet been selected, MANAGE-N will give you the opportunity to select the target timer file. Otherwise, the 'Edit Timer File' window opens at once. This window is very simple, with basically only two parameters that can be edited. The first parameter is the starting time (STTIME) of the simulation (transplanting day), the second is the stop time of the simulation (FINTIM). The stop time should be set sufficiently high, like 600. or 999. because the simulation stops by itself when the date of the final harvest has been reached. Do not use FINTIM to stop the simulation prematurely.

As with the 'Edit DataFile' window, also for the timer file, a special comment can be placed at the top of the timer file. The comment can be edited by pressing the [Edit Comment Block] push button: the 'Edit Datafile Comment Block' window will pop up to facilitate easy editing (see Figure 7.7). You can change the name of the timer file. Press the

[Save] button to store the changed contents into another file. When using an existing file name, you will be asked to confirm overwriting the older file.

Pressing the push button [Exit] in the 'Edit Timer File' window leaves the window without saving any changes.



```

Edit Timer File

STTIME = 194.
FINTIM = 600.

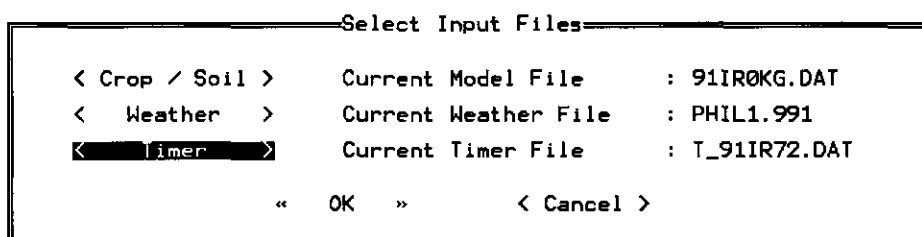
<Edit Comment Block>
Timerfile : T_91IR72.DAT
< Save > < Exit >

```

Figure 7.8 The 'Edit Timer File' window enables editing of the start date of the simulation.

Select Input Files

The 'Select Input Files' menu option enables you to choose the input files with which you will run the model. The 'Select Input Files' window will open, and in this window three types of input files must be chosen: a crop/soil file, a weather file, and a timer file.



```

Select Input Files

< Crop / Soil >   Current Model File   : 91IR0KG.DAT
< Weather >      Current Weather File  : PHIL1.991
< Timer >        Current Timer File    : T_91IR72.DAT

< OK >           < Cancel >

```

Figure 7.9 The 'Select Input Files' window.

Pressing any of the three file buttons [Crop / Soil], [Weather] or [Timer] will bring forward the item-selector. The item-selector displays a list of files from which you must pick one. After all three types of files are selected (watch for the square-root $\sqrt{}$ sign), the

[OK] button activates these selections and returns you to the main menu. Pressing [Cancel] returns you to the main menu without activating the selections. As long as all three files are not selected, the 'Start Run' menu option remains disabled.

Set Optimisation Parameters

The setting of parameters needed for the N optimisation runs is handled by the 'Set Optimisation Parameters' menu option. This is required before the optimisation procedure can commence. See also paragraph 5.5.4 on page 93.

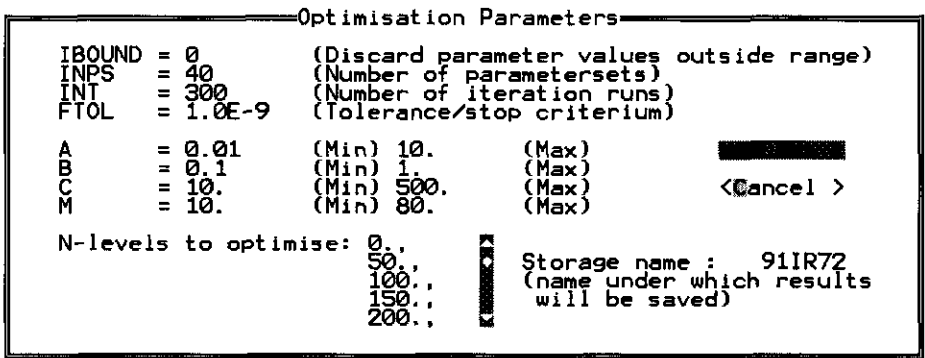


Figure 7.10 The 'Set Optimisation Parameters' window.

The 'Optimisation Parameters' window lets you edit parameters that influence the numerical aspects of N optimisation. The window has three main sections: the first includes parameters that deal with the technical side of the optimisation, the second section deals with the ranges of the parameters that describe the continuous N application curve (the parameters that will be optimised), and the last section covers information on the N levels for which the optimisation is to be performed. The first section holds four parameters that influence the optimisation:

INPS

The initial number of parameter sets. These sets (and the accompanying yields) are used to generate new sets, which will be simulated in the next standard model run (iteration run). If the outcome of such a model run is better (in terms of yield), then the new parameter set will replace the worst set in the initial collection of parameter sets. This initial collection is also called the 'vase'. A good rule of thumb is to choose a number that is at the very least 10 times the number of parameters that will be optimised (in this case, the number of parameters equals 4: A, B, C and M).

INT

The maximum number of iteration runs. These runs actually simulate the yield with regard to the newly computed parameter sets (from the 'vase') and replace parameter sets in the vase when the outcome is better than the worst case in the 'vase'. Typically the number of iteration runs should be at least 6 times the number of initial runs ($6 * INPS$).

FTOL

The tolerance level or stop criterion of the optimisation. If the difference between the minimum and maximum yield in the 'vase' (the actual tolerance level RTOL) is smaller than this tolerance level, then the optimisation is stopped.

The actual tolerance is calculated in this way:

$$RTOL = 2 * ABS(WRRMAX - WRRMIN) / (ABS(WRRMAX) + ABS(WRRMIN))$$

IBOUND

When computing a new parameter set from the sets located in the 'vase', it is possible that the new computed value of a parameter is out of its range as specified in the second section of this window. In that case, it is either possible to discard the parameter set and recompute a new set ($IBOUND = 0$), or to set the value of such a parameter to its boundary ($IBOUND = 1$).

By default, a new set will be generated ($IBOUND = 0$).

The second section of the 'Optimisation Parameters' window defines the boundaries for the four parameters that are used to describe the continuous N application curve. The predefined ranges have been tested to satisfy a wide range of environments and varieties. You should be careful when narrowing the range of a parameter: better solutions to the N application curve might be ruled out unintentionally.

Changing the ranges is possible for the current session only; when you exit MANAGE-N by 'Quit MANAGE-N' and then start MANAGE-N again, the default ranges are back in place.

The last section of the 'Optimisation Parameters' window determines the N levels for which the optimisation should be performed. You can add as many levels as needed, or remove them from the pre-defined list. The 0 kg N-level will always be included for reference purposes (it only computes the initial sets, though, and will not commence optimisation).

Another field in the last section is the 'storage name' field. You should enter a descriptive name for the optimisation that is about to run. MANAGE-N will use that descriptive name to create a new directory in which the results of the optimisation will be stored. These results can be reviewed at any time in the main menu. When you supply a name that already exists, MANAGE-N will warn you and ask whether to use that name anyway (and

overwrite any results that might have been stored within that directory) or to choose another name. If you do not supply a name, MANAGE-N will create one automatically (a random 8-digit number).

Press [OK] to store the modifications. [Cancel] will take you back to the main menu without saving. You are required to enter the 'Optimisation Parameters' window and store the settings before an N optimisation run can commence.

Set Phenology Dates

This option allows you to set the dates of several phenological crop stages. As a result, these crop stages will be indicated in the recommended APCUM graphs by dashed lines to provide an easy reference (see for an example Figure 6.6 on page 112). The 'Set Phenology Dates' menu option opens a window (see Figure 7.11) in which you can enter dates (in days after transplanting) of the phenological crop stages. These are Active Tillering (AT), Maximum Tillering (MT), Panicle Initiation (PI), First Flowering (FF), and Maturity (M).

Phenology Dates	
Active Tillering (AT)	DAT 35. 60. 94.
Maximum Tillering (MT)	
Panicle Initiation (PI)	
First Flowering (FF)	
Maturity (M)	
« OK »	
< Cancel >	

Figure 7.11 The 'Phenology Dates' window.

The input of these phenology dates is optional. If you do not enter them, no phenological stages are indicated in the APCUM graphs. If you enter only First Flowering, then only First Flowering is indicated.

When you enter this window for the first time after selecting the crop/soil file, MANAGE-N will extract the values for First Flowering and Maturity from the crop/soil file. If you do not want them included in the graphs, then remove the values.

A safety check has been built in. All dates should be in increasing order, or else MANAGE-N will not accept these values. Also, you cannot enter a date that is larger than

the harvest date (DATH) in the crop/soil file. Press the [OK] button to save, [Cancel] will ignore any changes you might have made.

Set Split Evaluation Parameters

The 'Set Split Evaluation Parameters' menu option opens the 'Split Evaluation Window' (Figure 7.12), which enables you to set various parameters, thus proposing a split dose fertiliser application scheme.

Split Evaluation Window

Optimisation name : IR91 <Choose Optimisation>
< Optimisation Info >

N-level: 200.0 < Graph >

DAT [d]	Split size [kg / ha]	DAT	K2
20.	50.	0.	0.1
40.	75.	999.	0.1
60.	75.		

<Use Data>
< Exit >

Figure 7.12 The 'Set Split Evaluation Parameters' window.

To help determine the size and timing of the split-dressings, you can 'consult' an optimised continuous curve obtained earlier. During the optimisation, nitrogen was applied through a continuous N application, of which the optimum application scheme for a certain N level can be seen in the APCUM (cumulative application) graph.

Press the [Choose Optimisation] button to retrieve an earlier optimisation and to make the retrieved graphs available to the 'Split Evaluation Window'. The [Optimisation Info] button can be used to review the various N levels analysed earlier and the respective values of the curve determining factors (A, B, C, and M) as well as the corresponding weight of the rough rice (WRR).

Choosing an optimisation is required when you want to display a continuous N application graph for a specific N level. The N level must be entered in the 'N-level' field. As soon as you do that, then the [Graph] button becomes available (on the condition that you had selected an optimisation case, indeed). Press [Graph] to view the APCUM graph of that N level. If the graph is not available (for example, you entered 125 while no optimisation had been performed for that N level), MANAGE-N displays a warning and returns gracefully to the 'Split Evaluation Window'.

To actually define the split application, you must supply a discrete N application scheme. Both timing and size have to be specified. Timing must be defined in days after transplanting. You should also enter the table K2 (fraction of N pool lost per day; versus DAT)). Valid values for K2 are 0 to 1.0. Make sure that at least two rows of data have been entered because the table is used for interpolation between input points, e.g.,

```
0., 0.1,
999., 0.1
```

Create the first column (days after transplanting) in such a way that it envelopes the complete simulation period. In the above example, K2 has been entered as a constant 0.1 for the complete simulation period. The value of 999 for DAT has been chosen to be well beyond the harvest date.

Press the [Use Data] button to store all this information in the generic crop/soil file. The 'Start Run' command must be issued to run the model with these split-dressings, and the 'View Graphs' menu option must be activated to review the results. In case you press the [Use Data] button and not all required information has been specified, MANAGE-N will point this out and give another chance to enter the data needed.

Toggle SWINLV

This menu option controls the value of the switch parameter SWINLV, indicating whether leaf nitrogen should be simulated (SWINLV = 0) or forced as observed values (SWINLV = 1). Toggling can only be accessed while the 'Single Run' *method* is active.

7.1.4 Run

The 'Run' menu pad has only one entry: the 'Start Run' menu option.

Start Run

The 'Start Run' menu option starts the ORYZA_0 model to execute the active *method* (either 'Single Run', 'Fit to Match', 'Optimise N-curve', or 'Split Evaluation').

'Start Run' is not available when the requirements for a specific *method* are not met, or when not all data files have been selected. 'Start Run' has to be executed to update the output results after any change in the input (whether these be in data files or as parameters through menu options).

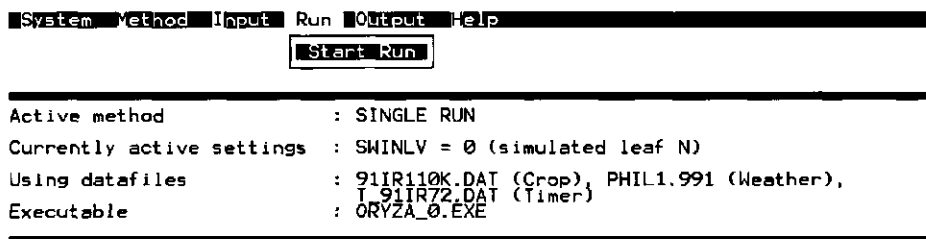


Figure 7.13 The 'Start Run' menu option, displayed just prior to a single run.

Before MANAGE-N actually starts the model run, it checks whether your input files are consistent: you cannot select a weather file of a year other than the year in which you conducted your experiment (MANAGE-N checks years of the observed time series with the selected year of the weather file). If this check cannot be passed, then MANAGE-N displays a warning and returns to the main menu.

For all *methods* except 'Fit to Match', you are directed to the 'Output' menu pad after a simulation run to inspect the results via 'View Tables', 'View Graphs' or 'View Optimisation Results'. If the current *method* is 'Fit to Match', activating the 'Start Run' menu option will commence 150 model runs. These model runs are required to find good values for the FSV factors. After these model runs, MANAGE-N will automatically open the 'Select Best Matchfactor' window. For details, see paragraph 5.4 and 6.2.

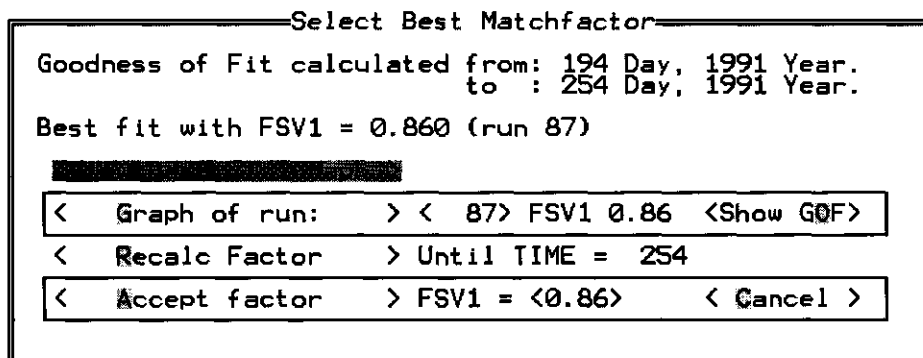


Figure 7.14 The 'Select Best Matchfactor' window. This window appears directly after a model run when the 'Fit to Match' method is in effect.

The first section of the window displays the period over which FSV1 (or FSV2) has been computed and the FSV value that gave the best fit between simulated and observed weight of the crop. The [View Best Fit Graph] button will display the graph that corresponds with the advised and best fit. The vertical dashed line indicates the break between FSV1 and

FSV2. This means that for FSV1, the GOF has been calculated from starting time of the simulation up to the dashed line; for FSV2, the period after the dashed line is taken into account. For the observed values plot (WCR_OBS), only measured data points are indicated.

- Viewing individual graphs

The next line in the 'Select Best Matchfactor' window displays two buttons and an edit field. The box around the buttons and fields indicates that the buttons within the box are connected. The [Graph of run] button will display the graph that belongs to the run (and FSV value) listed in the fields to the right of the [Graph of run] button. The run number can be changed by entering it manually or by using the tiny buttons < and >. After changing the run number, press the [Graph of run] button to view the results of that specific run. You can use this option when you are not satisfied with the advice MANAGE-N gave you with regard to the best FSV value: you can check out other graphs and decide which gives the best fit. The [Show GOF] button gives you the exact Goodness-Of-Fit of the specified run, as well as the number of data points used in calculating it.

- Adjusting the period

You can adjust the period over which FSV is to be calculated. Suppose the simulated graph fits the observed graph very well during the whole simulation, then consider moving the DATFSV value (the dashed vertical line in the graph, see Figure 7.15) to the end of the simulated period. That saves you another 150 model runs to determine FSV2.

To change DATFSV, enter a new TIME value in the field to the right of the [Recalc Factor] button. If you enter an invalid time (less than STTIME, larger than STTIME + DATH), MANAGE-N displays a warning message and you may correct your mistake. Then press the [Recalc Factor] button to let MANAGE-N re-compute the Goodness-Of-Fits for all 150 FSV1 values. When this is finished, press [View Best Fit Graph] again to inspect the results. Of course, it is also possible to decrease the TIME value if a bend in the graph occurs before the dashed line. Make sure that always at least two data points are located in the new period.

- Accepting the FSV values

When you are satisfied with the results, press [Accept Factor] to store the recommended FSV values in the crop/soil file. Otherwise, enter your preferred value in the field at the right of the [Accept Factor] button. Again, you can also use the tiny < and > buttons to decrease or increase the FSV value.

The [Cancel] button returns you to the main menu without updating the crop/soil file.

Sim. crop weight (WCR) vs. observed (WCR_OBS) FSV1=0.83

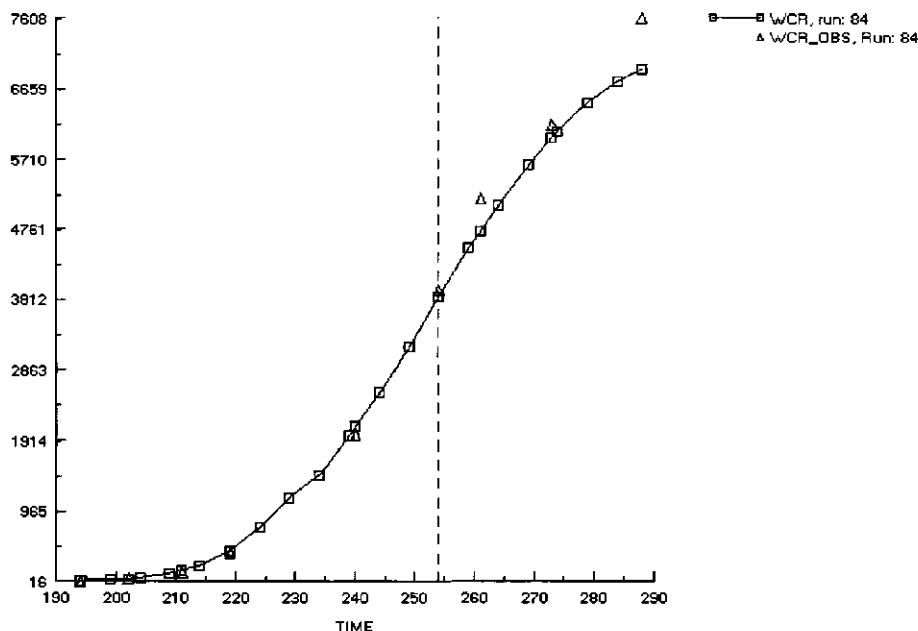


Figure 7.15 An example of the best fit graph for FSV1 calculation. The dashed line indicates the time (DATFSV) when the switch from FSV1 to FSV2 will be made. Observations are indicated by the triangle marker, without connecting line. The place of the rectangular markers (WCR) along the X-axis are determined by the number of days between consecutive simulation output and by the observation dates.

7.1.5 Output

This menu pad gives you access to the output functions of MANAGE-N. Within this menu, you can store or retrieve single run results, view tables or graphs, and display the optimisation results window.

Store Single Run Results

The menu option 'Store Single Run Results' enables you to store the dynamic (time-coursed) simulation results of a single model run in a directory. These results can be retrieved later on, or you can use another program like a spreadsheet, a statistical program or a plotting utility to process the data in a different way.

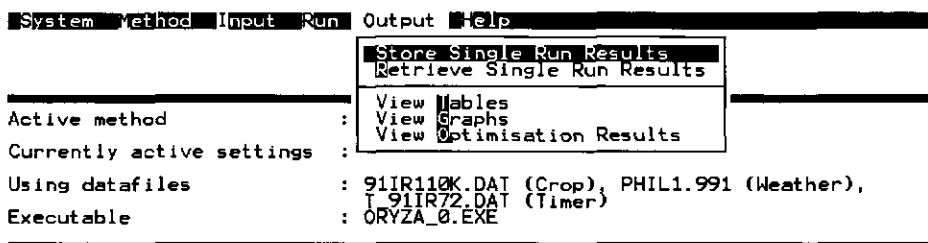


Figure 7.16 The contents of the 'Output' menu popup.

Not only are the ASCII (plain text) result files stored, but also a binary file that is required when you want to display time-sequenced graphs from within MANAGE-N. This binary file, RES.BIN, can be quite large. If you are sure that you do not want to view time-sequenced graphs through MANAGE-N's 'View Graphs' menu option, then you may delete it to save disk space. Deleting can only be done outside MANAGE-N. The storage directories are located within the MANAGE-N\SAVE directory. Storage directories of single runs have an extension of '.SR'.

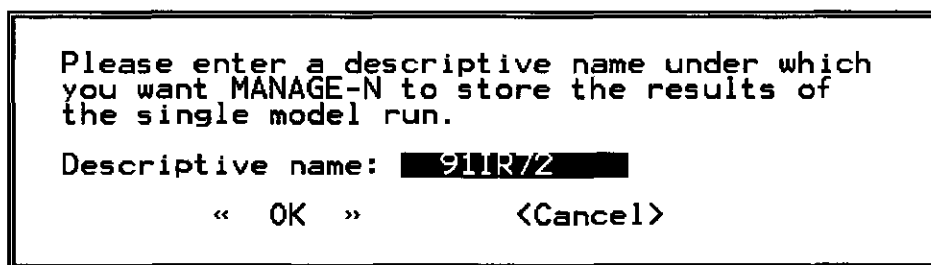


Figure 7.17 The 'Store Output' window.

This 'Store Output' window asks for a descriptive name under which the results of the just finished single run will be stored. The descriptive name, for example 91IR72, will be post-fixed with the extension '.SR' to indicate a single run, and then a directory with that name will be created in the MANAGE-N\SAVE directory. All result files of this single run will be stored in there. So, in this example, all result files will be stored in the MANAGE-N\SAVE\91IR72.SR directory.

The 'Store Single Run Results' menu option is only available when the active *method* is 'Single Run'. Results of the 'Optimise N-curve' *method* are automatically stored, using

the storage name that can be supplied in the ‘Set Optimisation Parameters’ window (page 126).

Retrieve Single Run Results

This allows you to retrieve previously stored results of a single run. The item-selector will pop up and you have to select one of the storage directories that are available. After the results have been retrieved, you can display them in a tabular (‘View Tables’) or graphical (‘View Graphs’) manner. Results of the ‘Optimise N-curve’ *method* can be retrieved via the ‘View Optimisation Graphs’ window.

View Tables

The menu option ‘View Tables’ lets you view the contents of the default model run result files. You can use [Ctrl+PageDown] to skip to the next file and [Ctrl+PageUp] to go to the previous file. Use cursor keys to scroll through the files (up and down as well as left and right).

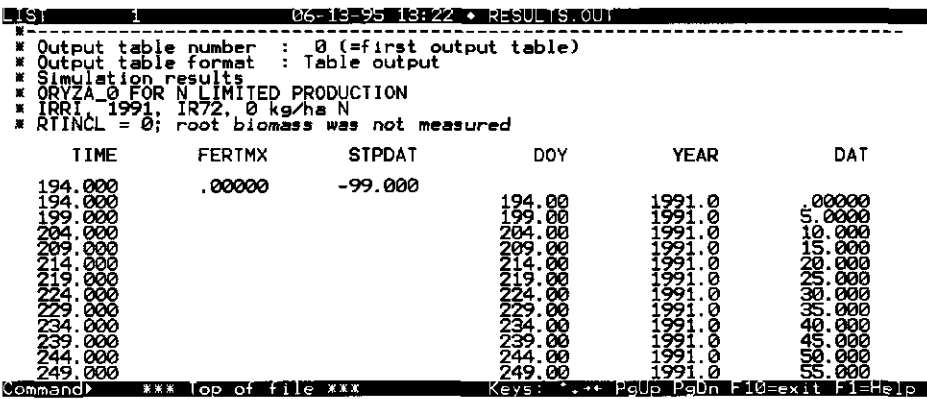


Figure 7.18 An example of the ‘View Tables’ menu option.

View Graphs

Press this menu option if you wish to see graphs of your model output. Depending on the type of run, you will either be guided to the plotting utility TTSELECT (‘Single Run’ and ‘Split Evaluation’ *method*, see paragraph 7.3) or to a special graph options screen in case of the ‘Optimise N-curve’ *method*. ‘View Graphs’ cannot be accessed when the ‘Fit to Match’ *method* is in effect because it has its own specialised graph window.

For the 'Optimise N-curve' *method*, the window displayed in Figure 7.19 will guide you to all the graphical results of the optimisation runs. The top-left box contains 4 push buttons:

- [Yield-response] displays the graph of maximum yield (dry weight of the rough rice, WRR) versus N level,
- [Biomass-response] shows the graph of maximum biomass (WCR) versus N level,
- [Scaled APCUM] shows the APCUM time curves for each N level, scaled individually to display relative differences in timing of the N application
- [APCUM per N level] displays the native APCUM graphs for each N level and, optionally, indicates several phenology stages. See also the 'Set Phenology Dates' menu option on page 128.

Examples of most of these graphs can be found in Chapter 6.

View Optimisation Graphs	
< Yield-response >	Moisture level of grains 14.0
< Biomass-response >	Cost of 1 kg N 1.00
< Scaled APCUM >	Price of 1 kg rice 1.00
< APCUM per N-level >	<Additional Income>
< Use RES.BIN > [View individual graphs using RES.BIN]	
Current optimisation graphs: IR91	<Retrieve Graphs>
	“ Exit ”

Figure 7.19 The 'View Optimisation Graphs' window.

The top right box (Figure 7.19) is reserved for the additional income graph. You have to supply the cost of 1 kg nitrogen, the price that 1 kg of rice fetches on the market, and the moisture level of the grains (as sold on the market). Cost and price are in local currency. The moisture level is needed because the grain yield in MANAGE-N is defined as being the dry grain yield. See also paragraph 6.4 (page 107).

After all prices are entered, pressing the [Additional Income] graph will display a graph of additional income versus N level. Use this graph to determine the N level that gives the best economic return.

The [Use RES.BIN] button in the 'View Optimisation Graphs' window is used to view graphs that are not predefined. Pressing [Use RES.BIN] activates the TTSELECT utility, and you can manually select the variables to be plotted. For each N level there is one model run for optimum A, B, C, and M, starting with run 0 for N level 0 kg ha⁻¹ (only native N soil supply). If the binary result file RES.BIN is not present, MANAGE-N displays a warning and returns to the 'View Optimisation Graphs' window.

The name of the currently 'active' optimisation case (of which graphs are to be shown) is displayed at the bottom of the window (you provided that name earlier). Next to this field

is the [Retrieve Graphs] button. You can use this button to retrieve graphs of other previously run optimisations. The item-selector will pop up and you have to select a storage directory. Press [OK] to activate that selection and to return to the 'View Optimisation Graphs' window. The [Exit] button returns you to the main menu.

View Optimisation Results

In addition to the graphical output, the 'View Optimisation Results' menu option gives you access to numerical results in the 'Optimisation Results' window. These are the final results of an optimisation case (name is displayed), grouped per N level.

Optimisation Results Window					
Results of optimisation : IR91					
N-level	A	B	C	M	WRR
0.0	1.4807	0.9078	300.7700	57.5790	3531.80
25.0	1.8686	0.3599	299.5600	65.0200	4317.10
50.0	3.0093	0.2909	55.6220	55.6360	4893.50
80.0	4.4286	0.1588	134.1200	62.6320	5186.70
100.0	7.4633	0.1578	196.9600	68.8610	5290.00
125.0	9.3609	0.1280	266.6900	61.0950	5379.00
150.0	9.1312	0.1281	296.0400	54.0740	5432.10
<Other optimisation>			<< Exit >>		

Figure 7.20 The 'View Optimisation Results' window.

The window in Figure 7.20 shows you the results of a specific optimisation. N level, the parameters belonging to the best N application curve, and the accompanying weight of the rough rice (WRR) are displayed. When more results are present than can be shown at once, you can scroll through the list using cursor keys. The [Other Optimisation] button enables you to pick another optimisation of which you like to view the results. The button, however, is disabled just after finishing an optimisation run. When you enter the 'Optimisation Results' window through the 'View Optimisation Results' menu option, you can access any results available.

[Exit] returns you to the main menu. If you just finished an optimisation run, you can view the currently active results in graphs by activating the 'View Graphs' menu option in the 'Output' menu pad.

7.1.6 Help

The 'Help' menu pad gives you access to the 'Help Index' and the 'About MANAGE-N' menu options.

Help Index

Pressing this menu option enables you to review all the topics that are available in the on-line help file. Topics range from theoretical background of MANAGE-N and N optimisation to general topics on the user-interface and copyright and to detailed context-sensitive topics on all the windows and menu options. The help topics cover the same information as those given in this Chapter but are less detailed.

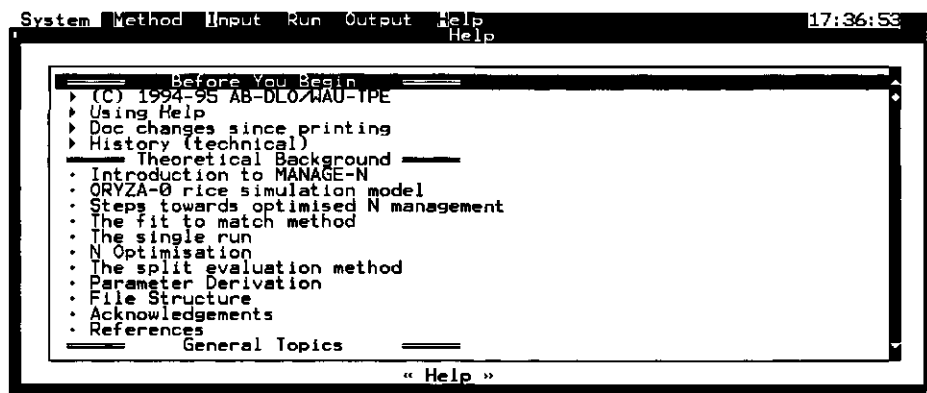


Figure 7.21 An example of the 'Help Index' screen.

You can scroll through the help index and get some idea of the topics. The first part of the help index is especially important: it contains the copyright and any changes that might have been made to MANAGE-N or its documentation after publishing this booklet.

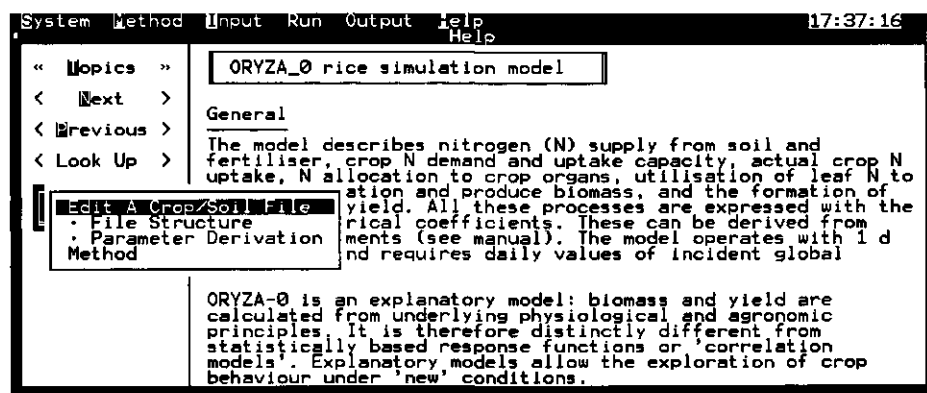


Figure 7.22 An example of a help topic with cross-references to other topics.

Help topics are linked (knowledgeable users know this as 'hyper-text') whenever details or related topics might be of interest to you. The [See Also] list button is enabled when other related topics are available. Move to that button and press [Spacebar] or [Enter], then pick a topic from the list that appears (see Figure 7.22) and press [Enter] again to jump to that topic. Help texts are often longer than one screen can contain. You can use the cursor keys to scroll through the text.

About MANAGE-N

This menu option (see the displayed window in Figure 7.23) gives you some key information on MANAGE-N. It displays the copyright, the address of the primary contact persons, the version of MANAGE-N, as well as some general system characteristics like free memory, the processor (CPU) that is available, and the graphics adapter. (See Figure 7.23.)

7.2 The Item-Selector

The screen called 'Item-Selector' is a window of paramount importance (see Figure 7.24). It enables you to make a selection from a given list of items. Items can be models, input files, parameters, directories, etc. Some items offer additional information.

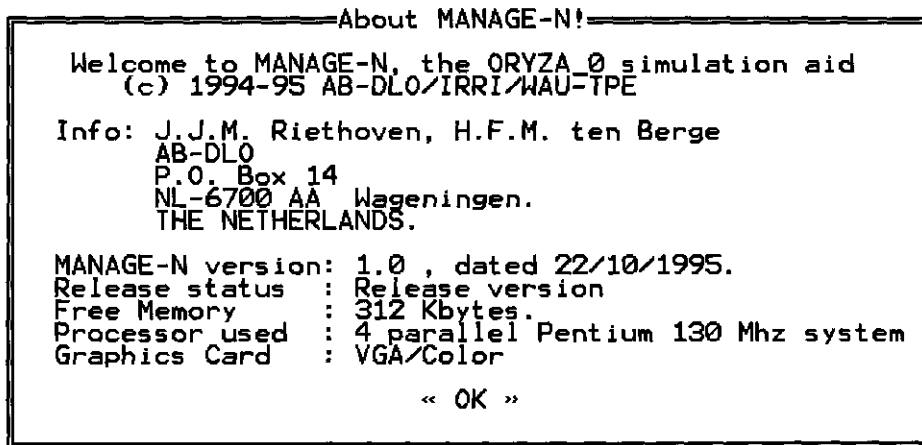


Figure 7.23 The 'About MANAGE-N' window.

The rectangular box displays the list of possible selections. If more items are present than can fit in the box, use the small arrows at the right side of the box to move (i.e., ‘scrolling’) through the list. Use your keyboard arrow keys, the [PageUp] and [PageDown] keys, or the mouse (click on the arrows). As you walk through the list, the highlighting bar moves on top of each current item. When this bar is positioned on top of an item and the [Enter] key is pressed, that item is chosen. A square root sign $\sqrt{}$ appears at the left side of the item. This indicates that you have successfully made that selection.

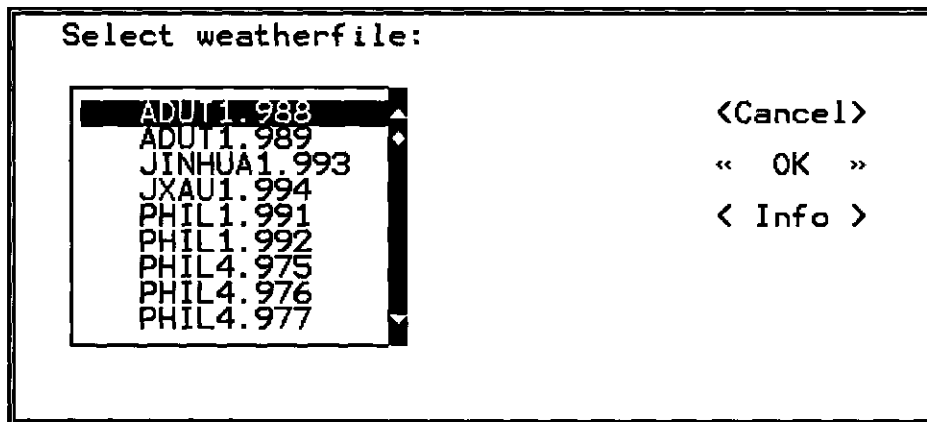


Figure 7.24 The Item-Selector, used in this example to select weather files.

If more than one choice is possible, one can move to the next item and press [Enter] again. The MANAGE-N tool displays a warning message when the user tries to choose more items than are allowed (a window appears in the upper right corner of the screen). In most cases, the Item-Selector only allows choosing a single item. When the selection has been made, the [Tab] key should be pressed until it highlights the [OK] button. Press this button to activate the selection. [Cancel] will not activate any selections made.

When information is available for a certain type of item, the [Info] button will be enabled (i.e. not dimmed). To get the information on a particular item, move the highlighting bar on top of it and then [Tab] to the [Info] button and press it (or use mouse). If the information can be retrieved, a window opens with the requested information. When more information is available than what the window can display at one time, the small arrows appear again. Scroll as described above.

The information window can be left by pressing the [Escape] key.

7.3 The plotting utility TTSELECT

The TTSELECT plotting utility (van Kraalingen, 1991b) is a program that displays graphs using data that the ORYZA_0 model has stored in a binary output file. This utility is

invoked by MANAGE-N to display custom-made graphs (like the yield-response to N level graphs) or when you press the 'View Graphs' menu option (during a *method* other than 'Optimise N-curve').

In the latter case, TTSELECT reads all the data in the binary output file and presents a list of the variables that are found (see Figure 7.25).

Then, if applicable, it also presents a list of the run numbers that have been made (for example, when you press the [Use RES.BIN] button in the 'View Optimisation Graphs' window, for each N level a run appears). Next, TTSELECT asks you to enter the names of the variables you want to plot. The first variable that you enter is the independent variable (X-axis), the next variable is the dependent (Y-axis). You can also enter more than one dependent variable but make sure the numerical ranges of the variables are not too different. Variables should be separated by a comma (,) or a blank.

If you like a custom title, enter it between double quotation marks. For example:

TIME, WCR, WCR_OBS, "Simulated & Observed crop weight"

(Only if applicable: after you press [Enter], TTSELECT will ask if you like to plot all runs or a selection. You enter the selection as a list of run numbers, separated by blanks.)

Then, TTSELECT asks you whether you wish to display only the end-of-run values. If you answer 'N' No, then you get time-coursed graphs. These are the default.

Now the graph will be shown. Press [Enter] to leave the graph. TTSELECT will ask you three questions about printing or storing graphs. The first question is whether you like the graph to be saved in a TTPlot (the native format of TTSELECT) file, the second question is whether you wish to save it in a screendump file, and the last question asks whether you like this plot to be printed on a HP DeskJet® or LaserJet® printer.

The first two questions are not of interest to you²⁵, the third may be if you have a DeskJet or LaserJet printer connected to your computer.

You can press three times [Enter] to skip all of the above three questions and return to the TTSELECT main menu for the next graph. Enter your next choice of variables to plot.

You can exit TTSELECT by pressing [Escape] until you are back in MANAGE-N.

²⁵ Except if you know how to process a screendump file (the explanation is beyond the scope of this booklet) or if you can use a native TTPlot file either for TTSELECT or for further processing in a spreadsheet.

TTSELECT version 2.21, October 1994 ...

The following variable names have been found:

```

=====
TIME      FERTMX      STPDAT      DOY      YEAR      DAT
DATEH     RDM        NL          WLV_OBS  FNLV_OBS  WCR
P         EPSIL      FSV1      FSV2     FSV        RNEFF
LNUC      GRUC       GCR       WSH      WCR_OBS   RDDSUM
RSR       RELTME     FITWCR    X        APSCOP    RECOV
NAVAIL    APCUM      SOLSUP    CUMNSS   SWIEXP    MAXUP0
MAXUP1    MAXUP2     FNMAX     MAXUP3   MAXUP4    MAXUP5
M1234S    DEMAND     NUPT      ANCR     FNACT     ANSOCH
ANLVCH    ANLV       WRR       HI       WCRFF     NAPPLD
ACTREC    NHRVST

```

Supply two or more variable names separated by a comma or a space.
First variable name will be used as X-value, type <ESC> to exit.

