Hans F. Ravn

The Balmorel Model Structure

Version 3.01 (June 2010)

(This document revised 2010.06.01)

Contents

1 Introduction **8**
- 1.1 This version . . . 8
- 1.2 Data structure, model and simulation . . . 11
- 1.3 A short introduction to GAMS terminology . . . 12
- 1.4 Naming conventions . . . 17
- 1.5 User interface . . . 19

2 File Structure **19**

3 Sets **23**
- 3.1 Geography . . . 23
  - 3.1.1 Countries: C, CCC . . . 25
  - 3.1.2 Regions: RRR . . . 26
  - 3.1.3 Areas: AAA . . . 26
  - 3.1.4 Urban and rural heat areas: AAAURBH, AAARURH . . . 26
  - 3.1.5 Relations between C, R and A: RRRAAA, CCCRRR . . . 26
  - 3.1.6 Relations to outside the modelled geography . . . 27
- 3.2 Time . . . 27
  - 3.2.1 The years: YYY, Y . . . 28
  - 3.2.2 Time segments within years: SSS, S, TTT, T . . . 28
  - 3.2.3 Daytypes: workdays and weekend . . . 30
  - 3.2.4 Ordered and unordered sets . . . 30
- 3.3 Generation technologies: GGG, G, GDATASET, GGCOMB . . . 31
  - 3.3.1 Acronyms . . . 32
  - 3.3.2 Short term storages . . . 32
- 3.4 Fuels: FFF, FDATASET, FKPOTSET . . . 35
- 3.5 New generation technology and area: AGKN . . . 36
- 3.6 Demand: DF_QP, DEF_..., DHF_... . . . 36
- 3.7 Emission policies: MPOLSET . . . 38
- 3.8 Internal sets . . . 38
  - 3.8.1 Static and dynamic sets . . . 38
  - 3.8.2 Areas simulated: IA . . . 38
  - 3.8.3 Rural and urban heat areas simulated: IARURH, IAURBH 38
  - 3.8.4 Country to area mapping: ICA . . . 39
  - 3.8.5 Regions simulated: IR . . . 39
  - 3.8.6 Electricity import-export: IRRRI, IRRRE, IRI, IRE . . . 39
  - 3.8.7 Season and time duplication: ISALIAS, ITALIAS . . . 39
  - 3.8.8 Season and time refinements: IST, ISTS, ISTT . . . 39
  - 3.8.9 Generation technology types . . . 39
  - 3.8.10 Investments in new technologies: IAGKN . . . 41
  - 3.8.11 Equation feasibility: IPLUSMINUS . . . 43
- 3.9 Restrictions on sets . . . 43

4 Parameters and scalars **43**
- 4.1 Scalars . . . 44
  - 4.1.1 PENALTYQ . . . 44
- 4.2 Parameters on the set YYY . . . 45
  - 4.2.1 YVALUE . . . 45
- 4.3 Parameters on the set SSS . . . 45
  - 4.3.1 WEIGHT_S . . . 45
  - 4.3.2 CYCLESINS . . . 45
- 4.4 Parameters on the set TTT . . . 45
  - 4.4.1 WEIGHT_T . . . 45
- 4.5 Parameters on the set CCC . . . 46
  - 4.5.1 ANNUITYC . . . 46

4.5.2 TAX_DE 47
4.5.3 TAX_DH 47
4.6 Parameters on the set RRR 47
4.6.1 DISLOSS_E 47
4.6.2 DISCOST_E 47
4.6.3 DEFP_BASE 47
4.7 Parameters on the set AAA 47
4.7.1 DISLOSS_H 47
4.7.2 DISCOST_H 47
4.7.3 DHFP_BASE 47
4.7.4 DISLOSS0E DISCOST0E, DISLOSS1E, DISCOST1E, DISLOSS0H, DISCOST0H, DISLOSS1H, DISCOST1H 48
4.7.5 WNDFLH 48
4.7.6 SOLEFLH 48
4.7.7 SOLHFLH 48
4.7.8 WTRRRFLH 48
4.7.9 WTRRSFLH 48
4.8 Parameters on the set product (FFF,FDATASET) 48
4.8.1 FDATA 48
4.9 Parameters on the set product (FFF,CCC) 48
4.9.1 TAX_F 48
4.10 Parameters on the set product (GGG,AAA) 49
4.10.1 GINVCOST 49
4.10.2 GOMVCOST 49
4.10.3 GOMFCOST 49
4.10.4 GEFFDERATE 49
4.11 Parameters on the set product (YYY,RRR) 49
4.11.1 X3FX 49
4.11.2 DE 50
4.12 Parameters on the set product (YYY,AAA) 50
4.12.1 DH 50
4.13 Parameters on the set product (GGG,GDATASET) 50
4.13.1 GDATA 50
4.14 Parameters on the set product (IRRRE,IRRRI) 51
4.14.1 XKINI 51
4.14.2 XINVCOST 52
4.14.3 XCOST 52
4.14.4 XLOSS 52
4.15 Parameters on the set product (FKPOTSETA,AAA) 53
4.15.1 FKPOTA 53
4.16 Parameters on the set product (FKPOTSETR,RRR) 53
4.16.1 FKPOTR 53
4.17 Parameters on the set product (FKPOTSETC,CCC) 53
4.17.1 FKPOTC 53
4.18 Parameters on the set product (AAA,SSS) 53
4.18.1 WTRRSVARS 53
4.18.2 HYPPROFILS 54
4.19 Parameters on the set product (AAA,SSS,TTT) 54
4.19.1 DH_VAR_T 54
4.19.2 SOLE_VAR_T 54
4.19.3 SOLH_VAR_T 54
4.19.4 WND_VAR_T 55
4.19.5 WTRRRVAR_T 55
4.19.6 DHFP_CALIB 55
4.20 Parameters on the set product (RRR,SSS,TTT) 55
4.20.1 DE_VAR_T 55
4.20.2 DEFP_CALIB 56

4.20.3 X3FX_VAR_T 56
4.21 Parameters on the set product (GGG,AAA,SSS) 56
4.21.1 GKDERATE 56
4.22 Parameters on the set product (YYY,AAA,GGG) 56
4.22.1 GKFX 56
4.23 Parameters on the set product (YYY,AAA,FFF) 57
4.23.1 FUELPRICE 57
4.24 Parameters on the set product (YYY,MPOLSET,CCC) 57
4.24.1 M_POL 57
4.25 Parameters on the set product (RRR,SSS,TTT,DF_QP,DEF) . . . 57
4.25.1 DEF_STEPS 57
4.26 Parameters on the set product (AAA,SSS,TTT,DF_QP,DHF) . . . 58
4.26.1 DHF_STEPS 58
4.27 Default values 58
4.28 Restrictions on parameter values. 59
4.29 Internal parameters and scalars 59
4.29.1 IBALVERSN 59
4.29.2 ISCALAR1, ISCALAR2, IOF 60
4.29.3 IANYSET 60
4.29.4 IWEIGHSUMS 60
4.29.5 IWEIGHSUMT 60
4.29.6 IHOURSINST 60
4.29.7 IDAYSIN_S 61
4.29.8 IHOURSIN24 61
4.29.9 IDE_SUMST 61
4.29.10 IDH_SUMST 61
4.29.11 IWND_SUMST 61
4.29.12 ISOLESUMST 62
4.29.13 ISOLHESUMST 62
4.29.14 IHYINF_S 62
4.29.15 IWTRRRSUM 62
4.29.16 IWTRRSSUM 62
4.29.17 IX3FXSUMST 62
4.29.18 IM_CO2 62
4.29.19 IM_SO2 62
4.29.20 IGKVACCTOY 63
4.29.21 IGKFX_Y 63
4.29.22 IXKINI_Y 63
4.29.23 IXKN 63
4.29.24 IAGK_Y 63
4.29.25 IFUELP_Y 63
4.29.26 ITAX_CO2_Y 63
4.29.27 ITAX_NOX_Y 64
4.29.28 ITAX_SO2_Y 64
4.29.29 ILIM_CO2_Y 64
4.29.30 ILIM_SO2_Y 64
4.29.31 ILIM_NOX_Y 64
4.29.32 IDE_T_Y 64
4.29.33 IDH_T_Y 65
4.29.34 IDHFP_T 65
4.29.35 IDEFP_T 65
4.29.36 IX3FX_T_Y 65

5 Auxiliary parameters for outputs **65**

6 Variables **66**

7 Equations and constraints **68**

8 Model and solve **70**

9 Calibration **70**
9.1 Calibration of fuel consumption 71
9.1.1 Calibration of GKDERATE 71
9.1.2 Calibration of GEFFDERATE 74
9.2 Calibration of demand functions 76
9.2.1 Calibration of DEFP_CALIB and DHFP_CALIB 76

10 Output **77**
10.1 Automatically generated GAMS output 77
10.2 Less automatical GAMS output and inspection 77
10.3 Balmorel auxiliary output 78
10.4 The compare facility 79

11 Errors and Log **82**
11.1 Errors automatically detected by GAMS 82
11.2 Errors detected by Balmorel auxiliary parts 83
11.3 Sequence of log and error observations 83

12 Model variants **84**
12.1 Why variants 84
12.2 Types of changes 84
12.3 Organising modifications 85
12.4 Addon 86
12.5 Selections and Add-ons 86
12.6 Control variables in the BUI environment 88
12.6.1 Control variables - file balopt.opt 91
12.7 Combination technologies 93
12.8 No new investments 95
12.9 No electricity transmission 95
12.10Price dependent electricity exchange with third countries 95
12.11Reserve generation capacity 98
12.12Finer subdivision of the year 98
12.13Making more parameters depend on the year 98
12.14 Making more parameters depend on geography 99
12.15 Market power 99
12.16 Markets for emission, renewable energy and JI 99
12.17 Investment costs depend on technical life time 99
12.18 Version numbering and referencing 100

13 User interface **101**

14 Overview of model structure components **101**

1 Introduction

This paper documents the Balmorel model structure and describes some of the technicalities in the model.

The Balmorel model was originally developed for the analysis of the power and CHP (combined heat and power) sectors in the Baltic Sea Region. The model was directed towards the analysis of policy questions in an international context to the extent that they contain substantial international aspects. The original ambitions and limitations have since long been superseded, to that the model is no longer tied to the original focus geography and issues.

The model is implemented in the GAMS modelling language. For the present, we assume that the reader is familiar with this. An ultra short introduction is given in Section 1.3. The files that contain the model as specified in the GAMS language is in fact a good documentation of the model. In the present document we have aimed at presenting a documentation that is structured differently and which presents additional information and overview relative to that in the GAMS model files.

In particular note that emphasis in the present document is on the model structure. By this is meant that actual values of parameters are not given nor are the actual members of the sets used in the model. However, the names of the parameters and sets are specified and so is their functioning in the model.

Further note that the present document mainly treats the GAMS part of the model. Various facilities are provided that permit working with i.a. a data base or spreadsheet environment, however, this will not be treated here.

For the exact documentation of the model, input data and set members, see the model files.

This document is part of a series that together documents the Balmorel model:

> Balmorel: A Model for Analyses of the Electricity and CHP Markets in the Baltic Sea Region (Main Report)
>
> The Balmorel Model: Theoretical Background
>
> The Balmorel Model Structure (this document)
>
> Balmorel: Getting Started

These documents and further information, including application examples and illustrative input data, may be found at the Balmorel homepage: www.Balmorel.com.

1.1 This version

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

The description given here is for version 3.01 (June 2010) of the model.

There have been a number of changes in the Balmorel model structure from the first version (from 2001) to the present one. The changes will not be documented here, however you may contact us for further information, cf. the Balmorel homepage.

The changes from the previous version, 2.14, are substantial and concern improvements in the functioning of the model, both in terms of new features and in terms of improvements in processing speed.

The model is implemented in GAMS, mainly Version 2.25, see further Section 1.3.

The model has been developed on PC/Windows, see further Section 1.3.

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. As of today all the above limitations have been eliminated. Probably Balmorel will in the future not be compatible with version 22.5.

THE FOLLOWING IS NEW FOR VERSION 3.01. The changes introduced with version 3.01 were relatively extensive. They may shortly be classified as follows:

- Changes in declarations, sequence of indexes: some parameters has their declarations changed with respect to the sequence of the involved sets in order to harmonise the declarations, and in order to improve efficiency of some operations on them. The indexes are basically used in the sequence Y < C < R < A < G < F < S < T. Changes involve FGMIN, FGMAX, GINVCOST, GOMVCOST, GOMVCOST, GOMFCOST, GOMFCOST. This will make version 3.01 incompatible with previous versions.
- Changed the names of Parameters DISLOSS_E, DISLOSS_H, XLOSS to DISLOSS1E, DISLOSS1H, XLOSS1; the intention is to introduce also parameters DISLOSS0E, DISLOSS0H, XLOSS0 to represent constant losses. This will make version 3.01 incompatible with previous versions.
- Added YYY index to PARAMETER XKINI(IRRRE,IRRRI), i.e. XKINI(YYY,IRRRE,IRRRI). This will make version 3.01 incompatible with previous versions.
- Added element FDRE 'Share of fuel counted as renewable energy (fraction)' to FDATASET. No handling yet.
- Added PARAMETER FMAXINVEST(CCC,FFF) 'Maximum investment (MW) by fueltype for each year simulated'.
- The label GDURBRUR in Set GDATASET and all its applications have been deleted.
- Introduction of acronyms to identify technology types (previously integer values were used): GCND, GBPR ,GEXT, GHOB, GETOH, GHSTO, GESTO, GHYRS, GHYRR, GWND, GSOLE, GSOLH. This version 3.01 will permit also to use the previous integer identification, however, it is foreseen that integers will become invalid in future versions.
- Introduction of possibility to use user defined acronyms to identify technology types (previously integer values were used). This version 3.01 will permit also to use the previous integer identification, however, it is foreseen that integers will become invalid in future versions.
- Moving some declarations and assignments of internal identifiers (IR, IA and others) in Balmorel.gms; this implies that it will no longer be possible to use these identifiers in data files.
- Any input data file now contains one and only one item (set, scalar, parameter, or acronym collection).
- All input data files are now supposed to be found in a folder named 'data', located at the same level as the 'model' folder (previously data files were located in the 'model' folder).
- The default folder for data input is the 'data' folder parallel to the 'model' folder; if a data file is not found here, the folder '../../base/data/' will be assumed.
- Extensive application of control variables and conditional compilation for control of model versions (the 'addon'-functionality, see file balopt.opt).
- Extensive use of GDX facilities for parts of data handling and analysis, including use af Microsoft Access for output data.
- A number of enhancements in model functionalities, reflecting development in relation to ongoing project applications of the model; most of them implemented as 'addons'.
- A number of changes that aim at interaction with the Balmorel user Interface (BUI).

Here follows a more detailed description:

- PARAMETER GEFFDERATE(GGG,AAA) changed to PARAMETER GEFFDERATE(AAA,GGG).
- PARAMETER FGMIN(RRR,FFF) changed to PARAMETER FGMIN(CCCRRRAAA,FFF).
- PARAMETER FGMAX(RRR,FFF) changed to PARAMETER FGMAX(CCCRRRAAA,FFF).
- PARAMETER GINVCOST(GGG,AAA) changed to PARAMETER GINVCOST(AAA,GGG).
- PARAMETER GOMVCOST(GGG,AAA) changed to PARAMETER GOMVCOST(AAA,GGG).
- PARAMETER GOMFCOST(GGG,AAA) changed to PARAMETER GOMFCOST(AAA,GGG).
- Introduced SET IGSOLH(GGG) 'Solar heat technologies' and code for defining it, but NO further handling.
- Introduced ACRONYMS GCND, GBPR ,GEXT, GHOB, GETOH, GHSTO, GESTO, GHYRS, GHYRR, GWND, GSOLE, GSOLH.
- Introduced application of acronyms in the definition of set IGCND, IGBPR, etc. Note: The integer codes used previously (where 1 signifies GCND, etc.) to identify technology types may become invalid in a future version.
- Introduced facilities for using user specified ACRONYMS to be used as fuel names.
- Introduced the element FDACRONYM in FDATASET.
- Introduced SET IGF(GGG,FFF) to hold the relations between technology and fuel, based on GDATA(G,'GDFUEL') and FDATA(FFF,'FDACRONYM'); code consequently adapted (by substitution expressions like: " $(GDATA(G,'GDFUEL') EQ FDATA(FFF,'FDNB') " with "$IGF(G,FFF)".) Note: The integer codes (held in FDATA(FFF,'FDNB')) used to identify fuel types may become invalid in a future version.
- Introduced PARAMETER XKDERATE(IRRRE,IRRRI) 'Transmission capacity derating (fraction)' Any input data file now contains one and only one item (set, scalar, parameter, or acronym collection).
- Deleted SETS DHF_U, DHF_D, introduced SETS DHF_U1, DHF_D1, DHF_U2, DHF_D2, DHF_U3, DH_D3; however only code for handling DHF_U1, DHF_D1 has been implemented.
- Introduced SCALAR IOF8784 'Multiplier 8784' /8784/;
- Introduced ALIAS(AAA,IAAAE,IAAAI); ALIAS(IA,IAE,IAI); ALIAS(FFF,IFFFALIAS); ALIAS (Y,IYALIAS); (ALIAS (S,ISALIAS);ALIAS (T,ITALIAS);?)
- Introduced $label ENDOFMODEL at the end of Balmorel.gms
- Introduced PARAMETER XKDERATE(IRRRE,IRRRI) 'Transmission capacity derating (fraction)'
- Folder /addon/combination/ renamed addon/combtech/
- Combination technologies features now changed into an addon, i.e., COMTECH is not reflected in execution unless setting "$setglobal COMBTECH yes" in balopt.opt.

- Introduced parameter CYCLESINS(S) 'Number of load cycles per season '; applied in QHSTOVOLT, QESTOVOLT (replaces previous QSTOVOLT_OneCycleInS.)
- Changed the names of Parameters DISLOSS_E, DISLOSS_H, XLOSS to DISLOSS1E, DISLOSS1H, XLOSS1.
- Introduced (but not done yet!) Parameters to represent constant losses:
  PARAMETER DISLOSS0E(RRR) 'Loss in electricity distribution (constant, UNIT?)'
  PARAMETER DISLOSS0H(AAA) 'Loss in heat distribution (constant, UNIT?)'
  PARAMETER XLOSS0(IRRRE,IRRRI) 'Transmission loss between regions (constant, UNIT?)';
- Changes in error files (error1.inc, error2.inc, error3.ind, error4.inc); details will not be given, see the files.
- Added control variables: (details to be given).

Limitations and expected modifications

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. We are not aware of any errors in the present version, however, it includes elements that are not yet completely integrated:

- The set IST(S,T) is introduced, however, advantage can not be taken of it since the implementation is not complete. (Not yet: SET ISTT(T); SET ISTS(S); defined by LOOP((S,T) $IST(S,T),ISTT(T)= YES); LOOP((S,T) $IST(S,T),ISTS(S)= YES). ISTS is used to hold the subsets of S for which there is at least on T such that (S,T) is in IST(S,T). Introduced to get more flexibility in definition of time segmentation of the year. Observe that it can not be used i definition of equations that involve lead or lags (typically i relation to storage)). Section 3.8.8. It is possible that IST will be eliminated i future versions.
- GDCH4 emission not fully implemented.
- FDN2O emission not fully implemented.
- GDLIFETIME not used (so for investments, the standard period implied by ANNUITYC is used).
- Minimum and maximum limits for hydro power storage works only correctly for problems without new investments permitted for this type.
- Heat and electricity short term storages only implemented with no new investments.
- The material related to price dependent electricity exchange with third countries (Section 12.10) is now implemented as standard. However, a few desirable features are not implemented yet, in particular print facilities.

Version for handling simulation on hourly basis throughout a year is now available (model Balbase3).

1.2 Data structure, model and simulation

We distinguish here between three concepts: that for which data structures exist, that which is modelled (i.e., that for which a meaningful data set has been entered into the data files), and that which is simulated.

When we refer to that for which data structure exist we have in mind what the data structures actually allow of data input, this could be seen as the potentials of the database. The restriction on this is the sets, parameters, etc. that are declared. For instance, the years for which demand may be given could be from 1995 to 2030. This set of years is given by the set YYY. Other triple letter sets, AAA, RRR, CCC, SSS, TTT, GGG and FFF have the same function.

When we refer to that which is simulated we refer to a specific simulation. Such a simulation will for instance only concern the subset Y of the above mentioned years, e.g. the years 1995 to 2010. The GAMS syntax requires that Y be a subset of YYY.

Further, in order to make a meaningful simulation, data must be available for the simulation. That for which data is available is referred to as that which is modeled (or that for which a data set exist).

Hence that which is simulated must be a subset of that which is modeled, and that again must be a subset of that for which data structures exist. Assuming that the user is reasonable, it is necessary only to distinguish between that which is simulated and that for which data structures exist.

Presently data structures cover the period 1995 to 2030, and for this period there has also been provided data. Hence, what is modeled is (as far as years are concerned) identical to that for which data structures related to years exist.

Observe that the aim of the present document is the description of the structure of the model. Therefore the actual parameter values and set members given here should be considered as examples, rather than that actually used. See further Section 12.18 for some specifications (and Sections 3.9 and 4.28 for exceptions).

1.3 A short introduction to GAMS terminology

GAMS is the acronym for General Algebraic Modeling System. The system is suitable for formulation, documentation and solution of large mathematical models.

Generally, we assume that the reader is sufficiently familiar with the GAMS language. A User's Guide, a Tutorial, and other relevant information about the GAMS modeling system may be found at www.gams.com.

For the purpose of the following description, we shall only point out a few basic things. The idea is therefore not to give a rounded presentation of the GAMS modeling language, implying e.g. that subjects that can be relatively easily understood by reading the model will not be explained.

GAMS version

The Balmorel model was originally implemented in version 2.25. This version has (minor) limitation relative to later versions. Version 2.25 was chosen to ensure compatibility with existing installations of GAMS. Later versions of GAMS have backwards compatibility such that Balmorel may execute on them.

The limitations following from restriction to version 2.25 as observed by us are (i) the length of identifiers and labels are restricted to ten characters, (ii) the use of WHILE is not possible, (iii) the use of FOR-TO and FOR-DOWNTO is not possible (see in Section 4.29.3 how this is circumvented), (iv) do not use tabulator. To check whether the syntax of version 2.25 is followed, a '$use225' may be inserted in the first line of the program, with the $ in the first position. (But we have observed that this feature unfortunately does not enforce limitation to ten characters.)

However, some few additions to the Balmorel model do not conform to the 2.25 version of GAMS.

The model is developed on PC/Windows. GAMS will also run under Unix and Linux. In order to facilitate Unix and Linux applications of the Balmorel model, file and path names have all been written in lower case letters. Note that according to the documentation, GAMS under Windows should use backward slash (\) in path names while under Unix and Linux forward slash (/) should be used.

However, it seem that the style we adopted (see Balmorel.gms) works for all three operating systems.

Sets

The GAMS language contains among other elements SETS, various parameter values (exogenously given) indicated by SCALAR or PARAMETER (and possibly entered in a TABLE), (endogenous) VARIABLES, and EQUATIONS. A set of EQUATIONS constitute a MODEL.

Sets are the basic building blocks of GAMS, corresponding to the indices in an algebraic representation of a model. The set is declared by SET or SETS, followed by the name (identifier) of the set, and possibly a description. The definition of the set is the specification of the contents of the set, i.e., the elements or the members of the set. If for example the model contains three countries, this may be specified as

SET COUNTRIES / DENMARK, NORWAY, SWEDEN, FINLAND /;

where it is seen that slashes (/) are used as delimiters of the definition.

As seen, in the GAMS system the creation of entities like SETS (but also PARAMETERS etc.) involve two parts: a declaration and an assignment or definition. Declaration means declaring the existence of something and giving it a name. Assignment or definition means giving something a specific value or form. Declaration and definition may be done in separate statements or (except for EQUATIONS) in the same statement (as above).

Sets may be given as subsets of previously defined sets, e.g.,

SET HYDROCOUNT(COUNTRIES) / NORWAY, SWEDEN, FINLAND /;

Sets may also be multi-dimensional, i.e., declared on the cartesian product of previously declared sets.

Sets may have their membership explicitly defined (i.e., the labels are given between slashes immediately following the declaration of the set) (in which case the sets are called static sets), or the membership may be defined by assignment (dynamic sets), see further Section 3.8.1.

A shorthand asterisk notation like SET /S1*S52/ may be used to indicate the labels S1, S2, ..., S52.

The entry order of the labels is the order in which the individual labels first appear in the program, either explicitly or as a result of using the shorthand asterisk notation. The entry order has implications for e.g. LOOP and DISPLAY statements. It also has implications in relation to ordered sets, see Section 3.2.4. Section 10.1 describes how a list describing the entry order may be obtained.

The ALIAS statement is used to define sets that are identical, but which have different identifiers (names). Hence, in relation to the above example, ALIAS (COUNTRIES,C) declares the set C and defines it to be identical to the set COUNTRIES.

Reference to individual members of sets may be given using quotation marks, thus in relation to the above set an individual country may be addressed as "DENMARK" or 'DENMARK'.

Sets may be one-dimensional or multi-dimensional and they may be ordered or unordered, see further Section 3.2.4.

Scalars and Parameters

The parameters and scalars are used to specify exogenous values.

Parameters are specified for some or all elements in a set, or for cartesian products of sets. The parameter DH, for instance, specifies the annual heat demand in an area (e.g., a city). Therefore this parameter is declared as DH(YYY,AAA), and hence it is clear that it refers to all combinations of elements (also referred to as the set product or cartesian product) in the sets YYY (all years) and AAA (all areas).

A parameter may be declared by the keyword "PARAMETER" (or "PARAMETERS") or the keyword "TABLE" followed by the name and the set(s), and possibly a description; finally a definition of values may be given. The difference between "PARAMETER" and "TABLE" only concerns the format in which data may initially be entered.

Scalars are also used to specify exogenous values, however, scalars are not related to any sets; it may be seen as a zero-dimensional parameter.

Parameter and scalar values may be specified directly by the user (often in a definition, i.e., immediately following the declaration). They may also be calculated (assigned)in the model from other values. Parameters and scalars that are not explicitly defined or assigned a value are automatically set to the default value zero.

Acronyms

An acronym is a special data type that allows the use of strings as values. Acronyms may be declared like e.g. "ACRONYM GHOB 'Heat only boiler technology', GCND Condensing type technology;" or "ACRONYM FUEL_GAS, FUEL_COAL, FUEL_LIGHTOILNOSULPHUR 'Almost sulphur-free oil';".

It is possible to assign acronyms to parameters and scalars, like e.g. GDATA('UNIT_MINE','GDTYPE')= GHOB. However, acronyms, which are character string values, can be used in logical conditions only with the 'EQ', '=', 'NE' and '<>' operators.

Variables

The variables of the model are endogenous, i.e., those entities that are determined internally in the model by solving the specified model. In the Balmorel model, a typical examples is the generation of electricity on a specific generation unit in a particular time period.

Variables are declared by the VARIABLE statement, followed by the name of the variable and the set(s), and possibly a description. Variables may be declared to be e.g. POSITIVE (meaning that they can attain only non-negative values), FREE (meaning that they can attain any real values), INTEGER or otherwise.

The values of the variables are to be found according to the problem type specified, typically by optimisation. However, variables may have their values fixed (by appending FX), or they may be bounded downwards and/or upwards (by appending .LO and/or .UP, respectively).

The numerical values of variables are referred to by the suffix L. Marginal values to equations are referred to by the suffix .M.

Naming restrictions

Identifiers are the names given to SETS, PARAMETERS, SCALARS, VARIABLES, EQUATIONS and MODELS. A label is the name of a set element. The types and number of characters of identifiers and labels are limited according to the GAMS syntax (page 12). In addition, conventions are applied in the Balmorel model (involving among other things the restriction to ten characters) (Section 1.4).

Obviously, words that have predefined meanings in the GAMS language (reserved words, key words) can not be used (e.g., MODEL, SET, INF, TABLE, LP).

And finally: GAMS is not case sensitive, thus e.g. the identifiers balmorel, Balmorel and BALMOREL are interpreted to be identical. (But observe, that the editor that the user applies may very well be case sensitive, and so may the operating system, cf. 1.3.)

Arithmetic expressions

The language permits the formulation of arithmetic expressions in a form that is fairly easily understood. Thus, e.g. the expression SUM(T, X(T)) can be read to mean the sum over the elements in the set T of the quantities X(T), where X is a vector with one element for each member in the set T. Similarly, PROD, SMAX and SMIN means the product, maximum value and minimum value, respectively, over the specified set. In contrast, MIN and MAX operate on lists of arguments.

The interpretation of the arithmetic operators "+", "-", "*" and "/" is straightforward. The traditional relational operators $<$, $\leq$, $=$, $\geq$, $>$, $\neq$ are specified as such, or as LT, LE, EQ, GE, GT, NE respectively, except in EQUATIONS, where $\leq$, $=$ and $\geq$ are specified as =L=, =E= and =G=, respectively. The "=" is used in assignments, e.g. PI=22/7.

Extended arithmetic

Extended arithmetic is allowed to include the value infinity, denoted INF. Thus, 6/INF is evaluated to zero, INF+INF is evaluated to INF, INF-100 is evaluated to INF, 8*INF is evaluated to INF and -INF is minus infinity. The expressions 0*INF and INF-INF are illegal. Also related to the implementation of extended arithmetic are NA (not available: thus e.g. 7+NA evaluates to NA), UNDF (undefined) and EPS. (EPS is an interesting item, numerically equal to zero in input, but yet different. For one use in input, see e.g. Section 4.15.1; in output EPS normally means zero or very close to it).

Conditional, logical, dollar expressions, exceptions

Various means may be used in order to formulate conditional expressions. Constructions using IF, ELSE and ELSEIF are similar to those found in common programming languages. Logical expressions may be made using NOT, AND, OR and XOR. Numerical values of parameters and scalars may be interpreted as logical values using the conventions that the value 0 means NO and other values means YES. GAMS further has the dollar ($) operator to permit conditional operations, loosely speaking corresponding to a conventional IF condition (but with subtle differences). An expression like SUM(X$MYPARM(X), ..) (where X is a set and MYPARM a parameter) is interpreted as summation of MYPARM over all those elements in X for which MYPARM(X) is not 0. (Due to the data representation used in GAMS, this is very efficient if MYPARM(X) is 0 for most elements in X.) Consult the GAMS User's Guide.

Sequence of statements, flow control

The sequence of the statements in GAMS is important. The statements of the model are normally executed sequentially.

However, control of this flow may be performed by using LOOP, IF-ELSE (including extensions using ELSEIF). The LOOP statement causes the execution of the statements within the scope of the loop for each member of the driving set(s) in turn. Thus e.g. "LOOP(C, ...)" is similar to "for all elements in turn in set C do ...". The order of execution within the loop is the entry order (Section 3.2.4) of the labels. The construction using FOR-TO, FOR-DOWNTO and WHILE statements are avoided, see page 12.

Entry of numerical data

Numerical data may be entered along with declaration of PARAMETERS or SCALARS or by assignment. For multi dimensional parameters, the TABLE is convenient. The layout of a TABLE is quite flexible. Thus, if a table has too many columns to fit nicely on a single line, then the columns that do not fit can be entered below (using the symbol "+" for continuation); thus, row labels, unlike column labels, may be duplicated. Data may be entered directly, or they may be calculated and assigned using "=". Observe that declarations can not come after assignments, and an assignment overwrites previous assignments. The extended arithmetic symbols INF, NA and EPS (but not UNDF) may be used in input. See e.g. Section 4.15.1 on the use of EPS.

Default data

It is very useful to note that if data are not entered for parameters or scalars then by default the value zero is assigned. (But note the special mean of EPS.)

However, not all elements can be given by default, at least one must explicitly be given a value, otherwise it is considered an error.

Numeric Issues

GAMS does not distinguish between integer and floating point numerical input or output, everything is treated as floating point. Variables may be specified as being integer, whether they are so in the found solution is a matter of the attained accuracy.

For reasons of numerical solution it is advantageous that the a model is well scaled. Most solvers will automatically scale the input, however it is possible to scale also from GAMS. This is done by letting GAMS do the scaling or the user may specify scaling for individual variables and equations. GAMS scaling is in most respects hidden from the user. See the GAMS User's Guide.

Comments and explanations

Comments may be entered in a line, if the line has a "*" in the first column. In particular, this may be used for commenting out a command. Comments may also be inserted between "/*" and "*/" (provided it is preceded by "$ONINLINE", preferably placed near the top of the BALMOREL.GMS file), or between "$ONTEXT" and "$OFFTEXT" (where the $'s must be in the first position of a line).

It is possible to associate explicative text with set element, parameters etc. For example, "SET CCC All Countries" or "PARAMETER DE(YYY,RRR) Nominal annual electricity demand". The explicative text may be given between a pair of quotes (and must be so, if special characters are used). Such text may be displayed in output, see Section 10.

Include files

In GAMS, the input may be split over several files. This is handled by include files. This means that the content of a file (typically with the extension "inc") may during compilation of the model be included in another file. Thus, for instance the contents of the file "TRANS.INC" is placed in this other file (e.g., BALMOREL.GMS) at the place where the statement "$INCLUDE TRANS.INC" (or "$INCLUDE "TRANS.INC";", but not "$INCLUDE TRANS.INC;") is found.

Equations and Model

See Section 8.

Solver

For the solution of the model a solver has to be used. Thus, the GAMS system passes the model to the solver, which solves the problem and passes the solution and related information back to the GAMS system, which in turn permits presentation of the solution and related information in various forms. Also default and error information, e.g. if the problem does not have a solution, will be returned.

Output, Errors etc.

See Section 10.

Control variables, conditional compilation

Conditional compilation and use of control variables may be used for switching between certain parts of the code. The advantages of control variables and conditional compilation (compared to more straightforward coding) are that specific parts of code are neglected during parsing, such that virtually no time is used in these parts. It is possible to have conflicting pieces of code in the same file, provided the control variables ensure that they are not in use at the same time. Further, control variables may provide for fundamental tests like existence or not of a file.

A control variable may be declared and initialised by a statement like "$setglobal MyControl yes", or a variant of this. It may be used to make a piece of GAMS code (on the remaining part of the line or on the line immediately following) active or not by statements like

$ifi %MyControl%==yes $include MyTestFile1.inc
$ifi not %MyControl%==RealRun $include MyTestFile2.inc
$ifi exist extracode.inc $include extracode.inc

The last "i" in "$ifi" makes the comparisons between text strings be case insensitive, in contrast to "$if". The condition is one of two types: a file operation or a string comparison.

1.4 Naming conventions

We have tried to select names for the various sets, parameters etc. to facilitate the recognition of the meaning from the name. Observe that names are limited to ten characters; this facilitates the printing of output and compatibility with older versions of GAMS, cf. page 12. The following conventions for names are used:

Single letters:

D: demand (e.g., DE: demand for electricity, DH: demand for heat)

E: electricity

F: fuel

F: flexible (e.g., DEF: flexible (i.e. elastic, depending on price) electricity demand, DHF: flexible heat demand)

G: generation, or generation technologies (e.g., GE: generation of electricity, GDINVCOST: investment cost for generation technology)

H: heat

I: internal (set, scalar or parameter)

K: capacity

M: emission

N: new

O: related to output from simulations

Q: equation

V: variable (see also VQ)

X: electricity transmission

Suffixes:

_T or T: the finest division of time is the subdivision of the season (e.g., one hour; or night-period, day-period, peak-hour) (and implicitly or explicitly also contains season and year index), cf. Section 3.2.2

_S or S: the finest division of time is the subdivision of the year into seasons (e.g. summer, winter; or Jan, ... , Dec) (and implicitly or explicitly also contains year index), cf. Section 3.2.2

_Y or Y: the year, annual

Further:

BPR: back pressure generating technology

CAL: calibration

CND: condensing generating technology

DIS: distribution

EXT: extraction generating technology

FLH: full load hours

FX: fixed, given, exogenous

HOB: heat only boiler generating technology

HY: hydro technology (GHYRS: with seasonal reservoir, GHYRR: run-of-river)

INI: initial

INV: investment

LIM: limit

OM: operation and maintenance (OMF: fixed, OMV: variable)

POL: policy (with respect to taxation, emission quota)

SOLE: solar, producing electricity

STO: storage (typically daily, HSTO: heat, ESTO: electricity)

VAR: variation over the time segments of the day and year

WND: wind

WTR: water (energy source)

VQ: variable that ensures feasibility in an equation

COMB: combination technologies

File extensions:

gms: gams file, the main file

inc: include file

out: output file

sim: simulation file

opt: option file

cmp: compare

med: intermediate file (from one program or execution to another)

mss: model and solver status print file

See also Section 10.3 concerning output files.

Index sequence

The sequence of indexes has been harmonized to some degree. The following sequence is generally used

Y < C < R < A < G < F < S < T

where < means "comes before" and the letters are to be interpreted intuitively (e.g. "R" can mean "RRR", "IR" etc.). For indexes not mentioned, no specific sequence is suggested, rather follow intuition when new code is developed.

BUT: e.g. (G,GDATASET), OK, special
(F,FDATASET) , OK, special
HYRSDS(AAA,HYRSDATAS,SSS) , OK, special (og lignende (AGS))
TAX_FHO_C(FFF,CCC), !!!
TAX_GH(AAA,G), !!!
TAX_FCHP_C(FFF,CCC) !!! ,
TAX_HHO_C(FFF,CCC),!!!
TAX_F(FFF,CCC)!!!
DEF_STEPS(RRR,SSS,TTT,DF_QP,DEF), OK, special, faktisk logisk
X3VPEX(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT), OK, special, og med 'geografi' paa rette plads
M_POL(YYY,MPOLSET,CCC) OK, special

Note that in GAMS later indexes run faster. For efficiency reasons this should be taken into account in loops, sums, and other indexed operations.

1.5 User interface

The GAMS program is contained in ascii files, hence any editor that can produce, read, modify and save such files may be used. Special editors suited for GAMS exist.

The user must make sure that the GAMS system is set up properly and in relation to the file structure described in Section 2.

THE FOLLOWING IS NEW FOR VERSION 3.01. A user interface BUI (Balmorel user Interface) is under development.

2 File Structure

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. ACTUALLY, IT IS ALMOST WITHOUT RELEVANCE! DO NOT READ IT and go directly to the next section 3. The model is distributed over a number of files. In this section we give an overview.

The files are ascii files, cf. Section 1.5.

In the base directory (e.g. taken from the home page www.balmorel.com at the Internet or from a diskette) you should find the following subdirectories (and possibly some more):

- model
- logerror
- printinc
- printout

This should be copied to the user's computer. The user may find it expedient to maintain this version unaltered, and therefore for an application a copy should be made, e.g. with the name "Balmorel-MyVersion1", containing the above mentioned subdirectory structure and the file structure mentioned in the sequel.

File and path names should not include special characters like æ, å, ő, ł, ß, œ, :, *, ?, ”, ¡, ¿, <, > or similar, and should be limited to eight characters. Observe that file and path names will be case sensitive in Unix and Linux. Cf. Section 1.1.

The above mentioned subdirectories, model, printinc, printout and logerror are mandatory in order to run the model. Other subdirectories may exist, e.g. 'documentation', 'data-pre-inc', but are not required for the functioning of GAMS.

Subdirectory Model

The Balmorel model is located in the subdirectory Model. Here the following files are found:

- BALMOREL.GMS: main model file (running this means running the model)
- SETS.INC: contains the declaration and definition of the static sets (see Section 3.8.1) in the model
- GEOGR.INC: contains most values specific for geographical entities
- FUEL.INC: contains values specific for the different fuel types
- TECH.INC: contains the parameters of the generation technologies in use
- TRANS.INC: contains the parameters related to electricity transmission between regions
- VAR.INC: contains daily and seasonal variations of all relevant parameters
- DE.INC: contains annual values of electricity demand
- DH.INC: contains annual values of heat demand
- FUELP.INC: contains annual values of fuel prices
- MPOL.INC: contains annual values of environmental policy
- GKFX.INC: contains annual values of exogenously specified generation capacities
- X3.INC: contains annual values of electricity exchange with countries (regions) not explicitly modelled
- balgams.opt: contains options for GAMS
- balbase1.sim: contains specification of the simulation/optimisation to be performed - balbase1 is without endogeneous investments.
- balbase2.sim: contains specification of the simulation/optimisation to be performed - balbase2 is with endogeneous investments..

These files together contain the model (balgams.opt does not hold any part of the real model). The BALMOREL.GMS file is the main file, while the others are include files (Section 1.3) which during compilation of the model automatically are inserted into the appropriate places in the BALMOREL.GMS file or elsewhere.

All user specified labels and numerical data are found in the include files.

The distribution of the different input data (sets, parameters etc.) over the files is done according to the following principles,

- SETS.INC contains declaration and definition of all static sets (see Section 3.8.1). Also the scalar PENALTYQ and the parameter YVALUE are in this file.

- TECH.INC contains global information about generation technologies, i.e., information that is not specific to geography. Therefore, all data structures that contain the index pair (GGG,GDATASET), and no other indexes are in this file.
- FUEL.INC contains global information about fuels, viz., information that is not specific to geography or time. Therefore, all data structures that contain the index pair (FFF,FDATASET) and no other indexes are in this file.
- TRANS.INC contains information about transmission conditions between pairs of regions. Therefore, all data structures that contain the index pair (IRRRE,IRRRI) and no other indexes are in this file.
- GEOGR.INC contains all information that is specific with respect to geography, except that which contains information relative to time (i.e. except that which contains indexes YYY, SSS or TTT), and except that which contains information about transmission between regions (i.e., except that which contains the index pair (IRRRE,IRRRI)).
- All data that depend on the year are placed in separate files:
  - DE.INC contains annual electricity demand for each year.
  - DH.INC contains annual heat demand for each the year.
  - GKFX.INC contains user specified installed generation capacity for each year.
  - X3.INC contains annual electricity exchange with third regions.
  - FUELP.INC contains fuel prices for each year.
  - MPOL.INC contains environmental policy data for each year.
- VAR.INC contains information about the variation of parameters within the year. Therefore, all data structures that contain indexes SSS and/or TTT are in this file.

See Section 14 for exact locations.

After running the model, GAMS will automatically have created two additional files that will be placed in the subdirectory Model (if the GAMS program has this directory as its base; if not, the files may be placed elsewhere):

- BALMOREL.LST
- BALMOREL.LOG

See Section 10 for more on this.

Subdirectory Data-pre-inc

This optional subdirectory contains various data and facilities for preparing some of the include files found in subdirectory Model. The user may find it expedient to create subdirectories. This will not be described further in the present document.

Subdirectory PrintInc

In addition to the files mentioned above there are in the subdirectory print auxiliary files that are not proper part of the model, but which provide various possibilities for generating output from successful model runs:

- PRINT1.INC: declares file names for predefined output and various parameters and sets that may be useful for creating output

- PRINT2.INC: declares parameter names for output that can be written for each year of the simulation

- prt3-bb1.inc: calculates the values of the parameters declared in PRINT2.INC relevant for model BALBASE1

- prt3-bb2.inc: like prt3-bb1.inc but for model BALBASE2

- prt4-bb1.inc: specifies for model BALBASE1 which output from the most recently simulated year to write to a file, by including files found in the subdirectory print

- prt4-bb2.inc: like prt4-bb1, but for model BALBASE2

- Several output generating files are found in the subdirectory print. They are all auxiliary include-files. They are controlled by the above PRINT*.INC files. See Section 5 and Section 10.

These files may be omitted (commented out) in the BALMOREL.GMS file without effecting the model itself. However, if they are not commented out, they must exist. We refer to such additional components as auxiliary parts. See Section 10.

Subdirectory LogError

This subdirectory contains auxiliary parts for checking the input data and monitoring the solution of the model.

- ERROR1.INC: declares file names for predefined output, ERRORS.OUT and LOGFILE.OUT

- ERROR2.INC: makes some simple checks of the input data and prints the conclusion to the file ERRORS.OUT. A summary is printed in the file LOGFILE.OUT.

- ERROR3.INC: makes some simple checks immediately before optimisation starts and prints the conclusion to the file ERRORS.OUT. A summary is printed in the file LOGFILE.OUT.

- ERROR4.INC: makes some simple checks immediately after optimisation starts and prints the conclusion to the file ERRORS.OUT. A summary is printed in the file LOGFILE.OUT.

- balbase1.mss: prints in the file LOGFILE.OUT a summary of the contents of the file ERRORS.OUT, and in addition model and solver status for model BALBASE1 (the extension 'mss' indicates 'model and solver status').

- balbase2.mss: like balbase1.mss, but for model BALBASE2.

- balbase23.mss: like balbase1.mss, but for model BALBASE3.

These files may be omitted (commented out) in the BALMOREL.GMS file (or wherever they are included) without effecting the model itself. However, if they are not all commented out, at least ERROR1.INC must be included. See Section 10 and Section 11.

Subdirectory Output

All the output specified by files in the print subdirectory is placed in the output subdirectory. The output generated from Error and Log files is placed in the LogError subdirectory.

Output generated automatically by the GAMS system will be placed in the Model subdirectory, cf. above.

In the output subdirectory additional facilities for presentation of output (e.g. in the form of spreadsheets) may be located. See Section 10.

Observe that by running GAMS any previous output in existing files may be overwritten. It is therefore recommended that the user creates a number of subdirectories (e.g. to the subdirectory Output) where output can be saved before the next GAMS run.

Subdirectory Documentation

In this optional subdirectory various documentation files may be placed. Also here the user may find it expedient to create subdirectories.

3 Sets

In this section we describe the sets in the model. According to the distinctions in Section 1.2 we shall describe the sets in the data structure, and the subsets that may be used for specific simulations. The names of the sets in the first group usually have triple letters (e.g., CCC, TTT).

Most of the sets have their members (elements, labels) specified by the user. We refer to those sets as input sets. Some sets are derived automatically from previously given sets, we refer to such sets as internal sets, Section 3.8. Section 3.9 lists some restrictions.

All sets are declared and defined in the file SETS.INC, see Section 2, except for internal sets (Section 3.8) that are declared and defined in the file BALMOREL.GMS.

3.1 Geography

The model permits specification of geographically distinct entities. The main types of geographical entities are Areas, Regions, and Countries. These entities are in relation to the data structure specified by the sets AAA, RRR and CCC, and for the subsets to be used for simulation by the subset C.

To ensure generality in geographical specification, all geographical entities, including AAA, RRR and CCC, are specified in set CCCRRRAAA.

Each country is constituted of one or more regions while each region contains zero or more areas. Any area must be included in exactly one region, and any region must be included in exactly one country, see also Figure 1 and Section 3.1.5.

The areas are the building blocks with respect to the geographical dimension. Thus, for instance all generation and generation capacities are described at the level of areas, and so are all aspects of heat demand; see the list below.

Areas are classified and grouped in a number of ways. The collection of subsets of areas into regions was described above, and further examples are given in Section 3.8.3.

Electricity balances are given on a regional basis. For each element in RRR electricity generation comes from the elements of AAA located in RRR. Hence, for each region an electricity balance must be fulfilled, but unlike heat, electricity may be exchanged between regions. Such transmission, and their constraints, losses and costs, are the motivation for the concept of regions. In contrast to this, transmission of heat between areas is not possible.

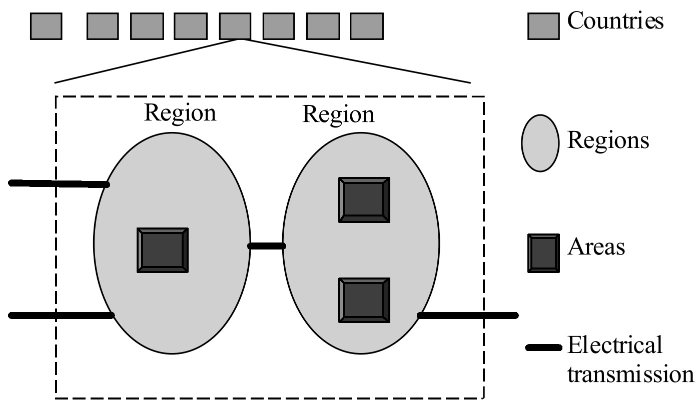

Figure 1: The geographical entities.

A number of regions (i.e., a nonempty subset of RRR) constitute a country. The country does not have any generation or consumption apart from that which follows as the sum over the regions in the country. However, a number of characteristics may be identical for all entities (e.g. generation units, demands, prices and taxes) in a country. A country is constituted of more than one region when needed to represent bottlenecks in the electricity transmission system within the country.

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. The following entities are related to countries:

- annuity
- taxes
- environmental policy
- availability of certain fuels

The following entities are related to regions:

- related to electricity demand:
  - annual nominal electricity demand
  - variation within the year of the nominal electricity demand
  - deviation from nominal electricity demand
  - consumer price base for electricity
  - variation within the year of the consumer price base for electricity
- related to electricity transmission and distribution:
  - losses in electrical distribution
  - cost of electrical distribution
  - cost of electrical transmission
  - losses in electrical transmission
  - electricity export to third countries

- variation within the year of the electricity export to third countries
- initial capacity on electrical transmission
- investment cost for new electrical transmission capacity

- related to energy and fuels:
  - availability of certain fuels

The following entities are related to areas:

- related to heat demand:
  - annual nominal heat consumption
  - variation within the year of the nominal heat demand
  - deviation from nominal heat demand
  - price base for heat
  - variation within the year of the consumer price base for heat
- related to heat distribution:
  - losses in heat distribution
  - cost of heat distribution
- related to technologies:
  - operation and maintenance cost for technologies
  - capacity reduction factor for technologies
  - efficiency reduction factor for technologies
  - initial capacities of generation technologies
  - investment cost for new technology
- related to fuels:
  - fuel price
  - availability of certain fuels
  - annual quantity and variation between the seasons of water availability for dispatchable hydro generation
  - annual quantity and variation within the year of non-dispatchable hydro generation
  - annual quantity and variation within the year of wind power generation
  - annual quantity and variation within the year of solar voltaic generation

The specification in Section 4 is structured according to the sets on which parameters are defined, hence refer to the Section 4 part of the table of contents to get an overview of the precise dependencies on geographical entities.

3.1.1 Countries: C, CCC

SET CCC contains the countries in the data structure (cf. Section 1.2), e.g.:

SET CCC /DENMARK, NORWAY, SWEDEN / ;

SET C(CCC) is the subset used to define those countries that are simulated. Observe, that if C is a proper subset of CCC then automatically the regions in the countries not included in C are excluded from the model, see Section 3.1.2 (and similarly with the areas not in regions in the countries in C, see Section 3.1.3). (An obvious implication of exclusion of a region is that electricity exchange with that region is not possible (therefore variables VX_T (Section 6) relative to the excluded region will not be included in the model).)

3.1.2 Regions: RRR

SET RRR contains the set of regions in the data structure, e.g.:

SET RRR / DK_E, DK_W, NO_R, SE_R / ;

As the choice of names indicates, Denmark is considered to consist of two regions, while Norway and Sweden each consists of one region.

If a restructuring of a country is desired, so that the number of regions is changed, this will involve restructuring of the associated data as well, see Section 12.

The simulated subset IR of RRR is described in Section 3.8.5 along with other variants of RRR.

3.1.3 Areas: AAA

SET AAA contains the set of all areas in the structure, e.g.:

SET AAA
/ DK_E_Copnh, DK_E_Other, DK_W_Odens, DK_W_Arhus, DK_W_Other, NO_R_Oslo, NO_R_Other, SE_R_Sthlm, SE_R_Rural / ;

If a restructuring of a region is desired, so that the number of areas is changed, this will involve restructuring of the associated data as well, see Section 12.

The simulated subset of AAA is described in Section 3.8.2 and other subsets are described in Section 3.1.4.

3.1.4 Urban and rural heat areas: AAAURBH, AAARURH

THE FOLLOWING WILL BE DELETED IN VERSION 3.01. THERE WILL BE NO DISTINCTION BETWEEN RURAL AND URBAN AREAS.

The set AAA of areas is classified according to a variety of principles. This is done by definition of a number of subsets (that may be overlapping) of AAA or subsets of set products that involve AAA. In the following the classification according to heat demand is described. Further examples are given in Section 3.5 and Section 12.

With respect to satisfaction of heat demand the areas are of two kinds, urban and rural. In the urban heat areas there may be an economic dispatch (i.e., a distribution of generation among the units varying over time according to economic principles) of heat between generation units. This is not the case in the rural heat areas where heat production is proportional between the different production units.

The set AAARURH of the rural heat areas in the data structure is defined as a subset of AAA as in the following example:

SET AAARURH(AAA) / DK_E_Other, DK_W_Other, SE_R_Rural /;

The set AAAURBH of the urban heat areas in the data structure is defined similarly. Note, that the two sets AAARURH and AAAURBH should not be overlapping.

The subsets IARURH and IAURBH used in simulation are then found automatically, see Section 3.8.3.

3.1.5 Relations between C, R and A: RRRAAA, CCCRRR

Given the definitions of the sets CCC, RRR, and AAA above the sets (mappings) RRRAAA and CCCRRR are defined in order to specify the connection between the sets, i.e., RRRAAA specifies which areas that belong to which regions, and CCCRRR specifies which regions that belong to which countries. Thus, RRRAAA(RRR,AAA) specifies the relation between RRR and AAA and CCCRRR(CCC,RRR) specifies the relation between RRR and CCC, as the following example shows:

```
SET RRRAAA(RRR,AAA)
/ DK_E.(DK_E_Copnh,DK_E_Other)
DK_W.(DK_W_Odens, DK_W_Arhus, DK_W_Other)
NO_R.(NO_R_Oslo,NO_R_Other)
SE_R.(SE_R_Sthlm,SE_R_Rural) /;

SET CCCRRR(CCC,RRR)
/ DENMARK .(DK_E,DK_W)
NORWAY .(NO_R)
SWEDEN .(SE_R) /;
```

Observe the use of the dot and the parentheses.

The internal set ICA(C,AAA) specifies the relation between AAA and C, see Section 3.8.4.

3.1.6 Relations to outside the modelled geography

It is possible to represent electricity exchange with places outside the modelled geography - i.e., with places that are not as extensively modelled as the basic parts of the model. See Sections 4.11.1 and 12.10.

3.2 Time

The description of the time dimension in the model may be divided into two parts: that which refers to the years and the relations between them, and that which refers to the aspects of time within the year.

The following entities are specified (exogenous) or found (endogenous) within a subdivision of the year:

- generation (exogenous and endogenous)
- relative weight of time segment
- capacity derating of generation units
- availability of hydro
- demands for electricity and heat
- calibration parameters relative to demands for electricity and heat
- flexible demands related parameters
- electricity exchange with third regions

The following entities are the same throughout each year, but may be different from one year to the next one:

- nominal generation capacities
- fuel prices
- emission limitations and taxes

The following entities are the same for all years in the data structure:

- characteristics of generation technologies (except that some may be available only from a certain year, and except for capacity derating)
- annuity
- distribution and transmission characteristics
- costs (except fuel costs)

- fuel potentials, including water availability
- taxes (except those related to emissions)
- variations within the year
- demand elasticities
- fuel characteristics (except prices)

The specification in Section 4 is structured according to the sets on which parameters are defined, hence refer to the Section 4 part of the table on contents to get on overview of the precise dependencies on geographical entities.

3.2.1 The years: YYY, Y

The years represented in the data structures are given by the SET YYY, e.g.:

SET YYY / 1995 * 2030 / ;

where the asterisk notation using "*" implies that the years from 1995 to 2030 are included.

The subset of years simulated is given by the SET Y(YYY).

Comment on naming conventions: The only labels consisting of digits only are those used for set elements YYY (and therefore also those in the subset Y(YYY)).

The sets YYY and Y are ordered, cf. the comments in Section 3.2.4 on ordered and unordered sets.

3.2.2 Time segments within years: SSS, S, TTT, T

The subdivision of the year into seasons is given by SET SSS specified e.g. as the following:

SET SSS / S1 * S4 / ;

and the subdivision of the time within the season of a season is given by SET TTT, e.g.,

SET TTT / T1 * T8 /

These examples mean that the year is divided into four seasons, and that each season has been subdivided into eight time segments (sub periods). We refer to the part of the year specified by (S,T) as a time segment (or more specifically as a time segment of the year) and to the part of the season specified by T as a time segment of the season.

(It is tempting to say that the set TTT represents a subdivision of the day - and we may actually do so sometimes. However, is not in general correct to say so, see Section 4.4.1.)

The extension - weight, duration - of each time segment in S and T is held in the parameters WEIGHT_S and WEIGHT_T, respectively, cf. Sections 4.3.1 and 4.4.1.

The seasons and time periods used in simulation are specified by the sets S(SSS) and T(TTT), respectively. S and T should be ordered, cf. Section 3.2.4.

Observe that all the descriptions of the subdivision of the year are the same for all the geographical entities (countries, regions, and areas, i.e., the sets CCC, RRR and AAA) and for all the years (the set YYY) in the model.

Comment on input data: It will as default be assumed that the year has 365 days and 8760 hours.

See also Section 4.29.7 and Section 4.29.8.

Comment on naming conventions: For chronological specifications of time segments the naming of the individual seasons will start from winter (i.e. the first season will include January 1st), and the naming of the time periods of the day will start at midnight, see also Sections 4.3.1 and 4.4.1. The labels should therefore be entered in such sequence, in particular in relation to application of ordered sets, cf. Section 3.2.4.

Obviously there are some interdependencies between the subdivision of the year into seasons and the further subdivision of the seasons, and this could expediently be reflected in the naming. The following convention may be used for naming the seasons: SET SSS may be defined as e.g.

SET SSS / S1 /; or
SET SSS / S2_1, S2_2 /; or
SET SSS / S1 * S4 /; or
SET SSS / S12_01 * S12_12 /; or
SET SSS / S52_01 * S52_52 /;

This gives the possibilities of representing the year with 1, 4, 12 or 52 seasons, respectively.

Similarly, SET TTT may be defined as e.g.

SET TTT / T1 /; or
SET TTT / T2_1, T2_ 2/; or
SET TTT / T8_1 * T8_8 /; or
SET TTT / T168_001 * T168_168 /;

giving the possibilities to represent the subdivision (the "day") of the season with 1, 2, 8 or 168 segments, respectively.

It is easy to aggregate time within the year such that the model uses only annual data, i.e., there is no subdivision into seasons nor any subdivision of the season into sub-periods. This is achieved by specifying the sets S and T to contain only one member each, e.g.: "SET S(SSS) /S1/ " and "SET T(TTT) /T1/". It is also easy to use other subsets S and T. E.g., if SSS is defined as "SET /S12_1 * S12_12/" to represent the twelve months of the year, then specifying SET S(SSS) / S12_1, S12_7 / means that only January and July will be used in the simulations to represent the whole year. With e.g.

SET SSS / S1 * S12/;
SET TTT / T1 * T12/;

a simulation with all 144 time segments will be specified as

SET S(SSS) / S1 * S12/;
SET T(TTT) / T1 *T12/;

and a simulation with the four time segments (S2,T1), (S2,T5), (S10,T1), (S10,T5), per year will be specified as

SET S(SSS) / S2, S10/;
SET T(TTT) / T1, T5/;

Further refinements are possible via the set IST, Section 3.8.8.