Which variable names :DAT, WCR, WCR_OBS "Simulated and observed crop weight"

Figure 7.25 An example of the menu system in the TTSELECT utility.

PART III

OTHER TOOLS

8 The FORTRAN Simulation Environment (FSE)

D.W.G. van Kraalingen¹

¹ DLO Research Institute for Agrobiological and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

8.1 Introduction

Most of the models in the SARP project are programmed in FORTRAN77, using a crop growth simulation driver called FSE (FORTRAN Simulation Environment; van Kraalingen, 1995). The simulation environment (Figure 8.1) consists of a driver routine that contains the control structure for rerun facilities, reading of weather data and the dynamic loop (integration, rate calculation, and time update), a framework for the major process-related routines, and a collection of utility routines that perform specific tasks such as reading of parameter values from data files and for generating model output.

This Chapter provides a concise introduction to FSE, its structure, and its features. As such, it is not meant to be a reference guide of FSE: van Kraalingen (1995) already covers that. The utility routines used are part of the FORTRAN77 library TTUTIL (Rappoldt & van Kraalingen, 1990). The reports by van Kraalingen (1995) and Rappoldt & van Kraalingen (1990) can be obtained from the DLO Research Institute for Agrobiological and Soil Fertility (AB-DLO), PO Box 14, 6700 AA Wageningen, The Netherlands.

8.2 Integration and time loop

The integration method used in the FSE program is the Euler (rectangular) integration method. The order in which calculations are executed and how reruns are implemented is shown in Figure 8.2. At the point where output is generated, values of state variables and rate variables refer to the same time. In the design of FSE, state and rate calculations are implemented in separate sections in the major subroutines. These major subroutines refer to processes that can be grouped together as a functional unit: for example, the distinction between plant and soil processes. The main program controls which section is activated through the concept of *task-controlled execution*. This is illustrated in Figure 8.3. The program lines of the plant and soil water subprocesses are separated into rate and state sections and only one of these sections is executed during a single call from the main program. Four tasks are distinguished: *initialisation* (ITASK=1), *integration* (ITASK=3), *rate calculation* (ITASK=2), and *terminal calculation* (ITASK=4).

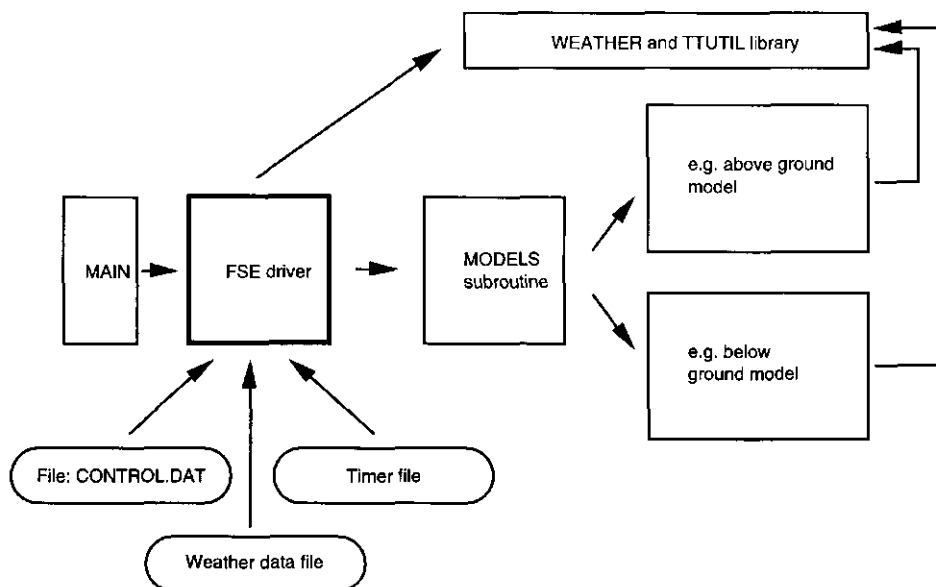


Figure 8.1 Overview of the generic structure of models that are programmed in FSE. Required data files for the FSE driver are indicated. The model(s) proper have their own localised access to required data. All models use the WEATHER and TTUTIL libraries.

After each time step, a decision is made if another time step is required or if the simulation should continue to the terminal section (Figure 8.2). One of the criteria to stop the simulation is that the predefined finish time (FINTIM) has been exceeded. In crop growth models, however, simulation has to be terminated when the crop is mature or if some other criterion has been met. It is thus necessary that the simulation loop can be terminated from within each of the submodels. This is implemented using a variable called TERMNL of the type LOGICAL, which indicates if the simulation loop should be terminated. The simulation loop continues as long as TERMNL=.FALSE. and the criterion is programmed as an emulated DO-WHILE loop (FORTRAN77 has no DO-WHILE control structure). The subroutine TIMER2 (in TTUTIL library) controls the time variables in the model. The basic actions of the subroutine TIMER2 are (i) (ITASK=1) check the values of FINTIM, DELT (time step of integration), TIME, etc. and copy these to variables local to the subroutine, switch on the output flag at the start of the simulation when TIME is a multiple of PRDEL (time interval for printed output) and when the simulation terminates, (ii) (ITASK=2) check whether the local time variables have the same value as the global time variables, add DELT to TIME, calculate the day number (DOY), flag if TIME is a multiple of PRDEL using the variable OUTPUT, flag if TIME has exceeded FINTIM using the variable TERMNL.

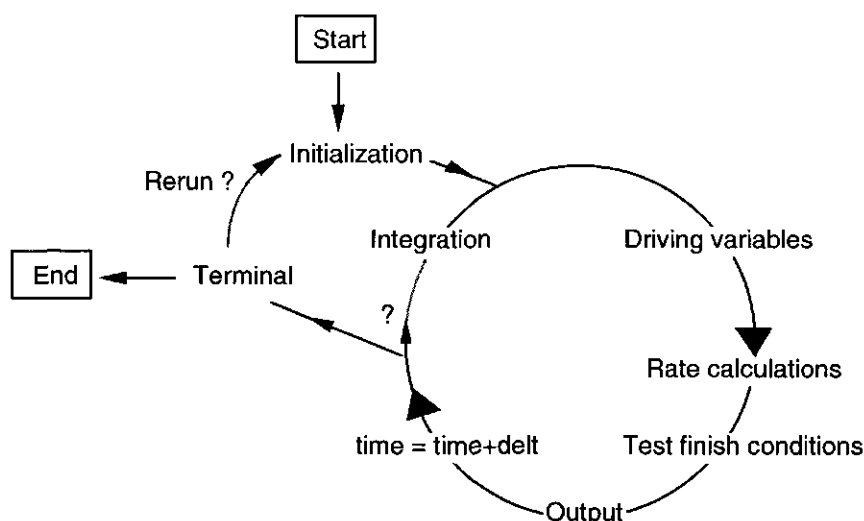


Figure 8.2 The FSE integration and time loop. Shown is the order in which calculations are executed when simulating continuous systems using the Euler integration, illustrating where to enter and leave the circle, and how reruns are implemented

The `TIMER2` routine which is used in the `FSE` program has some extra features: the year of simulation is automatically incremented and leap years are also recognised (`DOY` runs until 366), and the day number is available as an integer and as a real value (`IDOY` and `DOY`, respectively).

8.3 Initialisation of the states and parameters from external data files

All state variables in the model have to be initialised in the process subroutines. The rate variables do not have to be initialised because the model starts with the rate calculations after the initialisation. Integration is only performed if a rate calculation has previously been carried out.

8.4 The input data files: reading data

Most of the parameters and initial values of the state variables of the various subprocesses are read from data files. The data files `CONTROL.DAT` and `TIMER.DAT` (see listings in

Appendix 4) and the data files for the models (e.g., species and soil characteristics for the plant and soil model, respectively) have identical formats, and each variable in them may appear only once. The weather data files will be discussed separately. The values for the initial state variables and parameters are read from the data files using a set of TTUTIL subroutines whose names all begin with RD (e.g., RDSREA means 'read a single real value'). With these routines, the user can request the value by supplying the name of the requested variable (after having defined which data file to use). The statement:

```
CALL RDSREA ('WLVGI', WLVGI)
```

requests the subroutine RDSREA to extract the value of WLVGI from the data file and assigns it to the variable WLVGI. It does so by searching for the line: WLVGI = <value> in the data file. The data file is selected by the following statement:

```
CALL RDINIT (IUNITD, IUNITL, FILEI1)
```

which calls the routine that

- opens the file with variable name FILEI1 using unit=IUNITD+1 (FILEI1 is a character string that has been assigned to the name of the first type of input file (e.g., PLANT.DAT) in the calling program, then
- analyses the data file,
- creates a temporary file from the data file using unit=IUNITD,
- closes the data file (leaving IUNITD used for the temporary file), and
- sends all error messages that have been created to the screen and to a log file (with unit=IUNITL; this log file has been opened previously).

After this call, the plant subroutine can acquire the numerical values (including arrays) or character strings through the different RD routines, RDSREA (read single real), RDSINT (read single integer), RDSCHA (read single character string), RDAREA (read array of reals), RDAINT (read array of integers), and RDACHA (read array of character strings). The CLOSE statement deletes the temporary file that is created by the RD routines. A data file could be as follows:

```
* example data file
A = 10.                ! single real value
B = 0., 2., 3., 4.     ! array of four real values
C = 10., 20.,          ! array continued on next line
    30., 40.
D = 100*10.            ! array of 100 real values
E = 10.; F = 20.; G = 30. ! more than one parameter on single
                        ! line
H = 'PIET'             ! string value
```

```

I = 5*'KLAAS'           ! array of string values
VOLGN  OBS   CALC      REM    ! table
  1     10.4   10.1    'prachtig'
  2      7.8    8.     'te hoog'
  3      2.3    2.4    'ook mooi'

```

The following syntax rules apply to these input files:

- the file consists of variable names and one or more integer, real, double precision or string values, separated by an '=' sign. So: `PLMX = 20.`, is a valid specification, as is: `WTRDIR = 'NLD'`;
- the name of a variable cannot exceed 31 characters;
- for array variables, more than one value may follow the equal sign, separated by commas or spaces;
- identical numerical array values may be given as `n*<numerical value>`;
- variables may appear in the file in any order as long as this name is unique;
- comment lines start with '*' in the first column or '!' in any column (the rest of the line is ignored);
- continuation character is ',' on preceding line, applies to arrays only;
- names of variables and numerical values can be given on the same line if separated by a single semicolon ';';
- Only the first 80 characters of each line on the data file are read;
- Supported data types are REAL, INTEGER, and CHARACTER. Strings (data type CHARACTER) must be specified between quotes; INTEGER data types do not have a decimal dot.
- Arrays may be organised in tables;
- No tabs or other control and extended ASCII characters are allowed in the file.

8.5 The CONTROL.DAT file

The CONTROL.DAT file (Listing 8.1) contains the file names that are used during the execution of an FSE model. This file was not present in the FSE 1 version of FSE which made it impossible to do reruns on the names of input files. In FSE 2, reruns can now be done on the names of all *input* files specified in the CONTROL.DAT, except the name of the reruns file itself and the names of the output file. In this file, a distinction is made between names of input files (beginning with FILEI) and names of output files (beginning with FILEO).

The FILEON and FILEOL variables are assigned the names of 1) the normal output file (containing the output table) and 2) the log file, respectively. Both FILEON and FILEOL may be set to the same file name.

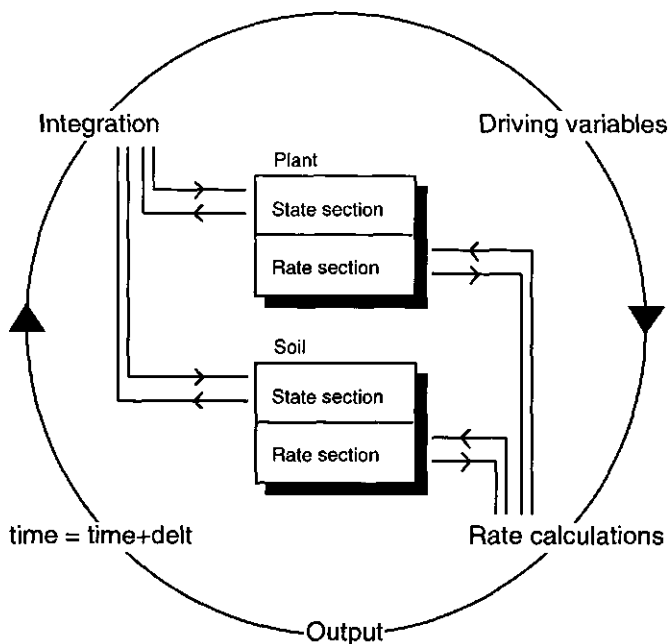


Figure 8.3 General FSE structure for incorporating several subprocesses into a single simulation model. The plant growth and soil water rectangles represent *one* subprocess description containing integration and rate calculations but called from different places in the main model with different task parameters.

Listing 8.1 Example of the contents of the CONTROL.DAT.

```
*****
FILEON='C:\USR\RESULTS.OUT'
FILEIR='RERUNS.DAT'
FILEIT='C:\USR\TIMER.DAT'
FILEI1='C:\USR\PLANT.DAT'
FILEI2='DUMMY.DAT'
FILEOL='RESULTS.LOG'
*****
```

The FILEIT and FILEIR variables are assigned the names of the timer file and the reruns file, respectively. The abovementioned file names are used by the FSE-driver. The file names that can be used by the model(s) are specified through the FILEI1, FILEI2, FILEI3, FILEI4, and FILEI5 variable names. These names are optional and can be specified only when they are needed.

8.6 The timer file

The timer file (e.g., `TIMER.DAT`) (Listing 8.2) specifies the value of the time variables such as time step of integration, time between different outputs to file, etc.; the directory in which the weather data are stored, the country code, station number and year; and some miscellaneous control variables. If many weather data files are used, it may be convenient to store these data in a separate directory. By assigning a directory name to the variable `WTRDIR`, the weather system is directed to read weather data from that directory, e.g.:

```
WTRDIR = 'C:\WEATHER\'
```

The country code is specified by the keyword `CNTR`:

```
CNTR = 'NLD'
```

For a list of available weather data files, their corresponding country codes and station numbers, see Stol (1994), e.g.:

```
NLD          <- country code for The Netherlands
PHIL         <- country code for the Philippines
```

These two character strings are read through the `RDSCHA` routine (in `TTUTIL` library) in the `MAIN` program.

The variables `ISTN` and `IYEAR` refer to the weather data and indicate the station number and year from a country. For example, when the country code is `NLD` (The Netherlands), `ISTN=1` and `IYEAR=1984`, the weather data from Wageningen 1984 will be used by the model. During execution, the weather system will try to open a file named `NLD1.984` on the given directory (`WTRDIR`).

The variables `STTIME`, `FINTIM`, `PRDEL`, and `DELT` represent the time parameters of the model. `STTIME` is the start day of the whole program; its value should be between 1 and 365. `FINTIM` is the finish time of the simulation, counted from the start of simulation. For example, when `STTIME=93.`, and `FINTIM=103.`, the simulation will continue until `TIME=103`. The variable `PRDEL` indicates the time between consecutive outputs to the output file (e.g., `RESULTS.OUT`). For example, when `PRDEL=5.`, output is given each time that `TIME` is a multiple of `PRDEL` (`TIME=5.,10.,15.`, etc.). If `PRDEL > 0`, then output is always given at the start of the simulation (`TIME=0`) and when the simulation is terminated (either `TIME=FINTIM` or some other finish condition). By giving `PRDEL` a high value (e.g., 1000), intermediate outputs are suppressed. Setting `PRDEL` to zero will suppress all output. The value of `DELT`, the time step of integration, is one day. This value cannot be changed because of the procedures used in the CO_2 assimilation subroutines that calculate daily rates using the Gaussian integration method.

Listing 8.2 Example of the contents of a timer file.

```
*****
* Defining the simulation run
*****
WTRDIR = 'C:\WEATHER\'
CNTR   = 'NLD'
ISTN   = 1      ! Station number of weather data
IYEAR  = 1985   ! Year of weather data
IFLAG  = 1101   ! Indicates where weather error and warnings
                  ! go (1101 means errors and warnings to log
                  ! file, errors to screen, see FSE manual)
*
* Time variables and output file options
STTIME = 121.   ! Start day of simulation
FINTIM = 148.   ! Finish time of simulation
PRDEL  = 5.     ! Time between consecutive outputs to file
DELT   = 1.     ! Time step of integration

IPFORM = 4      ! Format of output file
                  ! (0 = no output table, 4 = normal table,
                  ! 5 = Tab-delimited (for Excel), 6=TTPLT format)
DELTMP = 'N'    ! Switch variable what should be done with
                  ! the temporary output file:
                  ! 'N' = do not delete,
                  ! 'Y' = delete
COPINF = 'N'    ! Switch variable what should be done with
                  ! the input files:
                  ! 'N' = do not copy input files into
                  !         output file,
                  ! 'Y' = copy input files into output file

IOBSD  = 1985,182, 1985,194
                  ! List of observation data for which output
                  ! is required. 0 = no harvest data.
                  ! The list should consist of
                  ! pairs of <year>,<day> combinations.

PRSEL= 'DOY','DAT','WSHG','WCR','XXWCR','WSO','XXWSO','<TABLE>',
        'WLVG','XXWLVG','WSTS','XXWSTS','WRTL','XXWRTL','<TABLE>'
*****
```

The variable IPFORM defines if an output table is required (no output table: IPFORM=0) and if so, what the format should be. A multiple-column table (IPFORM=4) is sufficient for normal printing and viewing. Using IPFORM=5, a tab-delimited multiple column table, which is easily imported in spreadsheet programs such as EXCEL®, is generated. A two-column format is generated using IPFORM=6.

The variable DELTMP defines whether the file of temporary output data (RES.BIN) should be deleted at termination of the simulation (DELTMP = 'N', do not delete, DELTMP = 'Y', delete). This file is created during the dynamic phase of the simulation and is read during the terminal phase of the simulation to generate the output file RESULTS.OUT.

The variable COPINF defines whether the input files should be copied to the output file. When COPINF='Y', all input files are appended to the output file. When COPINF='N', nothing is done with the input files.

The variable IFLAG defines where errors and warnings from the weather system should be sent to (screen or file). For full documentation on the possible IFLAG values, see van Kraalingen et al. (1991c).

The array variable IOBSD can be used to force output at day numbers for which harvest data from the field are available. In many cases these harvest data will not coincide with output intervals in the model unless PRDEL is set to unity (which may cause large output files to be generated). A maximum of 50 year - day combinations can be defined here. A single value of zero indicates that no forced output is required, e.g.:

```
IOBSD = 0.                                <- No forced output
IOBSD = 1985,11, 1985,67 <- Output is forced on day 11 in 1985
                                and day 67.
```

The array variable PRSEL can be used to select a subset of the normal output variables without having to change the model. With PRSEL, several tables can be generated below each other. For example:

```
PRSEL = 'WSO', 'XXWSO', '<TABLE>', 'DVS', '<TABLE>'
```

generates a table with WSO and XXWSO after which a separate table with DVS is printed.

8.7 The model data files

With a crop growth model programmed in FSE, model data files contain the parameters, initial state values, and measured values of state variables. A crop growth model can have several types of data files, for example, files for crop characteristics, for soil parameters, for pest and disease data, and so on. The general syntax rules for data files as discussed above apply to these files as well.

8.8 Weather data

The weather data used in the model are read from external files. The weather data file definition, however, is different from those for the RD routines. The weather data system used has been developed jointly by the DLO Research Institute for Agrobiological Sciences and Soil Fertility (AB-DLO) and the Department of Theoretical Production Ecology of the Wageningen Agricultural University. It is especially designed for use in crop growth simulation models and has been documented in a separate report (van Kraalingen et al., 1991c; available on request).

The weather data system basically consists of two parts: the weather data files and a reading program to retrieve data from those files. A single data file can contain, at the most, the daily weather data from one meteorological station for one particular year. The

country name (abbreviated), station number, and year to which the data refer are reflected in the name of the data file (Listing 8.3).

Listing 8.3 Example of the contents of a WEATHER file (NLD1.985).

```

*-----*
* Station name: Wageningen (Haarweg), Netherlands
* Year: 1985
* Author: Peter Uithol -99.000: NIL VALUE
* Source: Natuur- en Weerkunde via Nel van Keulen
* Longitude: 5 40 E, latitude: 51 58 N, altitude: 7 m.
*
* Column Daily value
* 1 station number
* 2 year
* 3 day
* 4 irradiation (kJ m-2 d-1)
* 5 minimum temperature (degrees Celsius)
* 6 maximum temperature (degrees Celsius)
* 7 early morning vapour pressure (kPa)
* 8 mean wind speed (height: 2 m) (m s-1)
* 9 precipitation (mm d-1)
*-----*
5.67 51.97 7. 0.00 0.00
1 1985 1 660. 0.2 5.7 0.670 5.4 6.8
1 1985 2 2200. -2.9 0.7 0.490 2.2 0.1
1 1985 3 2280. -5.5 0.2 0.520 1.6 0.0
1 1985 4 3310. -18.4 -5.2 0.230 0.7 0.4
1 1985 5 4220. -19.5 -7.8 0.210 0.6 0.0
1 1985 6 1940. -13.8 -6.2 0.310 3.9 1.7
1 1985 7 4040. -21.4 -9.7 0.200 3.3 0.0
1 1985 8 2710. -21.3 -6.2 0.240 2.2 0.0
1 1985 9 930. -7.8 -3.8 0.380 4.1 0.1
1 1985 10 1540. -11.6 -3.6 0.390 2.1 0.6
<continued>

1 1985 362 940. -6.2 -3.0 0.430 1.1 0.0
1 1985 363 2000. -3.7 -1.2 0.460 1.7 0.0
1 1985 364 900. -5.3 -2.4 0.450 2.9 0.0
1 1985 365 3410. -6.2 -3.0 0.660 5.4 0.0

```

The reading program consists of a set of subroutines and functions, only two of which are intended to be called by the main program (STINFO and WEATHR). The others are internal to the reading program.

A call to the first subroutine (STINFO) defines the country (CNTR), station code (ISTN), year number (IYEAR), and the name of the directory containing the weather data (WTRDIR). This information is first read from the timer file, e.g. TIMER.DAT. A control parameter (IFLAG) is also supplied to indicate where possible messages of the system should be directed (screen and/or log file), and a name must be given to the log file if that name should differ from the default name WEATHER.LOG. The subroutine STINFO returns the location parameters (longitude LONG, latitude LAT, and altitude ELEV) of the selected meteorological station and, if the radiation is calculated from sunshine hours by the weather system, two coefficients of the Ångström formula (a and b) pertaining to the selected station. The value of a status variable (ISTAT) indicates a possible error or

warning (e.g., the data file requested does not exist). The location parameters can later be used to calculate day length (from the latitude) or average air pressure (from the altitude).

After this initialisation procedure, weather data for specific days can be obtained by calls to the second subroutine (WEATHR) with day number starting from January 1st as 1, as input parameter. The output of WEATHR consists of six weather variables for that day and the value of the status variable ISTAT indicating a possible error or warning (e.g. missing data, data obtained by interpolation, requested day is out of range, etc.). The six weather variables are daily incoming total global radiation (RDD), minimum and maximum air temperature (TMMN and TMMX), vapour pressure (VP), wind speed (WN), and rainfall (RAIN).

The subroutine STINFO can be called again at any time during program execution to change any of its input parameters. A call to STINFO with identical input parameters is also permitted (in fact this is done regularly in the FSE main program). Similarly, the subroutine WEATHR can be called repeatedly with any day number between 1 and 365 (or 366 in the case of a leap year).

8.9 Implementing reruns

The FSE program includes a rerun facility. The general idea behind the rerun facility is that the data files remain identical and that the changes in data are specified in a separate file -- e.g., RERUNS.DAT (Listing 8.4) which may contain the names and values of variables from any of the 'standard' data files that are read by the program. Thus, the file RERUNS.DAT may contain parameters from every type of the model data files (e.g. crop, soil, pest, but also timer file). In the first run, the values from the standard data files will be used. In subsequent runs, those values are then automatically replaced by the values from the rerun file. Execution will continue until all the rerun sets from RERUNS.DAT have been used. Before a rerun is started, the model checks if all the variables of the preceding set were used. If this is not the case, it is assumed that there is a typing error in the data files and the simulation is halted. The output of the different runs is merged in one output file. If the file RERUNS.DAT is absent or empty, the model will execute one single run, using the data from the standard data files. It is shown in Figure 8.2 that the control structure for the reruns is programmed as a loop around the actual model.

The format of the rerun files is identical to that of the other data files, except that the names of variables may appear in the file more than once. Arrays can also be redefined in a rerun file. The order and number of the variables should be the same in each set. A new set starts when the first variable is repeated. This is shown in the following example, where the variables STTIME and NPL from file TIMER.DAT are redefined:

```
STTIME = 90; NPL = 11., 5.           <- 1st rerun set
STTIME = 110; NPL = 11., 5.         <- 2nd rerun set
```

Important:

Each variable of which the value is changed somewhere in the rerun file should be assigned a value in each set, even if that value is identical to the value in the previous set.

Listing 8.4 Example of the contents of a RERUNS.DAT file.

```
*****
* RERUNS.DAT file
*
* When a value for an input variable is changed in a rerun,
* that value must be changed accordingly in each set, even
* if that value is identical in all sets.
* Lines with an asterisk (*) in the first column are comment lines
* Input variables can be separated by a semi-colon (;)
*
*****
SWINLV = 1 ! use measured amount of nitrogen in the leaves
FSV1 = 0.87 ! site variety factor before flowering

SWINLV = 0 ! use simulated amount of nitrogen in the leaves
FSV1 = 0.90 ! site variety factor before flowering
*****
```

8.10 Output of simulation results

Output is organised from each major subroutine separately. This avoids large argument lists to communicate output variables to the main program and limits the number of changes in the main program when, for instance, another plant routine with different output variables is used.

All subroutines write their output to the same output file (of which the name is defined in the FSE main program). By using a set of special routines (the OUT routines), output can be written as tables. It is also possible to add print plots of selected variables to that output file (for details, see Rappoldt & van Kraalingen, 1990). The use of the OUT routines considerably simplifies the generation of output files. The available routines are OUTDAT for output of single variables and OUTARR for arrays.

The OUTDAT routine also has a task parameter as input (the first argument in the call statement), similar to the subprocess descriptions. The first call (with ITASK=1) to OUTDAT (i.e., CALL OUTDAT(1,20,'X',0.)) specifies that X will be the independent variable and that unit 20 and 21 can be used for the output file. Subsequent calls with ITASK=2 (e.g., CALL OUTDAT(2,0,'X',X)) instruct OUTDAT to store the incoming names and numerical values in a temporary file (with the units from the ITASK=1 call). The number of combinations of name and value that can be stored depends solely on free disk space and not on RAM memory. The first call to OUTDAT below the DO-loop (with ITASK=4, CALL OUTDAT(4,0,'X',0.)) instructs the routine to create an output table using the information stored in the temporary file. Different output formats may be chosen, depending on the value of the task variable. Tab-delimited format (e.g., for the EXCEL[®] spreadsheet) can be generated by defining IPFORM in the TIMER.DAT file IPFORM=5, two-column format with ITASK=6. With any of these ITASK values, the string between quotes is written above the output.

The OUTARR call is actually an 'interface' call to OUTDAT. What the routine does internally is that it generates names (like A(1) and A(2)) and calls OUTDAT repeatedly for each of these name-value combinations. The range of array subscripts that should be generated by OUTARR is specified by the third and the fourth (last) subroutine arguments. This procedure can be repeated several times. The final call to OUTDAT (with 99) deletes the temporary file (i.e., CALL OUTDAT(99,0,' ',0.)).

8.11 Operation of the model

The model does not require interactive input during execution. The runs have been specified completely in the data files. During execution, the model will display run number, year number, and day number on the screen. During execution, errors and warnings may occur from the weather system and/or from the other modules of the model. They, generally, consist of one line of text. If simulation is terminated by an error during the dynamic section of the run, the outputs generated before the error in that particular run occurred, are written to the temporary file but are not yet written to the output file (RESULTS.OUT) until the terminal section of the model. Data can be recovered from the temporary file, using the OUTREC program (OUTput REcovery, see next paragraph on Error Recovery).

8.12 Errors and warnings from the FSE program

Several checks are performed by the model. All errors terminate the model execution and a message to that effect is displayed on the screen. In some cases, the error is also written to the output file (RESULTS.OUT). Warnings are displayed on the screen and are sometimes also written to the output file.

The weather system can also generate errors and warnings. Unlike errors from other sections of the model, the weather system itself never terminates the execution of the model. It is the FSE MAIN program that subsequently terminates the simulation run. Errors from the weather system are written to the screen and the log file WEATHER.LOG; warnings are written to the log file only.

If a run is terminated by some error from the model, the output file RESULTS.OUT will not contain the results of that specific run. But the results until the error occurred are written to the temporary file RES.BIN. This file can be converted into an output table by running the output recovery program OUTREC. This program requests an integer number from the user. A standard output table is generated by a '4' (the default); '5' generates a tab-delimited table (meant to be imported in EXCEL®) and '6' generates an output of only two columns at a time. The output table will be written to the file OUTREC.OUT so that any existing RESULTS.OUT file is not deleted.

9 The FORTRAN Simulation Translator (FST), a simulation language^{*}

D.W.G. van Kraalingen¹, C. Rappoldt¹ & H.H. van Laar²

¹ DLO Research Institute for Agrobiological Sciences (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

² Department of Theoretical Production Ecology, Wageningen Agricultural University, P.O. Box 430, 6700 AK Wageningen, The Netherlands

9.1 Introduction

The FST language and the corresponding FST software feature a powerful and easy-to-use simulation language providing clear error messages. The syntax of the FST language was based on the syntax of CSMP III (IBMTM, 1975). FST was developed because there was a need for a simple simulation language which, at the same time, allows the user to shift to the more powerful and flexible simulation environment that FORTRAN provides. This shift is made easy because the FST translator translates the FST source file into a clean and versatile FORTRAN program and also generates corresponding data files. When FST is used on IBM-compatible computers, an easy-to-use shell is available through which facilities are available for running the model, and graphical display of the model results among others.

Primarily, FST should be seen as a language for education and simple modelling purposes. The quality of the generated FORTRAN, however, provides an excellent starting point for users who need flexibility that FST does not provide. How exactly the generated FORTRAN routines work is outside the scope of this book but more information can be found in Rappoldt & van Kraalingen (1990), van Kraalingen (1991a), and van Kraalingen & Rappoldt (1989). FST will be fully documented by van Kraalingen & Rappoldt (in prep.). In this Chapter, a simple model for the simulation of logistic growth of yeast will be used to explain FST. Further, a more detailed description of the features of FST is given. The FST is available and requests can be sent to D.W.G. van Kraalingen.

9.2 The structure of the model

A simulation program in FST should start with a TITLE keyword containing a short identification of the program. In FST, three sections can be distinguished: INITIAL, DYNAMIC, and TERMINAL. These keywords indicate that the computations must be performed before, during, and after a simulation run, respectively. In using these sections,

^{*}) This Chapter is printed with permission from Kluwer Academic Publishers. Source: J. Goudriaan & H.H. van Laar, *Modelling Crop Growth Processes*, 219-230. © 1994 Kluwer Academic Publishers.

each section keyword closes the former section. The entire program is terminated by the keyword END (see Listing 9.1).

Calculations before the simulation starts are done in the INITIAL section, the simulation itself is executed in the DYNAMIC section, some single calculations after the simulation can be done in the TERMINAL section. If there is no section specified, FST considers the program to be in the DYNAMIC section.

In the INITIAL section, computations are executed only once per run. It is advisable to use the INITIAL section for the input data (initial conditions and parameters) and the time variables and to define data output and the integration method used in the model. Furthermore, the computation of results which are used as input for the dynamic section of the program may be executed here. The use of the INITIAL section is optional.

In the DYNAMIC section, the time course of the state variables is calculated. This section is, therefore, executed many times (roughly the simulation period divided by the time step of integration). It is normally the most extensive section in a model. The DYNAMIC section contains the complete description of the model dynamics, together with any other computation required during the simulation.

The TERMINAL section can be used for computations and specific output that can only be obtained at the end of the simulation run. This could be a computation based on the final values of one or more variables. As in the INITIAL section, the computations are executed only once. The TERMINAL section is also optional.

FST is provided with a sorting algorithm to free the user from the task of correctly sequencing the statements. Sorting of the statements and checking of the integrity are done for each section, separately.

9.3 FST example program simulating logistic growth

The steps to make in building an FST simulation model will be demonstrated by solving the following differential equation for logistic growth of yeast:

$$dY/dt = RGR \cdot Y \cdot (1 - Y/Y_{\max}) \quad (9.1)$$

In this example, Y is the amount of yeast (state variable) and dY/dt the rate of yeast growth (rate variable). In FST, this equation is programmed by the following two statements:

```
Y = INTGRL (IY,RY)
RY = RGR*Y*(1.-Y/YMAX)
```

The first line defines the name of the state variable (Y) and the name of the rate of change of Y (RY). The second line is the actual calculation of the rate of change of Y . Several additional values have to be defined to actually be able to do calculations with these two program lines, such as the value of the relative growth rate (RGR), the initial value of the quantity Y (IY), and the maximum value that the quantity of Y can attain ($YMAX$). The

quantities RGR and YMAX are called *parameters* because they should be constant during the model execution. The quantity IY is called an *initial condition* because it specifies the start value of a state variable. The value of model parameters is defined in a PARAMETER statement, initial conditions are defined in a INCON statement. Suppose we want to begin the simulation with a value of 1 for the state variable y , a relative growth rate of 0.2, and a maximum value of y (YMAX) of 20. The model will now be:

```
TITLE Logistic growth of yeast
INITIAL
  INCON IY = 1.
  PARAMETER RGR = 0.2, YMAX = 20.
DYNAMIC
  Y  = INTGRL (IY,RY)
  RY = RGR*Y*(1.-Y/YMAX)
END
```

The next step is to choose a solution scheme such as Euler (with a fixed time step of integration), or Runge-Kutta (a variable time step of integration) and to specify a start time, a finish time, and a time step of integration. Suppose we want to simulate from time equals 0 (Start TIME) to time equals 40 (Finish TIME) with a time step of integration of 0.5 (DELT) using the Runge-Kutta method (DRIVER='RKDRIV'). The model now becomes:

```
TITLE Logistic growth of yeast
INITIAL
  INCON IY = 1.
  PARAMETER RGR = 0.2, YMAX = 20.
  TIMER STTIME=0., FINTIM=40., DELT = 0.5
  TRANSLATION_GENERAL DRIVER='RKDRIV'
DYNAMIC
  Y  = INTGRL (IY,RY)
  RY = RGR*Y*(1.-Y/YMAX)
END
```

The model is now ready to run except that no output has been specified yet. If we would like to study the simulated values of Y and RY, we add a PRINT statement and we specify the necessary time interval between different outputs (PRDEL) to the TIMER statement (e.g., 5). All this is shown in Listing 9.1.

Listing 9.1 Simple FST program demonstrating simulation of logistic growth.

```
TITLE Logistic growth of yeast
INITIAL
  INCON      IY = 1.
  PARAMETER RGR = 0.2,  YMAX = 20.
  TIMER      STIME = 0., FINTIM = 40., DELT = 0.5, PRDEL = 5.
  PRINT Y, RY
  TRANSLATION_GENERAL DRIVER='RKDRIV'

DYNAMIC
  Y = INTGRL (IY,RY)
  RY = RGR*Y*(1.-Y/YMAX)
END
```

An important feature of FST is that the statements do not have to be in a computational order as would have been necessary with an ordinary programming language such as FORTRAN. An advantage of this is that it allows the user to define the model statements in a conceptual order. FST takes care of sorting the statements in a computational order while checking the integrity of the model. Execution of the model presented in Listing 9.1 gives the following output:

```
*-----
* Output table number : 0 (=first output table)
* Output table format : Table output
* Simulation results
* Logistic growth of yeast

      TIME          Y          RY
0.000000      1.0000      0.19000
5.000000      2.5032      0.43798
10.0000       5.6001      0.80641
15.0000      10.278      0.99923
20.0000      14.837      0.76605
25.0000      17.730      0.40244
30.0000      19.100      0.17182
35.0000      19.659      6.69605E-02
40.0000      19.873      2.51711E-02
```

Summarising the above process, in FST models the following types of statements can at least be recognised:

- 1 Statements that specify the mathematical model:
 - the state variables,
 - the initial conditions (=start values of the state variables),
 - the differential equations specifying the rates of change of state variables.
- 2 Statements specifying how to solve the model:
 - the integration algorithm,
 - the start and finish time of the simulation.
- 3 Statements specifying the output of the model:
 - the variables to be output,
 - the time interval between different outputs.

9.4 Comment lines and FST statements

Basically two types of program lines can be distinguished in FST, comment lines and FST statements. *Comment lines* are useful for the programmer to explain in words, e.g., what is calculated in a particular place in the program. Comment lines should always start with an asterisk (*) in the first column and they are usually written above the statements they belong to. *FST statements* can start in any column between 1 and 72. FST statements can be divided across several lines by putting three dots at the end of each line that is continued on the next line (the last dot may appear in column 72). An example is given in Listing 9.2.

Listing 9.2 An example showing the syntax of comment lines and statements.

* The differential equation for logistic growth	<- comment line
Y = INTGRL (IY,RY)	<- normal statement
RY = RGR*Y*(1.-Y/YMAX)	<- normal statement
PARAMETER RGR = 0.2,...	<- statement to be continued
YMAX = 20.	<- rest of continued statement

9.5 Rules for FST keywords, variable names, and values

In FST statements, several types of words can be distinguished such as FST keywords, variables, and values. Keywords are words that have a special meaning to FST such as PRINT, TIMER, PARAMETER, etc. and these keywords are reserved names. Variables are the words that the user defines to represent the problem, such as the state variables, the rates of change, model parameters, etc. There are some restrictions in FST on the spelling of keywords and variable names. First, all keywords and variable names should be in uppercase. The variable names are made up of letters and digits up to a total length of six characters and the first character must be a letter. They can be chosen freely except that the names of the keywords cannot be used as variable name. Also some reserved FST variables exist that the user cannot use for another purpose (e.g., TIME). In case of using reserved variable names, FST will give an error message. The format of the values that are specified within an FST program can be either a floating point format such as: 0.198 or 27.45 or a scientific format such as 2.342E-12.

9.6 Definition of input values of the model

As was shown in the example in Listing 9.1, values of model parameters are specified with a PARAMETER keyword, and initial values of the state variables are specified with INCON keyword. (Initial values of state variables can also be defined in a calculation but this is shown later). Two more keywords exist for specifying values relevant to the model. The CONSTANT keyword can be used for physical or mathematical constants (e.g., the ratio of

the circumference to the diameter of a circle, π). Often the need exists to have a parameter which varies in time or which depends on some other model variable. This can be achieved by the `FUNCTION` keyword. Behind the `FUNCTION` keyword, a table of (x, y) values can be assigned to a variable name. This name can be used for linear interpolation by the `AFGEN` function (Arbitrary Function GENERator). After a `FUNCTION` keyword, only one variable name can be defined whereas after the `PARAMETER`, `INCON`, and `CONSTANT` keywords, several variable names can be defined. Note that the `FUNCTION` keyword does not specify the variable name that is used to interpolate from the table nor to which variable the interpolated result is assigned; it simply assigns a (x,y) table to a name. (See Listing 9.3).