Comment on input data: Demand for electricity is specified for each region. If demand is not synchronous between the regions, this will in itself motivate exchange between the regions. Similarly, time zone differences impact on heat demand, and indeed all other characteristics related to the time of the day. In particular, this may be relevant for regions far apart in the east - west direction, because in this case there may be a discernable time zone difference. This is more outspoken the larger the difference in time zone is in relation to the length of the time segments in TTT. Table 1 illustrates for the Baltic Sea Region the local time zones relative to GMT.

	GMT	DK	EE	FI	DE	LV
Summer	0	2	2	3	2	2
Winter	0	1	1	2	1	1
	LT	NO	PL	RU (West)	RU (Kaliningrad)	SE
Summer	2	2	2	4	3	2
Winter	1	1	1	3	2	1

Table 1: Time zones in the Baltic Sea Region relative to GMT (based on http://time.greenwich2000.com/). Observe that these conventions need to be stable.

3.2.3 Daytypes: workdays and weekend

The time segments TTT may be seen as representing a typical week. For some purposes it may be useful to distinguish between different types of days, e.g. working days (Sunday through Friday) and weekend days (Saturday and Sunday). This may be accomplished using the set DAYTYPE 'Types of days within the week' /WORKDAY, WEEKEND/.

The assignment of time segments TTT to the two types of days is done in TWORKDAY and TWEEKEND, e.g. SET TWORKDAY 'Time segments TTT in workdays' /T1*T8/, and SET TWEEKEND 'Time segments TTT in weekends' /T9*T12/; or /T001*T120/, /T121*168/; respectively.

This assigment should be mutually exclusive and exhaustive, i.e., each label in TTT should appear exactly once, either in TWORKDAY or in TWEEKEND.

3.2.4 Ordered and unordered sets

Sets in GAMS may be ordered or unordered. Ordered sets are static in the sense that they are initialised by having their elements specified between "/" and "/" at the time of declaration, and the sets are never changed afterwards. They are ordered in the sense that the order in which the labels appear in the GAMS program is the same as the order in which they appear in the initialisation of the set (the entry order). See also Section 3.8.1.

For ordered sets, the elements have a sequence, viz., that given in the initialisation. Hence, for such sets it is possible to know if one element is "before" or "after" another one in the set, implying in relation to modelling that chronological phenomena may be represented. The operators '+' and '++' are used to indicate "the next" element, the latter further indicates a cyclical concept where "the first" is the successor to "the last".

The function ORD applied to an element in a one-dimensional static and ordered set returns the number of that element in the sequence. Thus for SET SEASONS /winter, spring, summer, autumn/, ORD("summer") attains the value 3. The function CARD returns the number of elements in a set (also for an unordered set), hence e.g. CARD(SEASONS) attains the value 4.

It is essential that the sets YYY, Y, SSS, S, TTT and T are ordered. For set S this may for instance be used for modelling of hydro power with reservoirs, where it is desired to represent that the contents of the reservoir at the beginning of a season equals the contents at the beginning of the previous season plus the inflow during the previous season, minus the water used for generation during the previous season. For set T this may similarly be used for modeling hydro power with reservoirs for shorter operation cycles (e.g. pumped storage suited for levelling of variations within the day or the week), or similarly for short-term heat storage. (See also Section 4.4.1.)

3.3 Generation technologies: GGG, G, GDATASET, GGCOMB

SET GGG is the set of generation technologies (i.e., hardware for transformation of energy) in the structure, given as e.g.

> SET GGG
> /CC-Cond1, ST-Cond1-G, ST-Cond1-O, CC-Co-B95,HO-Pump, HO-W-Old, HO-CHP-G, HYDRO, GWIND1, GWIND2 /;

SET G(GGG) is the set of generation technologies simulated, e.g.

> SET G(GGG) / ST-Cond1-O, HO-CHP-G, GWIND2 /;

Subsets of G are described in Section 3.8.9.
The set GDATASET is the set of attributes of generation technologies:

> SET GDATASET / GDTYPE, GDFUEL, GDCB, GDCV, GDFE, GDESO2, GDNOX, GDCH4, GDAUXIL, GDINVCOST0, GDOMVCOST0, GDOMF-COST0, GDFROMYEAR, GDLIFETIME, GDKVARIABL, GDSTOHLOAD, GDSTOHUNLD, GDCOMB, GDCOMBSK, GDCOMBSLO, GDCOMBSUP / ;

Descriptions are given in Section 3.8.9 and in Section 4.13.1.

Observe that the user should not change this set without proper knowledge of the functioning of the set. Thus, the set can not be reduced from that specified above since data will be needed in the model for each of the elements, see Section 4.13.1. The set may be enlarged with new elements, however then the user will have to specify in the model how these elements are to be used.

The data corresponding to the elements GDINVCOST0, GDOMVCOST0 and GDOMFCOST0 are considered as default values that may be overwritten, see Sections 4.13.1, 4.10.1, 4.10.2 and 4.10.3.

The data are further discussed in Section 4.13.1.

Comment on naming conventions: Observe that to distinguish technology from fuel (see Section 3.4) where similar labels (names) are tempting, the following is advocated: hydro power is called something indicating "hydro" as a technology, i.e., as an element in GGG, and something indicating "water" as a fuel, i.e., as an element in FFF (see Section 3.4). For all other ambiguous subjects, a prefixed "G" is advocated for elements in GGG and no prefixed "G" for elements in FFF. E.g., a particular wind turbine could be e.g. "GWIND-2300" as an element in G but not as an element in FFF.

In the Balmorel the specification of a generation unit is done by referring to its name (according to the technology catalogue given by set GGG) and its geographical location (according to the area catalogue given by set AAA). Thus, a specific kind of technology may be represented in more than one area. The capacity GKFX (Section 4.22.1) of a particular generation unit must therefore be specified with indexes reflection this, i.e. GKFX(*,AAA,G), where AAA represents geography and G represents technology kind.

The idea behind this is that for the geographical area considered (the Baltic Sea Region) it is not possible to get, nor sensible to use, precise information about all generation units. Therefore a limited number (approximately 50) of technology kinds have initially been specified in the set GGG. This moreover facilitates the aggregation of existing units into fewer but larger ones.

The expectation is, however, that with increased application of the model, some possibly with a national focus, more data will become available, and in specific applications there will be a desire to increase the level of detail in representation of technologies.

The GAMS syntax (Section 1.3) permits an explanatory text associated with (some or all) units in G. This identification is a possibility only, and therefore to use it systematically a convention is needed, and one such will now be described.

```
SET GGG        'All generation technologies'
/
ST-Cx-CO       "Steam condensing coal - new"
ST-Cy-CO       "Steam condensing coal - old"
ST-G-NOS       "The proposed new natural gas - Norway South"
NU-BBackSE     "Swedish nuclear Barseback"
ST-B8-CO       "CHP back pressure coal - old"
WI-L9          "Generic wind power - new"
WI-RS-DK       "New off shore wind - Roedsand, Denmark"
/;
```

Table 2: Illustration of a convention permitting various levels of identification of generation units.

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. As illustrated in Table 2 the text associated with the label in the set GGG is used to describe the technology, and this description indicates how specific each technology is. The GAMS syntax permits up to 80 characters, all on the same line, preferable enclosed in (single or double) quotes. To provide nice single line printouts, at maximum of 50 characters is advocated.

Observe, though, that in the mechanisms of the Balmorel model this text is not used. Therefore the user must make sure that the data entered for each technology is consistent with the intention indicated in the text for that technology.

Thus, for a technology which is intended to be located in one area only, the user must make sure that this technology appears with positive capacity GKFX(*,AAA,G) only in that area. If new investments are permitted the user must similarly make sure that the set AGKN(AAA,G) (Section 3.5) specifies that new capacity of the technology in question can only be established in the relevant area.

3.3.1 Acronyms

THE FOLLOWING IS NEW FOR VERSION 3.01. Acronyms are used to identify technology types, defined as follows: ACRONYMS GCND, GBPR, GEXT, GHOB, GETOH, GHSTO, GESTO, GHYRS, GHYRR, GWND, GSOLE, GSOLH;

An acronym is a special data type that allows the use of strings as values. It is possible to assign acronyms to parameters and scalars. However, acronyms, which are character string values, can be used in logical conditions only with the 'EQ', '=', 'NE' and '<>' operators.

Thus for example, the internal set IGCND(G), which holds all technologies of the extraction type (cf. Section 3.8.9) is defined as IGEXT(G) = YES$(GDATA(G,'GDTYPE') EQ GEXT);

The acronyms used for identifying technology types are like internal sets (cf. Section 3.8 page 38) in the sense that they should not be changed (unless accompanying code to handle the changes are also made); this is unlike the situation for the acronyms used for handling fuels, cf. Section 3.4 page 35, they are user defined.

3.3.2 Short term storages

There are three types of energy storages. One is hydro with reservoir, this type is intended for representation of energy storages as found in e.g. Norway, Sweden and Finland, where water is stored and used for generation of electricity. Typically, the energy may be stored for a year or more.

The other two types of storage are of a shorter term nature, intended for handling daily or weekly cycles. There are electricity and heat storages. Note that the characterisation as electricity storage only means that what comes out of and what goes into the storage count as electricity, irrespective of how the energy is physically stored; for instance it may be stored as hydrogen, as elevated water or as compressed air.

Short term storages for heat and electricity are implemented as follows. A storage is a technology, member of IGESTO (electricity storage) or IGHSTO (heat storage). The following description is given for heat storages, see the comments on electricity storages below.

It is assumed that there is at most one heat storage in any area.

The storage capacity is given in GKFX (MWh). Loading and unloading capacities (hours to load to or unload from full capacity) are given in GDATA(*,'GDSTOHLOAD') and GDATA(*,'GDSTOHUNLD'). The variable VGH_T(IA,IGHSTO,S,T) (MW) takes heat from the storage (unloading).

The variable VHSTOLOADT(IA,S,T) (MW) takes heat to the storage (loading). This heat comes from all heat production in IA.

The heat content QHSTOVOLT at the beginning of time segment T+1 is equal to the heat content at the beginning of time segment T plus loading during time segment T minus unloading, this is expressed in equation QHSTOVOLT.

The expression is cyclical over T in each S, i.e. the last label T in any S is linked to the first label T in the same S, as if the first label followed after the last one (see also Section 3.2.4). This has two important consequences. One is that during one S there is complete balance between what goes into the storage and what goes out (under the assumption of no loss). The good thing about this is that there is no need to pay special attention to end conditions to ensure this reasonable property.

The other consequence is that the storage contents at the end of one season need not equal the storage contents at the beginning of the following season. The good thing of this is precisely that it does not enforce cyclical patterns that are linked between seasons. This is relevant if for instance the time resolution within the year is chosen to specify four representative typical weeks (spring, summer, autumn, winter), each in full hourly resolution (in total 672 time segments). Here it seems inappropriate to enforce that the initial storage content of the summer week equals the final storage content of the spring week, since in real time there are several other weeks between those two weeks. (Using an addon (to be described later) it will be possible to enforce the equality of storage contents over season borders.)

It is assumed that there is no derating for IGHSTO (implemented such that GKDERATE is not used, but GKDERATE must be positive). It is assumed that there is no loss in the storage. Further, there is no associated fuel (GDATA(IGHSTO, 'GDFUEL') is not used but must be 0), no fuel consumption (GDATA(IGHSTO, 'GDFE') is not used but must be positive), no emission (IM_CO2(IGHSTO) and IM_SO2(IGHSTO) and GDATA(IGHSTO,'GDNOX') are not used but must be 0), etc.

It is essential to note the intention and caveats of the implementation of the short term storages. This is linked to the user's intention with and representation of the time structure within the year.

In an extreme case where card(S) and card(T) are one, i.e., there is only one time segment within each year, it does not make sense to have short term storage (but there is no need to delete it from the model, due to the cyclical nature of the storage energy balances will remain). Similarly, if e.g. card(S) is 52 and card(T) is 168, intended to represent 168 one-hour time segments over each week of the year, it makes sense to represent the storage in full hourly detail. Hence, some care must be given to the segmentation of time and the relationship to storage modelling. Consistent with Balmorel's potential for a flexible representation of time various possibilities are implemented.

If card(S) and card(T) are sufficiently large (meaning that all essential loading and unloading of the heat storage will be visible on that time representation) it will make sense to represent the short terms storage in the intuitive way. In this case the consistency between power (measured in MW), energy contents (measured in MWh) and time (measured in hours) is ensured by the use of IHOURSINST (in equations QHSTOVOLT and QESTOVOLT), which holds the length (hours) of each time segment (S,T). Thus, for model Balbase3, the values of elements in IHOURSINST are essentially one hour.

For the intermediate case, for instance if the time structure within a year is represented by 12 seasons S each subdivided into 18 T, it would not make much sense to attempt to represent the approximately 30 real daily cycles within each S by 30 model cycles. On the other hand, it might be desirable to have some representation of a storage. This may be achieved by an approach where it is assumed that there are a number of identical storage cycles within each season and that it suffices to represent one of these.

In order to have consistency between power (measured in MW), energy contents (measured in MWh) and time (measured in hours) in this case it is necessary to adjust the influence of the time segment.

Consider a case with card(S)=4, card(T)=42, meant to represent 4 seasons within a year (each season consisting of approximately 13 weeks or 2160 hours). The average length of T will be 2160/42 or approximately 51 hours. Suppose that the demand pattern given on T has 7 hills and 7 valleys and that this should be interpreted to mean that one typical week is represented on each of the 4 seasons. Hence, the "week" is understood to be repeated 13 times during one season (i.e., there are 13 "cycles").

Suppose that the dynamic equation for the short term storage were given as indicated here

$$Vol_{S,T+1} = Vol_{S,T} + (load_T - unload_T) * weight_{S,T}$$

Then the changes in storage volume ($Vol_t - Vol_{t+1}$) during one time segment would be about 13 times larger than intended, since the intended length of the representative average time segment T is not 51 hours but approximately 4 hours.

Therefore the dynamic equation should more appropriately be given as

$$Vol_{S,T+1} = Vol_{S,T} + (load_T - unload_T) * weight_{S,T}/CYCLESINS(S)$$

where parameter CYCLESINS(S) is the number of cycles within one season, 13 in this example.

If the above mentioned demand pattern (DH_VAR_T for heat, DE_VAR_T for electricity) given on T (card(S)=4, card(T)=42) had one hill and one valley then this could be interpreted to mean that one typical day is represented on each of the 4 seasons. In this case CYCLESINS(S) would take a value of approximately 90. In the other above mentioned case (card(S)=52, card(T)=168) CYCLESINS(S) would take the value 1.

From the above equations for the storage volume it follows that the reasonable balances (i.e., no storage energy disappears nor emerges from nothing) will be kept irrespective of the exact value of CYCLESINS(S), as long as CYCLESINS(S) is positive and less than infinity. However, as also seen, the value of CYCLESINS(S) will for given $(load_T - unload_T) * weight_{S,T}$ influence the magnitude of $Vol_{S,T}$. If therefore the storage capacity (volume) is of importance, the value of CYCLESINS(S) also is. This is in particular the case if analysis of the size of the storage is in focus, e.g. in relation to investments in new storage capacity. Note that for a storage with lower and upper limits that are to be respected for all time periods it is essential that a representative sequence of the time periods within each season is chosen. Also note that the above mechanism will be consistent with GDATA(*,'GDSTOHLOAD') and GDATA(*,'GDSTOHUNLD'), where these input values are given in hours.

Electricity storages are very similar to heat storages, mainly the names have a 'E' instead of a 'H'. The one important real difference is that operation of electricity storage may imply generation of heat, which is used in district heating. This feature may be relevant for e.g. hydrogen storage where part of the loss may be exploited as heat. The amount of heat generation is given as electricity generation divided by GDATA(IGESTO,'GDCB'), and this quantity enters equation QHEQ. If no such heat should be included, let GDATA(IGESTO,'GDCB')=INF.

3.4 Fuels: FFF, FDATASET, FKPOTSET

SET FFF is the set of fuels in the structure, given as e.g.

> SET FFF /NUCLEAR, NGAS, COAL-HIGHS, COAL-LOWS, LIGNITE, FUELOIL, SHALE, PEAT, WIND1800, WIND2300, WATER, BIO, SUN, ELEC, GARBAGE /;

Observe that in all simulations the whole set FFF is used. If therefore a particular fuel is not desired, the technology that uses it could be excluded from GGG and G, or from G.

The set of fuels is divided into three subsets by definitions of SET FKPOTSETC(FFF), set FKPOTSETR(FFF) and SET FKPOTSETA(FFF). The subsets need not be mutually exclusive, nor need they together constitute FFF.

The subsets indicate whether the members have their potentials specified at the level of country, region or area, respectively.

The following could the examples of definitions:

> SET FKPOTSETC(FFF) / NUCLEAR, LIGNITE, SHALE, PEAT /;

> SET FKPOTSETR(FFF) / WIND, WATER, SUN, BIO /;

> SET FKPOTSETA(FFF) / NGAS, WASTE /;

Thus, if e.g. COAL and FUELOIL are included in the set FFF, no limit will be placed on the use of these fuels.

SET FDATASET is the set of attributes of fuels:

> SET FDATASET / FDNB, FDCO2, FDSO2 , FDN2O / ;

The FDNB contributes to the coupling between generation technology and fuel. In GDATASET (Section 3.3) the elements GDFUEL for each technology contains an integer that points to the FDNB for the fuel that the technology uses, cf. also Sections 4.8.1 and 4.13.1.

Observe that the user should not change the set FDATASET without proper knowledge of the functioning of the set. Thus, the set can not be reduced from that specified above since data will be needed in the model for each of the elements, see Section 4.8.1. The set may be enlarged with new elements, however then the user will have to specify in the model how these elements are to be used.

Comment on naming conventions: See page 31.

Acronyms for fuels

THE FOLLOWING IS NEW FOR VERSION 3.01. Acronyms are used to handle fuel names. These acronyms are user defined (in contrast to acronyms for technology types, cf. Section 3.3.1 page 32).

The user should ensure that the acronyms for fuels precisely match the elements in set FFF

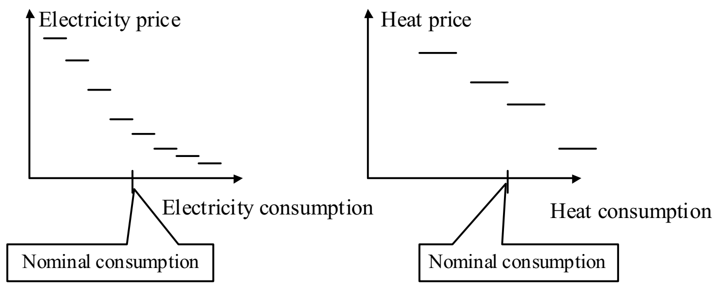

Figure 2: Elastic demand, illustration of price elasticities.

3.5 New generation technology and area: AGKN

Investment in new generation capacity may be determined endogenously. The specification of where new technology capacity of a particular type can be placed must therefore be determined. This is done by specifying the product set AGKN(AAA,G) that hold those combinations of areas and technologies that permit new investment.

Implicit restrictions on new investments, following from the information given in FKPOTC, FKPOTR, FKPOTA and GDATA(G,"GDKVARIABL"), Sections 4.17.1, 4.16.1, 4.15.1, 4.13.1, will automatically be used (through the derived set IAGKN, Section 3.8.10) , therefore it is not necessary to specify AGKN(AAA,G)=NO for such (AAA,G).

See Section 3.8.9 concerning specification of capacity according to heat or electricity side.

Other possibilities are described in Section 12.

3.6 Demand: DF_QP, DEF_..., DHF_...

Demand for electricity is specified for each region and demand for heat is specified for each area.

The specification may be considered to consist of three elements, see also Figure 2 and Figure 3:

- A nominal value, specified for each year in the simulation period as an annual quantity, parameters DE and DH, Sections 4.11.2 and 4.12.1.
- A nominal profile, i.e., a distribution of the annual quantity over the time segments of the year, specified in DE_VAR_T and DH_VAR_T, see Sections 4.20.1 and 4.19.1.
- An elasticity function which specifies the relationship between quantity and price for deviations from the nominal profile. Parameter values for this are given as described in Sections 4.6.3, 4.7.3, 4.19.6, 4.20.2, 4.29.34, 4.29.35, while the related sets are specified in the following.

The sets related to elastic demands specify steps relating quantities and prices:

SET DF_QP /DF_QUANT, DF_PRICE /;

Observe that the user should not change this set without proper knowledge of the functioning of the set. Thus, the set can not be reduced from that specified above since data will be needed in the model for each of the elements, see below.

The individual steps in the electricity demand function are specified by SET DEF given as e.g.:

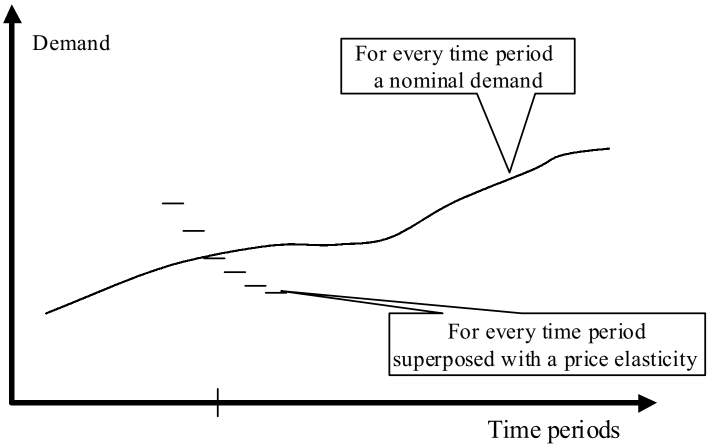

Figure 3: Elastic demand, illustration of development over time.

SET DEF / DEF_D1_4, DEF_D1_3, DEF_D1_2, DEF_D1_1, DEF_U1_1, DEF_U1_2, DEF_U1_3 / ;

This example shows 7 steps.

The entry order (Section 3.2.4) of the labels in DEF is important, cf. Section 4.25.1.

SET DEF_D1(DEF) and SET DEF_U1(DEF) are subsets used to distinguish between steps for regulation downwards (decreased demand, in this example 4 steps) and upwards (increased demands, in this example 3 steps) of electricity demand relative to nominal demand:

SET DEF_D1(DEF) / DEF_D1_4, DEF_D1_3, DEF_D1_2, DEF_D1_1/ ;

SET DEF_U1(DEF) / DEF_U1_1, DEF_U1_2, DEF_U1_3 / ;

If the sets DEF_D1(DEF) and DEF_U1(DEF) are empty then the intention is that demand is inelastic, according to the interpretation relative to DEF_STEPS, see Section 4.25.1.

Four other subsets of DEF are DEF_D2(DEF), DEF_U2(DEF), DEF_D3 and DEF_U3. The functioning of these sets is the same as for DEF_D1(DEF) and DEF_U1(DEF). The difference is the way the numerical values entered in DEF_STEPS will be interpreted, see Section 4.25.1.

In order to permit empty sets in relations to inelastic demand, the dollar control option $ONEMPTY must be set.

Similarly for the steps in the heat demand function (the example has 5 steps, of which 4 are down and 1 is up; set DHF_U2 is empty):

SET DHF / DHF_D1_2, DHF_D2_1, DHF_U1_1, DHF_D2_1, DHF_U2_2 /;

SET DHF_D1(DHF) / DHF_D1_2, DHF_D1_1/ ;

SET DHF_U1(DHF) / DHF_U1_1 / ;

SET DHF_D2(DHF) / DHF_D2_1, DHF_U2_2 / ;

SET DHF_U2(DHF) / / ;

SET DHF_D3(DHF) / DHF_D3_1, DHF_U3_2 / ;

SET DHF_U3(DHF) / / ;

The same comments apply to DHF_D1, DHF_U1, etc. as to the similar sets for electricity.

3.7 Emission policies: MPOLSET

SET MPOLSET contains elements for specification of environmental policies for each country,

TAX_CO2 "CO2 emission tax (Money/t CO2)"

TAX_SO2 "SO2 emission tax (Money/t SO2)"

TAX_NOx "NOx emission tax (Money/kg NOx)"

LIM_CO2 "Annual CO2 limit (t CO2/year)"

LIM_SO2 "Annual SO2 limit (t SO2/year)"

LIM_NOx "Annual NOx limit(kg NOx/year)"

Observe that the user should not change this set without proper knowledge of the functioning of the set. Thus, the set can not be reduced from that specified above since data will be needed in the model for each of the elements, see Section 4.24.1. The set may be enlarged with new elements, however then the user will have to specify in the model how these elements are to be used.

3.8 Internal sets

A number of sets are defined and their members defined automatically, i.e., they are not specified explicitly by the user. We refer to these as internal sets, in contrast to input sets. The names of these sets start with I, Section 1.4. The internal sets are dynamic sets in the sense explained next.

In a certain sense also the acronyms holding technology types, cf. Section 3.3.1 32 are internal (although not sets), since the shall not be handled by the user as input.

3.8.1 Static and dynamic sets

In the GAMS terminology, static sets are sets that have their membership declared as the SET itself was declared and the membership was never changed. In contrast, dynamic sets have their membership changed because of assignments. Hence, membership of dynamic sets may change during the execution of the program.

In assignments, constructions of sets may be done using the symbols "+", "-", "*" and "NOT" to provide the set operations union, difference, intersection and complement, respectively. Constructions using YES and NO may be used, see below for examples.

Dynamic sets are not ordered, Section 3.2.4. They can not be used in declarations but can be used in definitions, see page 13.

3.8.2 Areas simulated: IA

SET IA(AAA) is the subset used to define those areas that are simulated. This subset is derived automatically as that subset of AAA that is relevant for the simulated countries C:

SET IA(AAA) = YES$(SUM(C,ICA(AAA,C)));

3.8.3 Rural and urban heat areas simulated: IARURH, IAURBH

THE FOLLOWING WILL BE DELETED IN VERSION 3.01. The subset IARURH used in simulation is found automatically as that subset of AAARURH, Section 3.8.3, that is relevant for the simulated countries C:

SET IARURH(AAARURH) = YES$(SUM(C,ICA(AAARURH,C)));

The subset IAURBH of urban heat areas is similarly automatically defined as the set of the urban heat areas.

3.8.4 Country to area mapping: ICA

The internal set ICA(C,AAA) specifies the relation between AAA and C. It is derived automatically from the sets RRRAAA(RRR,AAA) and CCCRRR(C,RRR), to assign consistently the areas in AAA to the countries in C:

ICA(C,AAA)=YES$(SUM(RRR, (RRRAAA(RRR,AAA) AND CCCRRR(C,RRR))));

3.8.5 Regions simulated: IR

SET IR(RRR) is the subset of regions that are simulated. This subset is derived automatically for the simulated countries C as:

SET IR(RRR) = YES$(SUM(C,CCCRRR(C,RRR)));

3.8.6 Electricity import-export: IRRRI, IRRRE, IRI, IRE

For description of transmission relations between pairs of regions copies of the sets are necessary. They are obtained by the ALIAS statement as:

ALIAS(RRR,IRRRE), ALIAS(RRR,IRRRI), ALIAS(IR,IRE), ALIAS(IR,IRI)

This permits the reference to pairs of regions, e.g., (IRI,IRE). As seen, IRRRE and IRRRE are the sets of regions in the data structure, and IRE and IRI are the subsets of regions in the simulation. The final E and I are used to indicate exporting and importing regions, respectively.

3.8.7 Season and time duplication: ISALIAS, ITALIAS

Copies of the sets S and T are obtained as "ALIAS(S,ISALIAS)" and "ALIAS(T,ITALIAS)".

3.8.8 Season and time refinements: IST, ISTS, ISTT

The set IST(S,T) identifies the time segments selected. In the standard version IST(S,T) holds all combinations of elements in S and T, i.e., IST is defined as IST(S,T)=YES, but a subset of this may be defined. Combined with overwriting of the standard definition of IHOURSINST(S,T) (Section 4.29.6) this permits maximal flexibility with respect to the time structure.

The set ISTS(S) holds the time segments S for which there is at least one element T.

The set ISTT(T) holds the time segments T appearing in at least one seasons S.

3.8.9 Generation technology types

A number of convenient subsets of generation technology types have been defined. There are 11 basic types: Condensing, Back pressure (sometimes also called intermediate take out and condensing), Extraction, Heat-only boilers, Electric heaters/heatpumps, Heat storage technologies, Electricity storage technologies, Hydro power with storage, Hydro power without storage (run-of-river), Wind power, Solar power (producing electricity). Some technologies may be combined into one, see Section 12.7.

The above names of the technologies are only indicative, and indeed may be misleading. E.g., a gas turbine may also in this context be characterised as a "back pressure" technology type. ¿From a functional point of view the characterisation below, as e.g. in Table 3, is more precise.

All technologies of each basic type is in a particular set. The names of these sets are, respectively, IGCND(G), IGBPR(G), IGEXT(G), IGHOB(G), IGETOH(G),

ACRONYM	CND	BPR	EXT	HOB	ETOH	WND	SOLE	SOLH	HYRS	HYRR	ESTO	HSTO
Electricity output	x	x	x			x	x		x	x	x	
Electricity input					x						(x)	
Heat output		x	x	x	x			x			(x)	x
Heat input												(x)
Storage									x		x	x
Dispatchable	x	x	x	x	x				x		x	x
Fixed electricity to heat relation		x			x						(x)	

Table 3: Functionality of the technology types.