Listing 9.3 Illustration of the syntax of the `PARAMETER`, `INCON`, `CONSTANT`, and `FUNCTION` keywords.

```

PARAMETER A=5., B=1.E10, C=0.51E-3
INCON A=0., H=10.
CONSTANT PI=3.141592
FUNCTION TMTB=0.,2000., 1.,3000.,...          <- table of four (x,y) pairs
                2.,2000., 3.,1000.
Y = AFGEN (TMTB,TIME)                        <- Linear interpolation of table TMTB,
                                           e.g.: if TIME=1.5, Y will get the value 2500.

```

9.7 Operation hierarchy in expressions and FST and FORTRAN functions

In FST expressions, the same hierarchical rules apply as with FORTRAN expressions. For convenience they are repeated here. Expressions are evaluated in the order of hierarchy of the operators, and then from left to right if the operators have the same hierarchy, except for exponentiation, which is evaluated from right to left (the expression $A^{**}B^{**}C$ is evaluated as $A^{**}(B^{**}C)$). Parentheses can be used to overrule the regular hierarchy of the operators. The hierarchy of operators is given in Table 9.1.

In expressions next to the abovementioned arithmetic operators, functions can be called to perform specific tasks. As FST is translated to FORTRAN, many of the FORTRAN intrinsic functions can be used in FST. Some special FST functions also exist that can be used in expressions. The use of several of such functions is presented in Listing 9.4.

Note that each function has its own number of arguments which must all be present. The only exceptions to this rule are the functions for taking the minimum or maximum of a set of variables: `MIN` and `MAX`. An argument may be a value or a variable name. A list of FST and FORTRAN functions is given in Table 9.2.

Table 9.1 The order of hierarchy of arithmetic operators in FST.

Operator	Function	Order of hierarchy
()	Grouping of operations	First
**	Exponentiation	Second
*	Multiply	Third
/	Divide	Third
+	Add	Fourth
-	Subtract	Fourth
=	Assignment	Fifth

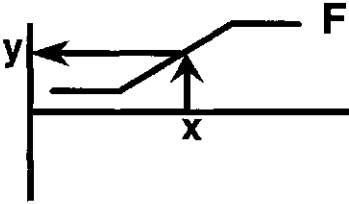
Listing 9.4 Use of FST and FORTRAN intrinsic functions.

```

* FST intrinsic functions
A = LIMIT(B, C, A)          <- A is limited between the values of B and C
C = INSW(D, C1, C2)         <- C is C1 if D < 0 else C2
* FORTRAN intrinsic function
A = MAX(10., B)             <- A is assigned 10 or B whichever is the largest
F = G*SIN(H)*COS(I)         <- F is assigned result of g*sin(h)*cos(i)

```

Table 9.2 FST and FORTRAN intrinsic function lists.

FST function	Explanation
Integrator: Y = INTGRL (IY , RY) Y = integrated value = Y_t IY = initial value = $Y_{t=0}$ RY = rate of change = dY/dt	$Y_t = Y_{t=0} + \int_{t=0}^t dY/dt \cdot dt$
Linear interpolation from (x,y) table: Y = AFGEN (F , X) Y = interpolation result F = (x,y) table specified with FUNCTION keyword X = start point of interpolation	
Input switch: Y = INSW (X , Y1 , Y2)	$y = y_1$ if $x < 0$ $y = y_2$ if $x \geq 0$
Limit between two values Y = LIMIT (YLOW , YHIGH , X)	$y = x$ if $y_{low} \leq x \leq y_{high}$ $y = y_{low}$ if $x < y_{low}$ $y = y_{high}$ if $x > y_{high}$
Y = NOTNUL (X)	$y = x$ if $x \neq 0$ $y = 1$ if $x = 0$
Y = REAAND (X1 , X2)	$y = 1$ if $x_1 > 0$ and $x_2 > 0$ $y = 0$ else
Y = REANOR (X1 , X2)	$y = 1$ if $x_1 \leq 0$ and $x_2 \leq 0$ $y = 0$ else

FORTRAN intrinsic function list:

FORTRAN function	Explanation	Mathematical notation	Restrictions
ABS (X)	absolute value of x	$ x $	
ACOS (X)	arccosine of x in range $[0, \pi]$	$\text{acos}(x)$	$-1 \leq x \leq 1$
ASIN (X)	arcsine of x in range $[-\pi/2, \pi/2]$	$\text{asin}(x)$	$-1 \leq x \leq 1$

ATAN (X)	arctangent of x in range $[-\pi/2, \pi/2]$	atan(x)	
COS (X)	cosine of x , x in radians	cos(x)	
COSH (X)	hyperbolic cosine of x	cosh(x)	
EXP (X)	exponential function	e^x	
LOG (X)	natural logarithm of x	$\log_e(x)$	$x > 0$
LOG10 (X)	base 10 logarithm of x	$\log_{10}(x)$	$x > 0$
MIN (X1, X2, . . . , Xn)	returns minimum value of arguments	min(x_1, x_2, \dots, x_n)	more than one argument
MAX (X1, X2, . . . , Xn)	returns maximum value of arguments	max(x_1, x_2, \dots, x_n)	more than one argument
MOD (X, Y)	remainder of x/y with same sign as x	$x \bmod y$	$y \neq 0$
SIN (X)	sine of x , x in radians	sin(x)	
SINH (X)	hyperbolic sine of x	sinh(x)	
SQRT (X)	square root of x	\sqrt{x}	$x \geq 0$
TAN (X)	tangent of x , x in radians	tan(x)	$x \bmod \pi/2 \neq \pi/4$
TANH (X)	hyperbolic tangent of x	tanh(x)	

9.8 FST keywords for output

In FST, several possibilities are present to get output of the model. The first one is the TITLE keyword. Any text that is written behind the TITLE keyword is written to the output file just above the output table (if any). Variables behind the PRINT keyword are written in a tabular format to the output file. The first eight variables are printed together in the first block; the second eight, in the second block; etc. The maximum number of variables per PRINT keyword is limited to 100. Different PRINT keywords generate different tables. This can be useful if you want to print the states in a separate table from the rates. Variables behind an OUTPUT keyword are print-plotted together in the same print-plot against TIME using the same Y-scale. The maximum number of variables per OUTPUT keyword is 25, also different OUTPUT keywords can be used to generate different print-plots. The time interval between different model outputs has to be specified behind the TIMER keyword by writing PRDEL='time_interval'. The TIMER keyword is discussed in more detail in the following section.

9.9 FST run control keywords

A model is not complete without information about the integration method (solution scheme), start time, finish time, time step of integration, and time interval for model output. The choice of the solution scheme is firstly determined by the type of translation

chosen. The FORTRAN Simulation Translator can generate two different FORTRAN routines. A model routine that can run under a general simulation driver when the keyword `TRANSLATION_GENERAL` is used, or a FSE compatible model routine, when the keyword `TRANSLATION_FSE` is used (FSE is a standard method of crop simulation in FORTRAN, developed at AB-DLO and TPE-WAU in Wageningen, see van Kraalingen (1991a), see also Chapter 8). When `TRANSLATION_FSE` is used, the solution scheme defaults to a fixed time step of the Euler integration. With `TRANSLATION_GENERAL`, the solution scheme has to be specified explicitly behind the keyword with `DRIVER='RKDRIV'` for a variable time step of the Runge-Kutta integration or `DRIVER='EUDRIV'` for a fixed time step of the Euler integration. The time information has to be specified behind the `TIMER` keyword with simple assignments to variable `STTIME` for start time, `FINTIM` for finish time, `DELT` for time step of integration, and `PRDEL` for time interval between outputs. See the example program for logistic growth. Note: when a Runge-Kutta solution scheme is chosen, the value of the time step (`DELT`) is used as a first guess at the start of the simulation. The actual time step is controlled by an error criterion.

When `TRANSLATION_GENERAL` is chosen, the variable `TRACE` can be given behind the `TIMER` keyword. The value of `TRACE` determines the amount of integration information that is logged to a log file. The default value of `TRACE` is zero, a value of four gives a full log of everything.

In some models, the termination of the run should not take place when `TIME` exceeds `FINTIM` but does occur when a state or rate variable exceeds a certain criterion. This can be achieved by using the `FINISH` keyword. Behind this keyword, two expressions are compared by an '<' or an '>' sign. When this expression is true, the simulation run is finished (e.g. `FINISH A>10.` or `FINISH A*B<C/E`). Several finish conditions can be used. Note that the `FINISH` keyword is usually not very effective to avoid division by zeros, etc. because the finish expressions are not evaluated before each expression of the model is evaluated.

9.10 Weather data in FST programs

Often weather data will be used in FST programs. One could define the weather data with large `FUNCTION` tables but this is not a very efficient solution. With the `WEATHER` keyword, weather data from the Wageningen-AB/TPE weather data library (see Listing 9.5) can become available to the FST model. Provided the data file(s) are present and contain no missing data, the variable names in Table 9.3 can be used (reserved variable names!) within the FST model (Note that the radiation is already multiplied by 1000 to get $\text{J m}^{-2} \text{d}^{-1}$!!):

Behind the `WEATHER` keyword, assignments should be written for the directory of the weather data files (`WTRDIR='directory'`), the country code (`CNTR='country code'`), the

station number within the country (ISTN=number), and the year to which the weather data pertain (IYEAR=year). The country code, the station number, and the year are combined into the file name of the weather data. The call for the weather data file NLD1.985 present on the directory C:\SYS\WEATHER\, with data from Wageningen (station no 1) 1985, The Netherlands is specified in FST as:

```
WEATHER WTRDIR='C:\SYS\WEATHER\' , CNTR='NL' , ISTN=1, IYEAR=1985
```

If the WEATHER keyword is used in the model and the weather system reports one or more of the weather variables as missing, the run is terminated only when these variables are actually used in the model. For a full description of the weather data file format, etc., see van Kraalingen et al. (1991c).

Table 9.3 Weather variables that can be used in FST.

Variable name	Meaning	Unit
RDD	Total daily global radiation	J m ⁻² d ⁻¹
TMMN	Minimum air temperature	°Celsius
TMMX	Maximum air temperature	°Celsius
VP	Vapour pressure at 9 a.m.	kPa
WN	Average wind speed	m s ⁻¹
RAIN	Total daily rainfall	mm d ⁻¹
DOY	Day of year (=time)	d
LAT	Latitude of the site	degrees

9.11 Rerun facility, the END keyword

Often the study of the effect of changing a parameter or initial condition is wanted. This requires the modification of one or more lines of the program and running the program again from the beginning. In FST, an elegant option exists to specify reruns in addition to the standard run as defined in the FST program. Any variable behind a PARAMETER, INCON, FUNCTION, TIMER, DRIVER, or WEATHER keyword can be given a new value on lines below an END keyword. A new rerun is then carried out automatically with only those specifications changed. An END keyword below allows the definition of yet another rerun.

Listing 9.5 Example of the contents of a weather data file (NLD1.985).

```

-----*
* Station name: Wageningen (Haarweg), Netherlands
* Year: 1985
* Author: Peter Uithol -99.000: NIL VALUE
* Source: Natuur- en Weerkunde via Nel van Keulen
* Longitude: 5 40 E, latitude: 51 58 N, altitude: 7 m.
*
* Column Daily value
* 1 station number
* 2 year
* 3 day
* 4 radiation (kJ m-2 d-1)
* 5 minimum temperature (degrees Celsius)
* 6 maximum temperature (degrees Celsius)
* 7 early morning vapour pressure (kPa)
* 8 mean wind speed (height: 2 m) (m s-1)
* 9 precipitation (mm d-1)
-----*
5.67 51.97 7. 0.00 0.00
1 1985 1 660. 0.2 5.7 0.670 5.4 6.8
1 1985 2 2200. -2.9 0.7 0.490 2.2 0.1
1 1985 3 2280. -5.5 0.2 0.520 1.6 0.0
1 1985 4 3310. -18.4 -5.2 0.230 0.7 0.4
1 1985 5 4220. -19.5 -7.8 0.210 0.6 0.0
1 1985 6 1940. -13.8 -6.2 0.310 3.9 1.7
1 1985 7 4040. -21.4 -9.7 0.200 3.3 0.0
1 1985 8 2710. -21.3 -6.2 0.240 2.2 0.0
1 1985 9 930. -7.8 -3.8 0.380 4.1 0.1
1 1985 10 1540. -11.6 -3.6 0.390 2.1 0.6
1 1985 11 2380. -5.3 -3.5 0.430 2.0 0.0
1 1985 12 2310. -7.9 1.7 0.470 3.3 1.6
1 1985 13 3800. -13.9 -5.5 0.270 2.6 0.0
1 1985 14 2880. -12.9 -3.9 0.310 2.7 0.1

.
1 1985 362 940. -6.2 -3.0 0.430 1.1 0.0
1 1985 363 2000. -3.7 -1.2 0.460 1.7 0.0
1 1985 364 900. -5.3 -2.4 0.450 2.9 0.0
1 1985 365 3410. -6.2 -3.0 0.660 5.4 0.0
-----*

```

9.12 FORTRAN subroutines with FST, the STOP keyword

Another powerful aspect of FST is that FORTRAN subroutines can be called from within the INITIAL, DYNAMIC, and TERMINAL sections using a CALL keyword. These routines can be part of the FST program itself but they can also be linked from an object library. FORTRAN subroutines can be written below a STOP keyword with the usual FORTRAN syntax (columns 7-72). The STOP keyword itself appears below the lines where reruns can occur. The STOP statement marks an important point in the FST program because from there, only a syntax check of the SUBROUTINE statement is carried out by the FORTRAN Simulation Translator. This is because any line below a STOP is assumed to be having a much more complex FORTRAN syntax. During the translation, any FORTRAN line is just appended to the generated model routine.

The rules for a CALL from a FST program to a subroutine are the same as for a call from a FORTRAN program, i.e., the number and sequence of the arguments must be identical both in the CALL and in the SUBROUTINE statement. In FST, as we have seen, statements from INITIAL, DYNAMIC, and TERMINAL sections are automatically sorted. A statement with a CALL, however, does not provide sufficient information for the sorting process because it is not clear which arguments are input and which are output. To resolve this lack of information of the sorting process, it is obligatory to define on top of the FST programs the input/output relations of the CALLs that are used in the program. This is done with the DEFINE_CALL keyword. Suppose we want to use a subroutine SUB1 with four arguments, the first three being input, the last being output:

```
SUBROUTINE SUB1 (X1,X2,X3,Y)
Y = X1*X2*(1.-X2/X3)
RETURN
END
```

The CALL from the FST program can be:

```
CALL SUB1 (RGR,Y,YMAX,RY)
```

The input/output relations of the subroutine are defined on top of the FST program by:

```
DEFINE_CALL SUB1 (INPUT, INPUT, INPUT, OUTPUT)
```

When a routine comes from a link library, only the FORTRAN subroutine itself should be omitted, the CALL and the DEFINE_CALL remain necessary.

A sample program of the discussed features of FST is given in Listing 9.6.

Listing 9.6 A sample program demonstrating the abovementioned features.

DEFINE_CALL SUB1(INPUT,INPUT,INPUT,OUTPUT)	<- Define input/output of subroutine
TITLE Logistic growth of yeast	<- Title for output file
INITIAL	<- Start initial section
IY = SQRT (4.)-1.	<- calculation of value for initial condition
PARAMETER RGR=0.2, YMAX=20.	
TIMER STTIME=0., FINTIM=40., DELT=0.5, PRDEL=5.	<- time information
PRINT Y, Y2	<- create an output table with Y and Y2
PRINT RY	<- create an output table with RY
OUTPUT Y	<- create a printplot of Y
DYNAMIC	<- Start dynamic section
Y = INTGRL (IY,RY)	
CALL SUB1 (RGR,Y,YMAX,RY)	<- call to subroutine
TRANSLATION_GENERAL DRIVER='RKDRIV'	<- generate general FORTRAN routine, use Runge Kutta
TERMINAL	<- Start section executed at end of each run
Y2 = YMAX/(1.+(YMAX-IY)*EXP(-RGR*TIME))	<- Calculation of analytical solution
END	<- Start of rerun section
PARAMETER YMAX=30., RGR=0.25	<- Definition rerun 1
END	
PARAMETER YMAX=40., RGR=0.30	<- Definition rerun 2
END	
STOP	<- Start section of FORTRAN subroutine(s)
SUBROUTINE SUB1 (X1,X2,X3,Y)	<- Start of a FORTRAN subroutine
IMPLICIT REAL (A-Z)	
Y = X1*X2*(1.-X2/X3)	
RETURN	
END	

10 RIGAUS

B.A.M. Bouman¹

¹ DLO Research Institute for Agrobiological and Soil Fertility (AB-DLO), P.O. Box 14, 6700 AA Wageningen, The Netherlands

The description of RIGAUS (Random Input Generator for the Analysis of Uncertainty in Simulation) in this Chapter is largely the same as that in Chapter 4 from Bouman et al. (1994b). 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. Values for different parameters/variables are drawn independently, i.e., without taking into account correlation between parameters/variables. RIGAUS has special provisions for drawing random input data for the soil water balance module SAHEL for rainfed upland environments, but these are not discussed here (see Bouman et al., 1994b). The use of RIGAUS in combination with crop growth models has been integrated in FSU (see Chapter 3), facilitating a user-friendly access to RIGAUS and its effects on model results. FSU handles all the communication with the RIGAUS tool; the user does not need to know the format of the input file and the names of special RIGAUS variables.

In this Chapter, however, a more technical description of RIGAUS is given. Thus, the information within this Chapter is especially of interest to those users who prefer running the RIGAUS tool stand-alone (outside FSU).

A separate manual (Bouman & Jansen, 1993) documents the RIGAUS tool in full detail, complete with the FORTRAN source code listings.

10.1 Statistical distributions

In RIGAUS, values can be generated randomly from uniform, beta or normal statistical distributions. In the current version, a maximum of 25 variables for a uniform distribution, 25 for a beta distribution and 25 for a normal distribution can be selected simultaneously (hence in total 75 variables/parameters). In principle, there is no limit to the number of draws that can be made for each variable. However, there is a limit to the number of reruns that can be made with a model in the FSE system. Therefore, the maximum number of draws for reruns under FSU is set to 999 (see also paragraph 10.3)

10.1.1 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 randomly generated values from a uniform distribution is given in Figure 10.1.

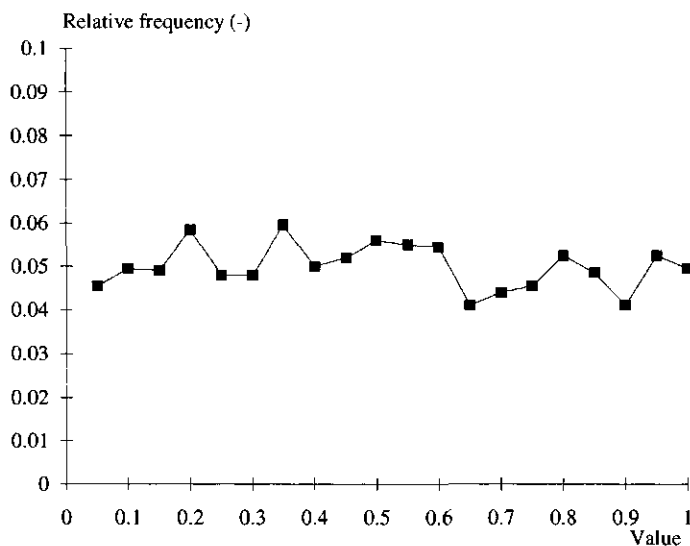


Figure 10.1 An example of uniformly distributed random value generation. Relative frequency distribution (fraction) of randomly drawn values from a uniform distribution between 0 and 1, using RIGAUS.

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)

10.1.2 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 Figure 10.2 are distributions of 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 10.1. 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 10.1 Beta distribution: the relation between shape parameters and mean and variance. Mean μ (upper number, bold) and variance σ^2 (lower number) of the beta distribution between 0 and 1 as a function of the shape parameters A and B.

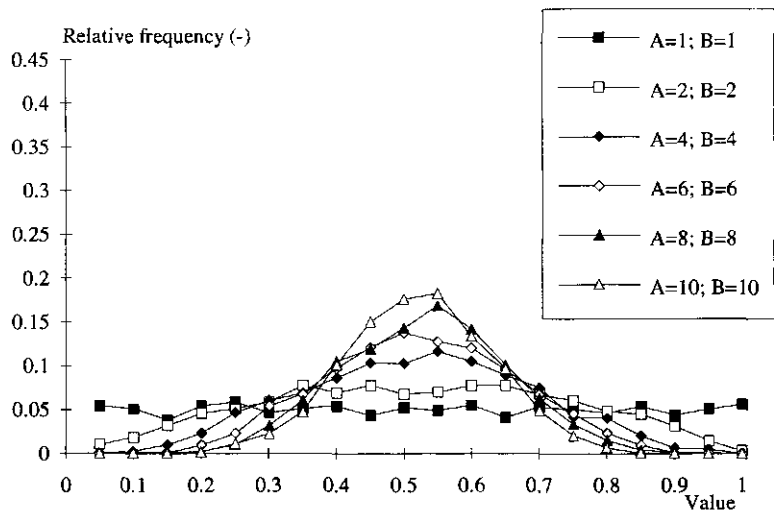
A \ B	1	2	4	6	8	10
1	0.500 0.083	0.333 0.056	0.200 0.027	0.143 0.015	0.111 0.010	0.091 0.007
2	0.667 0.056	0.500 0.050	0.333 0.032	0.250 0.021	0.200 0.015	0.167 0.011
4	0.800 0.027	0.667 0.032	0.500 0.028	0.400 0.022	0.333 0.017	0.286 0.014
6	0.857 0.015	0.750 0.021	0.600 0.022	0.500 0.019	0.429 0.016	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 that has to be supplied by the user for drawing from a beta distribution is (per parameter/variable):

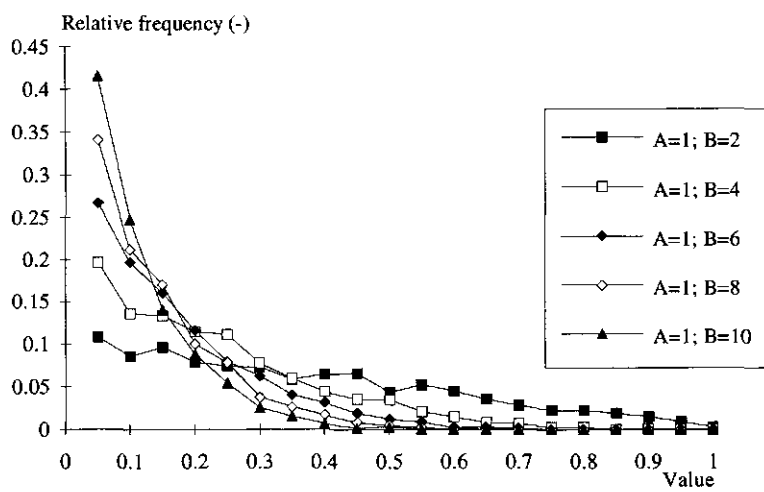
- Name of variable(s) for which random values have to be chosen (VBETA)
- Shape parameter A (ABETA) (in FSU, no capitalisation is used: a)
- Shape parameter B (BBETA) (in FSU, no capitalisation is used: b)
- Upper limit (BETAUP)
- Lower limit (BETALO)

10.1.3 Normal distribution

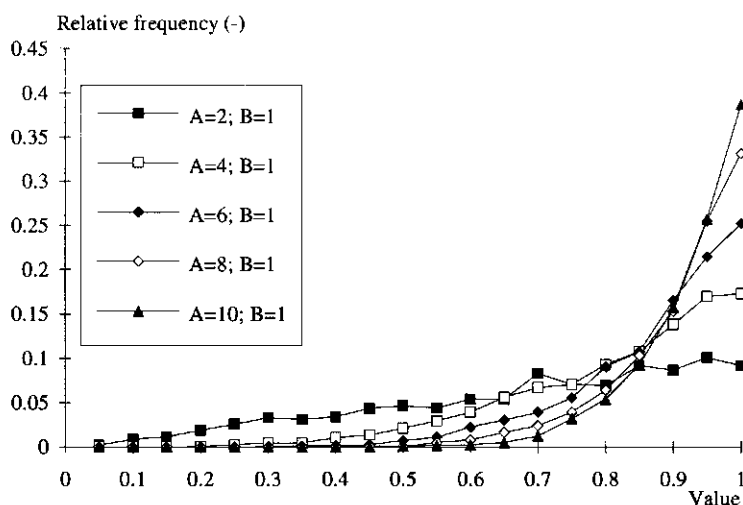
Random values for a normal distribution are generated using the function RGAU. This random generator is based on the Box-Muller method (Box & Muller, 1958). The normal distribution generated by RGAU has a mean of 0 and a variance of 1, but in RIGAUS, the mean and variance of the distribution can be set by the user. Examples of normal distributions with different means μ and variances σ^2 , as generated by RIGAUS, are given in Figure 10.3. Note that, on average, 95% of the values of a normal distribution lie between $\mu - 2\sigma^2$ and $\mu + 2\sigma^2$.



(10.2a)



(10.2b)



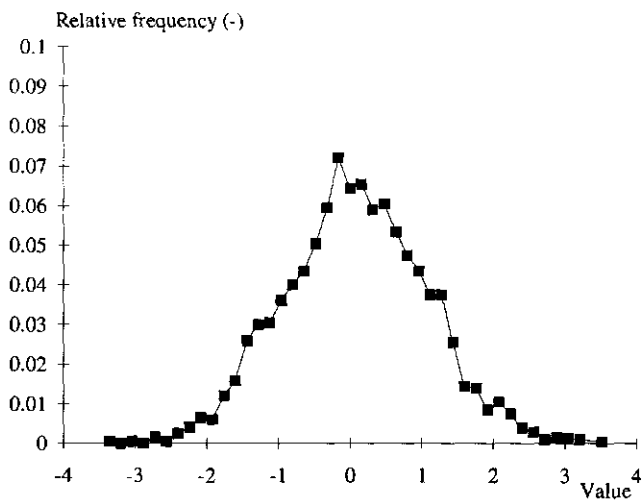
(10.2c)

Figure 10.2 Examples of randomly generated data with beta distributions. Relative frequency distribution (fraction) of randomly drawn values from a beta distribution between 0 and 1, using RIGAUSS. Different combinations of the A and B parameters are used in Figs. 10.2a, 10.2b, and 10.2c (see legends).

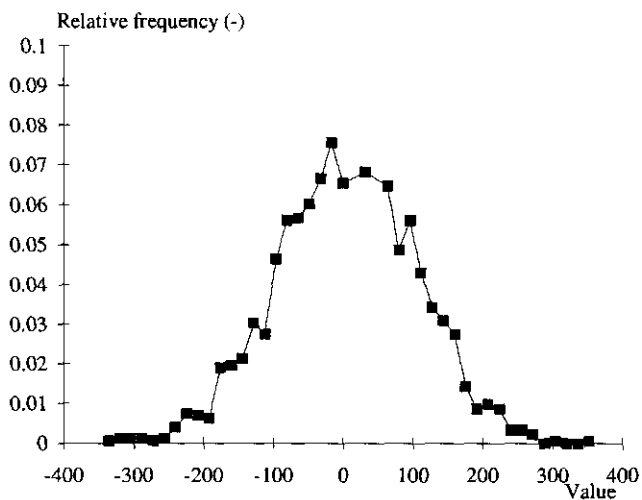
Warning: a normal distribution is not bound by preset minimum and maximum values. If values from a normal-type distribution have to be contained between fixed boundaries (as is often the case for model parameter values), a beta distribution with equal A and B values can be used (see paragraph 10.1.2).

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)



(10.3a)



(10.3b)

Figure 10.3 Examples of randomly generated values using the normal distribution. Relative frequency distribution (fraction) of randomly drawn values from a normal distribution, using RIGAUSS. In Figure 10.3a, the variance (VARU) of the distribution was 1, in Figure 10.3b, it was 100. The mean of the distribution (MEANU) was 0.

10.1.4 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 10.3.1). 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 10.3.2). 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. If 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.

10.2 Measured data

Random variables are uniformly drawn from a series of measured data using the RUNI function. 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 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

The total number of measured values for each parameter/variable may not be greater than 500 (see paragraph 10.3.1)

10.3 Running RIGAUS

Under FSU (previously known as the SARP-shell), the running of RIGAUS is facilitated with a menu system (see Chapter 3). If RIGAUS is run without FSU, this paragraph provides some useful information. RIGAUS (developed on an 486 IBMTM compatible PC) is written in the programming language FORTRAN77. A full listing of the source code can be found in Bouman & Jansen (1993). 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 a Microsoft® compiler subroutine, but a provision for the use of a VAX compiler subroutine is included in the source code.

The maximum number of draws per variable, of total drawn values (draws per variable times number of variables), of variables that can be selected from each statistical distribution type, and of measured data per variable are set in RIGAUS:

NDRAW = maximum number of draws (=999)
ILPREP = maximum number of total drawn values (=10000)
IMNP = maximum number of variables per statistical distribution type (=25)
KMNP = maximum number of measured data per variable (=500)

The maximum number of total drawn, ILPREP, is determined by the total number of parameter values that can be used in reruns of a crop growth model programmed using the TTUTIL library. For example, there may be 10000 draws for one single parameter, or 500 draws for 20 parameters each. Note that both the values of NDRAW and ILPREP are determined by the set of subroutines OPSYS that are to be linked with the model (see also paragraph 'OPSYS: end-of-season values system' on page 199)

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 15.

10.3.1 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:

General

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. If not applicable (see Bouman et al., 1994b), set it to 0.

ISWI = 0: special provisions are ignored (I)
ISWI = 1: special provisions are included (I)

The number of random draws should be set.

TND = ... (I)

The seed should be supplied.

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

UNIFORM distributions

NDU	= ...	Number of variables (maximum = 25) (I)
VUNI	= '...', '...', ...	List of variable names (max. = 25) (C)
UNILO	= ..., ..., ...,	Lower boundary of variable values, in the order of the variables specified above (max. = 25) (R)
UNIUP	= ..., ..., ...,	Upper boundary of variable values, in the order of the variables specified above (max. = 25) (R)

BETA distributions

NDB	= ...	Number of variables (maximum = 25) (I)
VBETA	= '...', '...', ...	List of variable names (max. = 25) (C)
ABETA	= ..., ..., ...,	A-value for beta distribution, in the order of the variables specified above (max. = 25) (R)
BBETA	= ..., ..., ...,	B-value for beta distribution, in the order of the variables specified above (max. = 25) (R)
BETALO	= ..., ..., ...,	Lower boundary of variable values, in the order of the variables specified above (max. = 25) (R)
BETAUP	= ..., ..., ...,	Upper boundary of variable values, in the order of the variables specified above (max. = 25) (R)

NORMAL distributions

NDN	= ...	Number of variables (maximum = 25) (I)
VNORM	= '...', '...', ...	List of variable names (max. = 25) (C)
MEANU	= ..., ..., ...	Mean of the normal distribution, in the order of the variables specified above (max. = 25) (R)
VARU	= ..., ..., ...	Variance of the normal distribution, in the order of the variables specified above (max. = 25) (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)

10.3.2 Program output

Two output files are generated by RIGAU: RERUNS.DAT and COLUMN.DAT. A third file, ERROR.LOG, is only created when a fatal error has occurred and contains messages on the nature of the error (see paragraph 10.3.3).

RERUNS.DAT

This file has the right format to serve as a reruns file in the FSE system. Appendix 15, paragraph 'RERUNS.DAT file created by RIGAUS' illustrates some sample output. 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'. 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 15, paragraph 'COLUMN.DAT file created by RIGAUS'. This file can be used in statistical programs such as GENSTAT or StatGraphics or in spreadsheets like EXCEL® for checking and evaluating the data, e.g. to check the generated distributions or the boundary values. If RIGAUS is operated under FSU (the 'Generate Monte Carlo rerun file' menu option), a plotting facility is available to check the generated distributions in COLUMN.DAT (the 'View Tables' and 'View Graphs' menu options). All output data (random values) are declared REAL and formatted in exponential notation E10.3.

10.3.3 Error and warning messages

A number of consistency checks on the input data are incorporated in RIGAUS. If inconsistencies are detected, either fatal error messages are given and the program is aborted, or warning messages are given. In the latter case, the program is still completed successfully. All error and warning messages are sent to the screen during program execution, whereas fatal error messages are also sent to a special output file, ERROR.LOG. If no fatal error messages occurred, ERROR.LOG will not be created (and previous ERROR.LOG files will be deleted).

What does RIGAUS check automatically?

Input data are checked on the maximum numbers allowed and on consistency. Some of RIGAUS' error checks are:

- The number of drawings TND exceeds 999 (NDRAW)
- The number of total drawn values (i.e., number of draws times number of variables selected), TND * (NDU+NDB+NDN+NMV) exceeds 10000 (ILPREP)
- The number of variables for uniform (NDU), beta (NDB), or normal (NDN) distributions exceeds 25

- 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 variables for the statistical distributions exceeds 25
- 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

Other checks that are performed by RIGAUS are found in Bouman & Jansen (1993). Error and warning messages can also be generated by the TTUTIL subroutines that are used in RIGAUS (Rappoldt & van Kraalingen, 1990). The program is aborted and an error message is given by the 'read' routines if:

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

Informative messages are also given if inconsistencies are detected but when RIGAUS can still complete successfully:

- 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).

What does the user have to check ?

- The format in which the input data are given should match the required format.
- The user has to check carefully the (input) boundary values for drawing from the uniform and beta statistical distributions. The same applies to the measured input data. If random draws are made from a normal distribution, there are in principle no limits to the range of possible values. Therefore, the results (randomly drawn values) have to be carefully checked for unrealistic values.
- It is advisable to check the generated distributions (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 plotting the generated values (option available under FSU).
- The standard format of the randomly generated parameter/variable values is REAL with exponential notation (E10.3). This format is compatible with almost all variables and input parameters used in the FSE-type of models. However, if this

format proves not compatible (i.e., INTEGER data are needed), either the output format in RIGAS 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).

References

- Berge, H.F.M. ten, M.C.S. Wopereis, J.J.M. Riethoven, T.M. Thiyagarajan & R. Sivasamy (1994a). The ORYZA_0 model applied to optimize nitrogen use in rice. In: Nitrogen economy of irrigated rice: field and simulation studies: proceedings of the 'International workshop on nitrogen management and modelling in irrigated rice', held at the Crop Experiment Station, Rural Development Administration, Korea, 1 - 10 November, 1993. H.F.M. ten Berge, M.C.S. Wopereis & J.C. Shin (Eds). SARP Research Proceedings. AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Baños: 235 - 253.
- Berge, H.F.M. ten, J.J.M. Riethoven & M.C.S. Wopereis (1994b). Numerical optimization of nitrogen fertilizer use in irrigated rice. SARP internal report. AB-DLO/Wageningen, IRRI/Los Baños: 59 pp + app.
- Berge, H.F.M. ten, M.C.S. Wopereis, J.J.M. Riethoven & H. Drenth (1994c). Description of the ORYZA_0 modules (version 2.0). In: ORYZA simulation modules for potential and nitrogen limited rice production. Drenth, H., H.F.M. ten Berge & J.J.M. Riethoven (Eds). SARP Research Proceedings. AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Baños: 43 - 55.
- Bouman, B.A.M. & M.J.W. Jansen (1993). RIGAUUS - Random Input Generator for the Analysis of Uncertainty in Simulation, CABO-TT Simulation report 34, AB-DLO/Wageningen: 26 pp.
- Bouman, B.A.M. & M.C.S. Wopereis (1994a). ORYZA_W (version 2.0), summary description. SARP internal report. AB-DLO/Wageningen: 45 pp.
- Bouman, B.A.M., M.C.S. Wopereis & J.J.M. Riethoven (Eds) (1994b). The use of crop growth models in agro-ecological zonation of rice. SARP Research Proceedings. AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Baños: 134 pp.
- Box, G.E.P. & M.E. Muller (1958). A note on the generation of random normal deviates. Ann. Math. Stat. 29: 610-611.
- Bratley, P., B.L. Fox & L.E. Schrage (1993). A guide to simulation. Springer-Verlag/New York: 397 pp.
- Drenth, H., H.F.M. ten Berge & J.J.M. Riethoven (Eds) (1995). ORYZA simulation modules for potential and nitrogen limited rice production. SARP Research Proceedings. AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Baños: 223 pp.

- L'Ecuyer, P. (1986). Efficient and portable combined pseudo-random number generators. Commun. ACM.
- Kraalingen, D.W.G. van & C. Rappoldt (1989). Subprograms in simulation models. CABO-TT Simulation Report no. 18. CABO-DLO/Wageningen, WAU-TPE/Wageningen: 54 pp.
- Kraalingen, D.W.G. van (1991a). The FSE system for crop simulation. CABO-TT Simulation Report no. 23. CABO-DLO/Wageningen, WAU-TPE/Wageningen: 110 pp.
- Kraalingen, D.W.G. van (1991b). The screen plotting utility TTSELECT. CABO-DLO internal report. CABO-DLO/Wageningen: 7 pp.
- Kraalingen, D.W.G. van, W. Stol, P.W.J. Uithol & M. Verbeek (1991c). User Manual of CABO/TPE Weather System. CABO/TPE internal report. CABO-DLO/Wageningen, WAU-TPE/Wageningen: 27 pp.
- Kraalingen, D.W.G. van (1995). The FSE system for crop simulation, version 2.1. QASA (Quantitative Approaches in System Analysis) report no. 1. AB-DLO/Wageningen, PE/Wageningen: 88 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.
- Price, W.L. (1979). A controlled random search procedure for global optimization. The Computer Journal 20:367 - 370.
- Rappoldt, C. & D.W.G. van Kraalingen (1990). Reference manual of the FORTRAN utility library TTUTIL with applications. CABO-TT Simulation report 20. CABO-DLO/Wageningen, WAU-TPE/Wageningen: 122 pp.
- 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. CABO-TT Simulation Report no. 24. AB-DLO/Wageningen, WAU-TPE/Wageningen: 24 pp. + app.
- Stol, W. (1994). Synoptic and climate data for agro-ecological research - the AB-MET database. CABO-TT Simulation report no. 37. AB-DLO/Wageningen, WAU-TPE/Wageningen: 103 pp.

Zheng Zhiming, H.F.M. ten Berge, Yan Lijiao, Wang Zhaoqian & Xu Zhaoben (1995). Optimization of nitrogen management for hybrid rice: a simulation approach. In: Nitrogen management studies in irrigated rice. T.M. Thiyagarajan, H.F.M. ten Berge & M.C.S. Wopereis (Eds). SARP Research Proceedings. AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Baños: 1 - 8.

APPENDICES

Appendix 1 Technical reference of FSU

The information contained within this Appendix is an extension to paragraph 1.5 'Quick installation' (page 9) and is meant for users who have more than average knowledge of computers. Furthermore, this Appendix provides information on how to add input files to FSU and explains the procedure of setting up FSU to enable printing of graphs to printers. Note: 'FSU', 'shell' and 'simulation environment' are interchangeable; they refer to the same entity.