IGHSTO(G), IGESTO(G), IGHYRS(G), IGHYRR(G), IGWND(G), IGSOLE(G), IGSOLH(G), IGCOMB1, IGCOMB2, IGNUC. IG2LEVEL.

Each technology is automatically allocated to one and only one of these sets according to the acronym (cf. Section 3.3.1 page 32) given in GDATA(GGG,'GDTYPE') as follows:

IGCND(G) = YES$(GDATA(G,'GDTYPE') EQ GCND);

IGBPR(G) = YES$(GDATA(G,'GDTYPE') EQ GBPR);

IGEXT(G) = YES$(GDATA(G,'GDTYPE') EQ GEXT);

IGHOB(G) = YES$(GDATA(G,'GDTYPE') EQ GHOB);

IGETOH(G) = YES$(GDATA(G,'GDTYPE') EQ GETOH);

IGHSTO(G) = YES$(GDATA(G,'GDTYPE') EQ GHSTO);

IGESTO(G) = YES$(GDATA(G,'GDTYPE') EQ GESTO);

IGHYRS(G) = YES$(GDATA(G,'GDTYPE') EQ GHYRS);

IGHYRR(G) = YES$(GDATA(G,'GDTYPE') EQ GHYRR);

IGWND(G) = YES$(GDATA(G,'GDTYPE') EQ GWND);

IGSOLE(G) = YES$(GDATA(G,'GDTYPE') EQ GSOLE);

IGSOLH(G) = YES$(GDATA(G,'GDTYPE') EQ GSOLH);

Combination technologies are in sets IGCOMB1 and IGCOMB2, see Section 12.7.

Observe that the sets are defined as subsets of G, not GGG.

The technologies in each set has specific properties in the model. The relative characteristics of each of the technology types are described in Table 3. As seen, all types are functionally different, except WND and SOLE. For these two the difference is only in the primary energy source (the wind and the sun, respectively), hence the reason for the difference is a matter of convenience in the distinction of input data like costs, production profiles over the day and the year, etc.

Since the allocation of the technologies to the sets is done according to the value of the ACRONYM in GDATA(GGG,'GDTYPE'), the user should not change the meaning associated with these ACRONYMS without proper understanding of the model. These properties are expressed as equations and/or lower and/or upper bounds and/or fixed values on the individual variables describing the generation from the technology. In other works, the classification of the technology into types is intrinsically linked to main functionalities in the model. See BALMOREL.GMS for details.

In addition, the technologies may be grouped into sets according to various characteristics. Some examples are: all technologies excluding electric heating; technologies for which the generation may be dispatched; technologies that produce electricity only i.e., not heat; technologies for which capacity is given with respect to electricity (IGKE) or heat (IGKH).

The identification of these technology sets are:

IGHH(G): producing only heat

IGEE(G): producing only electricity

IGE(G): producing electricity (with or without heat)

IGH(G): producing heat (with or without electricity)

IGEH(G): producing electricity and heat

IGHHNOSTO(G): type IGHH except heat storage

IGEENOSTO(G): type IGEE except electricity. storage

IGKH(G): capacity specified with respect to heat

IGKE(G): capacity specified with respect to electricity

IGKENOSTO(G): capacity given on electricity side, except el. storage

IGKHNOSTO(G): capacity given on heat side, except heat storage

IGNOTETOH(G): all except heat pumps

IGDISPATCH(G): dispatch may be made

IGEOREH(G): producing electricity (with or without heat)

IGKKNOWN(G): capacity can not be found endogeneously

IGKFIND(G): capacity can be found endogeneously

THE FOLLOWING WILL BE DELETED IN VERSION 3.01. IGHORHERUR(G): producing heat (with or without electricity), rural areas

Each technology is automatically allocated to these dynamic sets. For most of them, this is done according to the ACRONYM given in GDATA(GGG,'GDTYPE'), and/or using previously defined sets. Table 4 specifies the dynamic sets that are defined this way. The sets IGKKNOWN and IGKFIND are defined according to the values given in GDATA(G,'GDKVARIABL'). Other ways may be used. See BALMOREL.GMS for details.

??? HOVHOV: Further internal sets that may be in use are

IGCOMB1(G) = YES$(GDATA(G,'GDCOMB') EQ 1); (Section 12.7)

LOOP(IGCOMB1, IGCOMB2(G) = YES$((GDATA(G,'GDCOMB') EQ 2) AND (GGCOMB(IGCOMB1,G)))); (Section 12.7)

IGNUC(G)

IG2LEVEL(G)

HOVHOV???

The set IGGGALIAS is a duplication of the set GGG.

3.8.10 Investments in new technologies: IAGKN

The set AGKN, Section 3.5, specifies where new technologies may be invested. This information, supplemented as described in Section 3.5, is transferred to set IAGKN.

ACRONYM:	GCND	GBPR	GEXT	GHONLY	GETOH	GHSTO
Internal set:	IGCND	IGBPR	IGEXT	IGHONLY	IGETOH	IGHSTO
IGE	yes	yes	yes		yes	
IGH		yes	yes	yes	yes	yes
IGEH		yes	yes		yes	
IGEE	yes					
IGHH				yes		yes
IGEENOSTO	yes					
IGHHNOSTO				yes		
IGKE	yes	yes	yes		yes	
IGKH				yes		yes
IGKENOSTO	yes	yes	yes		yes	
IGKHNOSTO				yes		
IGNOTETOH	yes	yes	yes	yes		yes
IGDISPATCH	yes	yes	yes	yes	yes	yes

ACRONYM:	GESTO	GHYRS	GHYRR	GWND	GSOLE	GSOLH
Internal set:	IGESTO	IGHYRS	IGHYRR	IGWND	IGSOLE	IGSOLH
IGE	yes	yes	yes	yes	yes	
IGH						yes
IGEH						
IGEE	yes	yes	yes	yes	yes	
IGHH						yes
IGEENOSTO		yes	yes	yes	yes	
IGHHNOSTO						yes
IGKE	yes	yes	yes	yes	yes	
IGKH						yes
IGKENOSTO		yes	yes	yes	yes	
IGKHNOSTO						yes
IGNOTETOH	yes	yes	yes	yes	yes	yes
IGDISPATCH	yes	yes				

Table 4: Acronyms and internal dynamic sets with relation to generation technology types, i.e., depending on GDTYPE.

3.8.11 Equation feasibility: IPLUSMINUS

It is not practically possible to ensure that a model will have a feasible solution. And if it does not, it may be difficult to find the explanation why. Therefore a mechanism is introduced to ensure that a model will always be feasible, and to provide some kind of indication that may help in searching for a reason if it is not. The SET IPLUSMINUS /IPLUS, IMINUS/ is part of this, see further Section 4.1.1, Section 6. Note that the following convention is used: when placed on the left hand side of the relation sign (=L=, =E= or =G=) the sign to the 'IPLUS' and 'IMINUS' terms should be '+' and '-', respectively.

3.9 Restrictions on sets

Most of the sets have their members (elements, labels) specified by the user. The exceptions are:

- Some sets have their members derived automatically (they are the dynamic sets, or internal sets, Section 3.8.1); these sets are described in Section 3.8.
- For the sets GDATASET, FDATASET, DF_QP, MPOLSET, see Sections 3.3, 3.4, 3.6, 3.7, the members should not be changed without proper understanding of the functioning of these sets.
- THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. For the sets DEF, DEF_DINF, DEF_UINF, DHF, DHF_DINF and DHF_UINF there are restriction such that some specific elements must be present, see Section 3.6.
- The acronyms (which are not sets, actually) describing technology types should should not be changed without proper understanding of the functioning of these acronyms, cf. Section 3.3.1 page 32, cf. also e.g. Table 4 page 42.

The set members (labels) in relation to the three last items will be referred to as obligatory set members.

It is permitted that an element is member of more than one set. This is used e.g. in the declaration of subsets, and other examples are given in Section 3.8.9. However, apart from such intentional use, it should be avoided that elements are members of more than one set. Thus, it may be tempting to have e.g. a technology that is called HYDRO and also a fuel that is called HYDRO, but this should be avoided, see the discussion on naming conventions on page 31 in Section 3.3.

For similar reasons, the only labels consisting of digits only are those used for set elements indicating the years (i.e., in labels the sets YYY and Y).

Finally, as previously noted, the user should not change the internal sets, cf. Section 3.8.

4 Parameters and scalars

In this section we describe parameters and scalars that must be specified by the user while in Section 4.29 parameters that are automatically calculated will be treated. The former type will be referred to as input parameters while the latter type will be referred to as internal parameters.

Recall from Section 1.3 that parameters and scalars are used to specify exogenous values, and that parameters, unlike scalars, relate to sets.

Recall that the focus in the present document is on model structure and therefore the actual input data to be used is not in focus. However, occationally some comments on input data will be given. To avoid confusion between what is logically required within the model structure and what may reasonable be expected concerning numerical values of input the comments on input data values will be clearly identified as such.

The units used in parameters are: MW (megawatt), MWh (megawatthour), GJ, hours, days, kg (kilogram), t (ton), and Money where the latter may indicate e.g. Euro or USD. Prefixes like M (million), k (kilo) or m (milli) will also be used.

Note that the term Money is used in parts of the code, for instance in Balmorel.gms, as a generic term. This is to indicate that the code there is actually independent of whether the currency used is e.g. EURO, dollars or DKK. However, in the data files it does matter what currency is used!

For entities like 'loss in electricity distribution' (which must be given as a fraction) or 'the number of hours per year' the unit has been indicated as '(none)'. For some entities the important thing is their proportions, in this case also '(none)' may be specified, however, an indication may be given as concerns between which entities the proportions should be taken; e.g. '(none∼MW)' to signal that the proportions are between MW or similar, see Section 4.20.1.

The factor 3.6 indicates the usual relations between units using seconds and hours, respectively, e.g. between MWh and GJ. The meaning of the numbers 24, 365, 8760 and others are obvious.

Most data are entered using a list (for parameters) or a TABLE (for parameters with two or more dimensions). Observe that if entries are not given, or entry values are not filled in, the default value zero is automatically used. In some cases where individual data can not be found, or can not be found for all relevant entries, user specified default data may be entered by first giving a TABLE with those values that are known, and then filling all other entries with the user specified default data, see Section 4.27. See Section 4.15.1 on the use of EPS.

In general, the data in any model must be selected by the user to be consistent in the sense that model and data are, generally speaking, mutually dependent, and therefore the individual data elements are also interdependent. This topic will not be discussed in general terms here. However, in a few places it is crucial that there is a logical consistency between parameter values, this is discussed in Section 4.28.

4.1 Scalars

4.1.1 PENALTYQ

The scalar PENALTYQ is a penalty used in relation to securing feasibility in equations, it enters the equation QOBJ as coefficient to variables VQ..., cf. Section 6, Section 3.8.11.

It is the user's responsibility to supply an appropriate values to PENALTYQ, large enough to dominate any other costs, and small enough to avoid numerical problems.

The reason for introducing the variables VQ... is that it may be very hard to find out what to do in case of infeasibilities. If an infeasibility exists in a single equation (in combination with lower and/or upper bounds on the variables entering the equation) it is simple, and GAMS will detect and report this. However, if infeasibility is the result of an interplay between several equations, it is not possible on theoretical grounds to identify the reason or even to indicate where to look for one. In our experience the study of which VQ variables are positive may be very useful. If such variables are positive, an error message will be written, cf. Section 11.

Not all equations have a VQ variable. An idea has been that if the zero point (corresponding to all variables set to zero) is feasible in an equation, and if the zero point is feasible for all entering individual variables, then no VQ is necessary.

Note that there might be models where some of the variables VQ are not needed. For instance, the equation QEEQ, presently formulated as an equality, could be formulated as an inequality, expressing that total supply should not be smaller than total demand. This could be relevant if e.g. a large share of wind power is present, and electricity overflow might occur. In this case, only one of the VQ variables is needed (figure out yourself which one). But be aware that the question is not simple. If you for instance have also CHP in the model, electricity overflow could be the result of forcing satisfaction of heat demand, and it is in this case not obvious that the result should be electricity overflow and not heat underflow. (Indeed, the mentioned error messages may be incomplete for the same reason.)

In summary, therefore, remember that the VQ variables are introduced originally with the purpose of diagnostics of infeasibilities, but that there might be reasons to modify this.

4.2 Parameters on the set YYY

4.2.1 YVALUE

The parameter YVALUE(YYY) holds the numerical values related to the years in set YYY. Unit: (none). If e.g. set YYY is defined as /2001 * 2003/ then YVALUE('2002') has the value 2002.

4.3 Parameters on the set SSS

4.3.1 WEIGHT_S

PARAMETER WEIGHT_S reflects how much of the year each season represents expressed relatively between the seasons. Unit: (none), see next.

One way of doing it is to state the number of days in each season (could sum up to 365). Another is to give percentages (summing up to 100), and more (infinitely many, actually) ways are possible.

Example: Assume the definition SET SSS / S01*S04 /. Let WEIHGT_S('S01')=1, WEIHGT_S('S02')=1, WEIHGT_S('S03')=1, WEIHGT_S('S04')=1. This specifies that seasons have the same length, which with 8760 hours per year gives 2190 hours per season (8736 hours per year gives 2184 hours per season, etc.). With WEIHGT_S('S01')=1.5, WEIHGT_S('S02')=1.5, WEIHGT_S('S03')=1.5, WEIHGT_S('S04')=1.5 again there are 2190 hours per season, this illustrates that only the relative sizes of the numbers are important.

See also Section 4.4.1. The parameter WEIGHT_S is used only in the calculation of the parameters IWEIGHSUMS and DAYSIN_S, see Sections 4.29.4 and 4.29.7.

Comment on input data: The calculation of IDAYSIN_S will involve a division. This will be unproblematic if the data entered in WEIGHT_S contain no negative values and at least one positive value (which is natural).

(It is quite possible to specify the year to have 366 (or even 365.24) days. Just take the editor and replace 365 by 366 (or 365.24) in the BALMOREL.GMS file (and 8760 should be changed accordingly to 8784 etc.). This will make the numerical values change slightly while some interpretations will be harder.)

Comment on naming conventions: See Section 3.2.2.

Comment on input data: See the comments on input data in Section 4.4.1.

4.3.2 CYCLESINS

PARAMETER CYCLESINS(SSS) holds the number of cycles per season. It is of importance in relation to short term storages, see Section 3.3.2.

4.4 Parameters on the set TTT

4.4.1 WEIGHT_T

PARAMETER WEIGHT_T reflects how much of the season each time segment represents expressed relatively between the time segments. Unit: (none), see next.

	5%	6%	7%	8%	9%	10%	15%	20%
5	0.2310	0.2374	0.2439	0.2505	0.2571	0.2638	0.2983	0.3344
10	0.1295	0.1359	0.1424	0.1490	0.1558	0.1627	0.1993	0.2385
15	0.0963	0.1030	0.1098	0.1168	0.1241	0.1315	0.1710	0.2139
20	0.0802	0.0872	0.0944	0.1019	0.1095	0.1175	0.1598	0.2054
25	0.0710	0.0782	0.0858	0.0937	0.1018	0.1102	0.1547	0.2021
30	0.0651	0.0726	0.0806	0.0888	0.0973	0.1061	0.1523	0.2008

Table 5: Annuity as depending on interest rates and number of years (investment at the beginning of the first year, payments at the end of the years).

One way of doing it is to state the number of hours that each period represents, another is to state it as percentages (summing up to 100), and there are very many other ways, cf. Section 4.3.1.

The parameter WEIGHT_T is used only in the calculation of the parameters IWEIGHSUMT and IHOURSIN24, see Sections 4.29.5 and 4.29.8.

It tempting to say that the set TTT represents a subdivision of the day - and we may actually do so sometimes. However, is not in general correct to say so. Thus, if for instance SSS contains four elements (wither, spring, summer, autumn) and TTT contains 24 segments, the time segments in TTT need not represent a typical day. Rather, TTT should represent not only the "typical day" but also the week-end days. See also Section 4.29.8

Comment on input data: The calculation of IHOURSIN24 will involve a division. This will be unproblematic if the data entered in WEIGHT_T contain no negative values and at least one positive value (which is natural).

Comment on input data: The set T is ordered, cf. Section 3.2.4. If this is used in the model to represent chronological aspects then the sequence of numbers entered in TABLE WEIGHT_T matters. This is the case e.g. if pumped hydro reservoirs or heat storage shall be modelled. However, if this is not so, WEIGHT_T may be used to represent only the duration (weight) of the individual time segments. Then, if e.g. the electricity loads (demand) as expressed in PARAMETER DE_VAR_T appear in descending magnitude the load duration idea is applied. Similarly it is important that the set S is ordered if e.g. a hydro reservoir with seasonal storage capacity is to be modelled.

Comment on naming conventions: See Section 3.2.2.

4.5 Parameters on the set CCC

4.5.1 ANNUITYC

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER ANNUITYC indicates the transformation of an investment to an annual payment. Unit: (none).

Thus, for instance, an investment of 100 at the beginning of the first year, repaid over 20 years, with payment at the end of each year including the first one, assuming an interest rate of 5%, will imply an annual payment of 8.02, hence, ANNUITYC should in this case have the value 0.0802. Alternatively, the numbers may be interpreted to mean that the net present value (NPV) at the beginning of this year of payments of 0.0802 at the end of each of the following 20 years (starting with this year) is 1, assuming a calculation rent of 5%.

For electrical transmission investments between regions in two different countries, the average annuity between the annuities for the two countries in question will be used.

Table 5 illustrates the dependence of the annuity on various combinations of interest rates, 5, 6, 7, 8, 9, 10, 15, 20% and number of years, 5, 10, 15, 20, 25, 30 years. The formula is See also Section 12.

4.5.2 TAX_DE

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER TAX_DE holds consumers' tax on electricity consumption. Unit: Money/MWh.

Comment on input data: Observe that the tax must be specified as the weighted average over all consumer groups.

4.5.3 TAX_DH

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER TAX_DH holds consumers' tax on heat consumption. Unit: Money/MWh.

Comment on input data: Observe that the tax must be specified as the weighted average over all consumer groups.

4.6 Parameters on the set RRR

4.6.1 DISLOSS_E

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER DISLOSS_E holds the loss in electricity distribution, as a fraction of the electricity entering the distribution network. Unit: (none).

4.6.2 DISCOST_E

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER DISCOST_E holds the cost of electricity distribution, given relative to end consumption. Unit: Money/MWh.

4.6.3 DEFP_BASE

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER DEFP_BASE holds the annual average consumer price of electricity (including taxes) in the base year. Unit: Money/MWh.

Comment on input data: Observe that the average is to be taken over the whole year and over all consumer groups, and that the price is including taxes (that may differ between the different consumer groups).

Comment on input data: Also in the case of inelastic demand a reasonable value must be given, as the value in DEFP_BASE will be also in this case be taken as starting point for calculation of a "very high" price, used in case the demand can not be satisfied, see also Section 4.25.1.

4.7 Parameters on the set AAA

4.7.1 DISLOSS_H

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER DISLOSS_H holds the loss in heat distribution, as a fraction of heat generated (identical to the heat entering the distribution network). Unit: (none).

4.7.2 DISCOST_H

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER DISCOST_H holds the cost of heat distribution, given relative to end consumption. Unit: Money/MWh.

4.7.3 DHFP_BASE

PARAMETER DHFP_BASE holds the annual average consumer price of heat (including taxes) in the base year. Unit: Money/MWh.

Comment on input data: Similar comments as in Section 4.6.3 apply.

4.7.4 DISLOSS0E DISCOST0E, DISLOSS1E, DISCOST1E, DISLOSS0H, DISCOST0H, DISLOSS1H, DISCOST1H

THE FOLLOWING IS NEW FOR VERSION 3.01.

PARAMETER DISLOSS0E, PARAMETER DISCOST0E, PARAMETER DISLOSS1E, PARAMETER DISCOST1E, PARAMETER DISLOSS0H, PARAMETER DISCOST0H, PARAMETER DISLOSS1H, PARAMETER DISCOST1H

Replacing and refining parameters DISLOSS_E etc.

4.7.5 WNDFLH

PARAMETER WNDFLH holds the full load hours for wind power, i.e., the annual wind power production divided by the wind power capacity. Unit: hours.

4.7.6 SOLEFLH

PARAMETER SOLEFLH holds the full load hours for solar power, i.e., the annual solar power production divided by the solar power capacity. Unit: hours.

4.7.7 SOLHFLH

PARAMETER SOLHFLH holds the full load hours for solar heat, i.e., the annual solar heat production divided by the solar heat capacity. Unit: hours.

4.7.8 WTRRRFLH

PARAMETER WTRRRFLH holds the full load hours for hydro run of river power, i.e., the annual hydro run of river power production divided by the hydro run of river power capacity. Unit: hours.

4.7.9 WTRRSFLH

PARAMETER WTRRSFLH holds the full load hours for hydro with storage power, i.e., the annual hydro with storage power production divided by the hydro with storage power capacity. Unit: hours.

4.8 Parameters on the set product (FFF,FDATASET)

4.8.1 FDATA

PARAMETER FDATA contains information about emission characteristics of fuels. In addition it contains an integer code FDNB identifying the individual fuels. Units: kg/GJ (for FDCO2), kg/GJ (for FDSO2), (none) for FDNB.

THE FOLLOWING WILL BE DELETED IN VERSION 3.01. The FDNB contributes to the coupling between generation technology and fuel. In GDATASET (Section 3.3) the elements GDFUEL (Section 4.13.1) for each technology contains an integer that points to the FDNB for the fuel that the technology uses. Therefore two different fuels should not have identical FDNB.

THE FOLLOWING IS NEW FOR VERSION 3.01. Acronyms may be applied. Acronyms will replace FDNB.

4.9 Parameters on the set product (FFF,CCC)

4.9.1 TAX_F

PARAMETER TAX_F specifies fuel taxes on primary fuel types (i.e. neither electricity nor heat). This tax is applied on the fuel, IRRREspective of whether electricity, heat or both is produced. Unit: Money/GJ.

4.10 Parameters on the set product (GGG,AAA)

4.10.1 GINVCOST

PARAMETER GINVCOST holds the investment cost for new technology. Unit: MMoney/MW.

Observe the definition of the capacity (MW), Section 4.22.1.

Observe that if a zero or if nothing is specified for GINVCOST in TABLE GINVCOST (and therefore the default value zero is automatically assigned) then the value in table GDATA, Section 4.13.1, is used.

4.10.2 GOMVCOST

PARAMETER GOMVCOST holds the variable operating and maintenance costs. Unit: Money/MWh.

Observe that if a zero or if nothing is specified for GOMVCOST in TABLE GOMVCOST (and therefore the default value zero is automatically assigned) then the value in table GDATA, Section 4.13.1, is used.

4.10.3 GOMFCOST

PARAMETER GOMFCOST holds the annual fixed operating and maintenance costs. Unit: kMoney/MW.

Observe the definition of the capacity (MW), Section 4.22.1.

Observe that if a zero or if nothing is specified for GOMFCOST in TABLE GOMFCOST (and therefore the default value zero is automatically assigned) then the value in table GDATA, Section 4.13.1, is used.

4.10.4 GEFFDERATE

PARAMETER GEFFDERATE represents an adjustment of efficiency. Unit: (none).

Comment on input data: This parameter is intended for catching some of the shortcomings in the modeling of the individual units. Thus, the information on efficiency given in GDATA may be seen as general information with validity irrespective of where the unit is located, this is then made geographically specific through GEFFDERATE. The value of GEFFDERATE will probably be slightly below unity (but despite the name it may also be above unity). See also Section 9.1.2.

4.11 Parameters on the set product (YYY,RRR)

4.11.1 X3FX

PARAMETER X3FX contains the annual net electricity export to third regions. Unit: MWh.

Comment on input data: Observe that the values in X3FX must be specified to be consistent with the values in X3FX_VAR_T, Section 4.20.3. X3FX is only used to calculate IX3FX_T_Y, see Section 4.29.36.

Observe that this exchange (intended to be positive for export, negative for import, but from Section 4.29.36 it follows that some care is necessary to get consistency) is specified by the user, and that there is no other exchange possibilities with regions or countries not in the model (i.e., not in the sets C or R). Also observe that no payment is associated with this exchange. No capacity is to be given for the transmission lines supposed to carry the exchange (and hence no interaction with e.g. XKINI, Section 4.14.1).

If the set C is a proper subset of the set CCC X3FX may be used to represent exchange between a region in C and a region which is in CCC but not in C.

Exchange between regions in the model (i.e., between members in the set IR) will be found during the simulation as the endogenous value of variable VX_T, see Section 6.

Electricity exchange with regions not in IR, depending on import-export prices, is described in Section 12.10.

4.11.2 DE

PARAMETER DE contains the nominal annual electricity consumption. Unit: MWh.

The value should be the end consumption, since distribution and transmission losses are accounted for separately.

The nominal annual consumption will be distributed over the time segments over the year, see Section 3.2. If demand is elastic, there may be deviation from this nominal value, see Section 3.6.

4.12 Parameters on the set product (YYY,AAA)

4.12.1 DH

PARAMETER DH contains the nominal annual heat consumption in those areas that are heat areas, Section 3.1.4. Unit: MWh.

The value should be the end consumption, since distribution losses are accounted for separately.

The nominal annual consumption will be distributed over the time segments over the year, see Section 3.6. If demand is elastic, there may be deviation from this nominal value, see Section 3.6.

4.13 Parameters on the set product (GGG,GDATASET)

4.13.1 GDATA

PARAMETER GDATA contains information about the individual generation technologies.

> THE FOLLOWING WILL BE DELETED IN VERSION 3.01. GDTYPE: This is an integer. According to the value of this integer, the technology is uniquely placed in one of the internal sets specified in Section 3.8.9. According to the value of the integer (and hence according to which of those sets the technology belongs), the technology has specific properties, see Section 3.8.9.
>
> THE FOLLOWING IS NEW FOR VERSION 3.01. An acronym will take the place of GDTYPE.
>
> GDFUEL: This is an integer indicating which fuel the technology uses, corresponding to the fuel number FDNB given in FDATA (Section 4.8.1).
>
> GDCB: This value specifies the Cb-value for back pressure and extraction type technologies.
>
> GDCV: This value specifies the isofuel constant Cv for extraction type technologies
>
> GDFE: Fuel efficiency
>
> GDESO2: Degree of desulphuring
>
> GDNOX: NOx-factor (mg/MJ)
>
> GDAUXIL: Used for various additional information. For CHP it denotes central (urban) or decentral (rural) technology

GDINVCOST0: Default investment cost (MMoney/MW) (default value). Will be used, if nothing is specified for GINVCOST, Section 4.10.1. Observe the definition of capacity, Section 3.8.9

GDOMVCOST0: Default variable operating and maintenance costs (Money/MWh) (default value). The cost is specified with respect to the total energy (electricity plus heat) Will be used, if nothing is specified for GOMVCOST, Section 4.10.2.

GDOMFCOST0: Default annual operating and maintenance costs (kMoney/MW) (default value). Observe the definition of capacity, Section 3.8.9 Will be used, if nothing is specified for GOMFCOST, Section 4.10.3.

GDFROMYEAR: technology available from the beginning of this year

GDKVARIABL: Capacity is a variable to be found for each year (0: no, 1: yes)

GDSTOHLOAD: number of hours to fully load the short term storage from empty

GDSTOHLOAD: number of hours to fully unload the short term storage from full

GCOMB: the technology is a combination technology, see Section 12.7.

GDCOMBSK: relevant for a combination technology, see Section 12.7.

GDCOMBSLO: relevant for a combination technology, see Section 12.7.

GDCOMBSUP: relevant for a combination technology, see Section 12.7.

The data corresponding to the elements GDINVCOST0, GDOMVCOST0 and GDOMFCOST0 are considered as default values that may be overwritten, see Sections 4.10.1, 4.10.2 and 4.10.3.

Observe that the following will be specified automatically in the model:

GDATA(IGBPR,'GDCV')=1;

GDATA(IGHOB,'GDCV')=1;

This ensures homogeneous ways of calculating fuel consumption. Therefore these obligatory values need not be given by the user (i.e., the corresponding entries in TABLE GDATA may be left blank).

4.14 Parameters on the set product (IRRRE,IRRRI)

4.14.1 XKINI

PARAMETER XKINI contains the initial electrical transmission capacities between pairs of regions. Unit: MW.

The electrical transmission capacity is the capacity disregarding an eventual loss (see XLOSS, Section 4.14.4). Thus, if there is a loss XLOSS, a maximum of XKINI MW may be sent into the transmission line, but at most (XKINI*(1-XLOSS)) MW may be extracted.

Observe that the initial transmission capacity between two regions need not be the same in both directions. (But new transmission capacity will be symmetric.)

4.14.2 XINVCOST

PARAMETER XINVCOST contains contains information about the investment cost in new electrical transmission capacity between pairs of regions. It also contains information about where it will at all be possible to establish new transmission lines. Unit: Money/MW.

Observe the definition of transmission capacity, Section 4.14.1.

If INF is entered in the table, this means that no transmission capacity can be established between the two associated regions.

The information in XINVCOST must be given only for the lower triangle of the table, and not for diagonal entries, i.e. only for (ORD(IRRRE) GT ORD(IRRRI)).

Note: in most cases the intention will be that no investment will be possible.

Compare the tables providing the other information relating to transmission (XKINI, XCOST, XLOSS).

4.14.3 XCOST

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER XCOST contains information about the electrical transmission cost between pairs of regions. Unit: Money/MWh.

The electrical transmission cost is applied to the electricity entering the transmission line, cf. Section 4.14.1.

Observe that the cost need not be the same in both directions.