Installing FSU (first time)

Before you proceed with the installation of FSU, make a backup of your original FSU disks. Please refer to the manuals accompanying your computer for instructions on how to do this. Ask a local computer expert if you are unsure. Also, if any of the following instructions appear unclear to you, contact your local computer support staff, or, if they do not know either, call the authors at the AB-DLO institute, Wageningen, The Netherlands. Store the original disks at a safe place and use your newly made backup disks to install the shell.

Follow these steps to perform a first-time installation on your system:

1. Be sure you have plenty of space on your hard disk (about 3 Mb); FSU extracts with example data and weather files.
2. Disk 1 (of 2) contains a file called `INSTALL1.EXE`. This file creates its own directory named FSU in which it places all necessary files and subdirectories. To install, move to the directory you want the new directory FSU to be created in. For example:

```
C:
CD \
A: INSTALL1.EXE
```

to install the shell in the directory C:\FSU.

```
D:
CD SARP
A: INSTALL1.EXE
```

to install the shell in the directory D:\SARP\FSU. Of course, it is assumed that the directory SARP exists.

3. After INSTALL1.EXE finishes, put disk 2 (of 2) into the disk-drive and enter the command

A: INSTALL2.EXE

After INSTALL2.EXE finishes, a directory named FSU has been created. This directory is the main directory for FSU.

4. **For most users, the above steps shall have successfully completed the installation of the shell. Go to step 11.**

There are, however, configurable options that you can set. These options are listed in the FSU.CFG file in the main FSU directory. If you wish to change any of the settings described below, start your editor and load FSU.CFG.

5. If you do not own a colour monitor or when the default colour display gives unpredictable results (for instance, on some portables), you might want to work in a two-colour environment. To facilitate this, change the line DISPLAY=COLOR into DISPLAY=MONOCHROME (or to DISPLAY=MONO).
6. Removing the entry USERNAME=FSU User will cause the shell to prompt for a username. If you like, you can fill in your own name: for example, USERNAME=Jean-Jack.
7. The more advanced users can specify the temporary path, meaning the place where FSU puts its working files that are only needed for a short time. Pointing the temporary path to a fast device like a RAM disk can speed up execution. About 200 Kb storage space must be available. It is generally not recommended to set the temporary path to a network drive.

Do not change anything if you are not sure. If you know exactly what temporary paths mean, then you can remove the exclamation mark before the TEMPDIR label and enter the path after the equal-sign. Remember to put a backslash '\ ' after the path as is shown in the sample configuration file on page 209. The shell will warn you at startup if the temporary path is not available.

8. It is possible to choose the text viewer of your preference. It must be able to load and view multiple files. One should be able to rotate through the files that were specified on the command line. If you are unsure, let it be, because installing a text viewer that is not capable of these things will hamper the correct working of FSU.

Summarising, the viewer must facilitate the next command line:

<viewer> <file1> <file2> <file10>

All specified <filex>s must be accessible right away within the viewer.

To change to a viewer of your preference, change the VIEWER= field in the FSU.CFG configuration file. At the right side of the = sign, enter the name of your viewer. Place the viewer in the main FSU directory. Do not specify a path in the VIEWER= field. For example:

VIEWER=VIEWME.EXE

and not:

`VIEWER=C:\SYS\UTIL\VIEWME.EXE`

9. Save the file (refer to your MS-DOS[®] or editor manual) you just edited.
10. You now have completed the additional configuration.
11. To start the FSU simulation environment, enter `FSU.BAT`.

Partial (renewed) installation of FSU

It might happen that you inadvertently delete a vital file of the standard FSU configuration (for example, the graphics utility or the RIGAU program). In such a case, you can do a partial installation of the shell:

1. Move to the FSU directory (it is where the `FSU.BAT` file is situated and all the needed `*.EXE` files).
2. Move to the parent directory:

`CD ..`

3. Insert the FSU installation disk 1 in drive A: and enter:

`A: INSTALL1`

4. During the installation process, the program will notice that several files are already present (i.e., already installed). It will ask you whether to overwrite this file. Press 'y' to do so, 'n' to skip the file (= not to install it again).
Files that were deleted (are not present any more in the FSU directory) will be installed automatically again.
5. Be careful that you do not overwrite data files (by pressing 'y') that you have changed and want to keep.
6. Do the same with the `INSTALL2.EXE` file (disk 2).
7. After these programs are completed, the deleted files are again retrieved.

Adding or modifying input files

Although it is actually discouraged to modify existing input files²⁶ like crop and soil files, adding new input files to FSU is no problem at all. When new input files (e.g., crop, soil

²⁶ This is due to the consistency of input files as related to model output. Suppose that a model has been run with soil file `CLAY.DAT` and the output has been stored. If one changes directly in the original `CLAY.DAT` file, one will not be able to relate this file anymore to the output previously stored, or even compare the previous output with new output created by using the modified `CLAY.DAT`. An exception to this 'rule' is during 'developing' and 'fine-tuning' a new input file.

and weather files) are available, they can be implemented within the simulation environment by simply copying them to their directories. For example, .copy weather files to the \DATA\INPUT\WEATHER directory, crop files to the \DATA\INPUT\CROP directory, and soil files to \DATA\INPUT\SOIL. That completes making these files available for use within FSU. Note that other types of input files may require different directories (see also Appendix 2, paragraph 'Adding new input types to FSU' on page 195).

If a model has been run stand-alone (outside the shell) and is brought under FSU for convenience, the timer file requires some special attention. The timer file should first be copied to the \DATA\INPUT\TIMER directory. Then, the timer file must be edited and all lines should be removed except comment lines and lines referring to parameters STTIME, FINTIM, MULTIP, ANGA and ANGB. All other parameters are already present in the RUNCTRL.DTA file, which is used to build an up-to-date timer file just prior to the model run. If ever the model displays error messages stating that a certain parameter occurred twice in the same data file, then probably the above procedure has not been carried out.

Printing graphs

It is possible to make a hard copy (plotting a graph to the printer) of the graphs that the shell displays. However, to do this some requirements have to be met. First, you should connect an Epson®-compatible printer to your PC. Second, you must execute the MS-DOS® program GRAPHICS.COM (or GRAPHICS.EXE on some computers) prior to running FSU. To do this, either place an extra line in your AUTOEXEC.BAT file or remove the 'rem' statement in front of the GRAPHICS.COM line in the FSU.BAT file. If you use MS-DOS® version 5 and up, it is possible to print on a small selection of other printers too. However, GRAPHICS.EXE requires some additional parameters then. These so-called command-line parameters can be found in the MS-DOS® manual, or, if your MS-DOS® version supports on-line help, it can be found by typing HELP GRAPHICS. Please contact a local computer expert if you encounter problems.

The above instructions only have to be executed once. From then on, you will be able to press [PrtScr] or [Shift+PrtScr] while the shell is displaying a graph to print it.

Appendix 2 Structuring your model for FSU

The process of adding new models to FSU is relatively easy. This Appendix explains in a nutshell what criteria should be met before placing a model under the shell and what actions should be taken to guarantee a correct co-operation between model and FSU. The assumption is being made that you have some knowledge of FORTRAN and are able to compile and link new programs. If not, then it is wise to consult a local expert on this matter.

The following restrictions apply with regard to the models:

- 1) Models need to be programmed using the FSE 2.0 (or higher) environment. See van Kraalingen (1995) and Chapter 8. If not, the shell will not be able to properly communicate with the model.
- 2) Do not make any changes to the FSE 2.0 code (the controlling part of the model).

The Microsoft FORTRAN Compiler/Linker version 4.1 or higher is required to successfully create the model executable.

After compiling the model, you need to link the thus created main object file(s) with the TTUTIL, WEATHER, and FSU libraries. TTUTIL version 3.3 or higher is required in all cases.. The FSU version 1.3 or higher library is needed in case you implement the OPSYS or OBSSYS system (see later). All these libraries are supplied with the FSU distribution in the MODEL.EXE directory. Some options in FSU require additional code to be implemented in the model. Paragraph 'OBSSYS: simulate/force system' on page 195 and paragraph 'OPSYS: end-of-season values system' on page 199 describe how to add them to the models. If you successfully generate a new model executable, copy it to the model executable directory of FSU (this directory is default called MODEL.EXE). Next, you need to create a so-called model definition file. Paragraph 'The model definition file' on page 200 explains the format of this file.

OBSSYS: simulate/force system

The simulate/force system OBSSYS enables a model to switch in a transparent way between simulating or forcing (using observed data) a state variable. Menu option 'Switch Simulated/Forced' requires this system to operate.

How is this system working? Well, first the data files are scanned for observed data for state variables. Data have to be formatted in a special way in the input files. Observed data must have the same name as the state variable, but with a postfix '_OBS'. Trigger information has an extension '_TRG' and the master switch for controlling whether to simulate or to force is appended with '_FRC'. Thus, for a state variable WLVG, the input file should include two arrays, named WLVG_OBS and WLVG_TRG, and a parameter WLVG_FRC. The arrays all have three columns, the first two being the year and the day of

year. The last column represents the observed value or simulation mode, respectively (see also paragraph 'Switch Simulated/Forced' on page 47). WLVG_FRC indicates simulate (=0), force (=2), or use triggering table (=1). In the first two cases, the triggering table is ignored.

The first thing to change in the source code of the model is to include a so-called common block in the main routine of the FSE driver. In the listing below, the line **COMMON /FSECM1/ YEAR,DOY,IUNITD,IUNITL,TERMNL** (printed in bold characters) has been inserted.

```
SUBROUTINE FSE

      IMPLICIT REAL (A-Z)

*-----Standard declarations for simulation and output control

      INTEGER  ITASK    , INSETS, ISET  , IPFORM, IL, ILEN
      LOGICAL  OUTPUT   , TERMNL, RDINQR, STRUNF, ENDRNF
      CHARACTER COPINF*1, DELTMP*1
      INTEGER  INPRS    , STRUN  , ENDRUN

      INTEGER  IMNPRS
      PARAMETER (IMNPRS=100)
      CHARACTER PRSEL(IMNPRS)*11

*-----Declarations for time control
      INTEGER  IDOY, IYEAR
      REAL     DELT, DOY, FINTIM, PRDEL, STIME, TIME, YEAR

*-----Declarations for weather system
      INTEGER  IFLAG    , ISTAT1, ISTAT2 , ISTN
      REAL     ANGA     , ANGB  , ELEV   , LAT  , LONG, RDD
      REAL     TMMN     , TMMX  , VP     , WN   , RAIN
      LOGICAL  WTRMES   , WTRTER
      CHARACTER WTRDIR*80, CNTR*7, WSTAT*6, DUMMY*1

*-----Declarations for file names and units
      INTEGER  IUNITR   , IUNITD , IUNITO  , IUNITL  , IUNITC
      CHARACTER FILEON*80, FILEOL*80
      CHARACTER FILEIC*80, FILEIR*80, FILEIT*80
      CHARACTER FILEI1*80, FILEI2*80, FILEI3*80, FILEI4*80, FILEI5*80

*-----Declarations for observation data facility
      INTEGER  INOD , IOD

      INTEGER  IMNOD
      PARAMETER (IMNOD=100)
      INTEGER  IOBSD(IMNOD)

*-----For communication with OBSSYS routine
      COMMON /FSECM1/ YEAR,DOY,IUNITD,IUNITL,TERMNL

*-----Unit numbers for control file (C), data files (D),
*   output file (O), log file (L) and rerun file (R). File name for
*   control file and empty strings for input files 1-5.
*   WTRMES flags any messages from the weather system

      DATA IUNITC /10/, IUNITD /20/, IUNITO /30/
      DATA IUNITL /40/, IUNITR /50/
      DATA FILEIC /'CONTROL.DAT'/
      DATA FILEI1 /' ', FILEI2 /' ', FILEI3 /' '
      DATA FILEI4 /' ', FILEI5 /' '
      DATA WTRMES /.FALSE./
```

To scan the data files, one has to insert two lines in the model source code. The first line (printed in bold characters) is in the FSE-part of the model and is just after the start of the reruns loop:

```

*=====
*
*                               Initialization section
*
*=====

      ITASK = 1
      TERMNL = .FALSE.
      WTRTER = .FALSE.

*-----Read names of timer file and input files 1-5 from control
*   file (these files can be used in reruns)
      CALL RDINIT (IUNITC,IUNITL,FILEIC)
      CALL RDSCHA ('FILEIT', FILEIT)
      IF (RDINQR ('FILEI1')) CALL RDSCHA ('FILEI1', FILEI1)
      IF (RDINQR ('FILEI2')) CALL RDSCHA ('FILEI2', FILEI2)
      IF (RDINQR ('FILEI3')) CALL RDSCHA ('FILEI3', FILEI3)
      IF (RDINQR ('FILEI4')) CALL RDSCHA ('FILEI4', FILEI4)
      IF (RDINQR ('FILEI5')) CALL RDSCHA ('FILEI5', FILEI5)
      CLOSE (IUNITC)

*-----Read time, control and weather variables from timer file
      CALL RDINIT (IUNITD , IUNITL, FILEIT)
      CALL RDSREA ('STIME', STIME)
      CALL RDSREA ('FINTIM', FINTIM)
      CALL RDSREA ('PRDEL', PRDEL )
      CALL RDSREA ('DELT', DELT )
      CALL RDSINT ('IYEAR', IYEAR )
      CALL RDSINT ('ISTN', ISTN )
      CALL RDSINT ('IPFORM', IPFORM)
      CALL RDSCHA ('COPINF', COPINF)
      CALL RDSCHA ('DELTMP', DELTMP )
      CALL RDSCHA ('WTRDIR', WTRDIR)
      CALL RDSCHA ('CNTR', CNTR)
      CALL RDSINT ('IFLAG', IFLAG)

*-----See if observation data variable exists, if so read it
      INOD = 0
      IF (RDINQR('IOBSD')) THEN
        CALL RDAINT ('IOBSD', IOBSD, IMNOD, INOD)
        IF (IOBSD(1).EQ.0) INOD = 0
      END IF

*-----See if variable with print selection exists, if so read it
      INPRS = 0
      IF (RDINQR('PRSEL')) CALL RDACHA ('PRSEL',PRSEL,IMNPRS,INPRS)

      CLOSE (IUNITD)

*-----Initialize TIMER and OUTDAT routines
      CALL TIMER2 (ITASK, STIME, DELT, PRDEL, FINTIM,
&                IYEAR, TIME , DOY , IDOY , TERMNL, OUTPUT)
      YEAR = REAL (IYEAR)
      CALL OUTDAT (ITASK, IUNITO, 'TIME', TIME)

*-----Open weather file and read station information and return
*   weather data for start day of simulation.
*   Check status of weather system, WTRMES flags if warnings or errors
*   have occurred during the whole simulation. WTRTER flags if the run
*   should be terminated because of missing weather

      CALL STINFO (IFLAG , WTRDIR, ' ', CNTR, ISTN, IYEAR,
&                ISTAT1, LONG , LAT, ELEV, ANGA, ANGB)

```

```

CALL WEATHR (IDoy , ISTAT2, RDD, TMMN, TMMX, VP, WN, RAIN)
IF (ISTAT1.NE.0.OR.ISTAT2.NE.0) WTRMES = .TRUE.
WSTAT = '444444'
IF (ABS (ISTAT2).GE.111111) THEN
  WRITE (WSTAT,'(I6)') ABS (ISTAT2)
ELSE IF (ISTAT2.EQ.0) THEN
  WSTAT = '111111'
END IF

*-----initialize OBSSYS routine
  IF (ITASK.EQ.1) CALL OBSINI

*-----Conversion of total daily radiation from kJ/m2/d to J/m2/d
  RDD = RDD*1000.

*-----Call routine that handles the different models
CALL MODELS (ITASK , IUNITD, IUNITO, IUNITL,
& FILEIT, FILEI1, FILEI2, FILEI3, FILEI4, FILEI5,
& OUTPUT, TERMNL,
& DOY , IDoy , YEAR , IYEAR ,
& TIME , STIME, FINFIM, DELT ,
& LAT , WSTAT , WTRTER,
& RDD , TMMN , TMMX , VP , WN, RAIN)

```

In the above code section, you entered the line containing the CALL OBSINI statement. The OBSINI procedure will initialise and set up the observation system. The next step is to change all the INTGRL statements into INTGR2 statements. Furthermore, two arguments have to be added: the name of the input file in which it should be located and the name of the state variable as a string. For example:

```
STATE = INTGR2 (STATE, RATE, DELT, FILEI2, 'STATE')
```

The INTGR2 statements take care of whether to simulate or force state variables.

Besides this, a by-product of this development is that you do not have to calculate observed data in two steps any more (reading AFGEN-tables and then using the function LINT to interpolate between data points). You can now get the (interpolated) observed value by using the GETOBS function. For example:

```
XXWCR = GETOBS (FILEI1, 'WCR')
```

in which FILEI1 is the name of the data file in which the observed values for the state variable WCR (WCR_OBS) are located..

To use the OBSSYS simulate/force system, you need to link your model with the FSU library and the 3.3 version (or higher) of the TTUTIL library. The first is included in the FSU distribution; TTUTIL must be obtained separately from D.W.G. van Kraalingen at the AB-DLO institute.

OPSYS: end-of-season values system

Models that require communication with the 'Regional Studies' menu must have implemented the OPSYS end-of-season values system. Basically, this system will output the value of chosen state variables at the end of a simulation run to a special output file named OP.DAT. When performing many reruns, like for example, via the 'Generate Monte Carlo Reruns File' menu option or via zonation studies, the values throughout the growing season are mostly not interesting. Only the last value for certain state variables are taken into account in most studies (e.g., final yield, water content of the soil). Some lines should be added to the source code of the model to make this system work (printed in bold characters):

```
*-----See if rerun file is present, and if so read the number
*      of rerun sets from rerun file

      CALL RDSETS (IUNITR, IUNITL, FILEIR, INSETS)

*      Initialise logfile for end-of-year state values
      CALL OPINIT

*=====*
*=====*
*                                     *
*      Main loop and reruns begin here      *
*                                     *
*=====*
*=====*

      DO 10 ISET=0,INSETS
      WRITE (*, '(A)') ' Initialize model...'
```

In the above source code, the line **CALL OPINIT** was inserted.

And then, in the termination section of your model (ITASK = 4), you add a line for each state variable that you want to output in this manner:

CALL OPSTOR('VARNAME', VARNAME)

The structure looks a lot like the regular OUTDAT routines of the FSE system. A maximum of 20 variables can be outputted in this way.

There is one modification (printed in bold characters) left that you should insert at the very end of the subroutine FSE:

```
*-----Close log file (if used)
      IF (FILEOL.NE.FILEON) CLOSE (IUNITL)

*-----Close log file of weather system
      CLOSE (91)

*-----Write end-of-year data to human-readable output table
      CALL OPWRIT (IUNITO)
```

```
RETURN
END
```

The line `CALL OPWRIT (IUNITO)` was inserted in this piece of code. To actually use this system, you have to compile and then link the model with the FSU library (besides `TTUTIL` and `WEATHER`).

The model definition file

In the same directory in which you placed your model executable, create a file called `MODEL.DEF` (use the name of the model executable instead of 'MODEL'; for example, `ORYZA_X.DEF` if you named the model `ORYZA_X.EXE`) and add the following line to this file:

```
[MODELINFO]
```

At the next lines, enter any information you like the user to see when he/she presses the [Info] button in the item-selector. Note that lines starting with an exclamation mark (!) are considered comments (i.e., will not be displayed). An example is listed in Appendix 4, on page 211.

Some models use so-called switches to activate or deactivate certain parts of the model. These switches (in FSU, they are called model options) are set by the user at the beginning of a model run. With these switches, the user can, for example, select for a potential production level or a water-limited production level. Switches are appended to the timer file by FSU, just prior to the model run. To communicate correctly with FSU, your model must thus read these switches from the timer file. If your model uses this kind of switch, then you must add an additional section to the model definition file:

```
[MODULES]
```

The above line signals the beginning of the part of the model definition file that describes the model options that are present within the model. The lines after `[MODULES]` have the following format:

```
<number>
<Description of Section 1>
<Description of Section n>

#<number><.1>
<Module description 1> , <switchname>=<value>
```

```
<Module description 2> , <switchname>=<value>
```

```
#<number><.2>
```

```
<Module description 1> , <switchname>=<value>
```

```
<Module description 2> , <switchname>=<value>
```

If you have a complex model with many parts that can be switched on or off, then you should make some functional division between the different modules and options. The correct way to do this is to define sections. Each section defines in a few words the options that can be set if one enters that section (see also paragraph 'Select Model Options' on page 55). The main section is numbered #1. For example, name a section 'Photosynthesis' if you have several switches that apply to the way in which photosynthesis-related processes are calculated. The actual switches are described in a separate paragraph, which is numbered according to the place of the section in the section just one higher in the hierarchy (.1 if it was listed first, .2 if it was listed second, etc.). For example, if the 'Photosynthesis' equals <Section 1>, then the paragraph which deals about photosynthesis is number #1 . 1. In that paragraph, one defines the switches and their values, for example:

```
#1.1
```

```
Use method 1 photosynthesis calculation , swipho=1
```

```
Use method 2 photosynthesis calculation , swipho=2
```

The above lines tell the shell which switch value corresponds to a description that the user selects in the 'Select Model Options' window. Note that instead of describing switches, one can also define new sections. See the example of the model definition file on page 211.

Each paragraph is sharply defined, starting with # and ending with a blank line. You may not mix sections and switch definitions in one paragraph. Paragraphs should also be listed in descending order of hierarchy, so one starts with #1, #1 . 1, #1 . 2, #1 . 1 . 1, #1 . 1 . 2, #1 . 2 . 1, and so on.

Another issue that needs to be defined is the data requirement. What types of input files does the model need? You must specify the required input files in a separate section in the model definition file:

[DATAFILES]

After this header, you list the input file types that the model needs:

```
<file type 1>, <file type 2>, <file type x>
```

Valid file types that are predefined by FSU are **TIMER**, **CROP**, **WEATHER**, **PEST**, and **SOIL**. A sample entry could look like this:

```
[DATAFILES]  
CROP, TREE, SOIL, WEATHER, TIMER
```

The above lines will create buttons for these file types in the 'Set Definition' windows of the shell. If your model uses file types that are not predefined by FSU (like the **TREE** file type in the above example), you must also enter them in this section. Note, however, that you must have a matching entry for that specific file type in your **FSU.CFG** file (see paragraph 'Adding new input types to FSU' on page 204).

In the event that you would like to use the model in conjunction with the Monte Carlo rerun file generation option (sensitivity/uncertainty/risk analysis), you must add a part that defines the parameters that can be used for each model option. Take a look at the example model definition file on page 211. The first thing to do is to include the line:

```
[RIGAUS]
```

After this label, the shell knows what to expect: definitions for the Monte Carlo rerun system. For each model switch you can list the parameters that are of importance to that section of the model in this format:

```
<switchname>=<value>  
<parameter name 1>  
<parameter name 2>  
<parameter name n>
```

For example, if switch **SWIWLP = 0** denotes an irrigated lowland production system, you can set up parameters for this system in the model definition file like this:

```
! These are the parameters for the irrigated lowland  
! production system  
SWIWLP=0  
! Soil parameters  
PERCS  
! Management parameters  
STTIME  
DTRP  
BUNDHI  
WL0M  
RIGIFT
```

RIPUD
DVSTE

(the above example is part of the FSU configuration of the ORYZA_WZ model). As you notice, you can put remarks on every line. Remarks start with an exclamation mark '!'. For each setting of a model switch, you can create a list like the one above; the list may consist of the same but often also of different parameters (to stick with the example: parameters that specify characteristics of puddled soil may only be listed at the model option that uses them).

However, sometimes the models are more complex and have more than one model switch and each switch can have a lot of different values. The shell supports these complex models in such a way that it is not necessary to supply parameters for each possible combination of switches and switch-values. The rule is simple: define the most specific parameter list first, the least specific list at the end. More specific means listing more switches and their values to define the environment for the following list of parameters. For example:

```
SWILAI=0
SWINPR=1
SWINPH=1
SWIPJJ=7
STTIME
DTRP

SWILAI=0
SWINPR=1
SWINPH=1
! insert another list here
TKPI

SWINPR=1
! list some general parameters here
NPH

SWINPR=0
! if the model options do not match any of the above
! switch combinations/values, then this one is
! compared. Since there are only two values of
! SWINPR, this list is taken if everything else fails.
NH
NPH
STTIME
```

When entering the 'Generate Monte Carlo Reruns File' window, FSU tries to match the actual model options (switches) with the more specific switches listed at the top of the model definitions file. If they do not match, the shell tries the next settings of the switches until a proper match has been found. The parameter names directly following the switches are listed in the 'Generate Monte Carlo Reruns File' window²⁷. If no proper match occurs, the shell displays a message that you have not properly set up the system to work with Monte Carlo reruns. Adding entries to the end of the list of a switch that can only have two values (like SWINPR in the above example) assures you that you always have a list of parameters in the 'Generate Monte Carlo Reruns File' window. Of course, use a very general list of parameters for the least specific list, only those parameters that are always used in the model whatever option is chosen. For example, STTIME (starting time of the simulation, which coincides with day of transplanting or day of seeding in most models) is a good bet.

It is the responsibility of the model developer to document the model in such a way that it can be used with the shell. As a user, you are however free to alter the definitions in the [RIGAUS] part as you see fit.

Adding new input types to FSU

Models that are incorporated within FSU use data files. These data files provide the required input for the models. In most cases, data are ordered into several data files, for example, all input related to a particular cultivar in a so-called crop data file, soil data in a soil data file, and so on. The groups into which these data files are ordered are called data types or file types. For data on crops, the data type is CROP, for soil data it is SOIL. Files for these input types are stored in different directories. The shell has 5 predefined file types: CROP, SOIL, PEST, WEATHER, and TIMER. Of course, it may happen that a model uses a file type that cannot be classified in one of the mentioned file types. In that case, a new file type must be defined.

²⁷ There is one other option that can be inserted in the [RIGAUS] section of the model definition file. However, this option refers to an internal quirk of the RIGAUS program and is only useful for a limited number of models. It enables, for puddled soils, to either generate random values for some parameters (referring to conditions in different layers of the soil) in an independent way or to generate these values by using an overall soil parameter and a predefined correlation matrix. As of now, only the ORYZA_W model uses these data, so it is probably of no interest to you. But, to be complete, this is what one should do to use the [Correlation] button in the 'Generate Monte Carlo Reruns File' window.

It boils down to adding a switch (ISWI) to the already defined lists. If ISWI equals 1, correlation is taken into account; if ISWI equals 0, all soil-layer parameters can be generated independently. Please refer to the sample model definition file on page 211. This switch is also described in Chapter 10 and in Bouman & Jansen (1993).

FSU's main configuration file FSU.CFG is the place to define these new file types. Include a line in the following format in the FSU.CFG file:

DATATYPE=<type>, <FSEname>, <path> [, <file mask>]

The <.> denotes required arguments, [...] denotes an optional argument. All identifiers and arguments are case-insensitive. 'File mask' can be any valid MS-DOS® file-mask (e.g. *.*, PU??M*.DA?, etc.). If no mask is specified, *.* (all files) is assumed. The <type> indicates how you want to name the new file type. A maximum of 8 characters is in effect. Make the type name as descriptive as possible - e.g., RICE, WHEAT, PUDSOIL, and so on. The default <FSEname> denotes how that particular file is opened within the FSE structure of your model. Most data files are internally called FILEIx (with x a number ranging from 1 to 6) in FSE. Most SARP ORYZA models have FILEI1 designated for the CROP data file. The shell needs to know the name of the FSE file handle so it can create all the control files in a correct way (see also Chapter 8 and van Kraalingen, 1995). The <path> points to the directory in which you place the new type of data files. The <file mask> is optional, you can use it to distinguish between file types if they all occupy the same directory.

For example, suppose that your model TREEGROW.EXE requires a file with data about trees. All your tree data files are stored in the directory D:\MYDATA\TREE. Your model uses the FILEI4 handle of FSE to open and use that particular data file. To define it for the shell, add the following line to the FSU.CFG file:

DATATYPE = TREE, FILEI4, d:\mydata\tree

If all your tree files have the extension *.DAT, you can specify that optionally like this:

DATATYPE = TREE, FILEI4, d:\mydata\tree, *.DAT

The above line makes the file type TREE accessible to FSU. It is now possible to refer to it in the model definition files. For your TREEGROW.EXE model, add or edit the following lines in the model definition file (see also paragraph 'The model definition file' on page 200):

**[DATAFILES]
TREE, WEATHER, TIMER**

This signals FSU to use input files of type TREE, WEATHER, and TIMER. The last two file types WEATHER and TIMER should always be present in the [DATAFILES] section; exclude one or both and the shell will warn you.

Optionally, you can also overrule the FSEname for a certain input file. Suppose that a file of type SOIL has been defined in your FSU.CFG file with a FSEname of FILEI2. One of your new models however requires that the SOIL file be read/handled by FILEI4. You can arrange this by editing the model definition file like this:

```
[DATAFILES]  
CROP, SOIL=FILEI4, WEATHER, TIMER
```

The additional '=FILEI4' structure is a local ('this model only') override of the default defined FSEname for type SOIL.

If you now start the shell, select your TREEGROW.EXE model and go right on to create a new set of input files. You will notice that FSU will ask for three file types: TREE, TIMER, and WEATHER in the 'Set Definition' window. The setup of these configurable types of data files is quite robust. The shell will warn you if you use file types that are not defined, or if you try to load sets of input files that do not belong to the current model (with regard to type of data files used).

Using predefined rerun files with FSU

Setting up your model to use predefined rerun files is easy. Make sure that the directory \DATA\INPUT\RERUNS exists under the main FSU directory: it should have been created automatically when you installed FSU. If it is not there, create it manually and consult local computer support staff if you do not know how.

Next, you can add files to this new RERUNS directory. Give the files a descriptive name, but use the extension of *.DAT. You must structure the content of the files in such a way that a valid reruns file is created (see Chapter 8).

In FSU, go to the 'Simulation Controls' menu option and check the 'Use pre-defined RERUNS.DAT file' button: a [X] should appear to indicate that the option is active. Next, go to the 'Create New Set(s) of Input Files' menu option and specify a new set. An entry for a reruns file is now also present. Store this set. When you subsequently run the model, FSU copies your predefined reruns file to the work area just prior to the model run. When the model starts, it uses your reruns file.

You must always specifically turn on the 'Use predefined RERUNS.DAT file' option before you select a set that contains a reruns entry. If this option is not active when selecting such a set, FSU warns you and will not use the reruns file.

Appendix 3 Problems with FSU

This Appendix lists common problems, their causes and gives possible solutions. These problems are specific to FSU. For generic problems that can arise with both the MANAGE-N decision support system and the FSU environment, refer to Appendix 5 on page 219. If a new problem (i.e., not listed below) occurs, contact the author of FSU.

After you activate menu options 'Retrieve Output!' and 'View Graphs', a red window pops up, displaying a.o. the text "Cannot find binary output file. Did you really run the model?" or "Cannot find file COLUMN.DAT"

<i>Cause:</i>	<i>Solution:</i>
FSU does not 'remember' which type of output you stored (single run, parameter modifications, multiple zoning run, or Monte Carlo runs). Therefore, it is possible for the 'View Graphs' option to get confused.	Create the same type of conditions that were in effect when you ran the model and stored the output. For example, choose any model, a single run set and 'View Graphs' again if output stored was a single run. Or, for example, select any model, a single run set and create any Monte Carlo rerun file and 'View Graphs' again if you stored output of a Monte Carlo multiple run.

When you run a model, a message appears "ERROR in RDDATA: variable name not in data file"

<i>Cause:</i>	<i>Solution:</i>
The model requires a variable that cannot be located in one of the data files.	Maybe you selected a wrong data file meant for another model. Check if the data files that you selected are indeed meant for the current model.

Appendix 4 Standard input files provided with FSU

This Appendix lists standard files that are used by the simulation models and are used or created by FSU. Most files are examples only and may differ considerably, depending on data entered by the user.

The FSU.CFG file

The file FSU.CFG specifies options that define the way FSU behaves in different computer environments. The example file listed below is the default as is distributed with the installation disks.

```
! Example configuration file of FSU version 2.0 (aka the SARP-SHELL)
! This file needs to be placed in the same directory as FSU.BAT and
! SCHIL.EXE
!
! You may place remarks after each option by inserting an exclamation
! mark. Everything after an exclamation mark will be ignored.
!
! Insert your own username after the equal sign. It can be up to
! 20 characters.
USERNAME=FSU User

! If you like monochrome screen-output instead of color, change the
! next line into DISPLAY=MONO or remove it altogether. Defaults to
! MONO
DISPLAY=COLOR

! If you want to use another temporary area other than the default,
! then define it here. Do not forget the backslash \
! (of course the directory must exist)
! TEMPDIR=D:\TMP\

! The default viewer (must be in the main FSU directory).
! If you like, you can specify another viewer here. But please, FIRST
! read the conditions that the viewer has to comply with. See Appendix
! "Technical Reference of FSU" in the FSU manual. Defaults to using LIST.COM.
! VIEWER=LIST.COM

! enter user-definable additional datafile types and directories here.
! Read the manual for additional information (Appendix 2).
! The lines displayed here are examples only, and will not work
! on your system.
! DATATYPE=JJTEST2, FILEI2, C:\USR\JEANJACK\
! DATATYPE=LOWSOIL, FILEI2, C:\USR\JEANJACK\FSU\DATA\INPUT\SOIL\, PU*.DAT
! DATATYPE=TREE, FILEI2, C:\workshop\FSU\DATA\INPUT\CROP
! DATATYPE=SOILAF, FILEI3, C:\workshop\FSU\DATA\INPUT\SOIL\RECAFS

! End of File.
```

The FSU.BAT file

This file is used to start FSU. If printing of plots is required, make sure that the GRAPHICS program is executed (remove the REM before GRAPHICS in this file).

```

@echo off
rem This batchfile controls the startup of shell and applications.
rem
rem
rem This batchfile executes the shell. If the shell is left, then
rem it checks on the occurrence of a temporary statusfile called
rem PROJECT.TMP. If this file does not exist, then a regular
rem 'Quit Shell' command is given. Else, a large application should
rem be started. The commandline of this application has been defined in
rem APPL_COM.BAT, which will be executed.
rem After execution, control is passed again to the shell.
rem ++++++

rem GRAPHICS.COM
rem GRAPHICS.EXE
rem One of the above lines will enable printing of graphs on an Epson-
rem compatible
rem printer (and others, especially if you use MSDOS 5.0 or up. Refer to
rem a local computer expert to set this up) if you remove the 'rem'
rem statement.

rem DO NOT CHANGE ANYTHING BELOW THIS LINE!
rem -----
:LOOP
    SCHIL.EXE
    IF NOT EXIST $status$.tmp GOTO THE_END
    CALL APPL_COM.BAT
    GOTO LOOP
:THE_END

```

The CONTROL.DAT file

The CONTROL.DAT file is used by the FSE (van Kraalingen, 1995 and Chapter 8) system in which the models are programmed and specifies which files are to be used during the simulation. CONTROL.DAT is automatically created by FSU just prior to a model run from selections that the user made.

```

* CONTROL.DAT created by FSU 2.0
FILEON='RESULTS.OUT'
FILEIR='RERUNS.DAT'
FILEOL='RESULTS.LOG'
FILEI1='C:\USR\JEANJACK\DEMO\FSU\DATA\INPUT\CROP\ORYZA_WZ.DAT'
FILEI2='C:\USR\JEANJACK\DEMO\FSU\DATA\INPUT\SOIL\LOAM.DAT'
FILEIT='T_ORYZAW.DAT'

```

The RUNCTRL.DTA file

The RUNCTRL.DTA file is part of the timer file that the models use. It contains various options that are not really of interest to the user. Most options are updated on-the-fly by the shell. Just prior to a model run, this file is joined with the timer file that the user chose and placed in the working directory of FSU.

```

*****
* RUNCONTROL FILE for FSU *
* THESE DATA SHOULD NOT BE CHANGED BY THE FSU USER *

```

```

* EXCEPT FOR MULTIP AND PRDEL
*****

* Weather data specification
WTRDIR = 'C:\USR\FSU\DATA\INPUT\WEATHER\'
CNTR   = 'PHIL'
ISTN   = 1      ! Station number of weather data
IFLAG  = 1100   ! Indicates where weather error and warnings go
                  ! (1100 means errors and warnings only to log file,
                  ! see FSE manual)

* Time variables
IYEAR  = 1991   ! Year of weather data
DELT   = 1.     ! Time step of integration

* Output options
COPINF = 'N'    ! Switch variable denoting what to be done with
                  ! input files:
                  !   'N' = do not copy input files into output file
                  !   'Y' = copy input files into output file

PRDEL  = 5.     ! Time between consecutive outputs to file,
                  ! (when PRDEL=0, no output is generated, when PRDEL is
                  ! very large (i.e. 10000) only initial and terminal
                  ! output is generated).
IPFORM = 4      ! (0 = no output table, 4 = normal table,
                  ! 5 = tab-delimited (for Excel), 6=TTTLOT format)
DELTMP = 'N'    ! Switch variable what should be done with the
                  ! temporary output file:
                  !   {0 = do not delete,
                  !   {1 = delete)

*****

```

The model definition file

A model definition file is used by FSU to acquire knowledge about the models. FSU cannot 'see' which kind of data files a model supports, what its history is, and what kind of parameters should be set up for use with the RIGAU program. The model builder must make a model definition file to properly 'introduce' the model to the shell. Placed in the same directory as the model, the model definition file also has the same name as the model, except for the extension. The extension of a model definition file is *.DEF. For example, if a model is called MYMODEL.EXE, then the model definition file should be named MYMODEL.DEF.

If the definition file is not available, then FSU will not be able to run the model. An example of such a definition file is found below. For a treatise on the options that make part of this file, the user is referred to Appendix 2, paragraph 'The model definition file' on page 200.

```

{MODELINFO}
*-----*
*
*                                ORYZA-W (2.1)
*      A Model for Irrigated and Rainfed Rice Production
*
* This version especially adapted from the FST-Version 1.3 of ORYZA1
* (April 1994) for use under FSU (aka the SARP-shell)
*

```

```

* Authors: SARP staff (this version: Bouman)
* Version: FSE-2.1
* Date: May 1994
*
*
* Reference: Bouman, B.A.M., M.S.C. Wopereis & J.J. Riethoven, 1994.
*           Crop growth modelling in agro-ecological zonation of rice,
*           SARP Research Proceedings, AB-DLO, Wageningen,
*           The Netherlands.
* Info:     B.A.M. Bouman, DLO Research Institute for Agrobiology and
*           Soil Fertility, PO Box 14, 6700 AA, Wageningen,
*           The Netherlands. e-mail: bouman@ab.agro.nl.
*-----

```

[MODULES]

```

#1
Irrigated Lowland           ,swiwlpl=0
Rainfed Lowland             ,swiwlpl=1
Rainfed Upland (non-puddled),swiwlpl=2

```

[DATAFILES]

TIMER, SOIL, CROP, WEATHER

[RIGAUS]

```

! RIGAUS DEFINITION FILE
! Parameters to be used for random drawing in RIGAUS
! Selection defined by switches for production environment
! (SWIWLPL) and for the correlation between soil moisture
! characteristics (ISWI)

```

```

! 1. Irrigated lowland
SWIWLPL=0
! 1.1. Soil parameters
SPSOIL
! 1.2 management parameters
STTIME
DTRP
NH
NPLH
NPLSB
NPLDS
WL0MXI
RIPUD
DVSIE

```

```

! 2. Rainfed lowland
SWIWLPL=1
! 2.1. Soil parameters
SPSOIL
SHRINK
WCSTP
WCCRAC
DDR
! 2.2 Management parameters
STTIME
DTRP
NH
NPLH
NPLSB
NPLDS
WL0MXI
TKLPI

```

```

! 3. Rainfed upland without soil correlations taken into account
SWIWLPL = 2
ISWI = 0
! 3.1 Soil parameters
TKL1
TKL2

```

```

TKL3
WCST1
WCST2
WCST3
WCFC1
WCFC2
WCFC3
WCWP1
WCWP2
WCWP3
FWCLI1
FWCLI2
FWCLI3
FRNOF
ZRTMS
EES
! 3.2 Management parameters
STIME
NPLDS

! 4. Rainfed upland with soil correlations taken into account
SWIWL = 2
ISWI = 1
! 4.1 Soil parameters
WCFC
FWCLI
FRNOF
ZRTMS
EES
! 4.2 Management parameters
STIME
NPLDS

```

Example crop file

The crop file listed here is part of the standard FSU distribution, and is used by the ORYZA_WZ 2.1 model.

```

*****
*                               ORYZA_Wz.DAT                               *
* Crop and management data for rice; models ORYZA1 (version 1.31)      *
* and ORYZA_W (version 2.1) [FORTRAN-FSE].                             *
* Experimental data: Parameters and Functions from: IRRI/APPA, 1992    *
* Oryza sativa cv.IR72, IRRI, Dry Season (M10) at 225 kg N            *
* plus data from Wopereis drought stress experiments at IRRI.         *
* This data file is for zonation (extrapolation) purposes.             *
*****

*****
* 1. Management parameters                                             *
*****
NPLH   = 5.      ! number of plants per hill
NH     = 25.     ! number of hills
NPLSB  = 1000.   ! number of plants in seed-bed
NPLDS  = 75.     ! number of plants/m2 direct-seeded
DTRP   = 12.     ! days in seed-bed (between sowing-transplanting)
*****
* 2. Crop data                                                         *
*****
* 2a. Crop data for both ORYZA1 and ORYZA_W (potential prod.         *
*****
* Initial data
LAPE   = 0.0001 ! initial leaf area per plant at emergence
DVSI   = 0.     ! initial development stage

```

```

WLVGI = 0.      ! initial leaf weight
WSTI  = 0.      ! initial stem weight
WRTI  = 0.      ! initial stem weight
WSOI  = 0.      ! initial weight storage organs
ZERO  = 0.      ! zero condition for integrals

* Model parameters
SHCKL = 0.25    ! parameter indicating relation between seedling-
                ! age and delay in leaf area development
SHCKD = 0.4      ! parameter indicating relation between seedling-
                ! age and delay in phenological development
RGRL  = 0.00800 ! relative growth rate of leaf area
FSTR  = 0.20     ! fraction carbohydrates allocated to stems that
                ! is stored as reserves
SCP   = 0.2      ! scattering coefficient of leaves for PAR
TBD   = 8.       ! base temperature for development (oC)
TBLV  = 8.       ! base temperature for juvenile leaf area growth
TMD   = 42.      ! maximum temperature for development
TOD   = 30.      ! optimum temperature for development
CO2REF = 340.    ! Reference level of atmospheric CO2
CO2    = 340.    ! Ambient CO2 concentration
TREF  = 25.      ! reference temperature
Q10   = 2.       ! factor accounting for increase of maintenance
                ! respiration with a 10 oC rise in temperature
DVRJ  = 0.000773 ! development rate during juvenile phase
DVRJ  = 0.000758 ! development rate during photoperiod-
                ! sensitive phase
DVRP  = 0.000784 ! development rate during panicle development
DVRP  = 0.001784 ! development rate in reproductive phase
MOPP  = 11.30    ! maximum optimum photoperiod
PPSE  = 0.0      ! photoperiod sensitivity
SFGF  = 64900.000 ! spikelet growth factor
WGRMX = 0.0000249 ! maximum individual grain weight

* table for GCM (General Circulation Model) temperature correction
TMCTB = 0.,0., 366.,0.

* table of extinction coefficient of leaves (KDF) as function of development
stage
KDFTB = 0.,0.4, 0.65,0.4, 1.,0.6, 2.1,0.6

* table of extinction coefficient of nitrogen profile in the canopy (KNF) as
* function of development stage (DVS)
KNFTB = 0.,0.4, 2.1,0.4

* tables of 'fraction of total dry matter allocated to shoot'
* (FSH) and of 'fraction of total dry matter allocated to roots'
* (FRT) as function of development stage
FSHTB = 0.0,0.50, 0.43,0.75, 1.0,1.0, 2.1,1.
FRTTB = 0.0,0.50, 0.43,0.25, 1.0,0.0, 2.1, 0.

* table of specific leaf area as function of development stage
SLATB = 0., 0.0047, 0.152, 0.0047, 0.336, 0.0033, 0.653,0.0028,
        0.787,0.0021, 1.011,0.0019, 1.431,0.0017, 2.10, 0.0017

* tables of partitioning factors for leaves (FLVTB), stems (FSTTB),
* and storage organs (FSOTB) as function of development stage.
FLVTB =
    0.000, 0.545,
    0.080, 0.545,
    0.245, 0.559,
    0.490, 0.542,
    0.720, 0.422,
    0.895, 0.053,
    1.230, 0.000,
    1.730, 0.,
    2.1, 0.
FSTTB =
    0.000, 0.455,
    0.080, 0.455,

```



```

0.245, 0.441,
0.490, 0.458,
0.720, 0.578,
0.895, 0.517,
1.230, 0.000,
1.730, 0.,
2.1, 0.
FSOTB =
0.000, 0.000,
0.720, 0.000,
0.895, 0.430,
1.230, 1.000,
1.730, 1.0,
2.1, 1.

* table of initial light use efficiency as function of temperature
EFFTB = 10.,0.54, 40.,0.36

* table of factor accounting for effect of temperature on AMAX
REDFTT = -10.,0., 10.,0., 20., 1., 37.,1., 43.,0.

* table of specific green stem area as function of development stage
SSGATB = 0.,0.0003, 0.9,0.0003, 2.1,0.

* table of nitrogen fraction in the leaves as function of
* development stage
NFLVTB =
0.00, 0.54,
0.16, 0.54,
0.33, 1.53,
0.65, 1.22,
0.79, 1.56,
1.00, 1.29,
1.46, 1.37,
2.04, 0.83,
2.10, 0.83

MAINLV = 0.02 ! maintenance respiration coefficient of leaves
MAINST = 0.015 ! maintenace respiration coefficient of stems
MAINSO = 0.003 ! maintenace respiration coefficient of storage organs
MAINRT = 0.01 ! maintenace respiration coefficient of roots

CRGLV = 1.326 ! carbohydrate requirement for leaf dry matter production
CRGST = 1.326 ! carbohydrate requirement for stem dry matter production
CRGSO = 1.462 ! carbohydrate requirement for storage organ " "
CRGRT = 1.326 ! carbohydrate requirement for root " "
CRGSTR = 1.11 ! carbohydrate requirement for stem reserves production

FCSTR = 0.444 ! mass fraction carbon in the stem reserves
FCLV = 0.419 ! mass fraction carbon in the leaves
FCST = 0.431 ! mass fraction carbon in the stems
FCRT = 0.431 ! mass fraction carbon in the roots
FCSO = 0.487 ! mass fraction carbon in the storage organs

LRSTR = 0.947 ! fraction of allocated stem reserves that is
! available for growth
TCLSTR = 10. ! time coefficient for loss of stem reserves

* table for leaf death coefficient as function of development stage
DRLVT = 0.,0., 0.6,0., 1.,0.015, 1.6,0.025, 2.1,0.05

*****
* 2b. Crop data for only ORYZA_W (water-limited production. *
*****
ZRTI = 0.0001 ! initial rooting depth
ZRTMC = 0.7 ! maximum rooting depth of crop
GZRT = 0.02 ! growth rate roots

* stress factors for drought before and at transplanting
ULLGTR = 1.1

```

```

LLLGTR = 1.1
ULLSTR = 0.51
LLLSTR = 0.26
ULDLTR = 0.31
LLDLTR = 0.14
ULRTTR = 0.63
LLRTTR = 0.0

* stress factors for drought after transplanting
ULLGTF = 0.94
LLLGTF = 0.94
ULLSTF = 0.77
LLLSTF = 0.34
ULDLTF = 0.43
LLDLTF = 0.14
ULRTTF = 0.77
LLRTTF = 0.14

*****
* 3. Switch parameters
* Note: DO NOT CHANGE SWINLV and SWILAI!!!
*****
SWINLV = -1. ! switch to use NFLV as function of DVS or daynumber
! -1: versus DVS; 1: versus DOY
SWITMP = -1. ! switch to use GCM temperature correction
! -1: do not use; 1: do use
SWILAI = 1. ! switch to use measured or simulated LAI (LAI leaves)
! -1: measured LAI; 1: simulated LAI
SWICOV = -1. ! switch to use plastic cover over seed-bed
! -1: do not use; 1: do use

*****
* 4. Measured data
*****
* Switches that indicate whether variables have been measured
* or not: 'Y' means measured and given; 'N' means not measured.
* Note: 'Y' and 'N' should be given in capital letters!

MLAI = 'N' ! measured leaf area index LAI
MNFLV = 'N' ! measured N-content leaves NFLV
MWTDW = 'N' ! measured weight total dry matter WTDW
MWST = 'N' ! measured weight stems WST
MWLVG = 'N' ! measured weight green leaves WLVG
MWLVD = 'N' ! measured weight dead leaves WLVD
MWPA = 'N' ! measured weight panicle WPA

* Measured data values: No data for zonation

*****
* Fabricated measured DVS data, used as an example only for
* FSU (aka the SARP-shell)
*****

DVS_OBS = 1992., 150., 0.,
          1992., 256., 1.,
          1992., 282., 2.

DVS_FRC = 0

DVS_TRG = 1900., 1., 0.,
          2000., 1., 0.

```

Example timer file

The file below is the part of the timer file the user can select or edit. This part is merged with the RUNCTRL.DTA file prior to the model run and placed in the work area of FSU.

The resulting merged file is used by ORYZA_WZ 2.1, the example model that is distributed together with FSU.

```
*****
*                                TORYZ_Wz.DAT                                *
* Timer file to be used with model ORYZA_Wz                                *
* Note: Angstrom parameters for humid tropical climate                      *
*****
ANGA = 0.29      ! Angstrom parameters: dry tropical A=0.25 B=0.45
ANGB = 0.42      !                                humid tropical A=0.29 B=0.42
                        !                                cold and temperate A=0.18 B=0.55

* Time variables
STTIME = 150.    ! Start time (day) of simulation (sowing)
FINTIM = 1000.   ! Finish time of simulation
*****
```

Example soil file

The crop file listed here is a part of the standard FSU distribution and is used by the ORYZA_WZ 2.1 model.

```
*****
* LOAM.DAT; Soil characteristics for a standard loam soil                *
* data from Penning de Vries et., 1989, page 152; soil no 13            *
* Soil data for SAHEL water balance                                       *
* DATA FOR MODELS ORYZA_Wz (UPLAND!) AND WHEAT_Wz                       *
*****

* Thicknesses of the soil compartments (m)
TKL1 = 0.2; TKL2 = 0.3; TKL3 = 0.5

* Water contents at field capacity (WCFC), wilting point (WCWP),
* air-dryness (WCAD) and saturation (WCST) for the three soil
* compartments (cm3/cm3):
WCFC1 = 0.355; WCWP1 = 0.108; WCAD1 = 0.007; WCST1 = 0.503
WCFC2 = 0.355; WCWP2 = 0.108; WCAD2 = 0.007; WCST2 = 0.503
WCFC3 = 0.355; WCWP3 = 0.108; WCAD3 = 0.007; WCST3 = 0.503

* Initial water content as fraction of WCFC, per layer:
FWCLI1 = 1.0; FWCLI2 = 1.0; FWCLI3 = 1.0

**SURFACE AND OTHER SOIL CHARACTERISTICS
* Fraction runoff:
FRNOF = 0.0

* Maximum rooting depth of soil (m)
ZRTMS = 0.9

* Evaporation extinction coefficient (1/m):
EES = 20.

SOILUP = 'Upland, non-puddled soil type'
```

Example weather file

Weather files in the FSU system should be formatted like the example weather file that is listed here. That includes the header with information! An exact description of the format

of the weather files used by the models can be found in van Kraalingen et al. (1991c) and Stol (1994).

```

-----*
* Station name: IRRI wet station site
* Year: 1979
* Author: Daniel van Kraalingen
* Source: Agroclimate Service Unit of IRRI
* Comments: Original name of data used in IRRI: ORWET
* Longitude: 121 15' E, latitude: 14 11' N, altitude: 21 m.
*
* Column Daily value
* 1 station number
* 2 year
* 3 day
* 4 irradiation (kJ m-2 d-1)
* 5 minimum temperature (degrees Celsius)
* 6 maximum temperature (degrees Celsius)
* 7 early morning vapour pressure (kPa)
* 8 mean wind speed (height: 2 m) (m s-1)
* 9 precipitation (mm d-1)
-----*
121.25 14.18 21. 0.00 0.00
1 1979 1 10656. 22.1 28.0 2.640 1.4 1.3
1 1979 2 7740. 21.5 27.0 2.430 1.4 0.0
1 1979 3 11700. 23.0 28.5 2.660 0.9 0.0
1 1979 4 14004. 21.5 28.5 2.530 0.8 0.0
1 1979 5 17316. 18.5 29.0 2.180 0.7 0.0
1 1979 6 16092. 18.7 29.0 2.400 0.6 0.0
1 1979 7 9396. 19.0 28.5 2.480 0.5 0.2
1 1979 8 13356. 21.0 29.5 2.870 0.3 0.0
1 1979 9 9180. 19.7 26.0 2.230 0.7 0.5
1 1979 10 16920. 21.0 29.0 2.760 0.6 0.0
1 1979 11 16920. 20.1 28.5 2.400 0.6 0.0
1 1979 12 16704. 19.5 29.3 2.480 0.5 0.0
1 1979 13 17748. 20.8 27.5 2.420 1.2 0.0
1 1979 14 12312. 18.0 27.0 2.040 0.5 1.8
1 1979 15 11700. 21.5 26.5 2.600 0.7 0.2
1 1979 16 10440. 21.0 26.7 2.400 1.1 0.0
1 1979 17 11268. 20.2 27.5 2.400 0.4 0.0
<continued>
1 1979 361 8352. 20.6 27.2 2.790 0.2 1.3
1 1979 362 12528. 21.6 28.9 2.740 0.5 0.8
1 1979 363 19224. 21.8 30.0 2.840 0.7 0.0
1 1979 364 13572. 20.1 29.8 2.910 0.2 0.0
1 1979 365 18180. 20.3 29.5 2.680 0.2 0.0

```

Appendix 5 Problems that may occur with MANAGE-N or FSU

This Appendix lists common problems, their causes, and their possible solutions. These problems can arise both with MANAGE-N and FSU. If the problem is not listed, try the specific Appendix dealing with problems in either MANAGE-N or FSU. If a new problem occurs (i.e., not listed here), please contact the author of MANAGE-N and FSU.

MANAGE-N or FSU refuses to start (or function properly) and displays the message 'Too many files open'	
<i>Cause:</i>	<i>Solution:</i>
There are too few available file handles. This is a MS-DOS® problem.	Edit the file CONFIG.SYS (located in the root directory of your hard disk) and enter the following line: FILES=40 If the line already existed, increase the number at the right side of the equal sign. Reboot your computer (press [Ctrl+Alt+Del] or turn the computer off and on). Now try again to start MANAGE-N or FSU.

At startup, MANAGE-N or FSU gives a warning that (some) data directories do not exist.	
<i>Cause:</i>	<i>Solution:</i>
Installation was not completed successfully	Refer to Appendix 6 (MANAGE-N) or Appendix 1 (FSU) and install anew or create the indicated directories manually.
You (or someone else) deleted the directories	Do not delete them in the future. Create the directories again by hand.
A backup or archiving program that does not handle empty directories removed them	Check if the backup software has an option that archives empty directories. Use that option. If not, do not use that particular program any more. It is not worth it. Create the directories manually.

At startup, MANAGE-N or FSU warns you that needed program files are missing.	
<i>Cause:</i>	<i>Solution:</i>
Incomplete installation	Refer to paragraph 'Partial (renewed) installation of MANAGE-N' on page 225 (MANAGE-N) or paragraph 'Partial (renewed) installation of FSU' on page 193 (FSU) and install the needed files anew.
You deleted them (unintentionally)	Make sure you do not delete them again. Refer to paragraph 'Partial (renewed) installation of MANAGE-N' on page 225 (MANAGE-N) or paragraph 'Partial (renewed) installation of FSU' on page 193 (FSU) and do a partial installation.

At startup, MANAGE-N or FSU warns you that the TEMPDIR directory is not present	
<i>Cause:</i>	<i>Solution:</i>
You configured an invalid directory in the MANAGE-N.CFG (MANAGE-N) or FSU.CFG (FSU) file after the TEMPDIR flag	1) Create the directory.
	2) Edit MANAGE-N.CFG (MANAGE-N) or FSU.CFG (FSU) and put the remark '!' back in front of the TEMPDIR flag.

A red window pops up, displaying a.o. the text "WINDOW NOT DEFINED"	
<i>Cause:</i>	<i>Solution:</i>
A bug in Microsoft® FoxPro®, the language in which MANAGE-N and FSU are programmed. Occurs when you try (too fast) to move from one window item to another with the mouse.	None. However, these red windows are not fatal, only moderately annoying. FSU and MANAGE-N recover correctly from these mishaps and you can continue without problems (press the button in the red window).

A red window pops up, displaying a.o. the text “OUT OF MEMORY”	
<i>Cause:</i>	<i>Solution:</i>
There is not enough memory available to successfully execute the requested operation.	1) Free up more memory for MANAGE-N or FSU (ask a local computer expert). When you start MANAGE-N or FSU, the ‘About’ window should display 80Kb or more free memory.
	2) If this error occurred when you tried to select input data files (weather files, crop files, etc.), then you have too many data files present at the same time. Try to move some of the data files to another directory for safe-keeping and try again.

Appendix 6 Technical reference of MANAGE-N

The information contained in this Chapter is mainly meant as a reference for users who like a more in-depth understanding of the workings of the MANAGE-N tool. The first section deals with installing and configuring MANAGE-N: it is an extension of paragraph 4.5 'Quick installation' (page 84) and is meant for users who have a more than average knowledge of computers.

Installing MANAGE-N (first time)

Before you proceed with the installation of MANAGE-N, make a backup of your original MANAGE-N disks. Please refer to the manuals accompanying your computer on how to do this. Ask a local computer wizard if you are unsure. Also, if any of the following instructions appear unclear to you, contact the local computer support staff or, if they do not know either, call the authors at the AB-DLO institute, Wageningen, The Netherlands. Store the original disks at a safe place and use your newly made backup disks to install MANAGE-N. Follow these steps to perform a first-time installation on your system:

1. Be sure you have plenty of space on your hard disk (about 3 Mb), MANAGE-N extracts with example data- and weather files!
2. Disk 1 (of 2) contains a file called `INSTALL1.EXE`. This file creates its own directory named MANAGE-N in which it places all necessary files and subdirectories. To install, move to the directory you want the new directory MANAGE-N to be created in. For example:

```
C:
CD \
A: INSTALL1.EXE
```

to install the MANAGE-N tool in the directory C : \MANAGE-N.

```
D:
CD SARP
A: INSTALL1.EXE
```

to install the MANAGE-N tool in the directory D : \SARP\MANAGE-N. Of course, it is assumed that the directory SARP exists.

3. After `INSTALL1.EXE` finishes, put disk 2 (of 2) into the disk-drive and enter the command

A: INSTALL2.EXE

After INSTALL2.EXE finishes, a directory named MANAGE-N has been created. This directory is the main directory for MANAGE-N tool.

4. **For most users the above steps have successfully completed the installation of the MANAGE-N tool. Go to step 10.**

There are, however, configurable options that you can set. These options are listed in the MANAGE-N.CFG file in the main directory. If you wish to change any of the below settings, start your editor and load MANAGE-N.CFG.

5. If you do not own a colour monitor, or when the default colour display gives unpredictable results (for instance on some portables), you might want to work in a two-colour environment. To facilitate this, change the line DISPLAY=COLOR into DISPLAY=MONOCHROME
6. The more advanced users can specify the temporary path, meaning the place where MANAGE-N puts its working files that are only needed for a short time. Pointing the temporary to a fast device like a RAM disk can speed up execution. About 200 Kb storage space must be available. It is generally not recommended to set the temporary path to a network drive.

Do not change anything if you are not sure. If you know exactly what temporary paths mean, then you can remove the exclamation mark before the TEMPDIR label and enter the path after the equal sign. Remember to put a backslash '\' after the path as is shown in the example configuration file on page 232. The shell will warn you at startup if the temporary path is not available.

7. It is possible to choose the text viewer of your preference. It must be able to load and view multiple files. One should be able to rotate through the files that were specified on the command line. If you are unsure, let it be, because installing a text viewer that is not capable of these things will hamper the correct working of the MANAGE-N tool.

Summarising, the viewer must facilitate the next command line:

```
<viewer> <file1> <file2> .... <file10>
```

All specified <file>s must be accessible right away within the viewer.

To change to a viewer of your preference, change the VIEWER= field in the MANAGE-N.CFG configuration file. At the right side of the = sign, enter the name of your viewer. Place the viewer in the main MANAGE-N directory. Do not specify a path in the VIEWER= field. For example:

```
VIEWER=VIEWME.EXE
```

and not:

```
VIEWER=C:\SYS\UTIL\VIEWME.EXE
```

8. Save the file (refer to your MS-DOS® or editor-manual) you just edited.
9. You now have completed the additional configuration.
10. To start the MANAGE-N simulation environment, enter MANAGE-N.BAT

Partial (renewed) installation of MANAGE-N

It might happen that you unintentionally delete a vital file of the standard MANAGE-N configuration (for example the graphics utility or the optimisation module). In such a case, you can do a partial installation of MANAGE-N:

1. Move to the MANAGE-N directory (that is the directory where the MANAGE-N.BAT file is situated, and all the needed *.EXE files).
2. Move to the parent directory:

CD ..

3. Insert the MANAGE-N installation disk 1 (of 2) in drive A: and enter:

A:INSTALL1

4. During the installation process, the program will notice that several files are already present (i.e., already installed). It will ask you whether to overwrite this file. Press 'y' to do so, 'n' to skip the file (= not to install it again).
Files that were deleted (are not present any more in the MANAGE-N directory) will be installed automatically again.
5. Be careful that you do not overwrite (when pressed 'y') data files that you have changed and want to keep.
6. Do the same with the INSTALL2.EXE file (disk 2 of 2).
7. After these programs finish, the deleted files are again retrieved.

Printing Graphs

It is possible to make a hard copy (plotting a graph to the printer) of the graphs that MANAGE-N displays. However, to do this some requirements have to be met. First of all, you should connect an Epson®-compatible printer to your computer, and second, you must execute the MS-DOS® program GRAPHICS.COM (or GRAPHICS.EXE on some computers) prior to running the MANAGE-N tool. To do this, either place an extra line in your AUTOEXEC.BAT file or remove the 'rem' statement in front of the GRAPHICS.COM line in the MANAGE-N.BAT file. If you use MS-DOS® version 5 and up, it is possible to print on a small selection of other printers too. However, GRAPHICS.EXE requires some additional parameters then. Please either contact a local computer expert or us if you encounter problems.

The above instructions only have to be done once. From then on, you will be able to press [PrtScr] or [Shift+PrtScr] while the shell is displaying a graph to print it.

Appendix 7 Problems with MANAGE-N

This Appendix lists common problems, their causes, and gives possible solutions. These problems are specific to the MANAGE-N tool. For problems that can arise with both the MANAGE-N tool and the FSU environment, refer to Appendix 5 on page 219. If a new problem occurs (i.e., not listed here), please contact the author of the MANAGE-N tool (see paragraph 4.6 on page 85).

After you ran the 'Fit to Match' menu option, MANAGE-N warns you that the result file is incomplete.	
<i>Cause:</i>	<i>Solution:</i>
1) There is not enough disk space.	Exit MANAGE-N and free up you hard disk. The 'Fit to Match' menu option requires at least 2 Mb of free disk space.
2) The model runs were abnormally aborted. Maybe you manually stopped the execution, or the computer was turned off.	Activate the 'Fit to Match' menu option again.

After the optimisation of a first N input level, MANAGE-N warns you that it cannot find the 'PAR_TERM.DAT' and 'CONFTERM.DAT' files.	
<i>Cause:</i>	<i>Solution:</i>
1) There is not enough disk space.	After each optimisation for a specific N input level, MANAGE-N copies the result files to a new directory. If your hard disk is low on space, this may not succeed. Exit MANAGE-N and free up you hard disk.
2) The optimisation runs were abnormally aborted. Maybe you manually stopped the execution, or the computer was turned off.	Activate the 'Optimise N-curve' menu option again.

You use the so-called validity check on biomass (WCR0, WCRSLP, DATCR) when performing an optimisation. Results of the optimisation are erroneous and the values of A, B, C and M in the 'Optimisation Results' window are the same for every N input level.

<i>Cause:</i>	<i>Solution:</i>
You set a too rigid check on biomass. Together with the number of initial runs that may be too small, this causes that all initial runs are rejected with regard to potential yield level. Rejected runs are defined as having a yield of -99. When all initial runs are rejected, the 'tolerance level' of the optimisation is met (no difference between runs), and no further runs are done (i.e., no iteration runs are performed for N input levels higher than 0 kg).	Loosen up the validity check or increase the number of initial runs.

Appendix 8 Sample input files used by MANAGE-N

This Appendix lists examples of files used by ORYZA_0 (and derivatives) and files used or created by MANAGE-N. Most files listed are created by MANAGE-N. All files are examples only, the numerical values contained within those files have no inherent validity.

Example of a crop/soil file

The file below is an example of a crop/soil file as created by MANAGE-N. Only the actual data (numerical values) and the header comments are provided by the user. Specific comments and division into sections are handled by MANAGE-N.

```
*****
* R. Torres, IRRI, The Philippines
* 1991
* variety:   IR72
* treatment: 110 KG/HA
*****

*****
* Observations or parameters derived from an observation in this
* individual treatment.
*****

* Treatment identifier
TREATM = 'IRRI, 1991, IR72, 110 kg/ha N'
DATH = 94.      ! Final harvest in days after transplanting [d]
DATFF = 60.     ! Days after transplanting on which first flowering occurs [d]
WCRI = 16.0     ! Initial weight of the crop [kg/ha]
ANCRI = 0.4     ! Initial amount of nitrogen in the crop [kg/ha]

* Observed weights of the total leaves [kg/ha]
WLW_OBS = 1991.,180.,8.5,
1991.,194.,8.5,
1991.,202.,16.8,
1991.,211.,133.8,
1991.,219.,342.5,
1991.,240.,1503.3,
1991.,254.,1935.3,
1991.,261.,2028.5,
1991.,273.,1564.8,
1991.,288.,892.8

* Observed weights of the crop [kg/ha]
* Only shoot has been measured as weight of the crop.
WCR_OBS = 1991.,180.,16.0,
1991.,194.,16.0,
1991.,202.,26.8,
1991.,211.,222.3,
1991.,219.,618.3,
1991.,240.,2961.5,
1991.,254.,5439.0,
1991.,261.,7503.0,
1991.,273.,9322.5,
1991.,288.,11534.5

* Observed fractions (in %) of N in the leaves [-]
```

```

FNLV_OBS = 1991.,180.,3.02,
1991.,194.,3.02,
1991.,202.,4.35,
1991.,211.,4.02,
1991.,219.,3.91,
1991.,240.,2.69,
1991.,254.,2.09,
1991.,261.,2.09,
1991.,273.,1.79,
1991.,288.,1.26

* Switch: 0 = roots not included in WCR_OBS, 1 = roots included [-]
RTINCL = 0
DATFSV = 60.
FSV1 = 0.88
FSV2 = 0.88

*****
* Parameters selected from various treatments for generic use.
* Specific for variety/site/season.
*****

FNCLV = 0.50 ! Fraction of total N crop present in leaves (pre-flowering) [-]
FNSO = 0.01 ! Fraction of N in the storage organs [-]
ANLVMX = 100. ! Maximum amount of N in the leaves [kg/ha]

* Maximum absolute N uptake rate [kg/(ha*d)] vs. DAT
MXUP1R = 0.,5.,
800.,5.

RUR = 0.2 ! Relative nitrogen uptake rate [1/d]
NUPCO = 0.03 ! Maximum ratio of N uptake rate to growth rate [g/g]
HI5 = 0.551 ! Intercept of harvest index (HI) on WCRFF/RDDSUM = 5 [-]

* Slope of the relation between HI and WCRFF/RDDSUM [-]
HISLOP = 0.00767

* Daily nitrogen soil supply vs. DAT [kg/(ha*d)]
SST = 0.,0.6,
800.,0.6

* Maximum N concentration in crop vs. DAT [g/g]
FNMAXT = 0., 0.03,
40., 0.017,
60., 0.015,
80., 0.01,
130., 0.07

* Fertiliser N recovery vs. DAT [g/g]
RECT = 0., 0.0,
60., 0.8,
65., 0.4,
70., 0.0,
150., 0.

* Root-shoot ratio vs. DAT [-]
RSRT = 0., 0.4,
1.2, 0.15

SDLAGE = 10. ! Seedling age at transplanting [d]

*****
* Parameters defining continuous fertiliser application curve and
* discrete split dressings. The parameters A, B, C and M can be

```



```

* optimised (for maximum grain yield) for given FERTMX by
* MANAGE-N, but values may also be user-defined. Values of split-
* dressings can only be user-defined.
*****

FERTMX = 200. ! Maximum fertiliser input [kg/ha]
A = 1. ! Symmetry parameter of cumulative N application curve [-]
B = 0.1 ! Slope parameter of cumulative N application curve [1/d]
C = 300. ! Asymptotic level of cumulative N application curve [kg/ha]
M = 30. ! Time shift parameter of cumulative N application curve [d]

* Days after transplanting at which critical crop mass check takes place [d]
DATCR = 35.

* Intercept on Y-axis of the WCRCR vs. WRR relation [-]
WCR0 = 999999999.
WCRSLP = 0. ! Slope parameter of the WCRCR vs. WRR relation [-]

*****
* Dummy values.
*****

APCUMI = 0. ! Initial N appl. dummy value (should always be zero)

*****

```

Example of a timer file

The file below is an example of the part of the timer file that the user can select and edit. This part is automatically merged with the RUNCTRL.DTA file prior to the model run and placed in the working directory of MANAGE-N. The resulting merged file is used by ORYZA_0.

```

*****
* Timer file to be used by model ORYZA_0 *
*****
* Experiment '91, IRRI, 0kg

STTIME = 194. ! start time (day) of simulation (transplanting)
FINTIM = 600. ! finish time of simulation

*****

```

Example of a weather file

Weather files in the MANAGE-N system should be formatted like the example weather file listed here. That includes the header with information! An exact description of the format of the weather files used by the WEATHR system in the ORYZA_0 model can be found in van Kraalingen et al. (1991c) and Stol (1994).

```

*-----*
* Station name: Wageningen (Haarweg), Netherlands
* Year: 1985
* Author: Peter Uithol -99.000: NIL VALUE ,

```

```

* Source: Natuur- en Weerkunde via Nel van Keulen
* Longitude: 5 40 E, latitude: 51 58 N, altitude: 7 m.
*
* Column Daily value
* 1 station number
* 2 year
* 3 day
* 4 irradiation (kJ m-2 d-1)
* 5 minimum temperature (degrees Celsius)
* 6 maximum temperature (degrees Celsius)
* 7 early morning vapour pressure (kPa)
* 8 mean wind speed (height: 2 m) (m s-1)
* 9 precipitation (mm d-1)
*-----*
5.67 51.97 7. 0.00 0.00
1 1985 1 660. 0.2 5.7 0.670 5.4 6.8
1 1985 2 2200. -2.9 0.7 0.490 2.2 0.1
1 1985 3 2280. -5.5 0.2 0.520 1.6 0.0
1 1985 4 3310. -18.4 -5.2 0.230 0.7 0.4
1 1985 5 4220. -19.5 -7.8 0.210 0.6 0.0
1 1985 6 1940. -13.8 -6.2 0.310 3.9 1.7
1 1985 7 4040. -21.4 -9.7 0.200 3.3 0.0
1 1985 8 2710. -21.3 -6.2 0.240 2.2 0.0
1 1985 9 930. -7.8 -3.8 0.380 4.1 0.1
1 1985 10 1540. -11.6 -3.6 0.390 2.1 0.6
<continued>

1 1985 362 940. -6.2 -3.0 0.430 1.1 0.0
1 1985 363 2000. -3.7 -1.2 0.460 1.7 0.0
1 1985 364 900. -5.3 -2.4 0.450 2.9 0.0
1 1985 365 3410. -6.2 -3.0 0.660 5.4 0.0
*-----*

```

Note: MANAGE-N only requires radiation data. The other weather variables may be listed as dummies, if not available.

The MANAGE-N.CFG file

The file MANAGE-N.CFG specifies some options that can define the way MANAGE-N behaves in different computer environments. The example file listed below is the default as distributed with the installation disks.

```

! If you like monochrome screen-output instead of colour, change the
! next line into DISPLAY=MONO or remove it altogether.
DISPLAY=COLOR
! If you want to use another temporary area other than the default
! SYSPATH, then define it here. Don't forget the backslash \
! (of course the directory must exist)
! TEMPDIR=C:\TMP\

! The default viewer (must be in the main MANAGE-N directory).
! If you like, you can specify another viewer here. But please, FIRST
! read the conditions that the viewer must comply to. See Appendix
! "Technical reference of MANAGE-N" in the MANAGE-N manual.
! VIEWER=LIST.COM

```

The MANAGE-N.BAT file

This file is used to start the MANAGE-N tool. If printing of plots is required, make sure that the GRAPHICS file is executed (remove the REM before GRAPHICS in this file).

```
@echo off
rem This batchfile controls the startup of MANAGE-N and applications.
rem
rem
rem This batchfile executes the shell. If the shell is left, then
rem it checks on the occurrence of a temporary statusfile called
rem PROJECT.TMP. If this file does not exist, then a regular
rem 'Quit Shell' command is given. Else, a large application should
rem be started. The commandline of this application has been defined in
rem APPL_COM.BAT, which will be executed.
rem After execution, control is passed again to the shell.
rem *****

rem GRAPHICS
rem The above line will enable printing of graphs on an Epson-compatible
rem printer (and maybe others, especially if you use MSDOS 5.0. Refer to
rem a local computer expert to set this up) if you remove the 'rem'
rem statement.

:LOOP
    MANX-N.EXE
    IF NOT EXIST $status$.SC$ GOTO THE_END
    CALL APPL_COM.BAT
    GOTO LOOP
:THE_END
```

The CONTROL.DAT file

This is the file used by the FSE system (in which the ORYZA_0 model is programmed) that specifies which files are to be used during the simulation. CONTROL.DAT is automatically created by MANAGE-N just prior to a model run, from selections that the user made.

```
* CONTROL.DAT created by MANAGE-N 1.0
FILEON='RESULTS.OUT'
FILEIR='RERUNS.DAT'
FILEOL='RESULTS.LOG'
FILEI1='C:\USR\JEANJACK\DEMO\MANAGE-N\DATA\CROP\91IR110K.ALL'
FILEIT='T_91IR72.DAT'
```

The RUNCTRL.DTA file

This file is part of the timer file that ORYZA_0 uses. It contains various options that are not really of interest to the user. Most options are updated by MANAGE-N on-the-fly. Just prior to a model run, this file is joined with the timer file the user chose, and placed in the working directory of MANAGE-N.

```

*****
*   RUNCONTROL FILE for MANAGE-N   *
*   THESE DATA SHOULD NOT BE CHANGED BY THE MANAGE-N USER   *
*   EXCEPT FOR PRDEL IN SOME CASES   *
*****

* Weather data specification
WTRDIR = 'C:\USR\FSU\DATA\INPUT\WEATHER\'
CNTR   = 'PHIL'
ISTN   = 1           ! Station number of weather data
IFLAG = 1100         ! Indicates where weather error and warnings go
                        ! (1100 means errors and warnings only to log file,
                        ! see FSE manual)

* Time variables
IYEAR  = 1991        ! Year of weather data
DELT   = 1.          ! Time step of integration

* Output options
COPINF = 'N'         ! Switch variable denoting what to be done with
                        ! input files:
                        !   'N' = do not copy input files into output file
                        !   'Y' = copy input files into output file

PRDEL  = 5.          ! Time between consecutive outputs to file,
                        ! (when PRDEL=0, no output is generated, when PRDEL is
                        ! very large (i.e. 10000) only initial and terminal
                        ! output is generated).
IPFORM = 4           ! (0 = no output table, 4 = normal table,
                        ! 5 = tab-delimited (for Excel), 6=TPPLOT format)
DELTMP = 'N'         ! Switch variable what should be done with the
                        ! temporary output file:
                        !   (0 = do not delete,
                        !   1 = delete)

*****

```

The OPTIM.DEF file

Used during the optimisation procedure, the OPTIM.DEF file lists control variables that define how many initial and iteration runs must be made. Furthermore, it specifies the tolerance level and whether parameter values should be set to boundaries or discarded when values fall outside the range defined in the PARAM.DEF file. This file is generated by MANAGE-N and is only listed here for the sake of completeness.

```

IBOUND = 0
INPS   = 40
INT     = 300
FTOL    = 1.0E-9
IOUT    = 1, 1, 1, 1

```

The PARAM.DEF file

Like the OPTIM.DEF file, this file is also generated by MANAGE-N just prior to an optimisation run. It lists the parameters for which optimisation should commence, together with the valid ranges for these parameters. In this case, the parameters defining the continuous fertiliser N application curve are listed.

A	=	0.01	,	10.
B	=	0.1	,	1.
C	=	10.	,	500.
M	=	10.	,	80.

Appendix 9 Acronyms used in ORYZA_0

The following table lists all variables used in the ORYZA_0 model in alphabetical order. MANAGE-N uses different 'derivatives' or 'forms' of ORYZA_0, depending on the selected *method* (task). Some of the parameters and variables are only used in the source code as intermediate variables and cannot be found in the output files. However, they are listed too.

Table A9.1 Acronyms used in all ORYZA_0 sources under MANAGE-N.