Comment on input data: Unreasonable results may be found if there are neither cost nor loss associated with electrical transmission. Therefore for all non-diagonal entries the user must enter a positive number in either TABLE XCOST or in TABLE XLOSS, Section 4.14.4.

THE FOLLOWING IS NEW FOR VERSION 3.01. Will probably be replaced by PARAMETER XCOST1 .

4.14.4 XLOSS

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER XLOSS contains the loss in transmission expressed as a fraction of the electricity entering the transmission line. Unit: (none).

Observe that the loss need not be the same in both directions, Section 4.14.1.

Comment on input data: Results that are unreasonable, or difficult to interpret or compare between cases, may be found if there are neither cost nor loss associated with electrical transmission. Therefore for all non-diagonal entries the user must enter a positive number in either XLOSS or in XCOST, Section 4.14.3.

The following reasoning will apply to import, export and loss between two regions R1 and R2 in time segment (S,T). The (non-negative) transmission from area R1 to area R2 is given by the variable VX_T(R1,R2,S,T), and the (non-negative) transmission from area R2 to area R1 is given by the variable VX_T(R2,R1,S,T) (both measured in MW). Assuming that either the loss or the cost (or both) on transmission is positive, then either VX_T(R1,R2,S,T) or VX_T(R2,R1,S,T) (or both) is zero. Assume that VX_T(R2,R1,S,T) is zero at the optimal solution. Then VX_T(R1,R2,S,T) is the electricity leaving R1 towards R2, the loss is VX_T(R1,R2,S,T)*XLOSS(R1,R2), and the electricity entering R2 is VX_T(R1,R2,S,T)*(1-XLOSS(R1,R2)). To get the three entities in energy terms for the time segment (S,T) multiply each of them by IHOURSINST(S,T).

THE FOLLOWING IS NEW FOR VERSION 3.01. Will probably be replaced by PARAMETER XLOSS1 .

4.15 Parameters on the set product (FKPOTSETA,AAA)

4.15.1 FKPOTA

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER FKPOTA holds the fuel potentials specified at area level. Unit: MW.

The potential is specified as an upper limit on the area's installed generation capacity relative to the individual fuel type.

Comment on input data: The specification of the fuel potential as an upper limit on generation capacity is obviously a simplification. However, in many cases the fuel potential, defined in energy terms, is not known with precision and this motivates the simplification. See Section 12.

Note the following on input format and use of EPS. In GAMS the default value for parameters and scalars is 0, which is convenient. In some cases, however, it would be nice to be able to distinguish between a 0 which means 'this entry is not relevant' and a 0 which means 'this entry takes the value 0'. This is indeed possible. Consider for example FKPOTA(FKPOTSETA,IA). If input for an index combination ('NatGas','Stockholm') specifies 0, or nothing at all, the numerical value will be 0. Therefore a condition like "$FKPOTA(FKPOTSETA,IA)" will for ('NatGas','Stockholm') evaluate to "false". However, if FKPOTA('NatGas','Stockholm') is given the value EPS then "$FKPOTA(FKPOTSETA,IA)" will evaluate to "true". Consequently, if the entered value is 0 (due to entering this number in the input file, or due to entering nothing) then the equation definition

QKFUELA(IA,FKPOTSETA)$FKPOTA(FKPOTSETA,IA)..

will imply that no equation will be generated. But if FKPOTA('NatGas','Stockholm') is given the value EPS, then the equation will be generated for ('NatGas','Stockholm') - and FKPOTA('NatGas','Stockholm') will be numerically 0.

4.16 Parameters on the set product (FKPOTSETR,RRR)

4.16.1 FKPOTR

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER FKPOTR holds the fuel potentials specified at regional level. Unit: MW.

The potential is specified as an upper limit on the regions' installed generation capacity relative to the individual fuel type.

Comment on input data: The same comments as in Section 4.15.1 apply.

4.17 Parameters on the set product (FKPOTSETC,CCC)

4.17.1 FKPOTC

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. PARAMETER FKPOTC holds the fuel potentials specified at country level. Unit: MW.

The potential is specified as an upper limit on the countries' installed generation capacity relative to the individual fuel type.

Comment on input data: The same comments as in Section 4.15.1 apply.

4.18 Parameters on the set product (AAA,SSS)

4.18.1 WTRRSVARS

PARAMETER WTRRSVARS contains the description of the seasonal variation of the amount of water inflow to the hydro reservoirs with storage. Unit: (none∼MW).

The water is assumed available at the beginning of each season.

Note that in contrast to other variation profiles, this one is related to seasons S, not to time segments (S,T).

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1996	186.0	248.1	230.6	245.4	260.3	258.8	250.0	303.4	342.9	296.4	225.4	233.5
1997	227.6	163.2	126.0	123.1	113.5	111.6	87.5	132.2	104.6	129.2	158.0	173.0
1998	163.2	147.0	131.2	122.9	107.3	120.0	69.7	49.8	78.2	108.4	142.5	152.7
1999	139.4	127.1	105.6	86.5	87.1	74.3	53.9	105.1	131.1	134.5	125.8	140.5
2000	124.0	104.2	92.0	103.1	61.0	75.2	48.6	75.8	103.8	118.9	131.2	135.1
2001	168.7	220.8	211.1	215.4	191.7	200.7	177.9	170.5	173.5	150.7	168.7	185.2
2002	192.3	157.9	143.7	132.7	113.3	108.7	108.1	144.4	177.9	230.0	316.2	550.1
2003	532.5	365.2	318.1	256.8	234.1	195.9	228.3	274.4	266.3	288.8	301.7	266.4
2004	250.5	242.8	254.6	249.2	230.8	267.4	253.3	273.6	244.2	229.8	240.8	215.6
2005	189.1	208.9	241.2	251.4	250.0	207.3	228.4	245.6	229.3	251.8	238.7	273.2
2006	324.1	349.8	418.1	406.5	293.4	345.8	393.1	531.4	525.2	450.4	385.6	273.1
2007	228.1	233.2	193.7	182.2	174.0	190.9	140.1	131.9	197.6	281.5	362.7	369.0

Table 6: Monthly average prices (NOK/MWh) for the system spot price in the Nordpool area (source: www.nordpool.com).

Comment on input data: Sources for WTRRSVARS may often specify energy contents (i.e., ~MWh), not power (~MW). If the lengths of all seasons (as expressed in WEIGHTS_S, Section 4.3.1) are identical, this is no problem, otherwise some rescaling may have to take place for such sources.

4.18.2 HYPPROFILS

PARAMETER HYPPROFILS contains the description of the seasonal variation of prices in relation to production of electricity from hydro power with storage. Unit: Money/MWh.

The purpose of HYPROFILS is to force a price profile onto the 'water value', and have it reflected in the resulting average prices found in a simulation. Typically this will be high prices during winter and low prices during summer; this in turn will influence how the water is used over the year. Only the differences between the prices in the individual seasons are important.

HYPPROFILS is intended for use with equation QHYRSSEQ, Section 7.

As example, Table 6 gives the monthly average prices for the system spot price in the Nordpool area. One year's data may be used (after conversion into the same currency as used in other input data) for HYPROFILS. If 1999 data are used the monthly price profile resulting from simulation will be relatively flat, slightly lower in summer than in winter. If 2002 is used, the resulting monthly price profile will be similar to 1999 for the first half year, and then it rises sharply, reflecting the relatively low reservoir filling towards the end of the year. Note that irrespective of the HYPPROFILS chosen, all the available water will be used (e.g. for the 2002 profile therefore more will be used in the first half of the year, and less in the second half)

The yearly average price will be largely independent of HYPROFILS (it will depend more distinctly on the amount of water available during the year).

4.19 Parameters on the set product (AAA,SSS,TTT)

4.19.1 DH_VAR_T

PARAMETER DH_VAR_T contains the description of seasonal and daily variation of the heat demand Unit: (none~MW) (see description in relation to DE_VAR_T, Section 4.20.1).

4.19.2 SOLE_VAR_T

PARAMETER SOLE_VAR_T contains the description of seasonal and daily variation of the solar electricity generation. Unit: (none~MW) (see description in relation to DE_VAR_T, Section 4.20.1).

4.19.3 SOLH_VAR_T

PARAMETER SOLH_VAR_T contains the description of seasonal and daily variation of the solar heat generation. Unit: (none~MW) (see description in relation to DE_VAR_T, Section 4.20.1).

4.19.4 WND_VAR_T

PARAMETER WND_VAR_T contains the description of seasonal and daily variation of the wind power generation. Unit: (none∼MW) (see description in relation to DE_VAR_T, Section 4.20.1).

4.19.5 WTRRRVAR_T

PARAMETER WTRRRVAR_T contains the description of the seasonal and daily variation of the amount of water inflow to the hydro electricity generation without storage (run of river). Unit: (none∼MW) (see description in relation to DE_VAR_T, Section 4.20.1).

4.19.6 DHFP_CALIB

PARAMETER DHFP_CALIB is used to calibrate the price side of the demand function for heat in order to get demand consistent for a base year. Unit: Money/MWh.

The intention with this parameter is the following. Balance between supply and demand is obtained as a consequence of the costs on the supply side and the demand function. The demand function is exogenously specified, and the supply function is found in the model (in the sense that supply costs may be calculated). However, it may be unlikely that the model and data will reproduce accurately the situation in a base years, such that the simulated demand need not correspond to that observed in the base year. The parameter DHFP_CALIB may then be given a value different from zero to obtain such correspondence.

The parameter is used in the calculation of IDHFP_T, Section 4.29.34 and Section 9.

4.20 Parameters on the set product (RRR,SSS,TTT)

4.20.1 DE_VAR_T

PARAMETER DE_VAR_T contains the description of seasonal and daily variation of the electricity demand. Unit: (none∼MW) (see below).

DE_VAR_T is used to calculate IDE_SUMST (Section 4.29.9) and DE_VAR_T and IDE_SUMST in combination with DE are used to calculate IDE_T_Y (Section 4.29.32).

The values in DE_VAR_T are interpreted to be specified relatively (i.e. the values for each day or for all time segments do not have to sum up to something specific, only the relative values are important) within each region. One way to do this is to specify each season/time period value as a percentage of the yearly maximum power load. Another option is to specify the

MW-loads for each combinations.

In any case it is important to note that the values must be derived from data given with the dimension of power, i.e., energy per time unit, e.g. MW, GW, J/s MJ/s, and not with dimension of energy, e.g. MHh or MJ (this is the meaning of the "∼MW" in the specification of the unit none∼MW).

The relationship between DE_VAR_T, hourly and annual electricity demand demand is discussed in Section 4.29.32.

Comment on input data: The calculation of parameter IDE_T_Y, Section 4.29.32, involves a division. This will be unproblematic if the data entered in DE_VAR_T contain no negative values and at least one positive value (which is natural).

Observe that parameters DE_VAR_T, DH_VAR_T, WND_VAR_T, SOLE_VAR_T, X3FX_VAR_T will be handled in similar ways as DE_VAR_T (special care should be taken with X3FX_VAR_T, see Section 4.20.3).

4.20.2 DEFP_CALIB

PARAMETER DEFP_CALIB is used to calibrate the price side of the demand function for electricity in order to get demand consistent for a base year. Unit: Money/MWh.

See the explanation in relation to DHFP_CALIB, Section 4.19.6.

The parameter is used in the calculation of IDEFP_T, Section 4.29.35 and Section 9.

4.20.3 X3FX_VAR_T

PARAMETER X3FX_VAR_T contains the description of seasonal and daily variation in the fixed exchange with third regions. Unit: (none∼MW) (see description in relation to DE_VAR_T, Section 4.20.1).

A positive number is intended to indicate net export from the region in the set RRR, a negative number indicates net import to the region in the set RRR for the given time segment (S,T), however see Section 4.29.36 about consistency. The values are seen from the country in the set CCC, i.e., any losses are disregarded.

X3FX_VAR_T is only used to calculate X3FX_T_Y, see Section 4.29.36.

Comment on input data: Observe that the values in X3FX_VAR_T must be specified to be consistent with the values in X3FX, Section 4.11.1, cf. Section 4.29.36.

4.21 Parameters on the set product (GGG,AAA,SSS)

4.21.1 GKDERATE

PARAMETER GKDERATE represents a reduction in capacity. Unit: (none).

This reduction may represent e.g. forced and scheduled outages. It is used to reduce the capacity of each unit type in each area.

Observe that for all technology types the specification of GKDERATE has implication for capacity of both electricity and heat (if both are relevant).

Comment on input data: If GKDERATE is set ot zero for some paricular combination (G,A,S) then this represents an outage for the whole season, typically a planned outage.

To represent stochastic outages for thermal units a reasonable value will probably be close to but smaller than 1. For other types of units (e.g., wind, hydro, solar, heatpumps) the value should probably be equal to 1; however, this will depend on a number of factors, e.g. the data source. See also Section 9.1.1.

4.22 Parameters on the set product (YYY,AAA,GGG)

4.22.1 GKFX

PARAMETER GKFX holds the exogenously specified generation capacities. Unit: MW.

For electricity generation plants and co-generation plants capacity must be specified with respect to electricity generation. For heat only boilers and electrical heating units the capacity must be specified with respect to heat generation. See Section 3.8.9, in particular the sets IGKH and IGKE.

Observe that generation capacities are considered specified as net capacity, i.e., the generation unit is assumed to be able to deliver such amount to the network (distribution or transmission network for electricity, distribution network for heat). On the other hand, the delivery may be modified by GKDERATE, see Section 4.21.1.

For electricity generation plants and co-generation plants this must be specified with respect to electricity generation. For heat only boilers and electrical heating units the capacity must be specified with respect to heat generation. See Section 3.8.9.

	DEF_D1_3	DEF_D1_2	DEF_D1_1	DEF_U1_1	..	DEF_U1_9
..DF_QUANT	0.94	0.96	0.97	1.02	..	1.40
..DF_PRICE	1.29	1.18	1.10	0.90	..	0.81

Table 7: Illustration of TABLE DEF_STEPS

4.23 Parameters on the set product (YYY,AAA,FFF)

4.23.1 FUELPRICE

PARAMETER FUELPRICE contains fuel prices. Unit: Money/GJ.

Comment on input data: Fuels like wind, water, sun or electricity are not expected to be given a positive value (and hence if no value is assigned the default value zero will be used).

4.24 Parameters on the set product (YYY,MPOLSET,CCC)

4.24.1 M_POL

PARAMETER M_POL contains emissions policy data. Unit: See Section 3.7.

For each year in the simulation the data is transferred to internal parameters, see Sections 4.29.26 ff.

Observe the following

4.25 Parameters on the set product (RRR,SSS,TTT,DF_QP,DEF)

4.25.1 DEF_STEPS

PARAMETER DEF_STEPS describes the elastic electricity demands in relative terms, by quantifying the steps. Units: (none), or MW and Money/MWh. See below.

The numerical values may be specified in three ways. Each way is associated with particular subsets of DEF.

In the subsets DEF_D1 and DEF_U1 the values are specified relative to 1.0. Thus, with e.g. quantity values of 0.97 and 0.96 for DEF_D2 and DEF_D1, respectively, this means that the first step in decrease of demand has a magnitude of 3% of demand (relative to demand DE) and the second step has a magnitude of 1%, adding up to 4%. See Table 7. Corresponding to these quantity steps the price steps have to be specified, e.g. as 1.18 and 1.10, respectively. Thus the 3% decrease in consumption is the result of a 10% increase in price (relative to DEFP_BASE). Similar ideas apply to increasing demand and decreasing price.

Observe that the sequence of quantities should be increasing and the sequence of prices should be decreasing.

It should be obvious that the sequence of the labels in DEF (which determine the 'real' sequence (i.e., the ORD of a label) of the labels in DEF_D1 and DEF_U1) is important, since differences between values corresponding to neighboring labels will be used (using the ORD operator).

In the subsets DEF_D2 and DEF_U2 the numerical values are specified in absolute terms. Hence, quantities are specified in MW and prices are specified in Money/MWh.

town In the subsets DEF_D3 and DEF_U3 the numerical values are specified...[text to be given]

The following few lines of GAMS code will, in relation to DEF_D1 and DEF_U1, generate a linear function for electricity demand. It may be placed after the definition of DEF_STEPS and before DEF_STEPS if used.

```
/* The following specifies the elasticity in electricity demand as a linear function type (specifying price
as a function of quantity) on DEF_D1 and DEF_U1. ISCALAR1 is the steplength (0<ISCALAR1<1) in
price, where e.g. -0.1 (must be negative) means that an increase in consumption of one step (see below)
will be associated with a decrease in price of 10% of the price at nominal demand. ISCALAR2 is the
steplength (0<ISCALAR2<1) in quantity, where e.g. 0.05 means that each step is 5% of the quantity
```

```
at nominal demand. The elasticity at nominal demand is then approximately (ISCALAR1/ISCALAR2).
The set IANYSET must be sufficiently large to embrace all steps DEF_D1 and DEF_U1. */
SET IANYSET / IANYSET001 * IANYSET100/;
ISCALAR1=-0.1; ISCALAR2= 0.1;
LOOP(DEF_D1$(ORD(DEF_D1) EQ CARD(DEF_D1)),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1)=1-ISCALAR1;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1)=1-ISCALAR2;);
LOOP(IANYSET$((2 LE ORD(IANYSET)) AND (ORD(IANYSET) LE CARD(DEF_D1))),
LOOP(DEF_D1$(ORD(DEF_D1) EQ (CARD(DEF_D1)-ORD(IANYSET)+1)),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1)=DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1+1)-ISCALAR1;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1)=DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1+1)-ISCALAR2;
));
LOOP(DEF_U1$(ORD(DEF_U1) EQ 1),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1)=1+ISCALAR1;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1)=1+ISCALAR2;);
LOOP(DEF_U1$(ORD(DEF_U1) GE 2), DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1)=DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1-
1)+ISCALAR1);
LOOP(DEF_U1$(ORD(DEF_U1) GE 2), DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1)=DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1-
1)+ISCALAR2);
```

The following few lines of GAMS code will, in relation to DEF_D1 and DEF_U1, generate a CES (constant elasticity of substitution) function for electricity demand. It may be placed after the definition of DEF_STEPS and before DEF_STEPS if used.

```
/* The following specifies the elasticity in electricity demand as a CES (constant elasticity of sub-
stitution) type. ISCALAR1 is the elasticity, where e.g. -0.1 (must be negative) means that an in-
crease in price of 1% will imply a decrease in consumption of 0.1%. ISCALAR2 is the steplength
(0<ISCALAR2<1) in price, where e.g. 0.05 means that each step is 5% of the price. The set IANY-
SET must be sufficiently large to embrace all steps DEF_D1 and DEF_U1. */
SET IANYSET / IANYSET001 * IANYSET100/;
ISCALAR1=-0.1; ISCALAR2= 0.1;
LOOP(DEF_D1$(ORD(DEF_D1) EQ CARD(DEF_D1)),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1)=1+ISCALAR2/2;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1)=1+ISCALAR1*ISCALAR2/2;);
LOOP(IANYSET$((2 LE ORD(IANYSET)) AND (ORD(IANYSET) LE CARD(DEF_D1))),
LOOP(DEF_D1$(ORD(DEF_D1) EQ (CARD(DEF_D1)-ORD(IANYSET)+1)),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1)=
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1+1)*(1+ISCALAR2);
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1)=
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1+1)*(1+ISCALAR1*ISCALAR2);));
LOOP(DEF_U1$(ORD(DEF_U1) EQ 1),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1)=1-ISCALAR2/2;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1)=1-ISCALAR1*ISCALAR2/2;);
LOOP(DEF_U1$(ORD(DEF_U1) GE 2),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1)=
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_U1-1)*(1-ISCALAR2));
LOOP(DEF_U1$(ORD(DEF_U1) GE 2),
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1)=
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_U1-1)*(1-ISCALAR1*ISCALAR2));
```

4.26 Parameters on the set product (AAA,SSS,TTT,DF_QP,DHF)

4.26.1 DHF_STEPS

PARAMETER DHF_STEPS describes the elastic heat demands in relative terms, by quantifying the steps. Units: (none).

See the description of the similar construction for electricity demand, Section 4.25.1.

4.27 Default values

If no data is assigned to a parameter or scalar the default value zero is automatically assigned. Unless zero incidentally is a suitable value another method of assigning default values therefore has to be used.

The following construction may be applied, indicated by an example. As seen, a default value 0.9 is used and assigned unless where non-zero values have been assigned in the TABLE:

TABLE GEFFDERATE(GGG,AAA)

	DK_E_Urban	LT_R_Urban	
CC-Cond1	0.94	0.99	
ST-BP-2-C		0.87	;

GEFFDERATE(G,A)$(GEFFDERATE(G,A) EQ 0)=0.9;

(Observe that not all elements in a parameter can be given by default, at least one must explicitly be given a value (in a TABLE or by assignment), otherwise it is considered an error.)

See further the GAMS manual on "dollar on the left" and "dollar on the right"

THE FOLLOWING IS NEW FOR VERSION 3.01. To be given: a description of use of EPS in input data. scalar sc1 /9/, sc2 /0/; sc1 = ((1+2+3)/sc2)($sc2ne0$); /$*$ $OK, sc1isnotassignedhere, stillhasvalue9*/sc1 = ((1+2+3)/sc2)$sc2; /* OK, sc1 is not assigned here, still has value 9 */ sc2=eps; sc1 = ((1+2+3)/sc2)($sc2ne0$); /$*$ $OK, sc1isnotassignedhere, stillhasvalue9*/sc1 = ((1+2+3)/sc2)$sc2; /* Division by zero error */

4.28 Restrictions on parameter values.

To ensure the basic functioning of the model, some parameter values must be consistent or attain some obligatory values:

- The pointer GDFUEL from technology type to fuel type FDNB must be correct, cf. Section 4.8.1. Hence, if the user changes the set of fuels, and therefore also the FDNB to some or all of the individual fuels, this must be carefully checked.

- For some parameters undesirable consequences will be encountered for certain values. Thus, leaving certain values unspecified, such that the default values zero is assigned, will result in a division-by-zero error, eg. Sections 4.29.32, 4.29.33, 4.20.3.

- GDATA(IGBPR,'GDCV') and GDATA(IGHONLY,'GDCV') must attain unity value (assigned automatically), Section 4.13.1.

- PENALTYQ should definitely be positive, and it should be clearly larger than ay other values to which is is implicitly compared in optimization. Consult a textbook on optimization and numerical analysis to get wiser.

4.29 Internal parameters and scalars

A number of scalars and parameters have been defined as part of the model. In this section we describe those scalars and parameters, called internal parameters, for which the values are derived from other input data, i.e. the user is not supposed to specify the values of these internal parameters. The names of these internal parameters all start with the letter I.

Another group of internal parameters is used to hold values for printing output, see Section 5. These parameters are not part of the model in the sense that they influence model results, hence they are referred to as auxiliary parameters. The names of these auxiliary parameters all start with the letter O.

4.29.1 IBALVERSN

The scalar IBALVERSN holds the version number of the Balmorel model. The value 211.20040305 indicates version 2.11 from March 5th. 2004, etc.

4.29.2 ISCALAR1, ISCALAR2, IOF

The scalars ISCALAR1, ISCALAR2 etc. may be used to hold intermediate values of various kind, their meanings and units therefore being context dependent. The should be reset to zero or whatever relevant value where they are used. They shall not be used for transferring values between files.

A number of scalars have been defined, intended for holding the most common multipliers. The names start with the text string 'IOF' ('internal or output related factor'), this permits a convenient search for any of these constants in the editor. Some examples are: SCALAR IOF1000 'Multiplier 1000' /1000/; SCALAR IOF1000000 'Multiplier 1000000' /1E6/; SCALAR IOF0001 'Multiplier 0.001' /0.001/; SCALAR IOF0000001 'Multiplier 0.000001' /0.000001/; SCALAR IOF3P6 'Multiplier 3.6' /3.6/; SCALAR IOF24 'Multiplier 24' /24/; SCALAR IOF8760 'Multiplier 8760' /8760/; SCALAR IOF8784 'Multiplier 8784' /87684/; SCALAR IOF365 'Multiplier 365' /365/.

4.29.3 IANYSET

The set IANYSET may be used for any purpose, its meanings therefore being context dependent.

A particular use of IANYSET is for construction of loops backwards through sets. Newer versions of GAMS permit the use of the constructions "for(c=1 to 5, ...)" and "for(c=1 downto 5, ...)". In accordance with the principle on application of GAMS versions, cf. Section 1.3, the core part of the Balmorel model will not use these constructions. The loop backwards through a set (e.g., S) will in stead be coded as follows:

```
loop(IANYSET$(ord(IANYSET) le card(S)),
loop(S$(ord(S) eq (card(S)-ord(IANYSET)+1)),
...
));
```

4.29.4 IWEIGHSUMS

The internal parameter IWEIGHSUMS is used to hold the total weight of the time of each season in S, hence it is calculated from WEIGHT_S(S), Section 4.3.1, as

IWEIGHSUMS = SUM(S,WEIGHT_S(S))

Observe that the sum must be over set S, not SSS. Unit: (none), cf. Section 4.3.1.

4.29.5 IWEIGHSUMT

The internal parameter IWEIGHSUMT is used to hold the total weight of the time of each time period in T, hence it is calculated from WEIGHT_T(T), Section 4.4.1, as

IWEIGHSUMT = SUM(T, WEIGHT_T(T))

Observe that the sum must be over set T, not TTT. Unit: (none), cf. Section 4.4.1.

4.29.6 IHOURSINST

The internal parameter IHOURSINST(S,T) holds the length of the time segment (S,T) measured in hours. Unit: (hours). It is defined as THE FOLLOWING WILL BE DELETED IN VERSION 3.01.

IHOURSINST(S,T)=IDAYSIN_S(S)*IHOURSIN24(T)

For advanced use, it may be defined differently, cf. Section 3.8.8.

4.29.7 IDAYSIN_S

THE FOLLOWING WILL BE DELETED IN VERSION 3.01. The time structure within one year of the model is indicated by the division of the year into seasons SSS and the division of the day in any season into time periods TTT, cf. Section 3.2.

The internal parameter IDAYSIN_S(S) is defined from the weights, Sections 4.3.1 and 4.29.4, as

IDAYSIN_S(S) = 365*WEIGHT_S(S) / IWEIGHSUMS

It indicates the length of each season given as the number of days that are in each season. Unit: (none). As seen, it is assumed that there are 365 days in the year. Observe that the sum is over the set S, not SSS. See Section 4.29.8.

4.29.8 IHOURSIN24

THE FOLLOWING WILL BE DELETED IN VERSION 3.01.

The internal parameter IHOURSIN24(T) is defined from the weights, Sections 4.29.8 and 4.29.5, as

IHOURSIN24(T) = 24*WEIGHT_T(T) / IWEIGHSUMT

It indicates the length of each period of the season given as the number of hours that would be in each time segment, if the length of the season were 24 hours. Unit: (none). Observe that the sum is over the set T, not TTT, similarly to the case in the following parameters.

4.29.9 IDE_SUMST

The internal PARAMETER IDE_SUMST holds the annual amount of electricity demand as expressed in the units of the weights and demands used in IHOURSINST(S,T) and DE_VAR_T,

IDE_SUMST(IR) = SUM((S,T), IHOURSINST(S,T)*DE_VAR_T(IR,S,T))

Unit: (none∼MWh).

Observe that IDE_SUMST may be interpreted as the annual nominal electricity demand as contained in the specification of variation profile DE_VAR_T and time weighting as contained in IHOURSINST; thus, it is not identical to the nominal annual electricity demand contained in DE, see further Section 4.29.32.

4.29.10 IDH_SUMST

The internal PARAMETER IDH_SUMST holds the annual amount of heat demand as expressed in the units of the weights and demands used in IHOURSINST(S,T) and DH_VAR_T,

IDH_SUMST(IA) = SUM((S,T), IHOURSINST(S,T)*DH_VAR_T(IA,S,T))

Unit: (none∼MWh). See also Section 4.29.9, Section 4.20.1 and Section 4.29.32. The use is described in Section 4.29.33.

4.29.11 IWND_SUMST

The internal PARAMETER IWND_SUMST holds the annual amount of wind generated electricity as expressed in the units of the weights and demands used in IHOURSINST(S,T) and WND_VAR_T,

IWND_SUMST(IA) = SUM((S,T), IHOURSINST(S,T)*WND_VAR_T(IA,S,T))

Unit: (none∼MWh). See also Section 4.20.1 and Section 4.29.32.

4.29.12 ISOLESUMST

The internal PARAMETER ISOLESUMST holds the annual amount of solar generated heat as expressed in the units of the weights and demands used in IHOURSINST(S,T) and SOLE_VAR_T,

ISOLESUMST(IA) = SUM((S,T), IHOURSINST(S,T)*SOLE_VAR_T(IA,S,T))

Unit: (none∼MWh). See also Section 4.20.1 and Section 4.29.32.

4.29.13 ISOLHESUMST

The internal PARAMETER ISOLHSUMST holds the annual amount of solar generated electricity as expressed in the units of the weights and demands used in IHOURSINST(S,T) and SOLH_VAR_T,

ISOLHSUMST(IA) = SUM((S,T), IHOURSINST(S,T)*SOLE_VAR_T(IA,S,T))

Unit: (none∼MWh). See also Section 4.20.1 and Section 4.29.32.

4.29.14 IHYINF_S

The internal PARAMETER IHYINF_S holds the seasonal amount of inflow to hydro reservoirs. Unit: (MWh/MW).

It is calculated as THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

(WTRRSFLH(IARURH) * WTRRSVAR_S(IARURH,S) * IDAYSIN_S(S))
/ IWTRRSSUM(IARURH);

4.29.15 IWTRRRSUM

The internal PARAMETER IWTRRRSUM holds the annual amount of hydro run-of-river generated electricity, similarly as IWND_SUMST in Section 4.29.11.

Unit: (none∼MWh).

4.29.16 IWTRRSSUM

The internal PARAMETER IWTRRSSUM holds the annual amount of hydro run-of-river generated electricity, similarly as IWND_SUMST in Section 4.29.11.

Unit: (none∼MWh).

4.29.17 IX3FXSUMST

The internal PARAMETER IX3FXSUMST holds the annual amount of electricity exported to third countries as expressed in the units of the weights and demands used in IHOURSINST(S,T) and X3FX_VAR_T,

ISOLESUMST(IR) = SUM((S,T), IHOURSINST(S,T)*X3FX_VAR_T(IR,S,T))

Unit: (none∼MWh). See also Section 4.20.1 and Section 4.29.32.The use is described in Section 4.29.36.

4.29.18 IM_CO2

The internal parameter IM_CO2 attaches the CO2 emission coefficient for the fuel to the technology using that fuel. Unit: kg/GJ. See BALMOREL.GMS.

4.29.19 IM_SO2

The internal parameter IM_SO2 combines the SO2 emission coefficient for the fuel with the efficiency for technology using that fuel. Unit: kg/GJ. See BALMOREL.GMS.

4.29.20 IGKVACCTOY

The internal PARAMETER IGKVACCTOY holds the internally found generation capacity at the beginning of the currently simulated year. Unit: MW.

The value of IGKVACCTOY is equal to the sum of the generation capacities found endogenously by simulation in the years previous to the currently simulated year. Total capacity at the beginning of the currently simulated year is equal to (IGKVACCTOY+IGKFX_Y), see Section 4.29.21. Total capacity throughout the currently simulated year is VGKN.L (where VGKN.L is the level (value) that VGKN attains) larger than (IGKVACCTOY+IGKFX_Y). Compare IXKINI_Y in Section 4.29.22.

4.29.21 IGKFX_Y

The internal PARAMETER IGKFX_Y holds the externally given (parameter GKFX, see Section4.22.1) generation capacity at the beginning of the currently simulated year. Unit: MW.

Total capacity at the beginning of the currently simulated year and throughout this year are described in Section 4.29.20.

4.29.22 IXKINI_Y

The internal PARAMETER IXKINI_Y holds the electrical transmission capacity at the beginning of the year simulated. Unit: MW.

The capacity throughout the currently simulated year is VXKN.L (where VXKN.L is the level (value) that VXKN attains) larger than IXKINI_Y. Compare IGKVACCTOY in Section 4.29.20.

4.29.23 IXKN

The internal PARAMETER IXKN holds the information whether investment will at all be possible or permitted between pairs of regions. It is derived from information in XINVCOST, Section 4.14.2. Note that only the lower triangle of the matrix will be used.

4.29.24 IAGK_Y

The internal set IAGK_Y(AAA,G) holds those (IA,G) for which there is some generation capacities (MW) at the beginning of the year. It it updated at the beginning of the year based on the exogeneous capacity specified in GKFX (Section 4.22.1) and accumulated endogeneously found capacity IGKVACCTOY (Section 4.29.20) up to the beginning of the year.

4.29.25 IFUELP_Y

The internal parameter IFUELP_Y holds the fuel price in the year simulated, transferred from parameter FUELPRICE, Section 4.23.1. Unit: Money/GJ.

4.29.26 ITAX_CO2_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. The internal PARAMETER ITAX_CO2_Y indicates environmental policy parameter for a given year and country. Unit: Money/ton.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.27 ITAX_NOX_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. The internal PARAMETER ITAX_NOX_Y indicates environmental policy parameter for a given year and country. Unit: Money/kg.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.28 ITAX_SO2_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01. The internal PARAMETER ITAX_SO2_Y indicates environmental policy parameter for a given year and country. Unit: Money/ton.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.29 ILIM_CO2_Y

The internal PARAMETER ILIM_CO2_Y indicates environmental policy parameter for a given year and country. Unit: ton.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.30 ILIM_SO2_Y

The internal PARAMETER ILIM_SO2_Y indicates environmental policy parameter, for a given year and country. Unit: ton.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.31 ILIM_NOX_Y

The internal PARAMETER ILIM_NOX_Y indicates environmental policy parameter for a given year and country. Unit: kg.

During simulation the relevant values in MPOLSET, Section 4.24.1, will be transferred to this internal parameter.

4.29.32 IDE_T_Y

The internal PARAMETER IDE_T_Y holds the nominal electricity demand for each time segment in the current simulation year. Unit: MW.

It is calculated using the input parameters DE_VAR_T and DE and the internal parameter IDE_SUMST as

IDE_T_Y(IR,S,T) = (DE(Y,IR) * DE_VAR_T(IR,S,T)) / IDE_SUMST(IR);

See Section 4.11.2 and Section 3.6 in relation to electricity demand, and Section 4.20.1 and Section 4.29.9 in relation to the variation profile.

Observe that the quotient (DE(Y,IR)/IDE_SUMST(IR)) expresses the relation between the annual nominal electricity demand DE in year Y and the annual nominal electricity demand IDE_SUMST, where the latter is expressed through parameters DE_VAR_T and IHOURSINST, cf. Section 4.29.9.

Also observe that the nominal electricity demand (expressed in MWh) for time segment (S,T) may be derived from the above as

IHOURSINST(S,T)* IDE_T_Y(IR,S,T)

Comment on input data: The calculation of IDE_T_Y involves a division. This will be unproblematic if the data entered in DE_VAR_T contain no negative values and at least one positive value (which is natural).

Observe that parameters DE_VAR_T, DH_VAR_T, WND_VAR_T, SOLE_VAR_T, X3FX_VAR_T will be handled in similar ways as DE_VAR_T (special care should be taken with X3FX_VAR_T, see Section 4.20.3).

4.29.33 IDH_T_Y

The internal PARAMETER IDH_T_Y holds the nominal heat demand for each time segment in the current simulation year. Unit: MW.

It is calculated using the input parameters DH_VAR_T and DH and the internal parameter IDH_SUMST as

IDH_T_Y(IA,S,T) = (DH(Y,IA) * DH_VAR_T(IH,S,T)) / IDH_SUMST(IA);

See Section 3.6 and Section 4.29.32.

4.29.34 IDHFP_T

PARAMETER IDHFP_T holds the price levels of individual steps in the electricity demand function, transformed to be comparable with generation costs, taxes and distribution costs. Unit: Money/MWh.

Observe that the magnitudes of the quantity measure (MW) of the corresponding steps will be specified as upper bounds on the variables VDHF_T, cf. Section 8.

4.29.35 IDEFP_T

PARAMETER IDEFP_T holds the price levels of individual steps in the electricity demand function, transformed to be comparable with generation costs, taxes and distribution costs. Unit: Money/MWh.

Observe that the magnitudes of the quantity measure (MW) of the corresponding steps will be specified as upper bounds on the variables VDEF_T, cf. Section 8.

4.29.36 IX3FX_T_Y

The internal parameter IX3FX_T_Y holds the export to third countries for each time segment. It is calculated using the input parameters X3FX_VAR_T and X3FX and the internal parameter IX3FXSUMST as

IX3FX_T_Y(IR,S,T)=
(X3FX(Y,IR)*X3FX_VAR_T(IR,S,T)) /IX3FXSUMST(IR)

Hence the sign of IX3FX_T_Y will depend on X3FX(Y,IR), X3FX_VAR_T(IR,S,T)) and IX3FXSUMST(IR).

Comment on input data: Observe that the calculation will result in an error, if division by zero is attempted. This puts restrictions on IX3FXSUMST and in turn in X3FX_VAR _T from which it is derived. - If IX3FXSUMST(IR) is either strictly positive or strictly negative for all R this is not a problem. However, it could make sense to specify values of X3FX_VAR _T such that there is export in some time segments and import in others, but such that there a zero net annual export. This situation is not handled in the model.

5 Auxiliary parameters for outputs

We have found that it may be convenient to have some internal parameters to hold various results. For this purpose we have defined a number of parameters, called auxiliary parameters. These parameters are only used to hold intermediate results for printing output, in contrast to those internal parameters described in Section 4.29 that are used as proper parts of the model. Such parameters are located in those files that are not proper part of the model (i.e., they are not found in files located in the subdirectory Model, cf. Section 2). The names of such auxiliary entities start with the letter O.

A complete list will not be given, here are some:

- SCALAR OMONEY is used to convert the currency used in the input to the currency to be used in output. The numerical value given should indicate the exchange rate between money unit used in output per money unit used in input. E.g., with input in EUR90 and ouput in DKR, and exchange rate 8.2 between the two currencies (i.e., one EUR90 = 8.2 DKR) then specify: "PARAMETER OMONEY DKR / 8.2 /;". The text string after OMONEY should be enclosed in quotes. In the print files it is assumed that the text string has a length of five characters to give nice printouts. OMONEY is located in print1.inc.

- SCALAR OCASEID is used in print files, it may e.g. be used to identify the case. The text string after OCASEID should be enclosed in quotes, e.g. "SCALAR OCASEID 'Balmorel Demo-example'", it is this text which is printed.

Error and logging are described in Section 11.2, and model output is described in Section 10.3.

6 Variables

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

The following are the main variables (endogenously determined values) of the model.

VOBJ "Objective function value (MMoney)"

VGE_T(AAA,G,S,T) "Electricity generation (MW), existing units"

VGH_T(AAA,G,S,T) "Heat generation (MW), existing units"

VX_T(IRRRE,IRRRI,S,T) "Electricity export from region IRRRE to IRRRI (MW)"

VGEN_T(AAA,G,S,T) "Electricity generation (MW), new units"

VGHN_T(AAA,G,S,T) "Heat generation (MW), new units"

VGKN(AAA,G) "New generation capacity (MW)"

VXKN(IRRRE,IRRRI) "New electricity transmission capacity (MW)"

VDEF_T(RRR,S,T,DEF_STEPS) "Flexible electricity demands (MW)"

VDHF_T(AAA,S,T,DHF_STEPS) "Flexible heat demands (MW)" ;

VGHYPMS_T(AAA,S,T) 'Contents of pumped hydro storage (MWh)'

VHYRS_S(AAA,S) 'Hydro energy equivalent at the start of the season (MWh)'

VESTOLOADT(AAA,S,T) 'Loading of electricity storage (MW)'

VHSTOLOADT(AAA,S,T) 'Loading of heat storage (MW)'

VESTOVOLT(AAA,S,T) 'Electricity storage contents at beginning of time segment (MWh)'

VHSTOVOLT(AAA,S,T) 'Heat storage contents at beginning of time segment (MWh)'

VQEEQ(AAA,S,T,IPLUSMINUS) 'Feasibility in electricity equation QEEQ'

VQHEQURBAN(AAA,S,T,IPLUSMINUS) 'Feasibility in heat equation QHEQURBAN'

VQESTOVOLT(AAA,S,T,IPLUSMINUS) 'Feasibility in electricity storage equation QESTOVOLT'

VQHSTOVOLT(AAA,S,T,IPLUSMINUS) 'Feasibility in heat storage equation QHSTOVOLT'

VQHYRSSEQ(AAA,S,IPLUSMINUS) 'Feasibility of QHYRSSEQ'

VGE2LEVEL

In the GAMS language, the variables may be free, positive (i.e., non negative) or negative (i.e.g, non positive). The specification restriction of a variable according to this is done as indicated in the following: to non-negativity is specified by declaring the variables as POSITIVE VARIABLE as the following indicates:

FREE VARIABLE VOBJ;

POSITIVE VARIABLE VGE_T;

POSITIVE VARIABLE VGH_T;

etc.

Most variables are declared to be positive. The exception is VOBJ which cannot be constrained since it expresses the objective function value, for which the sign is unknown. In the GAMS language this is indicated by the specification FREE VARIABLE. All other variables are declared as positive.

Lower or upper bounds on the individual variables may be imposed by assignment of .LO and/or .UP, respectively. For certain constructions variables may have they values fixed, this may be done by assignment of .FX (or implicitly by assigning identical values for .LO and .UP).

Also the units in which the variables are measured are specified in the list above. There are only two kinds of units, related to money or power, respectively. The objective function variable VOBJ is in millions of Money terms (e.g., MEuro or MUSD) while all others are in MW terms.

The purpose of the variables with names starting with VQ is to secure feasibility in the equations, even if unfortunate values are entered for some of the energy system input. VQHSTOVOLT corresponds to equation QHSTOVOLT etc. The variables enter the objective function (specified in QOBJ, Section 7) with a large coefficient PENALTYQ, Section 4.1.1, and they will therefore only be positive if there will not otherwise be a feasible solution. If such variables are positive, an error message will be written, cf. Section 11.

The variables with names starting with VQ are declared also on IPLUSMINUS, see Section 3.8.11, i.e. there is one corresponding to 'plus' and one to 'minus'. These sign refer to the sign in front of the variable when it is placed on the right hand side of the relational operator (=L=, =E= or =G=) in the corresponding equation.

To each variable a number of attributes are associated, Section 10.1.

7 Equations and constraints

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

The constraints in the GAMS model may be on the individual variables, cf. Section 6. More general constraints are called equations. This refers to both equality constraints (indicated by =E=) and inequality constraints (indicated by =L= for 'less than or equal' and =G= for 'greater than or equal').

The model contains the following equations:

QOBJ 'Objective function (MMoney)'

QEEQ(RRR,S,T) 'Electricity generation equals demand'

QHEQ(AAA,S,T) 'Heat generation equals consumption'

QGCBGBPR(AAA,G,S,T) 'CHP generation (back pressure) limited by Cb-line'

QGCBGEXT(AAA,G,S,T) 'CHP generation (extraction) limited by Cb-line'

QGCVGEXT(AAA,G,S,T) 'CHP generation (extraction) limited by Cv-line'

QGGETOH(AAA,G,S,T) 'Electric heat generation'

QGNCBGBPR(AAA,G,S,T) 'CHP generation (back pressure) Cb-line, new'

QGNCBGEXT(AAA,G,S,T) 'CHP generation (extraction) Cb-line, new'

QGNCVGEXT(AAA,G,S,T) 'CHP generation (extraction) Cv-line, new'

QGNGETOH(AAA,G,S,T) 'Electric heat generation, new'

QGEKNT(AAA,G,S,T) 'Generation on new electricity cap, limited by cap'

QGHKNT(AAA,G,S,T) 'Generation on new IGHONLY cap, limited by cap'

QHNRURPROP(AAA,G,S,T) 'Proportional generation of heat in rural areas, new'

QGKNHYRR(AAA,G,S,T) 'Generation on new hydro-ror limited by cap and water'

QGKNWND(RRR,AAA,G,S,T) 'Generation on new windpower limited by cap and wind'

QGKNSOLE(RRR,AAA,G,S,T) 'Generation on new solarpower limited by cap and sun'

QHYRSSEQ(AAA,S) 'Hydropower with reservoir seasonal energy constraint'
QGKNHYRR

QHYMINRS(AAA,G,S) 'Hydropower reservoir - minimum level'

QHYMAXRS(AAA,G,S) 'Hydropower reservoir - maximum level'

QHYMING(AAA,G,S,T) 'Hydropower reservoir - minimum generation'

QESTOVOLT(AAA,S,T) 'Electricty storage dynamic equation (MWh)'

QESTOLOADT(AAA,S,T) 'Electricity storage loading less than heat production (MW)'

QHSTOVOLT(AAA,S,T) 'Heat storage dynamic equation (MWh)'

QHSTOLOADT(AAA,S,T) 'Heat storage loading less than heat production (MW)'

QKFUELC(C,FKPOTSETC) 'Total capacity using fuel FFF is limited in country'

QKFUELR(RRR,FKPOTSETR) 'Total capacity using fuel FFF is limited in region'

QKFUELA(AAA,FKPOTSETA) 'Total capacity using fuel FFF is limited in area'

QXK(IRRRE,IRRRI,S,T) 'Transmission capacity constraint'

QLIMCO2(C) 'Limit on annual CO2-emission'

QLIMSO2(C) 'Limit on annual SO2 emission'

QLIMNOX(C) 'Limit on annual NOx emission'

QECOMBGKL(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QHCOMBGKL(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNECOMBGKL(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNHCOMBGKL(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QECOMBSLO(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QHCOMBSLO(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNECOMBSLO(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNHCOMBSLO(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QECOMBSUP(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QHCOMBSUP(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNECOMBSUP(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QNHCOMBSUP(AAA,G,S,T): relevant for combination technologies, see Section 12.7.

QGE2LEVEL(AAA,G,S,T): relevant for technologies with slow regulation, changing only between workdays and weekend, see Section 3.2.3.

The specification of an EQUATION consists of a declaration, as seen above, and a definition which gives the details. The definition starts with the name of the previously declared equation followed by ".." and then the algebra. In equations the relational operators $\leq$, $=$ and $\geq$ are specified as =L=, =E= and

=G=, respectively. See the BALMOREL.GMS file for the details, and consult the GAMS User's Guide.

Most of the equations are expressed in MW. The exceptions are noted above.

Lower and upper bounds on the individual variables are described in Section 6.

8 Model and solve

In the GAMS language the word MODEL has the specific meaning of a collection of previously declared EQUATIONS. Hence, it is possible to declare more EQUATIONS than what are actually used in a specific model, and to specify several models from previously declared equations.

The specification of a model is done by stating MODEL followed by an identifier (the name of the model), possibly a short descriptive text and then, between "/" and "/", the equations to be included in the model. E.g. a very small model called "TINY", based on Section 7 could be

MODEL TINY Only for this example / QOBJ, QEEQ, QHEQ /;

It is possible to use ALL if all the declared EQUATIONS are to be used. Thus, one version of the Balmorel could (but should not!) be specified as

MODEL AllBal Baltic Model for Regional Energy Liberalisation /ALL/;

Observe that in some of the auxiliary parts the name of the model is important, cf. Section 11.2.

To specify the solution of the model the SOLVE statement is used, e.g.

SOLVE Balmorel USING LP MINIMIZING VOBJ;

In this, VOBJ is the variable that holds the objective function value, cf. Section 6, and it is to be minimised (the alternative is to specify MAXIMIZING). The problem class is specified to be LP (linear programming).

Various further options related to the solution process may be applied. Some may be included in the SOLVE statement, some specified by using the OPTION statement. Useful options may be specified in relation to RESLIM, ITERLIM, HOLDFIXED. See the GAMS User's Guide.

9 Calibration

Most of the numerical values in the Balmorel model will be taken directly from data sources. Care should be taken to ensure that they are consistent. In general, this is not easy, but a discussion is outside the scope of the present document.

The model contains a few calibration parameters that may be tuned in an attempt to attain certain consistency between model simulation results and historically observed values.

Basically, there are four calibration parameters: GKDERATE, GEFFDERATE, DEFP_CALIB and DHFP_CALIB. The basic information about these parameters is given in their respective sections, 4.21.1, 4.10.4, 4.20.2 and 4.19.6.

The first two are used to attempt consistency between fuel consumption as determined in the model and as specified in energy statistics, respectively. GKDERATE limits the generation of the units, and hence the baseload units (the high merit order units) are not generating at full nominal capacity all 8760 hours of the year. This implies a certain generation on other units (the medium to low merit order units) that would otherwise not generate. Thus, this parameter changes the relative amounts of generation between the units. GEFFDERATE changes the relation between fuel consumption and electricity and heat generation of the individual units. (It may also change the relative merit order of the units.)

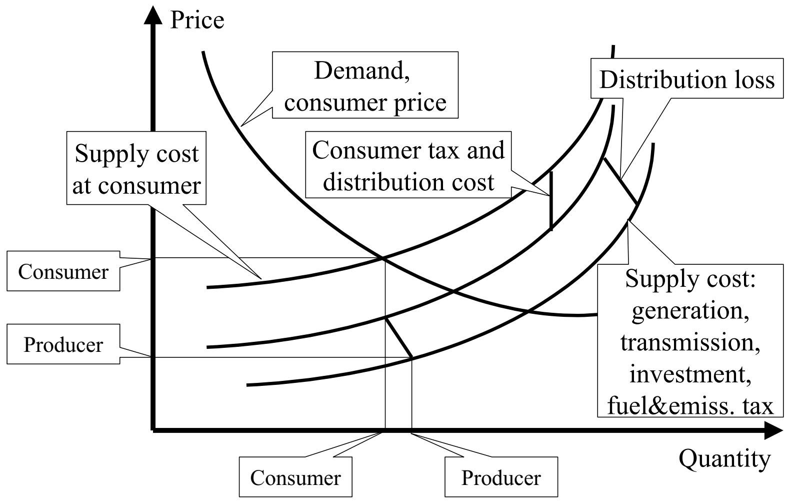

Figure 4: Illustration of elements in the calibration of demand functions

The last two parameters are used to attempt attainment of consistency between the consumption of electricity and heat as determined in the model and as specified in energy statistics, respectively. The point is that the model in principle has two methods to determine or specify prices of electricity and heat. One is to determine them through the marginal values related to cost components of generation (including fuel and emission taxes), distribution cost, correction for losses (in the case of electricity generation, possibly also transmission), and consumers' taxes on electricity and heat, respectively. The other way around the demand functions for electricity and heat, respectively, are specified directly through nominal values, profiles and elasticities, cf. Section 3.6. See Figure 4. Most probably, these two methods will yield prices that are not consistent. The effect of an inconsistency is that when a historical year is simulated (e.g., 1995) then the consumption of electricity and heat, respectively, as found in the model will not be the same as that observed historically. Moreover, the model permits identification of marginal values that differ between the time segments of the year, while historically most consumers have seen a price that has been constant over long time periods.

The parameters DEFP_CALIB and DHFP_CALIB permit a modification in the demand functions for electricity and heat in order to get such consistency.

The suggested sequence of calibration is first GKDERATE then GEFFDERATE and finally DEFP_CALIB and DHFP_CALIB.

9.1 Calibration of fuel consumption

9.1.1 Calibration of GKDERATE

The calibration of GKDERATE may be done with departure in a variety of sources. The overall purpose is to ensure a reasonable balance between generation on the various units. Data sources may be any, e.g. statistics over planned and forced outages of generation units. For thermal generation units, typical values for forced outages could be in the range 0.03 to 0.15. Scheduled outage could maybe be 2 to 8 weeks per year. Hence values of GKDERATE could typically be found in the range 0.7 to 0.95.

For wind power the apparent efficiency and capacity for a group of turbines will in general be different from that which may be immediately derived from the individual turbines. For instance for an installed capacity of 1000 MW, dispersed over a certain areas, the maximal generation will most probably never reach 1000 MW, due to forced and planned outages, and due to the fact that the wind speed is not the same all over the area. Such phenomena may be reflected in GKDERATE.

The following explains in more detail the reasoning related to stochastic outages of dispatchable units.

Modeling of stochastic outages

Consider the problem of modelling stochastic outages in the electricity system, more specifically the expected generation of the individual units.

Thus, assume n units with capacities $\overline{x}_i$, sorted in merit order. Each has a rate of forced outage of r_i. Consider one time period with demand d.

We take three models:

1. The "true" stochastic model
2. No consideration of outages
3. Capacity reduction model, i.e., the capacity $\overline{x}_i$ is substituted by $\overline{x}_i(1 - r_i)$, and the problem is then solved deterministically.

Example

Consider an example where $\overline{x}_i = 100$MW and $r_i = 0.1$ for all i. Let $d = 300$MW and consider a time period of one hour.

Model 1.

Unit 1	on	on	on	off	on	off	off	off
Unit 2	on	on	off	on	off	on	off	off
Unit 3	on	off	on	on	off	off	on	off
Prob.:	0.729	0.081	0.081	0.081	0.009	0.009	0.009	0.001

Table 8: Probability of on-off combinations for the first three units

With the specified capacities and load the first three units should always be on. Table 8 gives the probabilities for the on-off combinations of these first three units. As seen, there is a probability of 0.729 that they are all three on, and consequently a probability of 0.271 that at least one will be forced out. The expected generation of the units are then 90MWh, see the Table 9.

Unit:	1	2	3	4	5	6
Prob. of attempted on:	1.0	1.0	1.0	0.271	0.0271	0.00271
Prob. of actually on:	0.9	0.9	0.9	0.2439	0.02439	0.002439
Expected prod., MWh:	90	90	90	24.39	2.439	0.2439

Table 9: Results for Model 1

When at least one of the first three units is unit off, unit 4 will be attempted applied. This happens with a probability of 0.271. Unit 4 will in these situations produce at 100 MW with probability 0.9 and hence its expected energy generation will be 0.2439 times 100MWh, i.e., 24.39MWh.

If unit 4 fails when attempted turned on then unit 5 will be attempted turned on. This attempt will happen with probability 0.0271. With probability 0.9 unit 5 will then turn on, and its expected energy generation will then be 2.439MWh. Continuation of reasoning will lead to the figures given in Table 9.

Model 2:

The similar table for model 2 is shown in Table 10. Observe, that as the model is not actually stochastic the terms "probability" and "expected" are somewhat misleading.

Model 3:

Unit:	1	2	3	4	5	6
Prob. of attempted on:	1.0	1.0	1.0	0.0	0.0	0.0
Prob. of actually on:	1.0	1.0	1.0	0.0	0.0	0.0
Expected generation, MWh:	100	100	100	0	0	0

Table 10: Results for Model 2

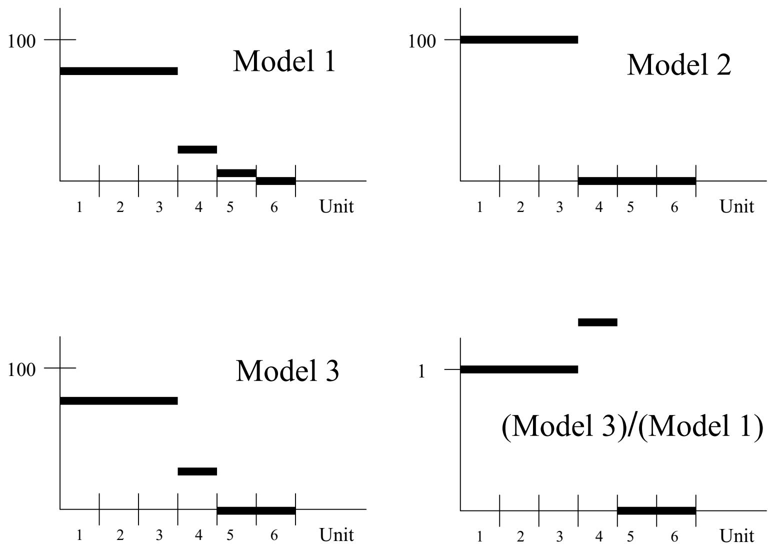

Figure 5: Expected generation (Models 1, 2 and 3) and comparison of results for Model 1 and Model 3

In this model, each of the units has a capacity of 90MW. Hence, the fourth unit will be applied with a generation of 30MW, and the figures are given in Table 11.

Unit:	1	2	3	4	5	6
Prob. of attempted on:	1.0	1.0	1.0	1.0	0.0	0.0
Prob. of actually on:	1.0	1.0	1.0	1.0	0.0	0.0
Expected generation, MWh:	90	90	90	30	0	0

Table 11: Results for Model 3

The graph in Figure 5 shows the expected energy generation for each unit in each of the three models. Taking Model 1 as the "true" model it is seen that Model 2 overestimates the energy generation for the first units (the good ones), and underestimates it for the last units. Model 3 comes closer to Model 1. Thus it has a correct representation of the expected energy generation for the first units, while it overestimates for the next unit and underestimates for the last units.

For Model 3 this is brought out more clearly in the last graph on Figure 5 which for each unit shows the expected energy generation of Model 3 in relation to that of Model 1.

Generalisations

We may generalise the results as follows. Define indexes i_1 and i_2 such that $\sum_{i=1}^{i_1} \overline{x}_i \leq d < \sum_{i=1}^{i_1+1} \overline{x}_i$ and $\sum_{i=1}^{i_2-1} \overline{x}_i(1-r_i) < d \leq \sum_{i=1}^{i_2} \overline{x}_i(1-r_i)$. As seen, for Model 3 all units with $i \leq i_1$ will be on with capacity $\overline{x}_i(1-r_i)$, and all units with $i_2 < i$ will be off. We define e_i^{M1} and e_i^{M3} to be the expected generation of unit i

under Model 1 and Model 3, respectively. We assume for simplicity that $0 < r_i < 1$ for all i, and give presentation of the results with a short argumentation.

Units with $i \leq i_1$ will be on all time in Model 3 with power $\overline{x}_i(1 - r_i)$, while in Model 1 they will be attempted on and therefore have expected power $\overline{x}_i(1 - r_i)$. Hence $e_i^{M1} = e_i^{M3}$ for $i \leq i_1$.

For $i_1 < i < i_2$ the power in Model 3 will be $\overline{x}_i(1 - r_i)$. In relation to Model 1 we see that all units with $i < i_1$ will be attempted run, and since $1 - r_i > 0$ they will in fact be running some of the time. Hence for units with $i_1 < i \leq i_2$ their expected power under Model 1 will necessarily be less than $\overline{x}_i(1-r_i)$ (which is the expected power when attempted on at full capacity), and therefore $e_i^{M1} \leq e_i^{M3}$ for $i_1 < i < i_2$.

For unit i_2, which is the "marginal unit" in Model 3, it may be shown that $e_{i_2}^{M1} < e_{i_2}^{M3}$ (if d is close to $\sum_{i=1}^{i_2-1} \overline{x}_i(1 - r_i)$), $e_{i_2}^{M1} = e_{i_2}^{M3}$, or $e_{i_2}^{M1} > e_{i_2}^{M3}$ (if d is close to $\sum_{i=1}^{i_2} \overline{x}_i(1 - r_i)$).