Acronym	Explanation	unit
A	symmetry parameter of cumulative application curve	-
ACTREC	fraction actual N recovery (total simulation period)	-
AFFLOW	switch indicating that flowering stage has been passed	-
ANCR	amount of N in the crop (live and dead material)	kg ha ⁻¹
ANLV(I)	amount of N in the leaves (initial)	kg ha ⁻¹
ANLVCH	rate of N allocation to the leaves	kg ha ⁻¹ d ⁻¹
ANLVMX	maximum amount of N in the leaves	kg ha ⁻¹
ANSOCH	rate of N allocation to grains	kg ha ⁻¹ d ⁻¹
APCUM(I)	cumulative N application curve (initial)	kg ha ⁻¹
APSLOP	daily fertiliser N application	kg ha ⁻¹ d ⁻¹
B	slope parameter of cumulative N application curve	d ⁻¹
C	asymptote level of cumulative N application curve	kg ha ⁻¹
CUMNSS	cumulative native soil N supply	kg ha ⁻¹
DAT	days after transplanting	d
DATCR	days after transplanting at which critical crop mass check takes place	d
DATEH	harvest date in days of year (DOY)	d
DATFF	date of first flowering in days after transplanting (DAT)	d
DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
DATH	harvest date in days after transplanting (DAT)	d
DELT	time step for integration	d
DEMAND	total crop N demand	kg ha ⁻¹ d ⁻¹
DOY	day of year=Julian date	d
EPSIL	initial radiation use coefficient	g MJ ⁻¹
FERTMX	total fertiliser N input	kg ha ⁻¹
FITWCR	simulated shoot or total crop weight, depending on field observations made (i.e. no roots measured, then FITWCR equals shoot weight). For comparison with WCR_OBS.	kg ha ⁻¹
FNACT	simulated whole-crop nitrogen fraction	g g ⁻¹
FNCLV	fraction of total crop N present in leaves (preflowering)	g g ⁻¹
FNLV	fraction of N in leaves	g g ⁻¹
FNLV_OBS	table of observed fraction (in %) of N in the leaves versus year and day of year ² . Used as forcing function in 'Fit to Match' and 'Single	

	Run', or as comparison with simulated fractions in 'Single Run' and 'Split Evaluation'.	g g^{-1}
FNMAX	maximum overall fraction of N in total crop biomass	g g^{-1}
FNMAXT	table of FNMAX versus days after transplanting (DAT)	g g^{-1}
FNSO	fraction of N in grains and panicles	g g^{-1}
FSV	site-variety match factor	-
FSV1	site-variety match factor before flowering	-
FSV2	site-variety match factor after flowering	-
GCR	crop growth rate (dry matter)	$\text{kg ha}^{-1} \text{d}^{-1}$
GRUC	growth per unit incident radiation	g (MJ)^{-1}
HI	harvest index (based on total biomass, including roots)	kg kg^{-1}
HI5	value of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals $5 \text{ kg ha}^{-1} / (\text{MJ m}^{-2})$	kg kg^{-1}
HISLOP	slope of the relation between the harvest index and WCRFF/RDDSUM	$\text{MJ m}^{-2} / (\text{kg ha}^{-1})$
IDOY	integer value of DOY	d
K1	Proportionality factor relating N pool size to potential uptake rate ¹	d^{-1}
K2	Proportionality factor relating N pool size to loss rate ¹	d^{-1}
K2T	table of K2 versus days after transplanting (DAT) ¹	d^{-1}
KNRAT	ratio of potassium to nitrogen in the crop (optional)	g g^{-1}
KURQMN	minimum amount of potassium uptake required (optional)	kg ha^{-1}
LARGE	dummy variable	-
LAT	latitude (south of equator negative values)	degree
LNUC	growth per day per unit leaf N	$\text{g g}^{-1} \text{d}^{-1}$
M	time shift parameter of cumulative application curve	d
M12345	crop N demand, equals the minimum value of all limitations	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXDAT	number of days in exponential N uptake phase (DAT)	-
MAXNCR	maximum amount of N in crop during exponential N uptake phase	$\text{kg ha}^{-1} \text{d}$
MAXUP0	maximum N uptake rate during exponential growth phase	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXUP1	maximum N uptake rate	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXUP2	N uptake rate as limited by the maximum N fraction in new dry matter	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXUP3	N uptake rate as limited by maximum overall fraction of N in total biomass	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXUP4	N uptake rate as limited by the maximum bulk amount leaf N	$\text{kg ha}^{-1} \text{d}^{-1}$
MAXUP5	N uptake rate as limited by nearing maturity stage	$\text{kg ha}^{-1} \text{d}^{-1}$
MXUP1T	table of MAXUP1 versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{d}^{-1}$
NAPPLD	total amount of N applied	kg ha^{-1}
NAVAIL	total N availability (soil and fertiliser)	$\text{kg ha}^{-1} \text{d}^{-1}$
NHRVST	total N uptake at harvest	kg ha^{-1}
NL	leaf N, as ANLV, but expressed in g m^{-2}	g m^{-2}
NLOSS	daily loss of N from the nitrogen pool ¹	$\text{kg ha}^{-1} \text{d}^{-1}$
NLOST	total amount of nitrogen lost from the pool ¹	kg ha^{-1}
NPOOL	pool of nitrogen in the soil ¹	kg ha^{-1}
NUPCO	maximum ratio of daily N uptake to growth	g g^{-1}
NUPT	N uptake rate	$\text{kg ha}^{-1} \text{d}^{-1}$
P	initial leaf nitrogen use coefficient	$\text{g g}^{-1} \text{d}^{-1}$
PNRAT	ratio of phosphorus to nitrogen in the crop (optional)	g g^{-1}

PURQMN	minimum amount of phosphorus uptake required (optional)	kg ha ⁻¹
RADTOT	cumulative radiation	MJ m ⁻²
RDD	radiation, daily total global, measured	J m ⁻² d ⁻¹
RDDSUM	sum of daily total global radiation from first flowering to end of simulation	MJ m ⁻²
RDM	radiation, daily total global, measured	MJ m ⁻² d ⁻¹
RECOV	best attainable recovery	kg kg ⁻¹
RECT	table of RECOV versus days after transplanting (DAT)	kg kg ⁻¹
RELTIME	relative time, equals 0 at seeding, equals 1 at DATFF. Required for interpolation of RSRT table	-
RNEFF	growth per unit radiation and unit leaf N	g g ⁻¹ (MJ m ⁻²) ⁻¹
RSR	root:shoot ratio	g g ⁻¹
RSRT	table of RSR versus relative time (RELTIME), required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS.	g g ⁻¹
RTINCL	switch to indicate whether root biomass was included in measured crop biomass WCR_OBS	-
RUR	relative nitrogen uptake rate	d ⁻¹
SDLAGE	seedling age at transplanting, required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS.	d
SOLSUP	native soil N supply	kg ha ⁻¹ d ⁻¹
SPLIT	table of sizes of discrete N split applications	kg ha ⁻¹
SPLDAT	table of days at which discrete N splits are applied (in DAT)	d
SST	table of SOLSUP versus days after transplanting (DAT)	kg ha ⁻¹ d ⁻¹
STPDAT	contingency cut-off date of N application (DAT)	d
STTIME	starting time of simulation (day of year)	d
SWIEXP	switch to signal end of exponential growth phase	-
SWINLV	switch for choosing measured or simulated amount of leaf N	-
TMN	minimum night temperature	°C
TMX	maximum day temperature	°C
TREATM	character string sent to the output file, containing the title of the experiment and treatment	-
WCR (I)	weight of the crop (shoot plus storage organs, roots) (initial)	kg ha ⁻¹
WCRFF	weight of the crop at first flowering	kg ha ⁻¹
WCRSLP	slope parameter of the WCRCR versus WRR relation	-
WCR0	intercept on Y-axis of the WCRCR versus WRR relation	kg ha ⁻¹
WCR_OBS	table of measured weight of crop versus year and day of year ² , aboveground (if RTINCL = 0) or total (if RTINCL = 1). Used for comparisons with simulated values, and for FSV calculation ('Fit to Match').	kg ha ⁻¹
WLV	weight of the leaves	kg ha ⁻¹
WLV_OBS	table of observed weight of the leaves versus year and day of year ² . Used as forcing function in 'Fit to Match' and 'Single Run', or as comparison with simulated leafmass in 'Single Run' and 'Split Evaluation'.	kg ha ⁻¹
WRR	weight rough rice	kg ha ⁻¹
WRT	weight of the roots (live and dead)	kg ha ⁻¹

WSHT	weight of the shoots	kg ha ⁻¹
X	derivative of logistic N application curve plus soil N	kg ha ⁻¹ d ⁻¹
XXWCR	measured weight of crop, aboveground (if RTINCL = 0) or total (if RTINCL = 1). Dummy name.	kg ha ⁻¹

¹ Used in 'Split Evaluation' *method* only.

² See paragraph 'Description of the input for MANAGE-N' on page 269 for an example of the required format of these observations.

Appendix 10 Required input for the MANAGE-N system

Three separate data files are required to run the ORYZA_0 modules that are part of MANAGE-N: a weather, a crop/soil data, and a timer file. This section lists input parameters and variables contained in those files. Other inputs to MANAGE-N, not used by ORYZA_0, are not listed here. A cross-reference of parameters and variables required per *method* is listed in Table A10.1. Table A10.2 provides a full list of all input parameters and variables, along with their description and unit. Examples of the data files can be found in Appendix 8.

Table A10.1 Acronyms of required MANAGE-N input, arranged by *method*..

Acronym	Default	Single Run	Fit to Match	Optimise N-curve	Split Evaluation
A		X		X ⁴	
ANCRI		X		X	X
ANLVMX [100.]	•	X		X	
B		X		X ⁴	
C		X		X ⁴	
DATCR [999.]	•			X	
DATFF		X	X	X	X
DATFSV		X	X ³	X	X
DATH		X	X	X	X
FERTMX		X		X ⁵	X
FNCLV		X		X	X
FNLV_OBS		X ⁸	X ⁹		X ⁸
FNMAXT		X		X	X
FNSO [0.012]	•	X		X	X
FSV1		X	X ³	X	X
FSV2		X	X ³	X	X
HI5 [0.55]	•	X	X ⁷	X	X
HISLOP [0.008]	•	X	X ⁷	X	X
K2T	•				X
[0., 0.1, 999., 0.1]					
KNRAT		X ⁶		X ⁶	X ⁶
M		X		X ⁴	
MXUP1T		X		X	X
NUPCO [0.05]	•	X		X	X
PNRAT		X ⁶		X ⁶	X ⁶
RECT		X		X	X
RSRT	•	X ¹	X ¹		X ¹
[0, 0.4, 1.2, 0.15]					
RTINCL		X	X		X
RUR [0.2]	•	X		X	X
SDLAGE		X ¹	X ¹		X ¹
SPLIT					X
SPLDAT	•				X
SST		X		X	X
STPDAT		X ²			
TREATM		X	X	X	X
WCRI		X	X	X	X

WCRSLP [0.]	•			X	
WCR0 [999999.]	•			X	
WCR_OBS		X ⁸	X ⁹		X ⁸
WLV_OBS		X ⁸	X ⁹		X ⁸

¹ Only needed to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included (i.e., roots are not included in WCR_OBS).

² No value should be specified, except when contingency cut-off of N application should be simulated.

³ Not required to enter values under this method: MANAGE-N will automatically change or add the value during FSV computation.

⁴ MANAGE-N will optimise this parameter. Not required for user to enter values.

⁵ FERTMX will be updated by MANAGE-N automatically, depending on the N levels that have been chosen by the user.

⁶ This is an optional parameter. Only required when minimum potassium and phosphorus requirements of the crop should be printed in the output.

⁷ Value is irrelevant to perform the 'Fit to Match' but must be supplied.

⁸ Observed values are only relevant for comparing 'Single Run' and 'Split Evaluation' simulated values with observed values.

⁹ Observed values are crucial in 'Fit to Match'.

• For these parameters, default values have been implemented in MANAGE-N. When the user does not have data to derive values for these parameters, the user is advised to use the defaults. The default values are listed with the parameter names in square brackets.

Table A10.2 Acronyms of required input variables and parameters for MANAGE-N.

Acronym	Explanation	unit
<i>Weather data file</i>		
latitude		degree
altitude		m
daily total global radiation during year	sunshine h or $\text{kJ m}^{-2} \text{d}^{-1}$	
<i>Crop/soil data file</i>		
A	symmetry parameter of cumulative application curve	-
ANCRI	initial amount of N in the crop	kg ha^{-1}
ANLVMX	maximum amount of N in leaves	kg ha^{-1}
B	slope parameter of cumulative N application curve	d^{-1}
C	asymptote level of cumulative N application curve	kg ha^{-1}
DATCR	days after transplanting at which critical crop mass check takes place	d
DATFF	date of first flowering in days after transplanting (DAT)	d
DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
DATH	harvest date in days after transplanting (DAT)	d
FERTMX	total fertiliser N input	kg ha^{-1}
FNCLV	fraction of total crop N present in leaves (preflowering)	g g^{-1}
FNLV_OBS	table of observed fraction (in %) of N in leaves versus year and day of year ¹	g g^{-1}
FNMAXT	table of maximum crop N concentration FNMAX versus days after transplanting (DAT)	g g^{-1}
FNSO	fraction of N in grains and panicles	g g^{-1}
FSV1	site-variety match factor before flowering	-
FSV2	site-variety match factor after flowering	-
HI5	value of harvest index where $\text{WCRFF/RDDSUM} = 5$	kg kg^{-1}
HISLOP	slope of the relation between the harvest index and WCRFF/RDDSUM	$\text{MJ m}^{-2} / (\text{kg ha}^{-1})$
K2T	table of N loss coefficient K2 versus days after transplanting (DAT)	d^{-1}
KNRAT	ratio of potassium to nitrogen in the crop	g g^{-1}
M	time shift parameter of cumulative application curve	d
MXUP1T	table of maximum N uptake rate MAXUP1 versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{d}^{-1}$
NUPCO	maximum ratio of daily N uptake to growth	g g^{-1}
PNRAT	ratio of phosphorus to nitrogen in the crop	g g^{-1}
RECT	table of potential recovery RECOV versus days after transplanting (DAT)	kg kg^{-1}
RSRT	table of root:shoot ratio RSR versus relative time (RELTIME)	g g^{-1}
RTINCL	switch to indicate whether root biomass was included in measured crop biomass WCR_OBS	-
RUR	relative nitrogen uptake rate	d^{-1}
SDLAGE	seedling age at transplanting	d
SPLIT	table of sizes of discrete N split applications	kg ha^{-1}
SPLDAT	table of days at which discrete N splits are applied (in DAT)	d

SST	table of native soil N supply SOLSUP versus days after transplanting (DAT)	kg ha ⁻¹ d ⁻¹
STPDAT	contingency cut-off date of N application in days after transplanting (DAT)	d
TREATM	treatment identifier	-
WCRI	initial weight of the crop	kg ha ⁻¹
WCRLP	slope parameter of the WCRCR versus WRR relation	-
WCRO	intercept on Y-axis of the WCRCR versus WRR relation	kg ha ⁻¹
WCR_OBS	table of measured weight of crop versus year and day of year ¹ , aboveground (if RTINCL = 0) or total (if RTINCL = 1)	kg ha ⁻¹
WLV_OBS	table of observed weight of the leaves versus year and day of year ¹	kg ha ⁻¹
<i>Timer file</i>		
STTIME	Start time of simulation in day of year (DOY)	d
FINTIM	Finish time: when the internal control variable TIME exceeds FINTIM, model simulation stops. TIME is initialised with value STTIME, then increases with every time step. Note: for cases that cross the year boundary, and stop on day x of the new year, FINTIM equals (365 + x) or (366 + x), depending whether the starting year is a normal or a leap year.	d
SWINLV	switch for choosing measured or simulated amount of N in the leaves	-

¹ See paragraph 'Description of the input for MANAGE-N' on page 269 for an example on the required format of these observations.

Appendix 11 **MANAGE-N** parameter derivation and **EXCEL®** sheet

The following pages describe in detail how input parameters can be derived for the ORYZA_0 model. Parameters are listed in the order that they appear in the MANAGE-N editing windows. Where applicable, pointers to a special input and parameter-deriving EXCEL® sheet have been incorporated (see section 'The EXCEL® sheet for ORYZA_0' on page 253). This sheet can be found on the installation disks of MANAGE-N, in a separate directory called SHEET.

The ORYZA_0 'derivatives' used in MANAGE-N differ slightly from the ORYZA_0 version described in Drenth et al. (1994). This also applies to some of the parameters. Differences between the two versions can be found in Appendix 13. To use MANAGE-N, the parameters listed below must be known or the recommended default values should be used.

Note: The units of input parameters in the model description (Drenth et al., 1994) may be different from those in model input files.

Crop and soil parameters

symbol	acronym	description	unit (in MANAGE-N)
f_N	FNCLV	fraction of total crop nitrogen present in leaves (pre-flowering)	$g\ g^{-1}$
$c_{max}(t)$	FNMAXT	maximum overall fraction of N in total crop biomass table versus days after transplanting (DAT)	$g\ g^{-1}$
r_N	RUR	relative nitrogen uptake rate	d^{-1}
q_N	NUPCO	maximum ratio of daily N uptake to growth	$g\ g^{-1}$
$N_{L,max}$	ANLVMX	maximum amount of nitrogen in leaves	$kg\ ha^{-1}$
n_p	FNSO	fraction of nitrogen in grain (incl. panicle)	$g\ g^{-1}$
$S_N(t)$	SST	table of native soil N supply versus time	$kg\ ha^{-1}\ d^{-1}$
$p(t)$	RECT	fertiliser N recovery table versus time	$g\ g^{-1}$
$u_N(t)$	MXUP1T	maximum nitrogen uptake table versus time	$kg\ ha^{-1}\ d^{-1}$

f_N (FNCLV)

To determine this parameter, all pre-flowering observations of total leaf nitrogen ($kg\ N\ ha^{-1}$ ground surface; Y-axis) are plotted versus total crop nitrogen uptake N_c ($kg\ N\ ha^{-1}$ ground surface; along X-axis). The average slope of the ensuing relation is f_N . It is important to exclude post-flowering data because the relation leaf N: crop N vanishes after flowering. The plot is prepared automatically with the help of the ORYZA_0.XLS

sheet. All N application levels can be used, preferably both low and high N treatments (see Figure A11.1).

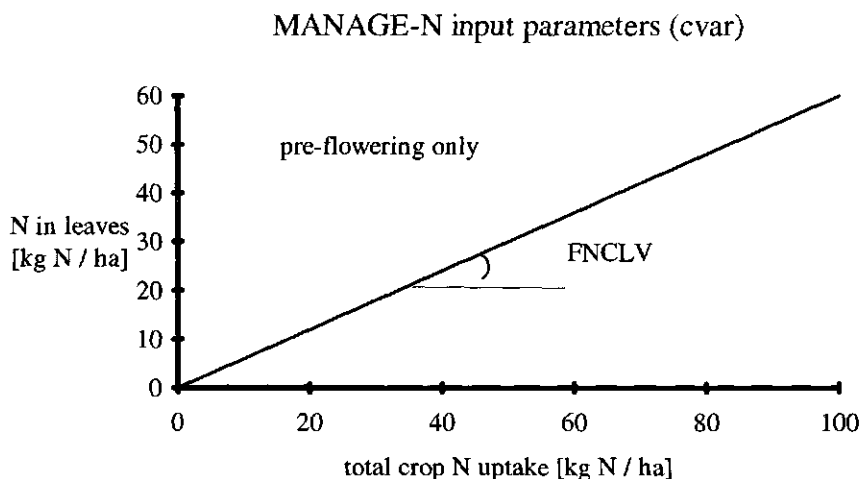


Figure A11.1 Derivation of the FNCLV parameter.

$c_{\max}(t)$ (FNMAXT)

The maximum N concentration (in the mean crop biomass) is calculated as total crop nitrogen uptake N_c (kg N ha^{-1}) divided by total crop biomass (including roots) W (kg ha^{-1}) (not as the mean value of the distinguished organ tissue concentrations!). It is plotted versus time. c_{\max} is a function of time and is defined in the model as a table, where the first value of each pair is time (in days after transplanting DAT) and the second is c_{\max} :

FNMAXT= 0.,0.030, 70.,0.020, 100.,0.015, 130.,0.010

The model obtains the daily values of c_{\max} by linear interpolation between these values with the help of the LINT function. Probably, this variable differs among varieties. Note: Since values are tabulated versus time - not development stage - these tables are different for varieties of different growth duration. The values can be derived from the ORYZA_0.XLS sheet by examining the graph of ANCR/WCR (Y-axis) versus time (X-axis). FNMAXT is defined as the boundary curve connecting all upper points in this graph (Figure A11.2). The graph has no significance if not derived from a treatment with very high N input.

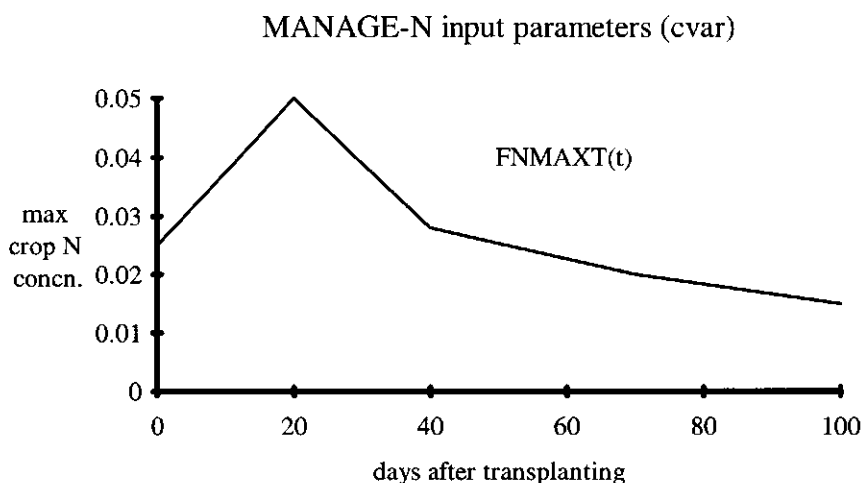


Figure A11.2 Derivation of the FNMAXT table.

r_N (RUR)

The relative nitrogen uptake rate is determined by plotting the natural logarithm of total crop N uptake, $\ln(N_c)$, versus time. The first part of this relation, covering 2-3 weeks, is usually a straight line. The slope of this linear section is r_N . This is done only for treatments where the crop had ample nitrogen available from transplanting onward. Values up to 0.2 d^{-1} have been found so far. This parameter is derived from the ORYZA_0.XLS sheet by examining the graph of $\ln(\text{ANCR})$ (Y-axis) versus time (X-axis). It is the slope of this graph over the first 20 days after transplanting. The graph has no significance if not derived from a treatment with very high N input.

q_N (NUPCO)

This maximum ratio of daily N uptake to growth is calculated again as a mean value over a sampling interval Δt . It is found by dividing the corresponding increment of crop N uptake, ΔN_c , by the crop biomass (including roots!) increment ΔW . A default value of 0.05 can be used.

$N_{L,\max}$ (ANLVMX)

The total amount of leaf N is probably bound to a maximum. The highest values found so far are roughly 100 kg N ha^{-1} . This value can be used as a default.

n_p (FNSO)

This parameter is directly found by analysing panicle plus grains for nitrogen. If no information is available, a default value of 0.012 can be used. It should be noted that n_p applies to the whole panicle and is therefore always lower than the grain N concentration.

$S_N(t)$ (SST)

The native supply of N by the soil is assessed from plots which received no fertiliser N. It must be specified as a function of time (Figure A11.3), but it can often be taken as a constant. As a first approximation, we suggest to take total crop N uptake N_c (kg N ha⁻¹) divided by the number of field days. This gives the mean daily N supply rate.

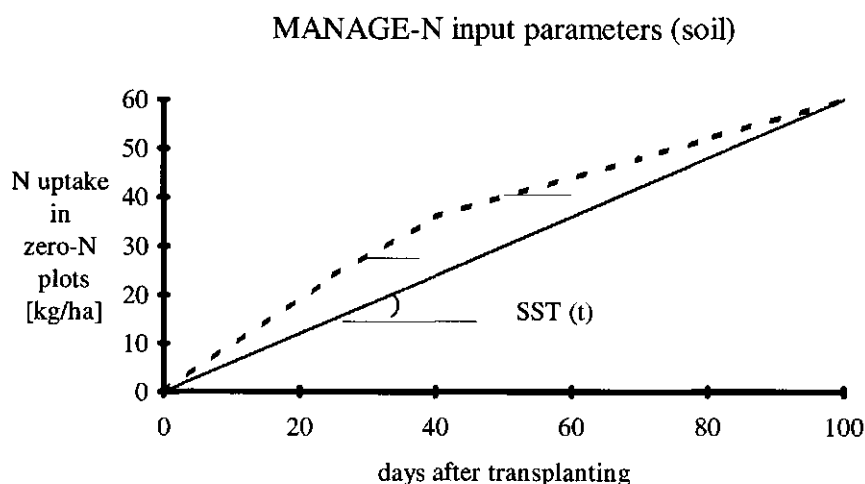


Figure A11.3 Derivation of the SST native soil supply table.

For example, in the next table, the first value is time (in days after transplanting, DAT), the second is S_N :

```
0., 0.6,
999., 0.6
```

$\rho(t)$ (RECT)

Fertiliser N recovery is the fraction of applied fertiliser absorbed by the crop. We correct for soil N supply by subtracting the amount of N taken up by an unfertilised crop from the total crop uptake, N_c . ORYZA_0 needs the recovery as a function of time, $\rho(t)$. A first

approximation of $\rho(t)$ can be obtained from particular experiments. Several examples of such experiments are now available in SARP. A requirement is that the experiment includes two treatments differing only in one of the split dose applications. All other applications should be identical for the two treatments. For example, in treatment T1, fertiliser is given at times t_1 , t_2 , and t_3 . In T2, the same amounts are given at t_1 and t_2 as in T1, but at t_3 a different amount is given -- for example, zero or any other amount different from what was applied at t_3 in T1. At the end of the season, total uptake N_c is determined for both T1 and T2. The extra uptake in the treatment which received most N (applied) is then divided by the extra amount applied in that treatment. The resulting ratio is the apparent recovery corresponding to t_3 . The best information is obtained when the application levels are not too high. Excessive N application obviously leads to low recovery, which then has no meaning at more reasonable input levels.

Another way of recovery analysis is possible: if treatments differ in more than one split dose application, but are identical up to a given time t_x , the difference in N uptake resulting from a treatment difference imposed at t_x may be assessed from sampling shortly after t_x -- i.e., still before the second different treatment event is imposed. This is only a reliable procedure if the amount applied at t_x is not excessive, and enough time is allowed after t_x to take up most of the 'recoverable' N from that application event, before sampling. ORYZA_0 interprets 'recovery' in a special way: it does not 'convert' a fraction $\rho(t_x)$ of N applied at t_x into uptake directly. Instead, a fraction $(1-\rho(t_x))$ is considered as unavoidable loss. So, $\rho(t)$ is a potential recovery. Some part of potentially recoverable N may be lost if the crop cannot absorb the whole fraction $\rho(t)$ at the time of application.

The information obtained in the manner described above (Figure A11.4) is then combined into a table (the first value equals time in days after transplanting DAT, the second value is ρ , and so on):

RECT= 0.,0.0, 40.,0.7, 70.,0.8, 75.,0.8, 85.,0.01, 135.,0.

This table is linearly interpolated with the help of the LINT function. A good default is to start with a value of 0 at transplanting (on the first day, no applied N would be taken up); then have a linear increase up to PI stage where a peak value is reached (e.g., a fraction $\rho = 0.7$ is potentially absorbed from a small amount of N applied at PI stage, provided the crop is not N saturated). Then, depending on soil and climate, ρ may be maintained at this high level for a while -- e.g., up to flowering -- and then quickly decrease to zero again at 10-20 days after flowering.

The $\rho(t)$ curve determines to a large extent the outcome of the optimisation procedure. Erroneous assumptions in $\rho(t)$ definitely lead to erroneous recommendations. This is unpleasant, but unavoidable: $\rho(t)$ varies in reality and has a strong effect on N uptake.

An important feature of $\rho(t)$ is the time when the peak value is reached and the length of the period during which a high ρ level is maintained. This pattern is more relevant to optimising fertiliser timing than the absolute level of $\rho(t)$.

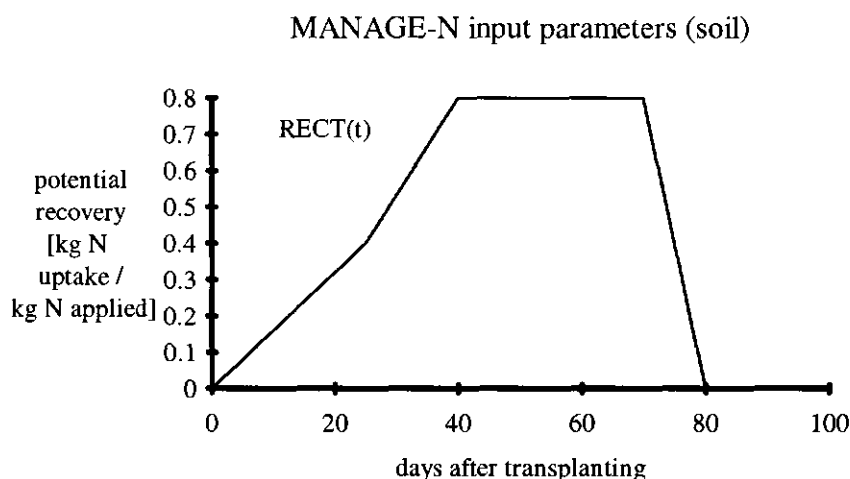


Figure A11.4 Derivation of the nitrogen recovery table RECT.

In composing the table, note MANAGE-N interpolates linearly between tabulated values. For example, if in reality no N can be taken up after flowering, the table should include a value 0.0 right after flowering, not two weeks later. Recovery would otherwise decrease gradually during the two weeks following flowering, and the (simulated) crop would (erroneously) absorb a considerable amount of 'extra' N in that period. Although much can be said against this approach of defining and utilising $\rho(t)$, the main advantage (its empirical nature allows taking into account site peculiarities) seems to outweigh possible disadvantages.

Although the ORYZA_0 .XLS sheet calculates crop N uptake, it does not deliver potential recovery. Potential recovery must be assessed from application and uptake in especially designed experiments with split applications as explained above. (Note: actual, as opposed to potential recovery, is an output of MANAGE-N.)

$u_N(t)$ (MXUPIT)

The time course of maximum nitrogen uptake rate $u_N(t)$ is determined by calculating the mean rate of uptake over time intervals. This is done by dividing ΔN_c (the total crop N increment between two sampling dates, in kg N ha⁻¹) by the duration of the sampling interval -- i.e., the time Δt between the two sampling dates. (The symbol Δ denotes a difference between two values.)

Climate and soil may have a strong influence on u_N , as well as crop development stage, although relations have not yet been established. Values as high as 8 kg N ha⁻¹ d⁻¹ are rarely found. Values of 3 - 5 kg N ha⁻¹ d⁻¹ are found more commonly. The ORYZA_0

simulation studies made on the basis of experimentally determined coefficients indicate that $u_N(t)$ is, in many cases, the most important limitation to N uptake during a major part of the growth cycle.

This parameter is plotted by the ORYZA_0.XLS sheet in the graph of DANCER/DTIME (Y-axis) versus time (X-axis). See Figure A11.5. Values must be derived for cases where ample N was available for uptake.

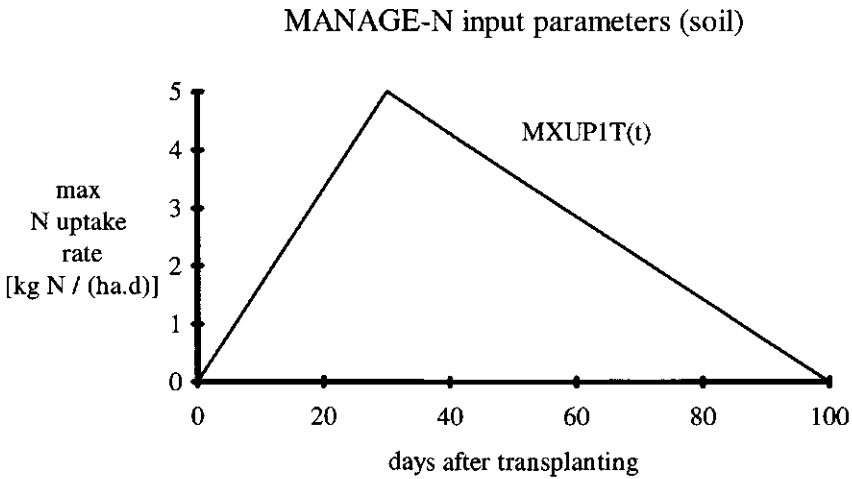


Figure A11.5 Derivation of the MXUP1T values.

For example, in the next table, the first value is time (in days after transplanting, DAT), the second is u_N :

```
0., 5.0,
999., 5.0
```

Fertiliser application parameters

Four parameters (A, B, C, and M) define the fertiliser application curve $A(t)$. A fifth parameter A_{tot} determines the absolute N input level which may cut off the $A(t)$ curve. Together, these parameters express N fertiliser management in the form of a continuous application curve. The curve parameters are not determined experimentally but are computed by MANAGE-N.

symbol	acronym	description	unit (in MANAGE-N)
a	A	affects symmetry of logistic function	-
b	B	affects slope at inflection point	d^{-1}
c	C	asymptote level of logistic function	$kg\ ha^{-1}$
m	M	affects t at inflection point	d
A_{tot}	FERTMX	absolute total N input level	$kg\ ha^{-1}$
W_0	WCR0	intercept on Y-axis of critical crop weight versus crop yield relation	-
W_s	WCRSLP	slope parameter of the critical crop weight versus crop yield relation	-

Optimum values of the first four parameters are computed by MANAGE-N under the 'Optimise N-curve' *method* (see also paragraph 5.5.4 on page 93). FERTMX is the user-defined total N input for which the 'Single Run' and 'Optimise N-curve' model runs are made.

The domain of valid (A, B, C, and M) points can be restricted for certain applications, by assigning limiting values to WCR0 and WCRSLP. This validity check in the optimisation module of ORYZA_0 (ORY0OPT.EXE) is optional and is not used here. The check is based on the reasoning that the crop weight (WCR) should have reached a minimum value before a user-specified date (DATCR) in order to reach at least a given grain yield. For a final yield WRR, this minimum required weight equals $(WRR - W_0) / W_s$. The solid line in Figure A11.6 depicts the relation between this minimum required weight (X-axis) and WRR.

Since no experimental values for W_0 and W_s are available at the moment, the validity check is inactivated by using default values of $W_0 = 999999999$ and $W_s = 0$. It is unlikely that this check will turn out to be necessary, but the option is included in the model in case future work would indicate so. Basically, this procedure ensures that sufficient 'sink capacity' is formed at early growth, to enable late growth and grain filling as dictated by the 'source' component (leaf N and radiation).

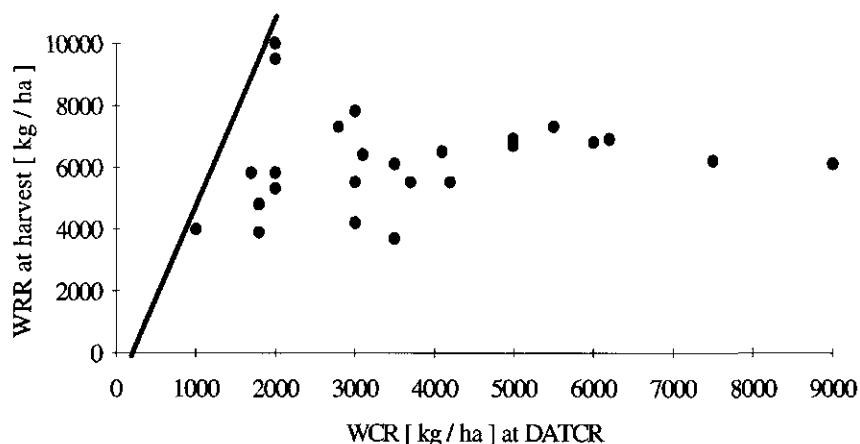


Figure A11.6 Derivation of the parameters that define the APCUM validity check. This Figure shows the weight of the crop biomass at a user-defined critical date (WCR_{CR}) versus rough rice yield (WRR) as could be observed over a range of (here hypothetical) experiments. The solid line represents a minimum required crop biomass at DATCR to reach a given grain yield level. Using the intercept (W_0) and slope (W_s) parameters of this line, a validity check of the APCUM curve can be performed by MANAGE-N during optimisation. Solutions leading to growth curves that would correspond to the domain left of the line are then discarded.

The EXCEL[®] sheet for ORYZA_0

On the distribution disks of MANAGE-N, an EXCEL[®] sheet called ORYZA_0.XLS is located in directory SHEET. This sheet can be used to arrange experimental data, to plot graphs, and to derive parameter values for ORYZA_0. The ORYZA_0.XLS sheet consists of shaded and normal columns. Only in the shaded columns are values to be entered. The values that can be derived from these basic inputs are computed automatically and displayed in the non-shaded columns. After entering sampling dates and the corresponding observed dry weights and nitrogen contents of plant organs (leaves, stems, roots, and panicles), a few graphs appear automatically in the sheet. These can be used to derive some of the other required parameters. Examples are shown below (Figures A11.7 and A11.8).

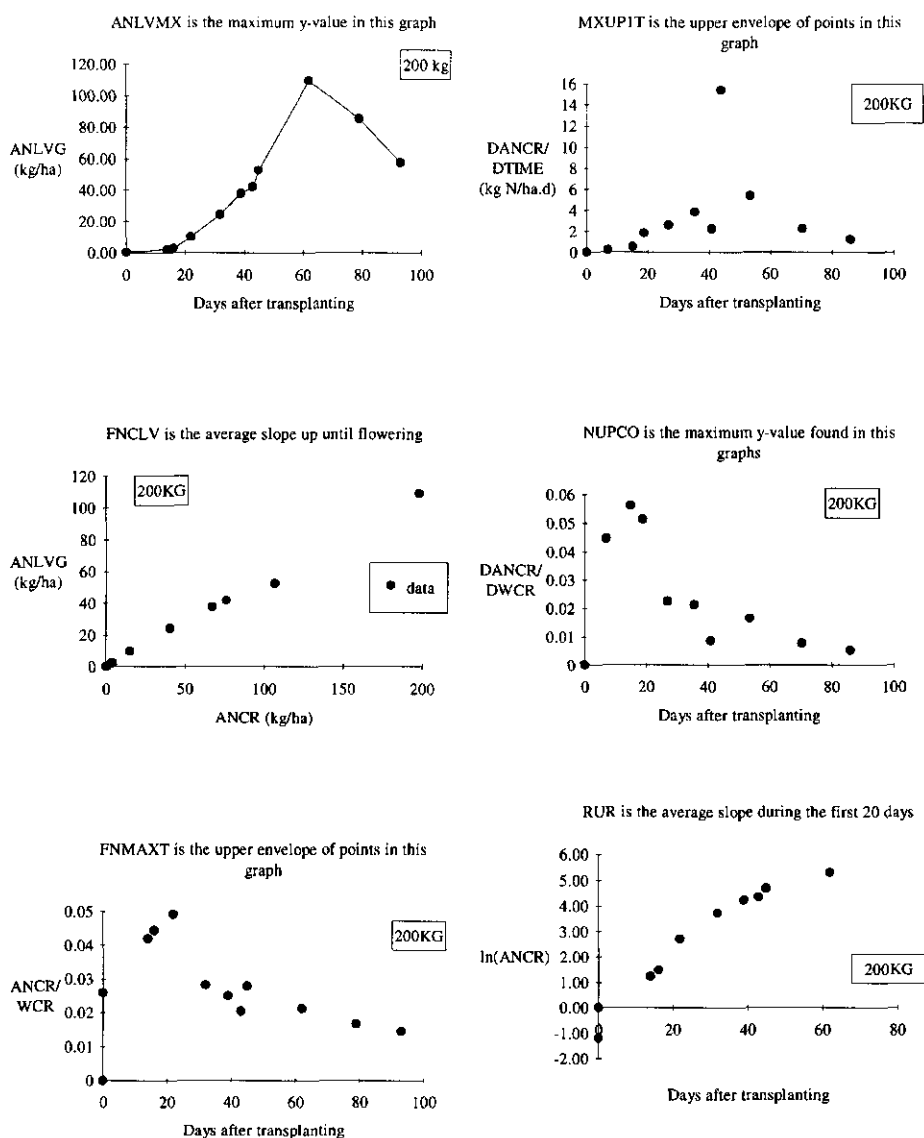


Figure A11.7 Examples of input-derivation graphs created by the ORYZA_0.XLS EXCEL® sheet.