For units $i_2 < i$, $e_i^{M3} = 0$. Further, $e_i^{M1} > 0$ since with positive probability all units with $i \leq i_2$ will fail when attempted run. Therefore $e_i^{M3} < e_i^{M1}$ for $i_2 < i$.

As seen, Model 3 provides an approximation to Model 1 such that "base load" units have the correct expected generations, "first reserve" units have overestimated generations, "second reserve" units have underestimated generations, while for the "marginal unit" the expected generation may be correct, overestimated or underestimated.

In terms of total costs it may be shown that the cost of Model 3 is less than or equal to that of Model 1.

9.1.2 Calibration of GEFFDERATE

Once GKDERATE has been given its values, GEFFDERATE may be specified. The aim could be that in the calibration year (e.g., 1995) there is, for the observed generation of electricity and heat, consistency between the fuel consumption as determined by the model and the fuel consumption observed historically.

The deviations between the two measures may be due to a variety of reasons. The following description assumes that the need for calibration can be meaningfully ascribed to the generation side, although also e.g. consumption, subdivision of the year into time segments, and unrealistic distribution losses could be responsible.

The reasons for deviations between the two measures may be many, even in relation to generation alone. For thermal units for instance, there will be a certain fuel consumption related to the start up of units; the efficiency for thermal units is not constant, but depends on the load; a certain additional loss may also be accredited to up- and down-regulation, relative to the loss encountered at steady level generation. Hence, the efficiency depends not only on technical factors, but also on the actual use of a unit.

For wind power the apparent efficiency and capacity for a group of turbines will in general be different from that which may be immediately derived from the individual turbines as discussed above. Such phenomena may be reflected in GKDERATE, but note that also for wind power there is inter-dependency with GEFFDERATE and GDFE, where the latter may be found in various ways. The specific way chosen to calibrate for wind power will therefore in part depend on the data sources. Similar considerations hold for hydro power.

The following description exemplifies the calibration for some thermal units, assuming that historical data are available at a regional level. The same value of GEFFDERATE will be assumed for all units participating in the calibration.

1. Define the set of generation technologies for which the calibration is to be performed. For instance, if backpressure and extration units will be considered, then specify "SET GEFFSETCAL(G); GEFFSETCAL(G)=NO; GEFFSETCAL(IGEXT)=YES; GEFFSETCAL(IGBPR)=YES;".

2. Define the set of fuels for which the calibration is to be performed. For instance, "SET FEFFSETCAL(F); FEFFSETCAL(F)=NO; FEFFSETCAL("NGAS")=YES; FEFFSETCAL("COAL")=YES;".

3. Select calibration year (e.g., 1995), and place the significant historical values for the calibration year in the relevant tables; essential are DE, DH, X3FX (and, if the relations between the fuel prices have changed significantly, also FUELP). Here, X3FX should be net electricity export in the calibration year to all other regions, IRRREspective of these being part of the model or not. DE should be that part of electricity consumption that is covered by the fuels and technologies selected above, multiplied by (1-DISLOSS_E(IR)). Similarly for heat demand.

4. Specify the set Y to contain only the calibration year.

5. Specify FIRSTYEAR to be the calibration year.

6. Exclude the possibility of electricity transmission, see Section 12.9.

7. Exclude the possibility of new investments, see Section 12.8.

8. Specify in SETS.INC the following sets: "SET DEF_D(DEF) / DEF_DINF/;", "SET DEF_U(DEF) / DEF_UINF/;", " SET DHF_D(DHF) / DHF_DINF /;" and "SET DHF_U(DHF) / DHF_UINF /;". This ensures inelastic demand, cf. Section 3.6.

9. For the fuels and technologies selected above let GEFFDERATE be 1.

10. Make sure that the files PRINT1.INC, PRINT2.INC, PRT3-*.INC, and PRT4-*.INC are included in the model by a $INCLUDE, and that there is a possibility to print out the fuel consumption for the fuels in question.

11. Run the model.

12. Calculate the quotient between the fuel consumption as found in the model and the historically observed fuel consumption.

13. If the values look reasonable (in particular they should be positive and not too far from 1) then let GEFFDERATE attain the value of the quotient for the generation technologies in question.

14. (The procedure may at this point be checked as follows: Run the model again. Now the fuel consumption in the calibration year should be close to that observed historically.)

15. Observe that the sets defined in Step 1 and Step 2 need not be the same for all countries. Therefore it may be necessary to run the calibration several times, one for each county, with changes between the sets specified in Step 1 and Step 2. Therefore the final table GEFFDERATE may have to be constructed by combining results from the individual runs.

16. Delete the sets constructed in Step 1 and Step 2, and specify the tables and sets that were modified above to have their desired contents, i.e., bring back the model to original status.

Since the values specified in Step 3 may differ between various runs, it may be convenient to keep the data separate. The auxiliary files GEFF_CAL1.INC and GEFF_CAL2.INC have been prepared for this purpose, and also in other ways there are deviations; however, the idea is the same. See the instructions in those files.

9.2 Calibration of demand functions

9.2.1 Calibration of DEFP_CALIB and DHFP_CALIB

The calibration of the demand function is only necessary if the demand is elastic. The purpose of the calibration is to get the model's demand to coincide with the observed values. This is done by modifying the base price IDEFP_T by DEFP_CALIB and IDHFP_T by DHFP_CALIB.

Calibration makes sense only if the geographical entity chosen contains those generation units that are price setting (marginal). Hence, if a region is heavy net exporter or importer of electricity, calibration can not meaningfully be made on this region alone.

In the following the calibration will be explained, assuming that the auxiliary file DFP_CALIB.INC, located in subdirectory PRINT-INC, is used.

1. Select the geographical entity that will be used for calibration (specify the set C).
2. Select a calibration year (e.g., 1995), and place the significant historical values for the calibration year in the relevant tables; essential are DE, DH, X3FX and FUELP.
3. Specify the set Y to contain only the calibration year.
4. Specify FIRSTYEAR to be the calibration year.
5. Exclude the possibility of new investments, see Section 12.8. Exclude the possibility of transmission, see Section 12.9.
6. Make sure that the files PRINT1.INC, PRINT2.INC, PRT3-*.INC, and PRT4-*.INC are included in the model by a $INCLUDE.
7. Let all the values in DEFP_CALIB and DHFP_CALIB be zero (e.g., place the statements "LOOP((IR,S,T), DEFP_CALIB(IR,S,T)=0);" and "LOOP((IR,S,T), DEFP_CALIB(IR,S,T)=0);" after DEFP_CALIB and DHFP_CALIB, respectively).
8. Include the statement "FILE DFP_CALIB /..\OUTPUT \DFP_CALIB.OUT/;" in file PRINT1.INC.
9. Include the statement "$INCLUDE "DFP_CALIB.INC";" in file PRT4-*.INC.
10. Specify in SETS.INC the following sets: "SET DEF_D(DEF) / DEF_DINF/;", "SET DEF_U(DEF) / DEF_UINF/;", " SET DHF_D(DHF) / DHF_DINF /;" and "SET DHF_U(DHF) / DHF_UINF /;". This ensures inelastic demand.
11. Run the model.
12. Take the table DEFP_CALIB in the file DFP_CALIB.OUT and let it replace the relevant parts of the existing table DEFP_CALIB. Take the table DHFP_CALIB in the file DFP_CALIB.OUT and let it replace the existing table DHFP_CALIB. (Recall to remove the "LOOP .." statements if they were introduced above.)
13. Modify the sets DEF_D(DEF), DEF_U(DEF), DHF_D(DEF) and DHF_U(DEF) to include the desired steps in down- and upwards directions.
14. (The procedure may at this point be checked as follows: Run the model again. Now the consumption of electricity and heat in the calibration year should be equal to that specified in DE_Y and DH_Y, respectively, i.e. all up- and down- regulation steps should be zero. The values in the tables in the file DFP_CALIB.OUT should now all be zero.)

15. Specify the tables that were modified in Step 1 to have their desired contents, specify possibilities of investment and transmission as desired (Step 5), delete or comment out the statements specified in Step 8 and Step 9, i.e., bring back the model to original status.

10 Output

There are two types of output, that generated automatically by the GAMS system, and that generated by auxiliary parts of the Balmorel model.

10.1 Automatically generated GAMS output

GAMS automatically generates two files after each run, the Balmorel.lst file and the Balmorel.log file. In addition, GAMS permits several ways of presenting output under various degrees of user control.

The log file

The log file contains an execution summary.

The lst file

The lst file contains a summary of model statistics and solution, including SOLVER STATUS, MODEL STATUS, OBJECTIVE VALUE. See the GAMS User's Guide for further information.

The lst file also as default contains an echo of the input with line numbers associated, usefull for identification. Errors detected by GAMS will be identified, see Section 11.1.

The lst file may also contain specific output if wanted by the user. Details may be controlled by various compiler directives in the form of OPTIONS and $ON / $OFF statements. For variables this will typically be their values.

Some options are OFFLISTING, OFFSYMXREF, OFFSYMLIST, OFFUELLIST OFFUELXREF, with the alternatives ONLISTING, ONSYMXREF, ONSYMLIST, ONUELLIST, ONUELXREF are used to control printing of listing and cross references. ONINLINE makes it possible to comment out parts between /* and */. LIMROW and LIMCOL specify the maximum number of rows and columns used in equations listing and inspection of details. SYSOUT controls the printing of the solved status in the list file. SOLPRINT controls the printing of the solution in the list file. ONDOLLAR will secure echoing of dollar control options in the listing file.

To each EQUATION in a solved model are associated a level and a marginal value, that may be referenced for printing, for instance for QOBJ as QOBJ.L and QOBJ.M, respectively. See the GAMS User's Guide for further information.

Similarly a number (six) of attributes are specified in relation to a variable, viz., the lower bound (.LO), the upper bound (.UP), a fixed value (.FX), a level (.L), a marginal or dual value (.M), a scale value (.SCALE) and a branching priority value (.PRIOR).

10.2 Less automatical GAMS output and inspection

GAMS provides additional environment for presentation and inspection of output. Here are a few options.

The DISPLAY statement

Specific items may be shown in the lst file using the DISPLAY statement. For instance, place the statement "DISPLAY 'MyOwnChoice: ', Y, C, G, DE, VGE_T.L

, QEEQ.M, 'End MyOwnChoice' ;" at the very end of the file 'Balmorel.gms'. Run the model and find the result in 'balmorel.lst' (near the end of the file; you may search for the text string 'MyOwnChoice').

GAMS Data Exchange, GDX

The GDX (GAMS Data Exchange) facilities and tools permit various inspection and data exchange possibilities. A GDX file is a file that stores the values of one of more GAMS symbols such as sets, parameters, variables and equations. GDX files can be used to prepare data for a GAMS model, present results of a GAMS model, store results of the same model using different parameters etc. See the GAMS homepage for further information.

As an example, place the statement "execute_unload 'balbase1.gdx';" at the very end of the Balmorel.gms file, then run the balbase1 model. Now the file balbase1.gdx is found in the model directory. Open it from the GAMS IDE and inspect the result and the possibilities.

Another example of use of GDX is for comparing data in two versions of a model using GDXDIFF: produce a GDX file for each of the two model versions, apply the GDXDIFF to them, and see the difference.

GTREE

GTREE is a tool to show the structure of the GAMS code in a tree. See the GAMS homepage for further information.

10.3 Balmorel auxiliary output

A number of auxiliary files have been designed to facilitate output from the Balmorel model. Four of these files are taken into the Balmorel model by the $INCLUDE statement, viz., PRINT1.INC, PRINT2.INC, PRT3-*.INC, PRT4-*.INC. See Section 2 for location. Prints are generated from these, or from other files that are in turn included in the model by a $INCLUDE statement in one of the mentioned files. See those files for further details.

Error checking is also reported and a log maintained, see Section 11.2.

Observe that quite a number of auxiliary include files for printing output are available. The user's computer may not be able to handle so many open files, therefore carefully specify those files to be produced in PRT4-*.INC and comment out the remaining ones.

It is important to note that the information printed by auxiliary parts is not reliable if errors occurred during the execution of the GAMS program!

A spreadsheet environment is provided for assistance in processing the output.

The print files in the Balmorel model contain different results of the model simulation. All output files are located in the folder "printout" and are generated after each simulation with the Balmorel model (please note that the output files overwrite old output files from the previous simulation).

The print files that are to be generated after each simulation are specified in the files "prt4_bb1.inc" and "prt4_bb2.inc" in the folder "printinc". Output files that are commented out in "prt4_bb1.inc" and "prt4_bb2.inc" are not printed. The exception is "inputout.inc", which should be $-included into the "balmorel.gms" file.

Contents of print-files

The maximum numbers of characters in the file name are 8 and the structure of the output file names are given by following letters:

E: Electricity

H: Heat

F: Fuels

M: Emissions (MCO2, MSO2 and MNOx)

G: Generation

D: Demand

P: Prices

K: Capacity

X: Transmission

O: Old "existing" plants

N: New plants

C: Country

R: Region

A: Area

Y: Year

S: Season

T: Time period within season

Z: Summation

The first letter(s) indicate(s) the subject of the output files. Then there is an underscore followed by letters which indicate the level of detail. C, R or A indicates the geographical level of detail. Y, S or T indicates the level of time segments. The letter G indicates that information for each particular technology is available.

The letter Z is used to summarise over the following letters. Ex. the file "EZGN_RY" contains information about the total electricity generation from all new plants in each region per year.

If there are no letters to identify the geographic level the file contains information for all countries that are simulated in Balmorel. If there is no letter to identify the time segments the file contains information for the whole time span that is simulated in Balmorel.

A list of available print files is given in Table 12.

10.4 The compare facility

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

In a number of cases it may be relevant to compare the results of different simulations (here called scenarios or variants). The present section describes a facility and the organization related to this. See also Section 12.3.

The file structure assumes that a base model and a number of scenario models are implemented. They are contained in parallel directories with the names "base" and e.g. "variant", "variant2", "variant3" etc. The purpose is to compare the result of the various scenarios with that of the base case. The simulations in the base case and in the scenarios are performed individually, and the basic idea is that the necessary results for comparison are saved in files that are used as input for the comparison.

The directory "base" contains the standard subdirectories ("model", "printinc"," printout", "logerror", "compare", cf. Section 2). The directory "compare" may have subdirectories according to the kind of comparisons that are made; here it will be assumed that "demo" is the relevant subdirectory to "compare", and that the base case and "variant" will be compared, using the demonstration files distributed with the Balmorel model.

The following indicates the directory structure within the 'Balmorel' directory:

File	Description	Geographic level	Time structure
inputout	Overview of various input data to the Balmorel model	—	—
bal_1	Energy balance	Country	Year
eg_cy	Electricity generation from all technologies	Country	Year
eg_gat	Electricity generation from all technologies individually	Area	Time period
ego_cy	Electricity generation from old technologies	Country	Year
egn_cy	Electricity generation from new technologies	Country	Year
ezgo_cy	Total electricity generation from all old technologies	Country	Year
ezgn_cy	Total electricity generation from all new technologies	Country	Year
gkn_ag	New investments in production capacities by area and technology	Area	Year
hg_cy	Heat generation from all technologies	Country	Year
hgo_cy	Heat generation from old technologies	Country	Year
hgn_cy	Heat generation from new technologies	Country	Year
ehf_ay	Fuel consumption from all technologies	Area	Year
ehf_ry	Fuel consumption from all technologies	Region	Year
ehf_cy	Fuel consumption from all technologies	Country	Year
ep_ry	Average electricity price	Region	Year
ep_rt	Electricity price	Region	Time period
hp_ay	Average heat price	Area	Year
hp_at	Heat price	Area	Time period
ehf2_cy	Fuel consumption distributed on each fuel	Country	Year
exk_ry	Transmission capacities (old plus new) by the end of the year	Region	Year
ex_ry	Net electricity export	Region	Year
mco2_cy	CO2 emissions	Country	Year
mso2_cy	SO2 emissions	Country	Year
mnox_cy	NOx emissions	Country	Year
epnxt_rt	Marginal electricity generation cost of 'the next' unit	Region	Time period
hsto_at	Heat storage	Area	Time period
hsto2_at	Heat storage contents at the beginning of each time segment	Area	Time period
esto_at	Electricity storage	Area	Time period
esto2_at	Electricity storage contents at the beginning of each time segment	Area	Time period

Table 12: Print files

- base
    - compare
        - demo
            - cmpinc
            - cmpout
        - otherpossibilities
            - cmpinc
            - cmpout
    - logerror
    - model
    - printinc
    - printout
- variant
    - compare
        - demo
            - cmpinc
            - cmpout
        - otherpossibilities
            - cmpinc
            - cmpout
    - logerror
    - model

    - printinc
    - printout

- variant2

- variant3

The idea in the process of making scenarios or variants is the following. Since only a small part of the input data will differ between the base case and the variant, most of the input should be reused, not copied, to prevent against accidental differences. The invariant input is given in the data files in the "/base/model/" directory, and these files are $-included both in the base case and in the variant case. The input data that differ will be placed in files in the "/variant/model/" directory and then $-included. The "Balmorel.gms" file is to be found in both the base and the variant model directories.

Thus, for all input files that are the same, use "$include ../../base/model/<filename>" in the "Balmorel.gms" files. For files that differ, use "$include ./../base/model/<filename>" in the base case and "$include <filename>" in the variant case.

The following procedure is advocated for handling of changes of data in the 'variant' model, cf. also Section 12.3.

All original data are taken from the model in the 'base' directory. This is handled by the '$-include ./../base/model/<filename>' statements, where <filename> is a data file, e.g. sets.inc, de.inc, trans.inc etc. This way all original data are entered. In order to insert new data, the old ones must be replaced. This is done by the statements '$if exist <filename> kill <identifier>' and '$if exist <filename> $include <filename>'. Here the <filename> is the name of a file holding the new data and <identifier> is the name of the identifier, typically a parameter.

Example: assume that you want to use other cost for electricity transmission. The original data are in the table XCOST in the file 'base/model/trans.inc'. Make a new file called 'xcost.inc' (i.e., the same name as the parameter) and place it in 'variant/model/' directory. The file starts with 'TABLE XCOST(IRRRE,IRRRI)' and the table holds the new data. Place the two statements '$if exist xcost.inc $kill xcost' and '$if exist xcost.inc $include xcost.inc' in the file 'variant/model/Balmorel.gms' immediately after '$include ../../base/model/trans.inc'. What will happen is that the '$kill' dollar control option removes all data in XCOST and resets XCOST, only the type and dimension are retained. The '$include' dollar control option then ensures that the file with new data is read. Observe that in order for the ' $kill' option to work, the dollar control option $ONMULTI must be set, this is best done in the file balgams.opt.

The files used for printing the results are found in the "/base/compare/demo/ cmpinc" directory. The declarations are given in "initbase.cmp", and this file also provides initialisation of gams2prm.gms (see below). Results to be used for comparison are transferred to intermediate parameters in "trnsbase.cmp". The specific prints from the base simulation are specified in "savebase.cmp". The result is printed to the file "baseout.cmp" in "/base/compare/demo/cmpout"

Observe that the option "$ONEMPTY" is specified in the printed file because all values may be zero, resulting in an empty parameter definition. When the printed file is $-included, the "$ONEMPTY" will become active (and remain so, unless overruled later by an "$OFFEMPTY").

In the base case (run from base directory), $-include the three files into "balbase1.sim" as follows: "initbase.cmp" at the very top, "trnsbase.cmp" immediately after "$INCLUDE ./../base/printinc/prt4-bb1.inc" and "savebase.cmp" at the very end. This ensures that declarations will be kept outside of loops. In the variant case the files specifying the process to make the comparison are found in "variant/compare/demo/cmpinc/". The names are "initvar.cmp", "trnsvar.cmp" and "savevar.cmp".

To proceed with a demonstration in relation to Balmorel do as follows:

- In the base case (run from "/base/model/" directory), $-include the three files into "balbase1.sim" as follows: "initbase.cmp" at the very top, "trnsbase.cmp" immediately after "$INCLUDE ./../base/printinc/prt4-bb1.inc" and "savebase.cmp" at the very end. Note that this ensures that declarations will be kept outside of loops.

- Run the Balmorel file in the base scenario.

- Now there should be a new file "baseout.cmp" in "../compare/demo/cmpout/ ".

- In the "variant/model/" directory there should be the files "balmorel.gms" and "balbase1.sim". Make sure that the latter file is $-included into the former (using "$include balbase1.sim", not "$include ./../base/model/balbase1.sim").

- In the variant case (run from "variant/model/" directory), $-include the three files into "balbase1.sim" as follows: "initvar.cmp" at the very top, "trnsvar.cmp" immediately after "$INCLUDE ../../base/printinc/prt4-bb1.inc" and "savevar.cmp" at the very end.

- Run the Balmorel model in the variant scenario.

- Find the results of the comparison in the files "variant.out" in the "variant/compare/demo/cmpout/" directory.

- The expected result is that differences are zero since input data in both simulations are identical. To introduce a difference, copy "de.inc" from "/base/model/" to "variant/model/"; change a number in the latter file for a relevant year and region; replace "$INCLUDE ../../base/model/de.inc" in "/variant/model/balmorel.gms" by "$INCLUDE de.inc"; run the variant model. Now there should be differences to be observed in "variant.out".

- Try to make similar exercise with respect to "balbase2.sim".

The implementation is based on the files gams2prm.gms (could also, with minor modifications, be gams2txt.gms). It is necessary that the directory "inclib" (a subdirectory to the GAMS-library) contains these files. The files may be downloaded from GAMS' home page (Contributed Software) (accessed 2006.01.07) and this site also contains documentation of the files.

See also in "/base/compare/documentation/" the demonstration program g2p-demo.gms concerning gams2prm.gms and g2t-demo.gms concerning gams2txt.gms.

11 Errors and Log

We distinguish between errors that are detected automatically by the GAMS system and errors that are detected by auxiliary parts of the Balmorel model.

11.1 Errors automatically detected by GAMS

If there are errors detected automatically by GAMS they will in the Balmorel.lst file be marked by four stars, hence a convenient way to locate them is to search for the string "****". If there are errors, then at the end of the Balmorel.lst file there will be a list of errors and a description of the possible cause of each error. User errors are indicated by the statement "**** USER ERROR(S) ENCOUNTERED". Other error types are marked by e.g. "**** EXECUTION ERROR", "**** MATRIX ERROR", or "**** PUT ERROR". See the GAMS User's Guide for further information.

11.2 Errors detected by Balmorel auxiliary parts

A number of error checks have been specified in the files ERROR1.INC, ERROR2.INC, ERROR3.INC. See Section 2 for location.

These files are included in the BALMOREL.GMS file by the $INCLUDE statement. If any of these files are included (i.e., they are not all commented out) the file PRINT1.INC must also be included (Section 10.3).

The error checking mainly concerns the numerical values of the input. The error checking tries to detect if the values specified are "reasonable". For instance, fuel efficiency would for most generation units be expected to take a value between, say, 0.3 and 0.9. On the other hand, it might be that values outside this range were relevant for some applications. In order to catch as many errors as possible, the range should be as small as possible, but in order not to indicate an error where there is none, the range should be large. Thus, a balance has to be achieved.

If an error (which, as just argued, need not be an "error") is detected, a specification is written to the file ERRORS.OUT. To see the exact reason for the identification of the error, see the appropriate ERROR*.INC file. The number of errors encountered is held in the parameter ERRORS. In any case, a summary is printed in ERRORS.OUT and also in the file LOGFILE.OUT.

(In a few cases the error check is not reported to ERRORS.OUT but rather results in a deliberate computation error. In this case follow the instructions given.)

The balbase1.mss file, see Section 2 for location, will print a summary of the model and solver status to the LOGFILE.OUT file for the BALBASE1 model. Similarly for other models, in particular BALBASE2.

Finally observe that the information in the auxiliary files described in this Section is not valid if there are user error(s) encountered, cf. Section 11.1.

11.3 Sequence of log and error observations

The very first step is to observe if the attempts to interpret the input and generate the model were successful. Therefore inspect the Balmorel.lst file to see if it contains the statement "**** USER ERROR(S) ENCOUNTERED", if this is the case then this should be fixed, cf. Section 11.1. The information in the output files described in Section 11.2 is in this case not valid.

After simulation, the user must first observe if the attempt was successful. Again, errors will be documented in the Balmorel.lst file, following "****". This for instance could be "**** EXECUTION ERROR". Also during printout errors may occur, indicated e.g. by "**** PUT ERROR". See Section 11.1.

If the execution of GAMS was successful, further information may be acquired by using the facilities provided in the auxiliary parts described in Section 11.2. This is intended to be more expedient, however, it can of course not be guaranteed that the auxiliary parts will be free of errors! If there are errors here, then using the GAMS standard output is the way to detect them. Moreover, some errors are detrimental to the intended functioning of the auxiliary parts, and therefore the information in the auxiliary parts is only reliable if there were no errors detected by the GAMS system. Also observe that if for instance no solution was found (which is also some sort of normal completion!) the information in the auxiliary parts may be unreliable.

If the auxiliary parts are used, the following procedure should be followed. First check the LOGFILE.OUT file, see Section 11.2. For a successful simulation there should be a declaration that there were no errors detected in the input, and that the solution efforts were successful. (Remember to check the date and time, because if there were errors detected by GAMS, the LOGFILE.OUT file may not be updated.)

If errors were detected in the input, then see the output file from the error detection, cf. Section 11.2.

If the solution is not successful then study the error messages in the LOGFILE.OUT and in the Balmorel.lst files. See the GAMS User's Guide for further information.

And finally: there are of course many modeling errors that neither GAMS nor the auxiliary parts can detect, but only the user.

12 Model variants

The above description refers to the "standard" version of the Balmorel model. In the following, a number of obvious modifications will be described.

12.1 Why variants

In any modeling work choices are made as to what to include in the model and what not. Many objectives are balanced in this process. Therefore, a model that may be suited for one purpose may be less appropriate for another. Moreover, if every possible application, and therefore the most detailed level of representation of the energy sector was attempted, the model would not be appropriate to any application.

To obtain maximum flexibility, the Balmorel model is coded in a high level modeling language (GAMS) and the code itself is available to any user. The user therefore has complete control over the model and therefore also over modifications. This permits a wider range of potential applications.

In the following a number of obvious modifications will be described.

12.2 Types of changes

Obviously, some modifications are easy while others are more complicated. The following classification may be suggested, where the first modifications are very simple, and the last one more complicated.

- Limit the scope of the model, while maintaining basic structure. Thus for instance with respect to geography, the model represents a number countries, as given by the set CCC. It is elementary to delete some of the countries from the model by declaring the set C to be a proper subset of CCC. It is not much more complicated to reduce the number of regions or areas within a country, although some consistency is required, see Section 3.1. Reduction of the number of years simulated is elementary and reduction of the number of time segments is discussed in Section 3.2.2. Reduction in the number generation technologies is elementary, Section 3.3. Reduction in the number of fuels is described in Section 3.4.

- Change the values of the numerical data entered. It is elementary to change the values of input parameters. Observe that the auxiliary parts involve some checking for "reasonable" values, see Section 11.2, and if therefore unanticipated values are entered, error messages may occur; in this case, the user is advised to revise the data and error checking.

- Enlarge the model with set elements (labels in the GAMS terminology) very similar to those that are already there. New countries, with their associated regions and areas may easily be introduced in the sets CCC and C. The model contains a number of energy transformation technologies in the set GGG and more - provided they are similar to one of the existing technology types, see Section 3.8.9 - may be added by copying the ideas in the representations already there and then filling in the required parameter values. Additional fuel types may be introduced in the set FFF (and the appropriate pointers introduced in GDFUEL, see Section 3.3). Additional years may be introduced into YYY without difficulties, and the number of time segments may also be increased, see Section 3.2.2 and Section 12.12.

- Change the model structure. The model structure consists of the parameters and variables in the model and the relations between them (see more specifically Section 14). On this issue it is not possible to specify the efforts involved as they depend heavily on the specific requirements. However, really many modifications to the structure can be made by an effort which is considerable smaller than that of acquiring the associated data.

The first items have explicitly or implicitly been covered in the preceding parts. In the sequel the last item is therefore addressed by way of examples. Section 12.3 briefly describes aspects of the organisation of modifications.

12.3 Organising modifications

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

Obviously there must be some internal consistency between the data in the model. Part of this is obtained by maintaining the proper sequence of the statements (cf. Section 1.3).

Observe that there are in this respect the following parts of the model (excluding auxiliary files):

- The first part of the BALMOREL.GMS file, down to but excluding the first $INCLUDE statement (not counting the auxiliary files)
- The SETS.INC file
- The declaration of internal sets in the BALMOREL.GMS file
- The other *.inc files (not counting the auxiliary files)
- The following part of the BALMOREL.GMS file

During execution, the model is composed by including the include files into the BALMOREL.GMS file. For this to work properly, the SETS.INC file must be included before any of the other include files.

However, the other include files may come in arbitrary order, since they only contain numerical values, contained in tables (or otherwise) directly specified by the user. (Possibly there may immediately after a table be an assignment modifying or complementing the values given in the TABLE (e.g. to user specified default values, see Section 4.27); in such cases the assignments should always follow immediately after the TABLE.) Hence, any table is independent of any other table, and the sequence does not matter.

All the internal parameters are derived after all the include files have been included (not counting the auxiliary files).

For this reason modifications in input data (i.e., what is contained in the *.inc files, not counting the auxiliary files) may be given quite flexibly. New static sets should be placed in the file SETS.INC. New tables should be placed in the appropriate include files (cf. Section 2), or they (and any new sets) may be placed in a new file that is taken into the BALMOREL.GMS file by an $INCLUDE statement; this statement should be placed after the $INCLUDE SETS.INC and before the handling of the internal sets and parameters (e.g., it may be placed immediately after the $INCLUDE SETS.INC statement).