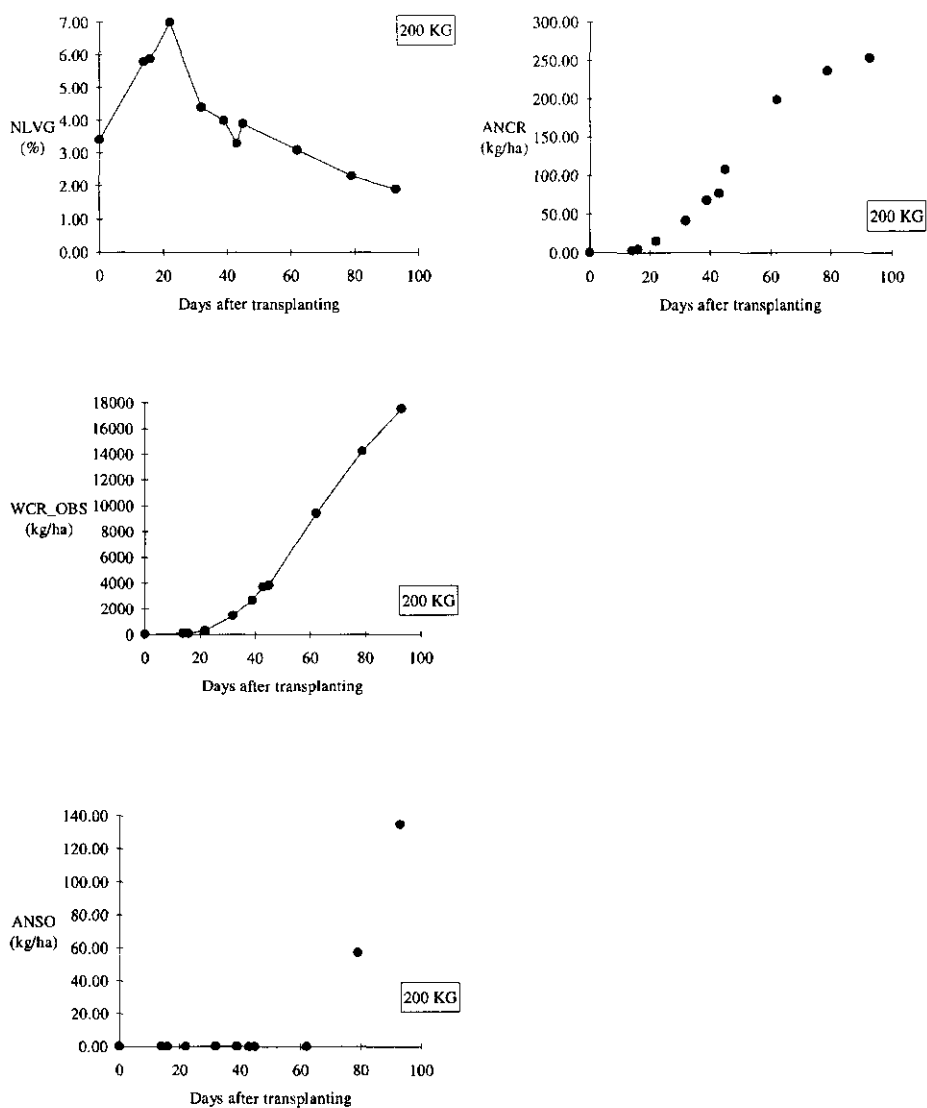


Figure A11.8 Examples of other graphs created by the ORYZA_0 .XLS EXCEL® sheet.

Appendix 12 Description of MANAGE-N input per *method*

All inputs required for MANAGE-N are listed below, per *method*. The parameters, observations and other variables are listed in the order in which they appear in the input windows (see 'Edit A Crop/Soil File' in Chapter 7), together with corresponding explanation and remarks.

The input is divided into three different types of input files: the crop/soil file, the weather file, and the timer file. Every *method* requires all types of input files. Only the content of the crop/soil file may differ per *method* because the input requirement is not the same for each *method*. Note that the crop/soil file for the 'Optimise N-curve' and the 'Split Evaluation' *methods* are more general regarding values of the crop and soil parameters, and thus these files are appended with an extension *.ALL to indicate this.

This Appendix starts with a section on generic inputs that are always required, no matter what *method* is used. Following that are concise listings of input specifications for 'Fit to Match' (page 260), 'Single Run' (page 262), 'Optimise N-curve' (page 265), and 'Split Evaluation' (page 267). The detailed descriptions and remarks of input parameters and variables can be found at the end of this Appendix (page 269).

A12.1 Generic input (all *methods*)

The generic inputs are located only in the weather and the timer file. The model uses only one daily weather variable, R:

symbol	acronym	description	unit
<i>R</i>	RDT	the incident daily global radiation	MJ m ⁻² d ⁻¹

(symbol as used in ten Berge et al., 1994c)

Radiation (Figure A12.1) is the main driving variable for dry matter production. Daily global radiation data are contained in the standard WEATHR data files and are inserted into the model with the help of the WEATHR system. Within the model, the unit is converted from kJ m⁻² d⁻¹ to MJ m⁻² d⁻¹. (If no measured global radiation data are available, daily sunshine duration (h) can be used. WEATHR takes care of the conversion to kJ m⁻² d⁻¹ when the values of the radiation conversion parameters *a* and *b* are supplied.) For more details on the WEATHR system, the reader is referred to van Kraalingen et al. (1991c).

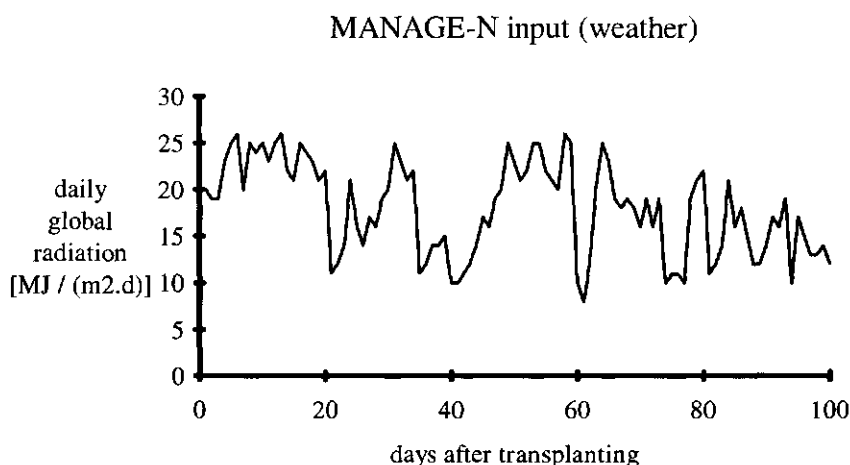


Figure A12.1 An example of the required incident daily global radiation RDT.

Other required inputs, all related to run control and simulation outputs, are situated in the timer file. Under MANAGE-N, the timer file is physically split into two parts: one part that contains control variables regulated by MANAGE-N (a file called `RUNCTRL.DTA`) and one that contains two parameters dealing with the start and the end of the simulation. The first part is hidden from the user, is maintained by MANAGE-N, and should never be modified by the user. The latter can be directly modified by the user via the 'Edit A Timer File' menu.

Table A12.1 lists all the inputs contained in the timer file. Most parameters are indirectly influenced by the user when a weather file is chosen, or when the 'Toggle SWINLV' menu option is used. Only two parameters are directly modifiable by the user: `STTIME` and `FINTIM`. `STTIME` defines the day of the year on which the rice crop is transplanted (starting time of the simulation run). The `FINTIM` parameter defines the absolute maximum time (date) on which a simulation run should terminate. This parameter should be set sufficiently high, like for example 999. Normally, a simulation run ends when the day of harvest (`DATH`) is passed. `FINTIM` is a security measure in case the simulation will not stop at the harvest date for one reason or another.

Table A12.1 Generic input for MANAGE-N, related to run control and output.

symbol	acronym	description	input handled by:
	COPINF	switch that denotes whether input files should be copied into the result files or not	MANAGE-N
	CNTR	country code (weather)	User ¹
	DELT	time step used in integration	MANAGE-N
	DELTMP	switch that specifies whether temporary files should be deleted or not	MANAGE-N
	FINTIM	maximum end time of the simulation	User
	IFLAG	output medium of errors and warnings	MANAGE-N
	IPFORM	format of result files	MANAGE-N
	ISTN	station number (weather)	User ¹
	IYEAR	starting year	User ¹
	PRDEL	step (in days) on which output should be generated	MANAGE-N
t_{TP}	STTIME	calendar day of planting (start of the simulation)	User
	SWINLV	switch that denotes simulation or forcing (observed values) of leaf nitrogen	User ²
	WTRDIR	directory in which weather files are located	User ¹

¹ Indirectly set by the user when choosing a weather file.

² Indirectly set by the user in the 'Single Run' *method* when the 'Toggle SWINLV' menu option is used.

A12.2 Input for the 'Fit to Match' method

The main purpose of the 'Fit to Match' *method* is to find accurate values for f_{SV} and DATFSV. The 'Fit to Match' *method* requires less input than other *methods*: it uses observed values of the fraction of nitrogen in the leaves and hence avoids calculations related to nitrogen uptake and distribution. All parameters listed in the first section of the following table can be found in the first inputscreen of the 'Edit A Crop/Soil File' menu option; all parameters grouped in the second section can be found in the second input screen. Note that the order of presentation in the table is identical with that in the input screens. The detailed description of the input can be found in paragraph 'Description of the input for MANAGE-N' on page 269.

Data for the 'Fit to Match' *method* should be stored in a file with extension *.DAT (see paragraph Edit A Crop/Soil File on page 121).

Table A12.2 Required input for the 'Fit to Match' *method* in MANAGE-N.

symbol	acronym	description	unit
	TREATM	character string sent to the output file, containing the title of the experiment and treatment	-
t_H	DATH	harvest date in days after transplanting (DAT)	d
t_{FF}	DATFF	date of first flowering in days after transplanting (DAT)	d
	WCR1	initial weight of the crop (shoot plus storage organs, roots)	kg ha ⁻¹
	SDLAGE	seedling age at transplanting, required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS.	d
	DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
f_{SV}	FSV1	site-variety match factor (before flowering)	-
	FSV2	site-variety match factor (after flowering)	-
	WCR_OBS	table of measured weight of crop versus year and day of year ² , aboveground (if RTINCL = 0) or total (if RTINCL = 1) ¹	kg ha ⁻¹
	WLV_OBS	table of observed weight of the leaves versus year and day of year ¹	kg ha ⁻¹
	FNLV_OBS	table of measured fraction (in %) of nitrogen in the leaves versus year and day of year ¹	g g ⁻¹
	RTINCL	switch to indicate whether root biomass was included in measured crop biomass WCR_OBS	-

RSRT	table of RSR versus relative time (RELTIME), required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS	g g^{-1}
HI5	value ² of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals $5 \text{ kg ha}^{-1} / (\text{MJ m}^{-2})$	kg kg^{-1}
HISLOP	slope ² of the relation between harvest index and the ratio of biomass at first flowering to post- flowering cumulative radiation	$\text{MJ m}^{-2} / (\text{kg ha}^{-1})$

(unit = as used in MANAGE-N)

(symbols as used in ten Berge et al., 1994c)

¹ See paragraph 'Description of the input for MANAGE-N' on page 269 for an example on the required format of these observations.

² Values must be supplied but are not used in this *method*.

A12.3 Input for the ‘Single Run’ method

The following table is divided into three sections, corresponding to the three input screens of the ‘Edit A Crop/Soil File’ menu option. The order of the parameters and variables corresponds also to the exact order in the input screens. ‘Single Run’ is used for detailed study of model behaviour.

Table A12.3 Required input for the ‘Single Run’ *method* in MANAGE-N.

symbol	acronym	description	unit
	TREATM	character string sent to the output file, containing the title of the experiment and treatment	-
t_H	DATH	harvest date in days after transplanting (DAT)	d
t_{FF}	DATFF	date of first flowering in days after transplanting (DAT)	d
	WCRI	initial weight of the crop (shoot plus storage organs, roots)	kg ha ⁻¹
	ANCRI	initial amount of N in the crop (live and dead material)	kg ha ⁻¹
	SDLAGE	seedling age at transplanting, required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS	d
	DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
f_{SV}	FSV1	site-variety match factor (before flowering)	-
	FSV2	site-variety match factor (after flowering)	-
	WCR_OBS	table of observed weight of the total crop (shoot plus roots) versus year and day of year ¹ , aboveground (if RTINCL = 0) or total (if RTINCL = 1). Used for comparisons with simulated values.	kg ha ⁻¹
	WLV_OBS	table of observed weight of the leaves versus year and day of year ¹ . Used as forcing function in ‘Single Run’, or as comparison with simulated leafmass.	kg ha ⁻¹
	FNLV_OBS	table of measured fraction (in %) of nitrogen in the leaves versus year and day of year ¹ . Used as forcing function in ‘Single Run’, or as comparison with simulated fractions.	g g ⁻¹

	RTINCL	switch to indicate whether root biomass was included in measured crop biomass WCR_OBS	-
	RSRT	table of RSR (root:shoot ratio) versus relative time (RELTIME), required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS.	g g^{-1}
f_N	FNCLV	fraction of total crop nitrogen present in leaves (pre-flowering)	g g^{-1}
$c_{\max}(t)$	FNMAXT	table of FNMAX (maximum fraction of N in total crop biomass) versus days after transplanting (DAT)	g g^{-1}
r_N	RUR	relative nitrogen uptake rate	d^{-1}
q_N	NUPCO	maximum ratio of daily N uptake to growth	g g^{-1}
$N_{L,\max}$	ANLVMX	maximum amount of nitrogen in leaves	kg ha^{-1}
n_D	FNSO	fraction of nitrogen in grains and panicles	g g^{-1}
	HI5	value of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals $5 \text{ kg ha}^{-1} / (\text{MJ m}^{-2})$	kg kg^{-1}
	HISLOP	slope of the relation between harvest index and the ratio of biomass at first flowering to post-flowering cumulative radiation	$\text{MJ m}^{-2} / (\text{kg ha}^{-1})$
$S_N(t)$	SST	table of native soil N supply versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{ d}^{-1}$
$\rho(t)$	RECT	table of fertiliser N recovery versus days after transplanting (DAT)	g g^{-1}
$u_N(t)$	MXUP1T	table of maximum nitrogen uptake versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{ d}^{-1}$
	STPDAT	contingency cut-off date of N application (DAT)	d
A_{tot}	FERTMX	total fertiliser N input	kg N ha^{-1}
a	A	symmetry parameter of cumulative application curve	-
b	B	slope parameter of cumulative N application curve	d^{-1}
c	C	asymptote level of cumulative N application curve	kg N ha^{-1}
m	M	time shift parameter of cumulative application curve	d
K_{rat}	KNRAT	ratio of potassium to nitrogen in the crop (optional)	g g^{-1}

P_{rat}	PNRAT	ratio of phosphorus to nitrogen in the crop (optional)	g g^{-1}
------------------	-------	---	-------------------

(unit = as used in MANAGE-N).

(symbols as used in ten Berge et al., 1994c).

¹ See paragraph 'Description of the input for MANAGE-N' on page 269 for an example on the required format of these observations.

A12.4 Input for the 'Optimise N-curve' method

The following table is divided into three sections, corresponding to the three input screens of the 'Edit A Crop/Soil File' menu option. The order of the parameters and variables corresponds to the exact order in the input screens.

The main purpose of the 'Optimise-N curve' *method* is to find optimum values for the fertiliser N application curve-determining factors A, B, C, and M. This is executed for several user-defined N-input levels (FERTMX).

Table A12.4 Required input for the 'Optimise N-curve' *method* in MANAGE-N.

symbol	acronym	description	unit
	TREATM	character string sent to the output file, containing the title of the experiment and treatment	-
t_H	DATH	harvest date in days after transplanting (DAT)	d
t_{FF}	DATFF	date of first flowering in days after transplanting (DAT)	d
	WCRI	initial weight of the crop (shoot plus storage organs, roots)	kg ha ⁻¹
	ANCRI	initial amount of N in the crop	kg ha ⁻¹
	DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
f_{SV}	FSV1	site-variety match factor (before flowering)	-
	FSV2	site-variety match factor (after flowering)	-
f_N	FNCLV	fraction of total crop nitrogen present in leaves (pre-flowering)	g g ⁻¹
$c_{\max}(t)$	FNMAXT	table of FNMAX (maximum fraction of N in total crop biomass) versus days after transplanting (DAT)	g g ⁻¹
r_N	RUR	relative nitrogen uptake rate	d ⁻¹
q_N	NUPCO	maximum ratio of daily N uptake to growth	g g ⁻¹
$N_{L,\max}$	ANLVMX	maximum amount of nitrogen in leaves	kg ha ⁻¹
n_o	FNSO	fraction nitrogen in grains and panicles	g g ⁻¹
	HI5	value of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals 5 kg ha ⁻¹ / (MJ m ⁻²)	kg kg ⁻¹
	HISLOP	slope of the relation between harvest index and the ratio of biomass at first flowering to post-flowering cumulative radiation	MJ m ⁻² / (kg ha ⁻¹)
$S_N(t)$	SST	table of native soil N supply versus days after transplanting (DAT)	kg ha ⁻¹ d ⁻¹

$\rho(t)$	RECT	table of fertiliser N recovery versus days after transplanting (DAT)	g g^{-1}
$u_{\text{N}}(t)$	MXUPLT	table of maximum nitrogen uptake versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{ d}^{-1}$
t_{CR}	DATCR	days after transplanting at which critical crop mass check takes place	d
W_0	WCR0	intercept on Y-axis of crop weight at DATCR (WCRCR) versus crop yield (WRR) relation	kg ha^{-1}
W_s	WCRSLP	slope parameter of critical crop weight at DATCR (WCRCR) versus crop yield (WRR) relation	-
A_{tot}	FERTMX	total fertiliser N input	kg N ha^{-1}
a	A	symmetry parameter of cumulative application curve	-
b	B	slope parameter of cumulative N application curve	d^{-1}
c	C	asymptote level of cumulative N application curve	kg N ha^{-1}
m	M	time shift parameter of cumulative application curve	d
K_{rat}	KNRAT	ratio of potassium to nitrogen in the crop (optional)	g g^{-1}
P_{rat}	PNRAT	ratio of phosphorus to nitrogen in the crop (optional)	g g^{-1}

(unit = as used in MANAGE-N).

(symbols as used in ten Berge et al., 1994c).

A12.5 Input for the ‘Split Evaluation’ method

The table below is divided into three sections, corresponding to the three input screens of the ‘Edit A Crop/Soil File’ menu option. The order of the parameters and variables in the table corresponds to the exact order in the input screens. The purpose of this *method* is to evaluate discrete split dressings and their effect on yield.

Table A12.5 Required input for the ‘Split Evaluation’ *method* in MANAGE-N.

symbol	acronym	description	unit
	TREATM	character string sent to the output file, containing the title of the experiment and treatment	-
t_H	DATH	harvest date in days after transplanting (DAT)	d
t_{FF}	DATFF	date of first flowering in days after transplanting (DAT)	d
	WCRI	initial weight of the crop (shoot plus storage organs, roots)	kg ha ⁻¹
	ANCRI	initial amount of N in the crop (live and dead material)	kg ha ⁻¹
	SDLAGE	seedling age at transplanting, required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS	d
	DATFSV	date (in days after transplanting, DAT) at which FSV shifts from FSV1 to FSV2	d
f_{SV}	FSV1	site-variety match factor (before flowering)	-
	FSV2	site-variety match factor (after flowering)	-
	WCR_OBS	table of observed weight of the total crop (shoot plus roots) versus year and day of year ¹ , aboveground (if RTINCL = 0) or total (if RTINCL = 1). Used for comparisons with simulated values.	kg ha ⁻¹
	WLV_OBS	table of observed weight of the leaves versus year and day of year ¹ . Used for comparison with simulated leafmass.	kg ha ⁻¹
	FNLV_OBS	table of measured fraction (in %) of nitrogen in the leaves versus year and day of year ¹ . Used for comparison with simulated fractions.	g g ⁻¹
	RTINCL	switch to indicate whether root biomass was included in measured crop biomass WCR_OBS	-

	RSRT	table of RSR (root:shoot ratio) versus relative time (RELTIME), required to calculate simulated shoot from simulated total crop weight for comparison with observed crop weight if roots were not included in WCR_OBS.	g g^{-1}
f_N	FNCLV	fraction of total crop nitrogen present in leaves (pre-flowering)	g g^{-1}
$c_{\max}(t)$	FNMAXT	table of FNMAX (maximum fraction of N in total crop biomass) versus days after transplanting (DAT)	g g^{-1}
r_N	RUR	relative nitrogen uptake rate	d^{-1}
q_N	NUPCO	maximum ratio of daily N uptake to growth	g g^{-1}
n_D	FNSO	fraction nitrogen in grains and panicles	g g^{-1}
	HI5	value of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals $5 \text{ kg ha}^{-1} / (\text{MJ m}^{-2})$	kg kg^{-1}
	HISLOP	slope of the relation between harvest index and the ratio of biomass at first flowering to post-flowering cumulative radiation	$\text{MJ m}^{-2} / (\text{kg ha}^{-1})$
	K2T	table of K2 (proportionality factor relating N pool size to loss rate) versus days after transplanting (DAT)	d^{-1}
$S_N(t)$	SST	table of native soil N supply versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{ d}^{-1}$
$\rho(t)$	RECT	table of fertiliser N recovery versus days after transplanting (DAT)	g g^{-1}
$u_N(t)$	MXUP1T	table of maximum nitrogen uptake versus days after transplanting (DAT)	$\text{kg ha}^{-1} \text{ d}^{-1}$
A_{tot}	FERTMX	total fertiliser N input	kg N ha^{-1}
K_{rat}	KNRAT	ratio of potassium to nitrogen in the crop (optional)	g g^{-1}
P_{rat}	PNRAT	ratio of phosphorus to nitrogen in the crop (optional)	g g^{-1}
	SPLIT	table of sizes of discrete N split applications	kg ha^{-1}
	SPLDAT	table of days at which discrete N splits are applied (in DAT)	d

(unit = as used in MANAGE-N).

(symbols as used in ten Berge et al., 1994c).

¹ See paragraph 'Description of the input for MANAGE-N' on page 269 for an example on the required format of these observations.

A12.6 Description of the input for MANAGE-N

TREATM

The purpose of this parameter is to include a comment in the crop/soil file as well as in the result files of the simulation runs. This one-line comment typically lists basic characteristics of a treatment. For example:

```
TREATM = 'IRRI, 1991, wet site, IR72, 125 kg/ha'
```

DATH, DATFF

Harvest date (DATH) is a key date in MANAGE-N: crop growth simulation terminates when that date has passed. Another important date is first flowering (DATFF): in ORYZA_0 it signals the start of the remobilisation of nitrogen from the leaves to the panicles. DATH and DATFF are expressed in number of days after transplanting.

WCRI, ANCRI

WCRI is the weight of the rice crop at transplanting. The value cannot be zero. If the crop is not transplanted but sown, the simulation should start at 2 – 3 weeks after germination. SDLAGE and WCRI are then age and biomass of the crop at the start of the simulation. The initial amount of nitrogen in the rice crop is specified as ANCRI.

SDLAGE

This parameter denotes the age (in days after germination) of the seedlings at the time of transplanting. SDLAGE is only required when no roots have been measured as part of the crop biomass. It is then used to interpolate the RSRT table (expressed versus relative crop age) in calculating simulated shoot weight from simulated total crop weight, for comparison with observed shoot weight.

DATFSV, FSV1, FSV2

The site-variety factors FSV1 and FSV2 serve as matching factors accounting for local conditions and are obtained by calibration using the 'Fit to Match' *method* (see paragraph 5.4 on page 90). Although these parameters should have a value before the 'Fit to Match' *method* is executed, the user can enter default values (set DATFSV to 99., and both FSV

values to 1.0). MANAGE-N will compute appropriate values for these parameters during the 'Fit to Match' *method* for these parameters and will enter these automatically into the crop/soil file.

WCR_OBS, WLV_OBS, FNLV_OBS

Observed weight of the crop (WCR_OBS), observed weight of the leaves (WLV_OBS), and the observed fraction (in %) of nitrogen in the leaves (FNLV_OBS) are very important to the execution of the 'Fit to Match' *method*. The simulated weight of the crop is compared with the observed weight of the crop. This is done for different FSV values, and the best fitting value is selected (advised on screen by MANAGE-N).

For other *methods* (except 'Optimise N-curve'), the observed values can be used to provide measured values for comparison with simulated output. Also, in the 'Single Run' *method*, if the menu option 'Toggle SWINLV' has been used to put MANAGE-N in the 'use measured leaf nitrogen' mode, both WLV_OBS and FNLV_OBS will be used to replace simulated leaf nitrogen by measured leaf nitrogen (i.e., WLV_OBS and FNLV_OBS are used as forcing functions).

The following format must be used when entering observed values for these state variables:

```
WCR_OBS = 1991., 180., 16.0,
          1991., 194., 16.0,
          1991., 202., 49.3,
          1991., 219., 394.0,
          1991., 273., 6176.0,
          1991., 288., 7609.0
```

The user should make sure that there is an initial value that coincides with the starting day of the simulation. In all tables, the first column is the year, the second column specifies the exact day (day of year, DOY) of the observation, and the third value is the observed value (here of the crop biomass) itself.

RTINCL, RSRT

The RTINCL switch signals to ORYZA_0 whether (= 1) or not (= 0) roots were included in the measured weight of the crop (WCR_OBS). This parameter is 'hidden' in the input screens of MANAGE-N: a check button must be turned on or off, setting these values internally.

The root-shoot ratio table is an array with relative time (= 0 at seeding, = 1 at DATFF) in the first column and the ratio of roots to shoots in the second column. It is used to calculate

simulated shoot weight from simulated crop weight for comparison with observed crop weight if no roots were included (i.e., roots are not included in WCR_OBS). An example of RSRT input is:

```
RSRT =  0.,    0.4,
        1.2,    0.15
```

FNCLV, FNMAXT

The fraction of total crop nitrogen in the leaves in the pre-flowering phase (FNCLV) can be derived from observations as explained in Appendix 11.

Maximum N concentration in the mean crop biomass (FNMAXT) is entered as a table of values versus time. See also Appendix 11. For example, first column is number of days after transplanting DAT, the second column is the FNMAX value:

```
FNMAXT =  0.,    0.025,
          25.,    0.04
          70.,    0.020,
          100.,   0.015,
          130.,   0.010
```

RUR, NUPCO, ANLVMX

During the exponential growth phase, actual nitrogen uptake by the crop is limited by the relative nitrogen uptake rate r_N . Values up to 0.2 d^{-1} have been found so far.

The maximum ratio of daily N uptake to growth, q_N , is used to limit nitrogen uptake. Nitrogen cannot be absorbed at an infinite rate because N is assimilated into structures requiring carbon. So the rate of carbon assimilation is likely to have a limiting effect on potential N uptake. Values for NUPCO are often around $0.035 \text{ kg N kg}^{-1}$ dry matter, but values up to $0.05 \text{ kg N kg}^{-1}$ have been observed. The default value for this parameter in MANAGE-N equals to $0.05 \text{ kg N kg}^{-1}$.

The total amount of leaf nitrogen $N_{L,\max}$ (ANLVMX) usually does not exceed 100 kg N ha^{-1} (default value).

All these parameters need to be derived from actual field experiments (see Appendix 11).

FNSO

The fraction of nitrogen in the grain (including the panicle) varies among N application levels and cultivars. If no value is known (see also Appendix 11), MANAGE-N's default value of 0.012 can be used.

HIS, HISLOP

Harvest index (I_H) is defined as final grain yield (0% moisture) divided by the total crop biomass (including roots). The harvest index can be obtained in two ways: it is computed dynamically by MANAGE-N using a regression relation between observed harvest index and weight of the crop at first flowering (WCRFF) divided by the sum of radiation after first flowering (RDDSUM); or it is set at a fixed value.

In the first method which uses sets of existing data, the linear relation between observed I_H and WCRFF divided by RDDSUM is expressed by two parameters: HIS and HISLOP. HIS is the value of I_H at WCRFF / RDDSUM = 5 ($\text{kg ha}^{-1} / (\text{MJ m}^{-2})$). HISLOP is the slope of the relation between I_H and WCRFF / RDDSUM.

Defaults based on observed data from several sites in the Philippines, India, and China are HIS = 0.55 and HISLOP = $0.008 \text{ MJ m}^{-2} / (\text{kg ha}^{-1})$ in MANAGE-N.

After a simulation run, the I_H is calculated using the following formula:

$$HI = HIS - HISLOP * (-5. + WCRFF/RDDSUM)$$

The second method enables a fixed setting of the harvest index. HISLOP must be set to 0, and HIS must be set to the appropriate value of I_H for the given genotype and location.

The results obtained in SARP so far have shown that I_H is always about 0.50 for short-duration cultivars. Lower values (0.4 and below) may occur in long-duration cultivars; under conditions of low postflowering radiation; and under extreme N application patterns. I_H can be analysed easily for each experiment and this should be done as a standard practice. If problems occur during grain filling (cold, pests, low spikelet viability, low spikelet numbers), the occurrence of a low I_H serves as a first signal.

HIS and HISLOP are not used during the 'Fit to Match' *method*. However, it is required to enter some values.

K2T

The parameter K2 represents the relative loss rate (d^{-1}) of N from the fertiliser N pool in the soil. It is expressed as a function of days after transplanting (DAT). Unless specified otherwise, MANAGE-N uses a default of 0.1 throughout the whole simulation period:

$$K2T = \begin{matrix} 0., & 0.1, \\ 999., & 0.1 \end{matrix}$$

This parameter is used to calculate the size of the available N pool, and, from it, actual uptake. It is used only under the 'Split Evaluation' *method*.

First, the fertiliser N recovery at time t $\rho(t)$ is calculated via the RECT table. Then, the parameter K1 (proportionality factor relating N pool size to potential uptake) is calculated from the following relation:

$$\rho(t) = \frac{K1}{K2 + K1}$$

The actual uptake is the minimum of N demand and $(K1 * N \text{ pool})$.

SST, RECT, MXUP1T

During the season, the native soil N supply SST can vary. SST is therefore specified as a table of native soil N supply versus time (in days after transplanting, DAT). Usually it is sufficient to maintain a fixed value throughout the season (first column is days after transplanting, second column is the native soil N supply value):

```
SST = 0.,    0.6,
      999.,  0.6
```

Fertiliser-N recovery $\rho(t)$ is defined versus time in the table RECT. ORYZA_0 does not 'convert' a fraction $\rho(t)$ of N applied directly into uptake. Instead, a fraction $(1-\rho(t))$ is considered as unavoidable loss. So, $\rho(t)$ is a potential recovery. Some part of potentially recoverable N may be lost if the crop cannot absorb the whole fraction $\rho(t)$ at the time of application. An example of such a potential recovery table RECT (first column is time in days after transplanting DAT, second column is the value of fertiliser N recovery):

```
RECT = 0.,    0.0,
       40.,   0.7,
       70.,   0.8,
       75.,   0.8,
       85.,  0.01,
      135.,  0.
```

The maximum nitrogen uptake μ_N is strongly influenced by climate and soil. Also, during the growing season this variable may change. Like RECT and SST, maximum nitrogen uptake is defined as a time-coursed table MXUP1T. All the above parameters must be derived from actual field experiments (see Appendix 11).

DATCR, WCR0, WCRSLP

These optional parameters are used to check the validity of a newly found N-application curve, by inspecting whether the biomass reached at a certain critical date (DATCR) would allow to realise the calculated final yield. This check is based on the reasoning that the crop should have reached a certain minimum weight before a specified date, to allow a particular grain yield later on (at the end of the simulation). See also Appendix 11. By default this option is not enabled.

STPDAT

This parameter is optional: it defines a contingency stop of the N application at a certain date after transplanting. Using this stop, one can immediately cut off the fertiliser N supply in case of, for example, pest and diseases and study the effects this will have on final yield. No value should be supplied for STPDAT when a normal simulation run is required, i.e., with no such contingency stop in the N application. The STPDAT parameter may only be used during the 'Single Run' *method*.

FERTMX

The fertiliser N level is defined by A_{tot} (FERTMX). This parameter is the total N input level in the treatment. During the 'Optimise N-curve' *method*, there is no need to set this value specifically in the crop/soil data file. The user is referred to the menu option 'Set Optimisation Parameters' to choose a range of N levels for which optimisation should be run. MANAGE-N inserts the FERTMX values automatically into the data files.

When selecting N input levels, note that local conditions strongly determine the absolute maximum amount of fertiliser that is still beneficial to the crop. Negative effects due to too high N input (e.g., sterility, lodging, pest and diseases) are not taken into account in the optimisation procedure and must be accounted for *a priori* by the user in defining FERTMX (crop/soil file) and the 'N-levels to optimise' field in the 'Set Optimisation Parameters' window.

A, B, C, M

The shape of the continuous N application curve is determined by the parameters A, B, C, and M. Whenever the 'Single Run' *method* is issued to study model behaviour under optimised conditions, the values for these four parameters should first be determined by optimisation ('Optimise N-curve' *method*) for that particular case and FERTMX. There is no need to enter any real values for these parameters in the 'Optimise N-curve' *method*

because MANAGE-N computes them. Output values of these parameters are stored in a file that can be displayed using the 'View Optimisation Results' menu option.

KNRAT, PNRAT

These optional parameters are used to give an estimate of the minimum uptake requirement of potassium and phosphorus. KNRAT is defined as the minimum ratio of K uptake to N uptake. PNRAT is defined as the minimum ratio of P uptake to N uptake. With these parameters and the total amount of nitrogen in the crop at the end of the simulation, KURQMN (minimum potassium uptake required per season) and PURQMN (minimum phosphorus uptake required per season) are calculated.

SPLIT, SPLDAT

The size of the discrete splits and the time on which they are applied are determined by, respectively, SPLIT and SPLDAT. The first is a vector of split sizes, the second is a vector of days after transplanting on which those splits are applied. The number of splits must equal the number of application days. For example:

```
SPLIT = 50., 75., 75.  
SPLDAT = 20, 40, 65
```


Appendix 13 ORYZA_0 differences

This Appendix describes the differences between the versions of ORYZA_0 that are being used in the MANAGE-N tool and the one described in the SARP Research Proceedings booklet 'ORYZA simulation modules for potential and nitrogen limited rice production' (Drenth et al., 1994).

Changes between the 'old' version and the versions used by MANAGE-N are treated per module in the same order as that printed in Drenth et al. (1994). Only the significant modifications are outlined; small changes or beautifications to the source code are not discussed. Furthermore, special provisions for MANAGE-N (e.g., source code to communicate with optimisation routines) are not treated, for these fall outside the scope of this booklet.

Time and environment (TIMENV)

All equations related to temperature and temperature sums have been removed. Heat units are no longer calculated and total radiation has been deleted. This module only computes stop time (via DATH) and outputs daily global radiation now.

Biomass accumulation by the crop (BIOMS2)

Changed names of the variables FNLV and WLV, which are measured data, to FNLV_OBS and WLV_OBS respectively. In the 'old' version, when measured leaf nitrogen is used as a forcing function, ORYZA_0 still goes through the whole process of calculating nitrogen application (NAP0), crop N demand (DEM0), crop N uptake (NUP0), and nitrogen allocation (NAL0). Of course, this procedure is completely obsolete. In the new version, all these modules are skipped when observed leaf nitrogen is used as a forcing function, thus speeding up the calculation process.

Dry matter accumulation (GROW0)

Harvest index (HI) is now calculated using the relation between observed HI and the observed weight of the crop at first flowering (WCRFF) divided by the observed post-flowering radiation (RDDSUM). The parameter HI5 (value of harvest index where the ratio of biomass at first flowering to post-flowering cumulative radiation equals $5 \text{ kg ha}^{-1} / (\text{MJ m}^{-2})$) and HISLOP (the slope of the relation) are read from the crop/soil file to calculate HI from the simulated WCRFF and RDDSUM. Observed harvest index can be used when HISLOP is set to zero: supply the observed harvest index as HI5.

In the 'old' version of ORYZA_0, DATEF was used as the date (in days after transplanting) when flowering commenced. This parameter has been renamed into DATFF: first flowering in days after transplanting.

The parameter SDLAGE (seedling age) is read from the crop/soil file when roots are not included in the observed weight of the crop (used to calculate simulated shoot from simulated total crop weight for comparison with observed crop =shoot weight).

XXWCR has been renamed in the output file to WCR_OBS (observed weight of the crop, with or without roots depending on the value of RTINCL).

N uptake: the fertiliser application curve, recovery and soil N supply (NAP0)

Native soil N supply is often not constant through the growing season. Hence, it must now be supplied as a table versus time (in DAT). Every day, ORYZA_0 interpolates between the existing measured (or derived) values to get an actual native soil N supply value for that day. The table is called SST (soil supply table), the interpolated value is called SOLSUP. In the new version, it is possible to cut off the nitrogen application (in case of a contingency): supply a value (in DAT) for STPDAT and after that date, no fertiliser nitrogen will be applied any more (i.e., APSLOP=0). If normal application (i.e., no contingency cut-off) is preferred, then no STPDAT must be specified in the crop/soil file. A number of new output variables have been added for better information regarding nitrogen dynamics. One of them is the actual N recovery (ACTREC). Another additional output is the cumulative native soil N supply CUMNSS, a variable that was introduced to help calculate actual recovery:

$$\text{ACTREC} = (\text{ANCR} - \text{CUMNSS}) / \text{NAPPLD}$$

ANCR is the total amount of nitrogen in the crop and NAPPLD is the total amount of fertiliser nitrogen that has been applied. At the end of the simulation, CUMNSS is the amount of nitrogen that the soil has supplied until seven days prior to harvest (there is no nitrogen uptake during the last week).

Nitrogen uptake: demand (DEM0)

The maximum daily N uptake rate (MAXUP1) is now computed using a time-coursed table (MXUP1T), since these uptake rates might change during phenological stages. Demand during the exponential growth phase has been redefined to also take the uptake limitation MAXUP1 into account. Minimum uptake requirements for potassium (KUMNRQ) and phosphorus (PUMNRQ) are computed at the end of the simulation, using the ratio of potassium (PNRAT) and phosphorus (PNRAT) to nitrogen in the crop. Both are optional

outputs: if the ratios are not available in the crop/soil file, then the minimum uptake requirements for these elements are not calculated.

Actual nitrogen uptake (NUP0)

The actual fraction of weight of the crop that consists of nitrogen (FNACT) is now calculated every day and written to the result file:

$$\text{FNACT} = \text{ANCR} / \text{WCR}$$

Allocation and redistribution of nitrogen (NAL0)

No changes in this module.

Support of discrete N-splits (NAP0SP)

A special provision made for MANAGE-N that is worth mentioning here is the modification of the nitrogen application module (NAP0) to use discrete N dressings (NAP0SP) instead of a continuous fertiliser N-application. MANAGE-N uses a version of ORYZA_0 with this module for the 'Split Evaluation' *method*. Because it is so completely different from the module described in Drenth et al. (1994), it is listed here in full. Paragraph 5.6 on page 97 describes the theoretical background of 'Split Evaluation'. When this module is used, BIOMS2 also has to be changed to call the correct new module (NAP0SP). Note that the NAP0SP module does not require, as opposed to NAP0, the intermediate variable X but does require NUPT in the argument list. The complete FST source code of the ORYZA_0 version that supports split-dose fertiliser N applications is listed in Appendix 14.

```
*-----*
* SUBROUTINE NAP0SP                                *
* *                                                  *
* * Author : SARP                                  *
* * Date   : May 1995                              *
* * Version: 1.0                                    *
* *                                                  *
* * Research Institute for Agrobiological and Soil Fertility (AB-DLO), *
* * P.O.Box 14, 6700 AA Wageningen, The Netherlands *
* *                                                  *
* * International Rice Research Institute, P.O. Box 933, *
* * 1099 Manila, The Philippines and *
* *                                                  *
* * Department of Theoretical Production Ecology, P.O. Box 430, *
* * 6700 AK Wageningen, The Netherlands. *
* *                                                  *
* * Purpose: Applies discrete quantities of N at set times to the *
* *           rice-crop system. *
* *                                                  *
```

```

* FORMAL PARAMETERS: (I=input, O=output, C=control, IN=init, T=time) *
* name type meaning units class *
* ----
* ITASK I4 Task that subroutine should perform - I *
* IUNITD I4 Unit that can be used for input files - I *
* IUNITL I4 Unit used for log file - I *
* FILEI1 C* Name of input file no. 1 - I *
* OUTPUT L4 Flag to indicate if output should be done - I *
* DELT R4 Time step of integration d I *
* DAT R4 Day after transplanting d I *
* NUPT R4 N uptake rate kg/ha/d I *
* ANCR R4 Amount of N in the crop kg/ha I *
* APSLOP R4 Daily fertilizer N application kg/ha/d O *
* RECOV R4 Best attainable recovery kg/kg O *
* NAVAIL R4 Total N availability (soil and fertilizer) kg/ha/d O *
* APCUM R4 Cumulative N application curve kg/ha O *
* NAPPLD R4 Total amount of N applied kg/ha O *
*
*
* Warnings : none *
* Subprograms called: none *
* File usage : IUNITD, IUNITL *
* Libraries used : TTUTIL *
*
*-----*
SUBROUTINE NAPOSP (ITASK , IUNITD, IUNITL, FILEI1,
& OUTPUT, DELT , DAT, NUPT , ANCR,
& APSLOP, RECOV, NAVAIL, APCUM, NAPPLD)

IMPLICIT REAL (A-Z)

* Formal parameters
INTEGER ITASK , IUNITD, IUNITL
LOGICAL OUTPUT
CHARACTER*(*) FILEI1
REAL DELT
REAL DAT, NUPT

* State variables, initial values and rates
REAL APCUM , APCUMI, APSLOP, NPOOL, NLOSS, NLOS

* Model parameter
REAL FERTMX, SUMSPL, K2, K1, DATH

* Auxiliary variables
REAL NAVAIL, RECOV , NAPPLD, NLOST, SOLSUP, CUMNSS, ACTREC
INTEGER ICNT, IDCNT

* AFGEN functions
*-----N-recovery table
REAL RECT
INTEGER IMRECT, ILRECT
PARAMETER (IMRECT = 40)
DIMENSION RECT (IMRECT)
*-----Table of N-split quantities [kg N/ha]
INTEGER IMNRST, ILNRST
REAL SPLIT
PARAMETER (IMNRST = 15)
DIMENSION SPLIT (IMNRST)
*-----Table of Days After Transplanting (DAT's) on which
* discrete N-appliance occurs.
INTEGER SPLDAT, ILDAST
DIMENSION SPLDAT(IMNRST)
*-----Table of DAT vs. proportional N-loss
REAL K2T
INTEGER IMK2T, ILK2T
PARAMETER (IMK2T = 50)
DIMENSION K2T (IMK2T)

```

```

*-----Table of DAT vs. native nitrogen soil supply
REAL SST
INTEGER IMSST, ILSST
PARAMETER (IMSST = 40)
DIMENSION SST (IMSST)

*
*   Used functions
REAL LINT, INTGR2

SAVE

IF (ITASK.EQ.1) THEN

*=====
*                               Initialization section
*=====
*
*   Open input file
CALL RDINIT (IUNITD, IUNITL, FILEI1)

*
*   Read initial states
CALL RDSREA ('APCUMI',APCUMI)

*
*   Read model parameters
CALL RDSREA ('FERTMX',FERTMX)
CALL RDSREA ('DATH ',DATH )

*
*   Read AFGEN functions
CALL RDAREA ('RECT ',RECT ,IMRECT,ILRECT)
CALL RDAREA ('SST ',SST ,IMSST,ILSST )
CALL RDAREA ('SPLIT ',SPLIT ,IMNRST,ILNRST)
CALL RDAINT ('SPLDAT',SPLDAT,IMNRST,ILDAST)
CALL RDAREA ('K2T' ,K2T ,IMK2T ,ILK2T )

CLOSE (IUNITD)

*
*   initially known variables to output
CALL OUTDAT (2,0,'FERTMX',FERTMX)

*
*   Initialize state variables
NPOOL = 0.
NLOS = 0.
APCUM = APCUMI
CUMNSS = 0.

*
*   Initialise rate variables
NUPT = 0.
NLOSS = 0.

*
*   Initialise auxiliary variables
SUMSPL = 0.
K2 = 0.
IDCNT = 1

*-----Check consistency of arrays
IF (ILNRST.NE. ILDAST) CALL ERROR ('NAP0S',
& ' Number of N-gifts does not equal number of days.')

DO 10 ICNT = 1,ILNRST
SUMSPL = SUMSPL + SPLIT(ICNT)
10 CONTINUE

WRITE (*, '(A,F9.2,A)') ' Sum of splits = ', SUMSPL,
& ' N kg/ha'
IF (SUMSPL.EQ. FERTMX) THEN
WRITE (*, '(A, F9.2)') ' This equals FERTMX = ', FERTMX
ELSE
WRITE (*, '(A, F9.2)')

```

```

&          ' This is not equal to FERTMX. Difference = ',
&          (FERTMX - SUMSPL)
ENDIF

ELSE IF (ITASK.EQ.2) THEN

*=====
*          Rate calculation section
*=====

*-----Fertilizer N gift (kg N ha-1)

      IF (SPLDAT(IDCNT) .EQ. INT(DAT)) THEN
        APSLOP = SPLIT(IDCNT)
        IF (IDCNT .LT. ILDAST) IDCNT = IDCNT + 1
      ELSE
        APSLOP = 0.
      ENDIF

*-----Best attainable recovery (kg uptake/kg applied)

      RECOV = LINT(RECT,ILRECT,DAT)

*-----Potential proportional N-loss with regard to uptake.

      K2 = LINT(K2T, ILK2T, DAT)
      K1 = (RECOV / (1. - RECOV)) * K2

*-----Native N soil supply

      SOLSUP = LINT(SST ,ILSST ,DAT)

*-----Total N availability (kg N ha-1 d-1)

      NAVAIL = SOLSUP + MAX(0., (K1 * NPOOL))

*-----N-loss from the nitrogen pool (kg N d-1)

      NLOSS = MAX(0., K2 * NPOOL)

*****
*-----Output section

      IF (OUTPUT) THEN

        CALL OUTDAT (2,0,'SOLSUP ',SOLSUP )
        CALL OUTDAT (2,0,'CUMNSS ',CUMNSS )
        CALL OUTDAT (2,0,'APSLOP ',APSLOP )
        CALL OUTDAT (2,0,'RECOV  ',RECOV  )
        CALL OUTDAT (2,0,'NAVAIL  ',NAVAIL )
        CALL OUTDAT (2,0,'APCUM   ',APCUM  )
        CALL OUTDAT (2,0,'NPOOL   ',NPOOL  )
        CALL OUTDAT (2,0,'NLOSS   ',NLOSS  )
        CALL OUTDAT (2,0,'K1      ',K1      )
        CALL OUTDAT (2,0,'K2      ',K2      )

      END IF

*****

      ELSE IF (ITASK.EQ.3) THEN

*=====
*          Integration section
*=====

*-----Cumulative N application

      APCUM = INTGR2 (APCUM, APSLOP, DELT, FILEI1, 'APCUM')

*-----Nitrogen pool and nitrogen loss

      NPOOL = INTGR2 (NPOOL, (APSLOP - MAX(0., (NUPT - SOLSUP)) -

```

```

&      NLOSS), DELT, FILEI1, 'NPOOL')
      NLOS  = INTGR2 (NLOS , NLOSS, DELT, FILEI1, 'NLOS')
*----Cumulative N soil supply (last 7 days no N-uptake)
      IF (DATH - DAT - 7.0 .GE. 0.)
&      CUMNSS = INTGR2 (CUMNSS,SOLSUP,DELT, FILEI1, 'CUMNSS')
*****

      ELSE IF (ITASK.EQ.4) THEN

*=====
*                               Terminal section                               *
*=====
*-----Total amount N applied (kg N ha-1)
      NAPPLD = APCUM

*-----Total amount of fertiliser lost (kg N ha-1)
      NLOST  = NLOS

*-----Actual recovery during the season ( - ), only if
*      nitrogen was applied.
      IF (NAPPLD .NE. 0.) THEN
        ACTREC = (ANCR - CUMNSS) / NAPPLD
      ELSE
        ACTREC = 0.
      ENDIF

*-----Terminal output
      CALL OUTDAT (2,0,'NAPPLD', NAPPLD)
      CALL OUTDAT (2,0,'NLOST',  NLOST)
      CALL OUTDAT (2,0,'ACTREC',  ACTREC)
*****

      END IF

      RETURN
      END

```


Appendix 14 FST source of the ORYZA_0 model

On the distribution disks of MANAGE-N, a directory ORYZA_0.FST exists. In this directory, the user can find the FST version of ORYZA_0 (file name ORYZA_0.FST) that is used in MANAGE-N. Furthermore, the FST version of the split-dose ORYZA_0 version (file name ORY0SPL.FST) is provided. Sample data files are also provided, in this case for the 210 kg N treatment of an experiment performed in Jinhua, China, 1993 (Zhiming et al., 1995). The FST versions of ORYZA_0 are not automatically installed when you execute the installation procedure of MANAGE-N: the user must manually copy all the files from the ORYZA_0.FST directory to a new location before running or modifying it. The first section of this Appendix lists the FST source code of the ORYZA_0 model proper (continuous fertiliser N application); the second section lists the FST source code of the split-dose version of ORYZA_0.

The FST source of ORYZA_0: continuous fertiliser N application

```
DEFINE_CALL RDDCAL(INPUT, INPUT, INPUT, INPUT, INPUT, OUTPUT, OUTPUT)
*-----*
*
*                                ORYZA0 (FST)
*
*                                SARP
*                                November 1995
*                                Version: 3.0
*
*
* Research Institute for Agrobiolgy and Soil Fertility (AB-DLO),
* P.O.Box 14, 6700 AA Wageningen, The Netherlands
*
* International Rice Research Institute, P.O. Box 933,
* 1099 Manila, The Philippines and
*
* Department of Theoretical Production Ecology, P.O. Box 430,
* 6700 AK Wageningen, The Netherlands.
*
*
* Purpose: Module for simulation of rice production under different
* N application levels and patterns. This FST version equals
* the version of ORYZA_0 that is used in MANAGE-N, with two
* exceptions: a contingency cut-off of the nitrogen
* application (via STPDAT) has not been build in, and there
* is no provision that indicates whether measured biomass
* includes roots or not (via RTINCL). Actions taken linked to
* the RTINCL switch are of course also not implemented in this
* FST version.
* For full documentation of parameters and variables, refer
* to the documentation of MANAGE-N (see References).
*
* References: Riethoven, J.J.M., H.F.M. ten Berge & H. Drenth (Eds),
* 1995. Software developments in the SARP project: a guide
* to applications and tools. SARP Research Proceedings.
* AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Banos:
* 320 pp.
*
* Libraries used      : TTUTIL, WEATHER, DRIVERS
*-----*
TITLE ORYZA_0 FOR N LIMITED PRODUCTION
INITIAL
```

```

*****
***** initial state variables *****
*****
INCON WCRI = 405.1
INCON ANCRI = 6.28
INCON APCUMI= 0.
INCON RADTOI= 0.
INCON CMNSSI= 0.
ANLVI      = FNCLV*ANCRI

*****
***** time parameters *****
*****
PARAMETER DATFF = 40.
PARAMETER DATH  = 91.
PARAMETER DATFSV= 49.

*****
***** site-season-variety match factor *****
*****
PARAMETER FSV1 = 0.87
PARAMETER FSV2 = 0.67

*****
***** crop parameters *****
*****
PARAMETER P      = 10.
PARAMETER EPSIL  = 2.5
PARAMETER MAXNCR = 35.
PARAMETER MAXDAT = 20.

PARAMETER FNSO   = 0.0159
PARAMETER FNCLV  = 0.473
PARAMETER ANLVMX= 100.
PARAMETER RUR    = 0.149
PARAMETER NUPCO  = 0.03

PARAMETER HI5    = 0.55
PARAMETER HISLOP= 0.008

FUNCTION FNMAXT = 0. ,0.0156,...
                21., 0.026,...
                49., 0.021,...
                70., 0.017,...
                91., 0.0152

*****
***** soil parameters *****
*****
FUNCTION SST    = 0. , 0.980,...
                999., 0.980

FUNCTION RECT    = 0., 0.2,...
                25.,0.85,...
                50.,0.85,...
                75.,0.,...
                91.,0.

FUNCTION MXUP1T = 0. , 7.9,...
                999., 7.9

*****
***** N application parameters *****
*****
PARAMETER FERTMX = 210.
PARAMETER A      = 1.
PARAMETER B      = 0.1
PARAMETER C      = 300.
PARAMETER M      = 30.

```



```

*****
***** K and P fertiliser parameters *****
*****

*---- When computation of minimum potassium and phosphorus
*      fertiliser requirements are not needed, then supply
*      dummy value of -99.
*      Otherwise, supply ratio of potassium (K) or phosphorus (P)
*      to nitrogen in the crop [g/g]
PARAMETER KNRAT = -99.
PARAMETER PNRAT = -99.

*****
***** run control *****
*****

TRANSLATION_FSE
TIMER STTIME =206., FINTIM=400., DELT = 1., PRDEL = 1.
FINISH TIME > DATEH
PRINT   DAT, WCR, XXWCR, P, GCR, ANLV, ANCR, LNUC, GRUC,...
        RNEFF, RADTOT, APCUM, CUMNSS, RDDSUM, HI, WRR,...
        NAPPLD, NHRVST, FNMAX, FERTMX, RECOV, NAVAIL, NUPT,...
        DEMAND, MAXUP0, MAXUP1, MAXUP2, MAXUP3, MAXUP4, MAXUP5,...
        ANSOCH, ANLVCH,...
        FNACT, ACTREC, PURQMN, KURQMN, FSV

PARAMETER TINY = 0.001, LARGE = 100.
* OPTION: set SWINLV to zero for simulated leaf N; to 1 for
* observed N% and leaf mass values as forcing function NL
PARAMETER SWINLV = 0.

*****
***** weather variables *****
*****
WEATHER WDIR='C:\SYS\WEATHER\', CNTR='JINHUA', IYEAR=1993, ISTN=1

*****
***** DYNAMIC *****
*****

***** time *****
*****
DAT  = TIME - STTIME
DATEH = STTIME + DATH - 1.

*****
***** radiation (MJ m-2 d-1) and *****
***** radiation use efficiency (g dry matter MJ-1) *****
*****
*-----radiation RDT in MJ m-2 d-1
RDT  = RDD*1.E-06
*-----cumulative radiation
RADTOT = INTGRL(RADTOI, RDT)

*****
***** crop growth *****
*****
*-----leaf nitrogen (g N per m2 ground surface)
*-----measured leaf N
NL1  = MAX(0., ANLV/10.)
*-----simulated leaf N
NL2  = MAX(0., (AFGEN(XNLVT,DAT)/100.)*AFGEN(XNLVT,DAT)/10.)
*-----leaf N as used in this model
NL   = INSW(0.5-SWINLV, NL2, NL1)

*-----FSV value may change at flowering: FSV1 for pre-,
*      FSV2 for post-flowering

```

```

FSV   = INSW(DAT-DATFSV, FSV1, FSV2)

*-----Growth per unit rad and unit leaf N (g g-1/(MJ m-2))
RNEFF = FSV*((P/(RDD*1.E-06))*(1.-EXP(-EPSIL*(RDD*1.E-06)/(P*NL+TINY))))

*-----Output only: overall leaf N use efficiency (g g-1 d-1)
LNUC  = RNEFF*(RDD*1.E-06)

*-----Output only: overall radiation use efficiency (g MJ-1)
GRUC  = RNEFF*NL

*-----Crop growth rate (with conversion to kg ha-1 d-1)
GCR   = 10.* RNEFF * (RDD*1.E-06) * NL

WCR   = INTGRL(WCRI, GCR)

*-----Observed crop biomass, for comparison only.
XXWCR = AFGEN(XWCRT, DAT)

CALL RDDCAL(WCR, RDT, DAT, DATFF, DELT, WCRFF, RDDSUM)

*****
***** N supply per day (kg N/ha.d) *****
*** derivative of logistic application curve plus soil N ***
*****

*-----N supply per day (kg N/ha.d),
* derivative of logistic application curve plus soil N
X      = EXP(-B*(DAT-M))

*-----Daily fertiliser N application (kg N ha-1 d-1)
APSLOP = MIN(FERTMX-APCUM, B*C*X*(1.+ A*X)**(-1.-1./A))

*-----Best attainable recovery (kg uptake/kg applied)
RECOV  = AFGEN(RECT,DAT)

*-----Native N soil supply (kg N ha-1 d-1)
SOLSUP = AFGEN(SST,DAT)

*-----Total N availability (kg N ha-1 d-1)
NAVAIL = RECOV*APSLOP + SOLSUP

*-----Cumulative applied amount of nitrogen (kg ha-1)
APCUM  = INTGRL(APCUMI, APSLOP)

*-----Cumulative native soil N supply upto one week before harvest
NSS    = INSW(DATH-DAT-7.0, 0., SOLSUP)
CUMNSS = INTGRL(CMNSSI, NSS)

*****
***** uptake limitations MAXUP due to limited demand *****
***** all in kg N/(ha.d) *****
*****

*-----max uptake during exponential phase
SWIEXP = REAAND(MAXNCR-ANCR, MAXDAT-DAT)
MAXUP0 = RUR*ANCR

*-----Maximum absolute uptake rate
MAXUP1 = AFGEN(MXUP1T, DAT)

*-----Uptake as limited by max fraction N per unit new dry
MAXUP2 = NUPCO * GCR

*-----Uptake as limited by max overall fraction of N in total
* existing biomass
FNMAX  = AFGEN(FNMAXT,DAT)
MAXUP3 = ((WCR+GCR*DELT)*FNMAX - ANCR)/DELT

```

*-----leaf nitrogen fraction (kg kg-1) in percent vs DAT

```
FUNCTION XNLVT = 0., 2.35,...
                7., 2.58,...
                14., 2.83,...
                21., 2.97,...
                28., 2.77,...
                35., 2.67,...
                42., 2.60,...
                49., 2.40,...
                56., 2.35,...
                63., 2.22,...
                70., 1.91,...
                77., 1.68,...
                84., 1.55,...
                91., 1.45
```

*-----total leaf mass (kg ha-1) vs DAT

```
FUNCTION XWLVT = 0., 149.2,...
                7., 322.1,...
                14., 777.7,...
                21., 1522.8,...
                28., 2476.6,...
                35., 2621.5,...
                42., 2886.3,...
                49., 2770.1,...
                56., 2667.4,...
                63., 2600.9,...
                70., 2485.1,...
                77., 2421.1,...
                84., 2388.5,...
                91., 2388.1
```

*-----total crop biomass (kg ha-1) vs DAT, for comparison only.

```
FUNCTION XWCRT = 0., 405.1,...
                7., 831.4,...
                14., 1790....
                21., 3449....
                28., 5411....
                35., 6529....
                42., 8257....
                49., 9916....
                56., 10793....
                63., 12364....
                70., 14014....
                77., 15225....
                84., 15618....
                91., 15615.
```

*-----For reruns, supply modified sets of input parameters here.

END

PARAMETER A= 10., B=0.1, C=300., M=30.

END

STOP

```
* RDDCAL computes intermediate variables that are required *
* for the calculation of harvest index from the *
* relation between observed harvest index and the *
* ratio of biomass at first flowering to post- *
* flowering cumulative radiation. *
* *
* IN WCR - weight of the crop, biomass (kg ha-1) *
* RDT - daily global radiation (MJ m-2 d-1) *
* DAT - days after transplanting *
* DATFF - first flowering in days after *
* transplanting *
```

```

*      DELT      - timestep of integration                      *
* INOUT WCRFF    - weight of the crop at first                 *
*               flowering (kg ha-1)                            *
*      RDDSUM    - post-flowering cumulative radiation          *
*****
      SUBROUTINE RDDCAL (WCR, RDT, DAT, DATFF, DELT, WCRFF, RDDSUM)

      REAL WCR, RDT, DAT, DATFF, WCRFF, RDDSUM, DELT
      LOGICAL AFFLOW

      REAL INTGRL

      IF (DAT.EQ.0.) THEN
        RDDSUM = 0.
        AFFLOW = .FALSE.
      ENDIF

*-----Crop weight at first flowering (kg ha-1)
      IF ((DAT.GE.DATFF).AND.(.NOT.AFFLOW)) THEN
        AFFLOW = .TRUE.
        WCRFF = WCR
      ENDIF

*-----Radiation sum upto flowering
      IF (AFFLOW) RDDSUM = INTGRL(RDDSUM, RDT, DELT)

      RETURN
      END

```

The FST source of ORY0SPL: split-dose fertiliser N application

```

DEFINE_CALL RDDCAL (INPUT, INPUT, INPUT, INPUT, INPUT, OUTPUT, OUTPUT)
DEFINE_CALL SPLDOS (INPUT, OUTPUT)
*-----*
*
*               ORY0SPL (FST)
*
*               SARP
*               November 1995
*               Version: 3.0
*
*
* Research Institute for Agrobiological and Soil Fertility (AB-DLO),
* P.O.Box 14, 6700 AA Wageningen, The Netherlands
*
* International Rice Research Institute, P.O. Box 933,
* 1099 Manila, The Philippines and
*
* Department of Theoretical Production Ecology, P.O. Box 430,
* 6700 AK Wageningen, The Netherlands.
*
*
* Purpose: Module for simulation of rice production under different
* discrete N split-dose schemes. This FST version equals
* the version of ORYZA_0 that is used in MANAGE-N under the
* 'Split Evaluation' method, with one exception: there
* is no provision that indicates whether measured biomass
* includes roots or not (via RTINCL). Actions taken linked to
* the RTINCL switch are of course also implemented in this
* FST version.
* For full documentation of parameters and variables, refer
* to the documentation of MANAGE-N (see References).
*
* References: Riethoven, J.J.M., H.F.M. ten Berge & H. Drenth (Eds),
* 1995. Software developments in the SARP project: a guide

```

```

*          to applications and tools. SARP Research Proceedings.      *
*          AB-DLO/Wageningen, WAU-TPE/Wageningen, IRRI/Los Banos:    *
*          320 pp.                                                     *
*
* Libraries used      : TTUTIL, WEATHER, DRIVERS                      *
*-----*
TITLE ORYZA_0 FOR N LIMITED PRODUCTION, DISCRETE N SPLIT-DOSE VERSION
INITIAL

*****
***** initial state variables *****
*****
INCON WCRI  = 405.1
INCON ANCRI = 6.28
INCON APCUMI= 0.
INCON RADTOI= 0.
INCON CMNSSI= 0.
INCON NPOOLI= 0.
INCON NLOSI = 0.
ANLVI      = FNCLV*ANCRI

*****
***** time parameters *****
*****
PARAMETER DATFF = 40.
PARAMETER DATH  = 91.
PARAMETER DATFSV= 49.

*****
***** site-season-variety match factor *****
*****
PARAMETER FSV1 = 0.87
PARAMETER FSV2 = 0.67

*****
***** crop parameters *****
*****
PARAMETER P      = 10.
PARAMETER EPSIL = 2.5
PARAMETER MAXNCR = 35.
PARAMETER MAXDAT = 20.

PARAMETER FNSO   = 0.0159
PARAMETER FNCLV  = 0.473
PARAMETER ANLVMX= 100.
PARAMETER RUR    = 0.149
PARAMETER NUPCO  = 0.03

PARAMETER HI5    = 0.55
PARAMETER HISLOP= 0.008

FUNCTION FNMAXT = 0. ,0.0156,...
                21., 0.026,...
                49., 0.021,...
                70., 0.017,...
                91., 0.0152

*****
***** soil parameters *****
*****
FUNCTION SST    = 0. , 0.980,...
                999., 0.980

FUNCTION RECT   = 0., 0.2,...
                25.,0.85,...
                50.,0.85,...
                75.,0,...
                91.,0.

FUNCTION MXUP1T = 0. , 7.9,...

```

```

999., 7.9

FUNCTION K2T      = 0. , 0.1,...
999., 0.1

*****
***** N application parameters *****
*****
PARAMETER FERTMX = 210.

*----Split-dose fertiliser N application is defined in
*   subroutine SPLDOS. See at the end of this program.

*****
***** K and P fertiliser parameters *****
*****

*---- When computation of minimum potassium and phosphorus
*   fertiliser requirements are not needed, then supply
*   dummy value of -99.
*   Otherwise, supply ratio of potassium (K) or phosphorus (P)
*   to nitrogen in the crop [g/g]
PARAMETER KNRAT = -99.
PARAMETER PNRAT = -99.

*****
***** run control *****
*****
TRANSLATION_FSE
TIMER STTIME =206., FINTIM=400., DELT = 1., PRDEL = 1.
FINISH TIME > DATEH
PRINT      DAT, WCR, XXWCR, P, GCR, ANLV, ANCR, LNUC, GRUC,...
           RNEFF, RADTOT, APCUM, CUMNSS, RDDSUM, HI, WRR,...
           NAPPLD, NHRVST, FNMALX, FERTMX, RECOV, NAVAIL, NUPT,...
           DEMAND, MAXUP0, MAXUP1, MAXUP2, MAXUP3, MAXUP4, MAXUP5,...
           ANSOCH, ANLVCH, NPOOL, NLOS, NLOST, K2, K1,...
           FNACT, ACTREC, PURQMN, KURQMN, FSV

PARAMETER TINY = 0.001, LARGE = 100.
*   OPTION: set SWINLV to zero for simulated leaf N; to 1 for
*   observed N% and leaf mass values as forcing function NL
PARAMETER SWINLV = 0.

*****
***** weather variables *****
*****
WEATHER WTRDIR='C:\SYS\WEATHER\' , CNTR='JINHUA' , IYEAR=1993, ISTN=1

*****
*****
***** DYNAMIC *****
*****

***** time *****
*****
DAT      = TIME - STTIME
DATEH    = STTIME + DATEH - 1.

*****
***** radiation (MJ m-2 d-1) and *****
***** radiation use efficiency (g dry matter MJ-1) *****
*****
*-----radiation RDT in MJ m-2 d-1
RDT      = RDD*1.E-06
*-----cumulative radiation
RADTOT   = INTGRL(RADTOI, RDT)

```

```

*****
***** crop growth *****
*****
*-----leaf nitrogen (g N per m2 ground surface)
*-----measured leaf N
NL1  = MAX(0., ANLV/10.)
*-----simulated leaf N
NL2  = MAX(0., (AFGEN(XNLVT,DAT)/100.)*AFGEN(XWLVT,DAT)/10.)
*-----leaf N as used in this model
NL   = INSW(0.5-SWINLV, NL2, NL1)

*-----FSV value may change at flowering: FSV1 for pre-,
*       FSV2 for post-flowering
FSV  = INSW(DAT-DATFSV, FSV1, FSV2)

*-----Growth per unit rad and unit leaf N (g g-1/(MJ m-2))
RNEFF = FSV*{(P/(RDD*1.E-06))*(1.-EXP(-EPSIL*(RDD*1.E-06)/(P*NL+TINY)))}

*-----Output only: overall leaf N use efficiency (g g-1 d-1)
LNUC  = RNEFF*(RDD*1.E-06)

*-----Output only: overall radiation use efficiency (g MJ-1)
GRUC  = RNEFF*NL

*-----Crop growth rate (with conversion to kg ha-1 d-1)
GCR   = 10.* RNEFF * (RDD*1.E-06) * NL

WCR   = INTGRL(WCRI, GCR)

*-----Observed crop biomass, for comparison only.
XXWCR = AFGEN(XWCRT, DAT)

CALL RDDCAL(WCR, RDT, DAT, DATFF, DELT, WCRFF, RDDSUM)

*****
***** N supply per day (kg N/ha.d) *****
*****

*-----Apply split-dose fertiliser N
CALL SPLDOS(DAT, APSLOP)

*-----Best attainable recovery (kg uptake/kg applied)
RECOV = AFGEN(RECT,DAT)

*-----Potential proportional N-loss with regard to uptake.
K2     = AFGEN(K2T, DAT)
K1     = (RECOV / (1. - RECOV)) * K2

*-----Native N soil supply (kg N ha-1 d-1)
SOLSUP = AFGEN(SST,DAT)

*-----Total N availability (kg N ha-1 d-1)
NAVAIL = SOLSUP + MAX(0., (K1 * NPOOL))

*-----N-loss from the nitrogen pool (kg N d-1)
NLOSS  = MAX(0., K2 * NPOOL)

*-----Cumulative applied amount of nitrogen (kg ha-1)
APCUM  = INTGRL(APCUMI, APSLOP)

*-----Nitrogen pool and nitrogen loss
DNPOOL = APSLOP - MAX(0., (NUPT - SOLSUP)) - NLOSS
NPOOL  = INTGRL (NPOOLI, DNPOOL)
NLOS   = INTGRL (NLOSI , NLOSS)

*-----Cumulative native soil N supply upto one week before harvest

```



```

NAPPLD = APCUM

*-----Total amount of fertiliser lost (kg N ha-1)
NLOST = NLOS

*-----total N uptake at harvest (kg N ha-1)
NHRVST = ANCR

*-----Actual recovery during the season ( - ), only if
*         nitrogen was applied. 0 if no nitrogen applied.
ACTREC = INSW(-NAPPLD, 1., 0.) * (ANCR - CUMNSS) / NOTNUL(NAPPLD)

*-----Calculate and output minimum Phosphorus (P) requirements [kg ha-1]
*         based on P/N ratio.
PURQMN = INSW(PNRAT, -99., PNRAT * ANCR)

*-----Calculate and output minimum Potassium (K) requirements
*         [kg ha-1] based on K/N ratio.
KURQMN = INSW(KNRAT, -99., KNRAT * ANCR)

*****
* these functions are used only if SWITCH = 1:
* NL as measured forcing function
* these specific values are to be filled out per treatment !!!
*****

*-----leaf nitrogen fraction (kg kg-1) in percent vs DAT
FUNCTION XNLVT = 0., 2.35,...
                7., 2.58,...
                14., 2.83,...
                21., 2.97,...
                28., 2.77,...
                35., 2.67,...
                42., 2.60,...
                49., 2.40,...
                56., 2.35,...
                63., 2.22,...
                70., 1.91,...
                77., 1.68,...
                84., 1.55,...
                91., 1.45

*-----total leaf mass (kg ha-1) vs DAT
FUNCTION XWLVT = 0., 149.2,...
                7., 322.1,...
                14., 777.7,...
                21., 1522.8,...
                28., 2476.6,...
                35., 2621.5,...
                42., 2886.3,...
                49., 2770.1,...
                56., 2667.4,...
                63., 2600.9,...
                70., 2485.1,...
                77., 2421.1,...
                84., 2388.5,...
                91., 2388.1

*-----total crop biomass (kg ha-1) vs DAT, for comparison only.
FUNCTION XWCRT = 0., 405.1,...
                7., 831.4,...
                14., 1790....
                21., 3449....
                28., 5411....
                35., 6529....
                42., 8257....
                49., 9916....
                56., 10793....
                63., 12364....
                70., 14014....

```

```

77.,15225.,...
84.,15618.,...
91.,15615.

*-----For reruns, supply modified sets of input parameters here.
END

STOP

*****
* SPLDOSE applies split-dose fertiliser N applications. *
* for the calculation of harvest index from the *
* relation between observed harvest index and the *
* ratio of biomass at first flowering to post- *
* flowering cumulative radiation. *
*
* IN DAT - days after transplanting *
*
* INOUT APSLOP - fertiliser N application on day DAT *
*****
SUBROUTINE SPLDOS (DAT, APSLOP)

*----Formal parameters
REAL DAT, APSLOP

*----Auxillary variables
INTEGER IDCNT, IMXSPL
PARAMETER (IMXSPL = 5)

*----Model parameters
* SPLIT - array with SORTED dates (in DAT) on which to apply
* a split-dose.
* SPLDAT - array with sizes of the split-doses (in kg/ha)
* NRSPL - the actual number of splits

REAL SPLIT, SPLDAT
DIMENSION SPLIT(IMXSPL), SPLDAT(IMXSPL)

*----Supply the sizes and dates of splits in the below
* variables. Use dummy values of -99. to fill up the array until
* the 10th element.
* Number of sizes in SPLIT and number of dates (in DAT) in SPLDAT
* must be equal to one another.
*
* No safety checks have been built-in, so make sure you supply the correct
* values and numbers.
DATA SPLIT /50., 50., 50., -99., -99./
DATA SPLDAT /20., 40., 60., -99., -99./

*----Initialisation
IF (DAT.EQ.0) THEN
  APSLOP = 0.
  IDCNT = 1
ENDIF

IF ((SPLDAT(IDCNT).EQ.INT(DAT)).AND.(SPLIT(IDCNT).NE.-99.)) THEN
  APSLOP = SPLIT(IDCNT)
  IF (IDCNT.LT.IMXSPL) IDCNT = IDCNT + 1
ELSE
  APSLOP = 0.
ENDIF

RETURN
END
*****

```

```

* RDDCAL computes intermediate variables that are required *
* for the calculation of harvest index from the *
* relation between observed harvest index and the *
* ratio of biomass at first flowering to post- *
* flowering cumulative radiation. *
*
* IN      WCR      - weight of the crop, biomass (kg ha-1) *
*          RDT      - daily global radiation      (MJ m-2 d-1) *
*          DAT      - days after transplanting *
*          DATFF     - first flowering in days after *
*                    transplanting *
*          DELT     - timestep of integration *
* INOUT    WCRFF    - weight of the crop at first *
*                    flowering (kg ha-1) *
*          RDDSUM    - post-flowering cumulative radiation *
*****
      SUBROUTINE RDDCAL (WCR, RDT, DAT, DATFF, DELT, WCRFF, RDDSUM)

      REAL WCR, RDT, DAT, DATFF, WCRFF, RDDSUM, DELT
      LOGICAL AFFLOW

      REAL INTGRL

      IF (DAT.EQ.0.) THEN
        RDDSUM = 0.
        AFFLOW = .FALSE.
      ENDIF

*-----Crop weight at first flowering (kg ha-1)
      IF ((DAT.GE.DATFF).AND.(.NOT.AFFLOW)) THEN
        AFFLOW = .TRUE.
        WCRFF = WCR
      ENDIF

*-----Radiation sum upto flowering
      IF (AFFLOW) RDDSUM = INTGRL(RDDSUM, RDT, DELT)

      RETURN
      END

```

Appendix 15 Examples of input and output of the RIGAUS program

This Appendix lists example input and output files that are used by RIGAUS program. These files can be created and handled by FSU if the user activates the 'Generate Monte Carlo Reruns File' menu option in the FSU environment. Otherwise, input files must be created manually. All files are examples only and may differ considerably, depending on data entered by the user.

The RIGAUS configuration and input file RIGAUS.IN

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

TND = 5           ! Number of Draws

ISEED = 27426    ! Seed for random drawing

***** UNIFORM *****
NDU = 1           ! Number of variables for drawing from UNIFORM
                  ! distribution (MAXIMUM = 10)

* Names of variables for UNIFORM distribution
VUNI = 'S1'

* Give lower and upper boundary, in sequence of the
* variables specified above.
UNILO = 5.
UNIUP = 10.

***** 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 = 0           ! 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 = 1      ! Number of measured variables (MAXIMUM = 3)

* Names of measured variables
NMVAR = 'YEAR'

* Give measured data of the above variables (MAX = 500)
MDATA1 = 1980., 1981., 1982., 1983., 1984., 1985., 1986., 1987., 1988.

```

RERUNS.DAT file created by RIGAUS

```

* ISEED = 83299
* This is rerun set      1
RIGIFT= .5442E+03
DTRP = .2807E+02
STTIME= .1189E+03

* This is rerun set      2
RIGIFT= .8243E+03
DTRP = .1178E+02
STTIME= .1432E+03

* This is rerun set      3
RIGIFT= .3382E+03
DTRP = .2746E+02
STTIME= .9997E+02

* This is rerun set      4
RIGIFT= .1474E+04
DTRP = .1286E+02
STTIME= .1567E+03

* This is rerun set      5
RIGIFT= .1281E+04
DTRP = .1436E+02
STTIME= .1548E+03

* This is rerun set      6
RIGIFT= .1404E+04
DTRP = .8504E+01
STTIME= .1667E+03

* This is rerun set      7
RIGIFT= .1395E+03
DTRP = .2097E+02
STTIME= .1584E+03

* This is rerun set      8
RIGIFT= .7231E+03
DTRP = .2126E+02
STTIME= .1270E+03

* This is rerun set      9
RIGIFT= .9895E+03
DTRP = .1767E+02
STTIME= .1717E+03

* This is rerun set     10
RIGIFT= .7552E+03
DTRP = .1681E+02
STTIME= .1439E+03

```

COLUMN.DAT file created by RIGAUS

RIGIFT	DTRP	STTIME
.5442E+03	.2807E+02	.1189E+03
.8243E+03	.1178E+02	.1432E+03
.3382E+03	.2746E+02	.9997E+02
.1474E+04	.1286E+02	.1567E+03
.1281E+04	.1436E+02	.1548E+03
.1404E+04	.8504E+01	.1667E+03
.1395E+03	.2097E+02	.1584E+03
.7231E+03	.2126E+02	.1270E+03
.9895E+03	.1767E+02	.1717E+03
.7552E+03	.1681E+02	.1439E+03