See also the description in Section 10.4.

Changes in the model structure are exemplified in the sequel.

Any modifications made should be in a form that is consistent with the other parts of the Balmorel model. Here is a checklist:

- Did you observe the conventions on notation? See Section 1.4 and see 'naming' in the index.

	BB1	BB2	BB3	SEL4	SEL5	SEL6	SEL7
BB1		-	-	-	+	+	
BB2	-		-	-	+	-	
BB3	-	-		?	+	-	
SEL4	-	-	?		RC	+	
SEL5	+	-	+	CR		RC	RC
SEL6	+	-	+	+	CR		
SEL7	-	-	+	+	CR		

Table 13: Relations between selections. Legend: +: Possible; -: impossible (alternatives); RC: possible to apply row selection without column selection, but not the other way; CR: possible to apply column selection without row selection, but not the other way; ?: Unknown, untested.

- Did you observe the conventions on location of sets, parameters, etc., Section 2?
- Did you remember to give a description of the sets, parameters introduced (including specification of the units or measurement wherever relevant, Section 4)?
- Did you observe the restriction on the use on GAMS version, Section 1.3? (You may check this by including a "$USE225" at the top of the Balmorel.gms file (with the $ in the first column). Unfortunately, this seems not to enforce the restriction of identifiers and labels to have a maximum of ten characters.)

12.4 Addon

THE FOLLOWING IS NEW FOR VERSION 3.01.

The Balmorel model contains a large and growing number features. Many features are developed in relation to specific projects, permitting focus and refinements that are relevant to the purposes of the project.

It is of course a strength that such features are made available for others to use, and users that develop new functionalities for the Balmorel model are encouraged to contribute them. However, in order not to pollute the model with specialised features that may seem irrelevant in other contexts, it is likewise important that they can be switched off when not desired. Remember: "Too many details kills any model".

In Balmorel such optional features are organised as addons. The mechanisms around this provide a strong tool for further enhancements of the model. [MORE TEXT TO COME]

12.5 Selections and Add-ons

THE FOLLOWING IS NEW FOR VERSION 3.01. THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

THIS SECTION CONTAINS UNFINISHED AND DEBATABLE MATERIAL. Think of it as an indication of the way the addons are implemented and may be applied.

Table 13 illustrates the selections.

Table 14 illustrates the possibilities for combining add-ons between them, and with (some of the? - f.eks. BBx??) selections.

Note: the + and - are symmetric; the RC and CR are "skew symmetric".

Note that one add-on is controlled by a control variables. Therefore one add-on may consist of several functional entities, e.g., two equations.

	BB1	BB2	BB3	AO4	COMB	HeatTrans	AO7
BB1		-	-	-	+	+	
BB2	-		-	-	+	-	
BB3	-	-		?	+	-	
AO4	-	-	?		RC	+	
COMB	+	-	+	CR		RC	RC
HeatTrans	+	-	+	+	CR		
AO7	-	-	+	+	CR		

Table 14: Relations between add-ons and between add-ons and selections. Legend: +: Possible; -: impossible (alternatives); RC: possible to apply row add-on/selection without column add-on/selection , but not the other way; CR: possible to apply column add-on/selection without row add-on/selection , but not the other way; ?: Unknown, untested.

Note that Table 14 contain only those columns for which not all entries has "+". Thus, the left column in Table 14 contains a list of all add-ons and (some?) selections, and the columns list those add-ons that do not go unconditionally (i.e., do not have "+" in all entries) with all other add-ons (and selections?).

Organising add-ons

The specification of whether to include an add-on or not is handled by a control located in file balgams.opt.

Suppose for instance that you have a functionality related to electricity exchange with third regions based in part on electricity prices in those regions. The control to this functionality will be called 'x3v'. Therefore in file balopt.opt you specify

$setglobal x3v yes

if you want to use it, and you specify

$setglobal x3v no

if you do not.

At the appropriate places in the GAMS code you add the necessary code changes, which will be something like

$ifi %x3v%==yes $include onenecessaryextra

In the base folder there is a subfolder called addons. Below that folder, add a folder called x3v. Below that again add folder model and folder datamine [? -. or data?], and you then have the following:

base/addon/x3v/model/
base/addon/x3v/datamine/

In folder 'base/addon/x3v/model/' you place a number of files with code parts, that have to go in various places in Balmorel.gms or *.sim. Typically one file for declaration of variables, one file for declaration of equations, one file for definition of equations, one file for inclusion of data, etc.

In folder 'base/addon/x3v/datamine/' you place files holding data.

There will be a number of places where the modifications can be placed within the existing code. You should try to keep the same order of thing as in the point of departure, in particular this means: keep the definitions and declaration of variables, equations and model in the Balmorel.gms file, close to those that are already there. Keep any looping in the *.sim file.

Naming conventions: When making an addon please try to signify by the naming of GAMS identifiers (parameters, sets, equation, variables, etc.) that they belong to that addon. E.g. if you have made an add-on related to unit commitment, you could call the add-on UNITCOMM, and all GAMS identifies could have a 'UC' in their name. You should also try to adhere to applicable parts of the conventions mentioned in Section 1.4. In particular note the prefix conventions related to functionalities:

V: variable

Q: equation

VQ: variable that ensures feasibility in an equation

O: related to output from simulations

I: internal

Names of GAMS identifiers introduced for UNITCOMM could then for variables be for instance VUCUP, VUCDOWN, for equations for instance QUCSTART, QUCMIN etc.

In the Balmorel core code the length of the GAMS identifiers are limited to 10 characters, cf. Section 1.4, in part to facilitate printing using put statements. Typically the printing (if any) from an add-on will be from code by the author of the add-on, and only there, so the limitation to 10 characters seems not essential for add-ons.

Do not forget to make code for errors and log related to the add-on. See Section 11.

12.6 Control variables in the BUI environment

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

Control variables are defined in three different ways: "$set varname value", "$setlocal varname value" or "$Setglobal varname value", where 'varname' is the name of the control variable. (Unfortunately GAMS allows one to define scoped local and global variables with the same name but treats them as different under some cases. In the face of this it may be advisable to only use one type of control variable in any application or at least only use a name once.)

Here only $Setglobal is used, it has the property that the definition has effect anywhere in the GAMS program following the definition.

A control variable will be used by checking its value (which here is supposed to be a text string) against another value, e.g. as follows for a control variable "ABC":

```
$ifi %ABC%==yes display "ABC was set to yes";
$ifi not %ABC%==yes display "ABC was not set to yes";
```

The '%'s indicate that the text string specified in the control variable definition is used, e.g. with "$Setglobal ABC yes" the text string is "yes" and GAMS execution will display "ABC was set to yes".

The BUI provides certain facilities to handle control variables. One aspect concerns the values that a particular control variable may take. These are specified using the !option keyword below the specification, e.g. as follows

```
$setglobal ABC yes
*!option yes
*!option no
```

or

```
$setglobal Balbase BB2
*!description Choice of model to apply
*!option BB1
```

*!description Annual simulation, exogenous investments
*!option BB2 !description Annual simulation, endogenous investments
*!option BB3
*!description Seasonal simulation, exogenous investments

As seen, a description may be entered. For the $setglobal line, the description is entered immediately below. For a *!option line the description may be entered in the following line, or it may be entered in the *!option line after the specification of the option.

The list of options should be extensive, such that the BUI can visualize all options and check the setting.

Often there are in reality only two options, like specifying "yes" and specifying "no". However, it may be convenient that one is seen as default, this may be handled as this

$setglobal ABC yes
*!option yes
*!select0or1

The intention here is that the control may be specified as "$setglobal ABC yes" or "$setglobal ABC ", and that in the application only "$ifi %ABC%==yes ..." and "$ifi not %ABC%==yes ..." are used.

Another possibility is that exactly one among the given list of options should be used, this may be specified by using *!select1. Both !select0or1 and *!select1 may be used if there are more than one options provided.

ALTERNATIV.

The list of options should be extensive, such that the BUI can visualize all options and check the setting.

Often there are in reality only two options, like specifying "yes" and specifying "no". Moreover, it may be intended that in the code only the "yes" is used, i.e. only these two forms are used: "$ifi %ABC%==yes ..." and "$ifi not %ABC%==yes ...". To signal this, the following should be used for a control variable ABC:

$setglobal ABC yes
*!option yes !singleoption?

ALTERNATIV SLUT

More complicated dependencies between control variable settings may be specified. For this, the following keywords for relations between setting of two control variables are provided:

!onlyif
!notif
!mustif
!mayif

Example:

$setglobal ABC A1
*!option A1 !onlyif EFG==EFG2
*!option B1
*!option C1 !mustif EFG==EFG3
*!option D1 !notif EFG==EFG4

$setglobal EFG EFG1
*!option EFG1
*!option EFG2 !notif ABC==C1 !notif ABC==B1
*!option EFG3
*!option EFG4 !notif ABC== D1

(Skal der vre procenttegn om?:EFG2 notif %ABC%==%C1%)

The information provided by !onlyif may in the example be seen as establishing an implication like "ABC==A1 ⇒ EFG==EFG2". Similarly the !onlyif may be seen to establish the implication "EFG==EFG3 ⇒ ABC==C1".

Following this idea, mutually exclusive control choices may be established by two !notif specifications, as above "ABC==D1 ⇒ not EFG==EFG4" and "EFG==EFG4 ⇒ not ABC==D1". Control choices that must be simultaneously chosen or not chosen may be specified using !onlyif for both; or !mustif for both; or !onlyif for one and !mustif for the other. Such expressions correspond to the equivalence relation "⇔".

The !mayif keyword is used to signify that the developer has considered simultaneous use of two controls and found it possible.

The expression following !onlyif, !notif, !mustif, !mayif will in the BUI as default be handled as case insensitive, implying e.g. that "EFG4 !notif ABC==D1" and "efG4 !notif AbC==d1" will lead to the same conclusion. If the control is to be treated as case sensitive, then this may be specified using the keyword !useifi (indicating that the control statemet will start with "$ifi", and not with "$if ").

Now compare with e.g. Table 13. Here the '+' signifies that two addons may be used simultaneously, this corresponds to using !mayif. On the other hand the '-' in the table corresponds to using !notif on both controls. The 'RC' and 'CR' in the table may be seen to correspond to suitable use of !onlyif and !mustif. Finally, '?' or an empty entry in the table correspond to not specifying any of those relational keywords.

(UD?) You may have a functionality that is kept as an add-on, e.g. UNITCOMM. When the add-on is active, there may be additional choices for the details, e.g. that UCstartup may be active or not, and that UCshutdown may be active or not. This may be expressed as e.g.:

```
$setglobal UNITCOMM yes
*!option yes !description Unit commitment will be represented
*!select0or1
$setglobal UCstartup yes
*!option yes !onlyif %UNITCOMM%==yes
*!select0or1
$setglobal UCshutdown
*!option yes !onlyif %UNITCOMM%==yes
*!description Shot-down costs for unit commitment will be considered
*!select0or1
```

As seen the controls...

Summary of syntax:

- Keywords: !option, !description, !select1??, !select0or1??, !onlyif, !notif, !mustif, !mayif, !singleoption??, !useifi. The keywords themselves are treated as case insensitive.

- Tilfojelse: der kan vare, at der er nogle control variables, der formelt er control variables, men i virkeligheden er 'faste' (f.eks. fordi de angiver praedefinerede stinavne). Det kan derfor vare fornuftigt (?) at indfore f.eks. !fixtext for saadanne.

- Og: det kan vare, at nogle control variables angiver user specifed names, dvs., ret frie - men de skal have control variable form ($setglobal etc.). Fro saadanne kan indfores !freetext

- og friere endnu: det er jo heller ikke sikkert, at vi onsker at bruge $setglobal, andet er vel ogs tankeligt(ikke mindst jo !$setlocal). Til dette formaal kunne man vel indfore en slags 'fri-tekst blok', f.eks.: !blockstart [nylinie] ¡noget tekst¿ [evt. flere linier med tekst][nylinie] !blockend. For at kunne navngive den i BUI, boer der vel vaere et !name eller !blockname. (I ovrigt giver jeg implicit alle $setglobal et name, nemlig navnet p kontrolvariablen. Dette kunne jeg jo saa valge at opfatte som et defaultname, der kan overskrives med et !name.)

- A keyword line must start with "*!" , where the "!" is part of a following keyword. Thus, a valid line may start as "*! !option", "*!!option" and "*!option".
- The *!option refers to the preceding $Setglobal statement. HALLO: skal der tilfojes, at der ikke maa vaere blanke linier (men gerne kommentarlinier) imellem?
- A *!description refers to the preceding *!option or $setglobal statement. [same line for option?]
- A description may also be entered in the *!option line; it extends from the !description and to either the end of the line or to the appearance of a '!', whichever comes first. Behov for quotes? Special characters?("The description may be enclosed in a pair of quotes, and must be so if it contains special characters (such as '!')" ?).
- gerne flere linier med keywords til hver kontrol??
- The control variable will be handled as case insensitive unless !useifi (!useif ??) is specified. Men er der behov for at skelne? Vi kan jo blot i BUI bevare den i !option inddaterede upper/lower?
- (No blank lines?? (if needed for visibility, you may add lines that contain only the initial *)
- Kan man bruge tomme? [se mine eksperimenter med gams-kode, hjemmePC]
- Skal der bruges % i f.eks. onlyif UNITCOMM==yes ?
- The use of control variables is more restricted here than in GAMS. Thus, only $setglobal may be used. The % symbol may be redefined in GAMS, but % must be used here. One may use control and environmental variables in GAMS, here only control variables may be used. In GAMS one either tests for the existence of a control variable or on the contents of the control variable, here only the contents of the control variable may be used. Quoted forms are not allowed here.

12.6.1 Control variables - file balopt.opt

- $Setglobal PRINTFILES !option yes
- $Setglobal bb1
- $Setglobal bb2
- $Setglobal
- $Setglobal COMBTECH
- $Setglobal MP
- $Setglobal GAS
- $Setglobal X3V
- $Setglobal QUAD

- $Setglobal WELFARE
- $Setglobal WATERVAL
- $Setglobal H2
- $Setglobal transport
- $Setglobal NAP
- $Setglobal DECOMP SKAL VEL HEDDE decomm?
- $Setglobal DECOMPEFF indicates that technology is decomissioned on the basis of profitability. When revenues can no longer cover both variable and fixed operating costs, capacity is decomissioned.
- $Setglobal CAES
- $Setglobal RENEWTARGET
- $Setglobal FOSSILTARGET
- $Setglobal WINDTARGET
- $Setglobal HEATTRANS
- $Setglobal NOELTRANSINV Disable investments in electricity transmission capacity
- $Setglobal WNDSHUTDOWN Allow wind turbines to shut down
- $Setglobal timeaggr
- $Setglobal quickfix
- $Setglobal MAKEWATER
- $Setglobal MAKEACCESS Create access file from the output gdx
- $Setglobal MAKEINVEST Used in a BB2 model to generate additional capacity for other models
- $Setglobal CREATETIME
- $Setglobal CREATEDTIME
- $Setglobal MOVERESULTS moveresults.bat
- $Setglobal ADDINVEST
- $Setglobal MAKEINVEST
- $Setglobal FOSSILTARGET
- $Setglobal RENEWTARGET
- $Setglobal WINDTARGET
- $Setglobal EXECUTEUNLOADINPUTDATAHans
- $Setglobal UnitComm
- $Setglobal RESEHARE(?)
- $Setglobal

12.7 Combination technologies

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.01.

The addon in relation to combination technologies is controlled by the control variable COMBTECH. To use the feature, specify "$Setglobal COMBTECH yes" in file balopt.opt, otherwise specify "$Setglobal COMBTECH".

The term combination technology will denote sets of two or more technology units that have some mutual bindings. The purpose of this is to introduce the possibility to mix linearly the properties of different technologies, in proportions that are results of the optimisation. Maximum and minimum shares on the proportion attained by each unit in a combination technology unit may be specified.

One example of an application is the combination of two technologies with different efficiencies but otherwise equal. In this case the optimisation will ensure that the less efficient technology will only be used if the more efficient one is already used at the specified maximum share; this may then be seen as a representation of one technology with an efficiency that decreases with increasing production level. Another example is that one technology using coal is combined with one technology using gas. It may be specified that the production based on natural gas can be at most e.g. 0.8 (i.e., 80%) of total production while coal can be at most 0.5. Further examples could specify minimum or maximum shares of total production from each of the combination technologies.

Application of combination technologies is assumed to be relevant only for technologies that traditionally are operated according to economic dispatch.

Here is the necessary input.

- The value in GDATA(GGG,'GDCOMB') is used to indicate combination technologies. The values are 0 (indicating that this is not a combination technology), 1 (indicating that this is a combination technology, and that it is the primary unit in a combination, 2 (indicating that this is a combination unit, and that it is a secondary unit in a combination (see below)).
- Define GGCOMB(GGG,IGGGALIAS) where the pair (GGG,IGGGALIAS) constitutes a combination technology combination, GGG must be a primary technology and IGGGALIAS must be a secondary technology. GGG and IGGGALIAS must be of the same technology type (as indicated in GDATA(GGG,'GDTYPE')). There can be more than one secondary technology to a primary technology, e.g., the definition "SET GGCOMB(GGG, IGGGALIAS)/ Gex-prim.(Gex-secA, Gex-sec2, Gex-secS) /;" is legal if the technologies Gex-prim, Gex-secA, Gex-sec2, Gex-secS are in GGG. A combination technology can be in only one combination technology combination.
- The technologies in a combination pair must be of the same type, specified in GDATA('GDTYPE').
- Technologies can not be combination technologies if they are not dispatchable, and storage technologies can not be combination technologies. Hence, a combination technology in G must be in the set (IGDISPATCH(G)- IGESTO(G) - IGHSTO(G)).
- The value GDATA(GGG,'GDCOMBSK') specifies the maximum fraction (value between 0 and 1) of derated installed capacity of the combination technology that this technology can produce. This is used to assign upper bounds on the production variables. The fraction is defined with respect to VGE_T if GGG is in IGKE, VGH_T if GGG is in IGHE.
- The value GDATA(GGG,'GDCOMBSLO') specifies the minimum fraction (value between 0 and 1) that the production from technology GGG must constitute of total production from the combination technology combination of which GGG is part. The production in question is VGE_T if GGG is in IGKE, VGH_T if GGG is in IGKH.

- The value GDATA(GGG,'GDCOMBSUP') specifies the maximum fraction (value between 0 and 1) that the production from technology GGG must constitute of total production from the combination technology combination of which GGG is part. The production in question is VGE_T if GGG is in IGKE, VGH_T if GGG is in IGKH.

- Secondary technologies must have zero values in GDATA(G,'GDINVCOST0') and GDATA(G,'GDOMFCOST'). This may be interpreted to mean that all investment costs and fixed operations and maintenance cost are related to the primary technology.

- The values in GDATA for 'GDFROMYEAR', 'GDLIFETIME', 'GDKVARIABL' must be the same for pairs of combination technologies in GGCOMB (GGG,IGGGALIAS).

Here are some details on the internal mechanisms.

- SET IGCOMB1(G): The set of primary combination technologies in G. Defined as IGCOMB1(G)=YES$(GDATA(G,'GDCOMB') EQ 1).

- SET IGCOMB2(G): The set of secondary combination technologies in G. Defined as LOOP(IGCOMB1, IGCOMB2(G)$((GDATA(G,'GDCOMB') EQ 2) AND (GGCOMB(IGCOMB1,G))) = YES).

- IAGKN will automatically be set to true for secondary technologies with GDATA(G,'GKVARIABL') equal to 1.

- The equation QECOMBGLK(AAA,IGKE,S,T) secures for each time segment (S,T) that total electricity production from an existing combination unit in IGKE does not exceed total derated capacity. Similarly, the equation QHCOMBGKL(AAA,IGKH,S,T) secures that total heat production from an existing combination unit in IGKH does not exceed total derated capacity. For new technologies the corresponding equations are QNECOMBGKL and QNHCOMBGKL.

- The equation QECOMBSLO(AAA,G,S,T) secures for each time segment (S,T) that the share of electricity production from an existing combination unit is at least as specified in GDATA(GGG,'GDCOMBSLO'). Similarly for equations QHCOMBSLO, QNECOMBSLO and QNHCOMBSLO.

- The equation QECOMBSUP(AAA,G,S,T) secures for each time segment (S,T) that the share of electricity production from an existing combination unit is at least as specified in GDATA(GGG,'GDCOMBSLO'). Similarly for equations QHCOMBSUP, QNECOMBSUP and QNHCOMBSUP.

- Observations, Possible errors:

  - Observation: In the print files the secondary units will not normally be identified as such.
  - Observation: Since secondary units have zero capacity it might be confusing that they may never the less have positive production.
  - Observation: If there are only two technologies in a pair it seems redundant to specify both GDATA(*,'GDCOMBSLO') and GDATA(*,'GDCOMBSUP') for both technologies, since one can be derived from the other. In this case, either specify GDATA(*,'GDCOMBSLO') as 0 for both technologies and the relevant values for GDATA(*,'GDCOMBSUP'), or specify GDATA(*,'GDCOMBSUP') as 1 for both technologies and the relevant values for GDATA(*,'GDCOMBSLO'). This will be beneficial for computation and interpretation of results.

– Possible error: For combination units the capacity entered in GKFX, and possibly found endogenously, refers to the primary combination technology. It is an error if secondary combination units have positive capacity in any area.

– Possible error: Inconsistency between GGCOMB(GGG,IGGGALIAS) and GDATA(GGG,'GDCOMB')

– Possible error: Check the values in GDATA(GGG,'GDCOMBSK'), GDATA(GGG,'GDCOMBSLO'), GDATA(GGG,'GDCOMBSUP').

– Possible error: In the definition of GGCOMB(G,IGALIAS), G must be a primary combination technology, and IGALIAS must be a secondary combination technology.

– Possible error: The dollar control option $ONEMPTY must be set.

12.8 No new investments

The model balbase2 is intended for analysis of long term development and therefore contains as an integrated feature the possibility to expand generation and electrical transmission capacities according to economic criteria.

User specified expansion and de-commissioning of generation capacity is specified in the parameter GKFX, Section 4.22.1.

Automatically generated expansion of generation capacity will be undertaken for those technologies for which a 1 is specified for GDKVARIABL in GDATA. If investment in new generation capacity should not be allowed this may be implemented by setting a 0 for GDKVARIABL in GDATA.

If investment in new electrical transmission capacity should not be allowed this may be implemented as follows. Include the line

```
VXKN.FX(IRE,IRI) = 0;
```

in the balbase2.sim annual updating section.

If no investments at all shall be allowed, use the model balbase1.

12.9 No electricity transmission

Electricity transmission between regions may be excluded by including the following statement

```
VX_T.FX(IRE,IRI,S,T) = 0;
```

in the BALMOREL.GMS annual updating section.

Observe that this will not influence transmission to third regions. Also observe that the above modification will prevent transmission within a country that has more than one region. To permit such transmission anyway, the above statement should be refined to the following form:

```
LOOP(C,VX_T.FX(IRE,IRI,S,T)$ (NOT(CCCRRR(C,IRE) AND CCCRRR(C,IRI)))=0);
```

[This feature is not properly tested.]

12.10 Price dependent electricity exchange with third countries

(Note: the material described in this section is now (version 2.12A) implemented as standard. However, a few desirable features are not implemented yet, in particular print facilities.)

Price dependent electricity exchange with places outside the simulated geographical scope ('third countries' or 'third regions') may be used together with the fixed electricity exchange with third countries. The price-quantity relationships are given as a piecewise step function. There are card(X3VSTEP) (Section 12.10) steps applied in simulation, and card(X3VSTEP0) steps may be given in the data. The length (MW) of each import step is X3VIMQ and X3VEXQ (Section 12.10) of each export step. The associated prices are X3VIMP and X3VIMP (Section 12.10), respectively. The prices are given on a yearly basis, the value for the currently simulated year are held in IX3VPIM_Y, and IX3VPEX_Y (Section 12.10). The exchange is assumed to be lossless and without transmission cost.

Potential places with which there may be price dependent electricity exchange are given in the set X3VPLACE0 (Section 12.10). The simulated price dependent electricity exchange transmission connections are specified in the set X3VX (Section 12.10). The set RX3VSUBSTI (Section 12.10) may be used to reduce the risk of user errors.

Associated variables are VX3VIM_T and VX3VEX_T, Section 12.10.

Observe that there is with this construction no relationship with the fixed electricity exchange with third countries (Section 4.11.1), i.e., the two types of exchange exist side by side.

Name	Domain	Type	Unit	Defined in	Page
IX3VPIM_Y	(RRR,X3VPLACE0,X3VSTEP0,S,T)	parameter	Money/MWh	BALMOREL.GMS	97
IX3VPEX_Y	(RRR,X3VPLACE0,X3VSTEP0,S,T)	parameter	Money/MWh	BALMOREL.GMS	97
RX3VSUBSTI	(RRR,X3VPLACE0)	set	-	SETS.INC	97
X3VPEX	(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	X3.INC	97
X3VPIM	(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	X3.INC	97
X3VPLACE0	-	set	-	SETS.INC	96
X3VQEX	(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	MW	X3.INC	97
X3VQIM	(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	MW	X3.INC	97
X3VSTEP0	-	set	-	SETS.INC	96
X3VSTEP	(X3VSTEP0)	set	-	SETS.INC	96
X3VX	(RRR,X3VPLACE0)	set	-	SETS.INC	96
VX3VIM_T	(RRR,X3VPLACE0,X3VSTEP0,S,T)	variable	MW	BALMOREL.GMS	97
VX3VEX_T	(RRR,X3VPLACE0,X3VSTEP0,S,T)	variable	MW	BALMOREL.GMS	97

X3VSTEP0

The set X3VSTEP0 holds the steps of the piecewise constant function giving the relationships between quantity and price for the price dependent electricity exchange with third countries. The definitions is like /X3VSTEP01*X3VSTEP03/ if three steps are used. X3VSTEP0 is used to hold data in the data base, while the subset X3VSTEP (Section 12.10) is used to indicate the steps used in simulation.

X3VSTEP

The set X3VSTEP holds the simulated steps of the piecewise constant function giving the relationships between quantity and price for the price dependent electricity exchange with third countries. The set is a subset of X3VSTEP0 (Section 12.10) and the definition is like SET X3VSTEP(X3VSTEP0) /X3VSTEP01*X3VSTEP02/ if two steps are used. (If no exchange is wanted, "IX3VSTEP(X3VSTEP)=NO" may be used, however, it is recommended that X3VX is used, Section 12.10.).

X3VPLACE0

The set of regions with which there can be price dependent electricity exchange, Section 12.10, is given by SET X3VPLACE0, defined e.g. like /X3VFARAWAY, X3VGERMAN, X3VPOLAND/. The set is used for holding data, while the set actually simulated is specified in the set X3VX (Section 12.10).

X3VX

The combinations of RRR and X3VPLACE0 that are to be simulated for price dependent electricity exchange, Section 12.10, is given by SET X3VX(RRR,X3VPLACE0). This set may be interpreted to specify the transmission lines that are assumed to be in operation between regions in the simulated geographical scope (set IR) and third countries (set X3VPLACE0). If e.g. the region 'DK_W' is in IR and 'X3VGERMAN' is in X3VPLACE0 then X3VX('DK_W','X3VGERMAN')=YES will specify that a transmission line is assumed to be in operation between the two places, and X3VX('DK_W','X3VGERMAN')=NO that it is not. Observe that the X3V kind of price dependent electricity exchange only will be possible for pairs of regions for which one region in IR, i.e. regions that are in the set C of simulated countries.

RX3VSUBSTI

The set RX3VSUBSTI is used in relation to the price dependent electricity exchange with third countries, Section 12.10. It indicates (by assigning YES) if elements in X3VPLACE0 (Section 12.10) is a substitute for a region in RRR. If it is, the price dependent exchange should by assumption only be used if the region is NOT included in a country in set C, i.e. the set RX3VSUBSTI(IR,X3VPLACE0) (where IR is a region in C) should be empty. Observe that the only function of the set RX3VSUBSTI is to help the user to avoid errors by printing an error message if relevant.

The declaration is SET RX3VSUBSTI(RRR,X3VPLACE0). It there are no substitutes then define RX3VSUBSTI(RRR,X3VPLACE0)=NO, otherwise give the real information, if any, e.g. RX3VSUBSTI('DE_R','X3VGERMAN')=YES;

X3VQIM, X3VQEX, X3VPIM, X3VPEX

The parameters X3VQIM and X3VPIM hold the quantity-price relationship for import in relation to price dependent electricity exchange, Section 12.10, and the parameters X3VQEX and X3VPEX hold the quantity-price relationship for export in relation to price dependent electricity exchange. Unit: Money/MWh for X3VPIM and X3VPEX, MW for X3VQIM and X3VQEX. The declarations are:
X3VPIM(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)
X3VQIM(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)
X3VPEX(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)
X3VQEX(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)

X3VQIM holds the limit (upper bound, step length) on import and X3VQEX on export for each particular step corresponding to the price X3VPIM and X3VPEX, respectively.

Comment on input data: It will be assumed that prices should be positive. For import the prices should be weakly increasing with ord(X3VSTEP0), for export the prices should be weakly decreasing with ord(X3VSTEP0). The prices for price dependent export to third countries should be greater than or equal to the prices for price dependent import from third places.

IX3VPIM_Y, IX3VPEX_Y

The internal parameters IX3VPIM_Y(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT) and IX3VPIM_Y(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT) hold the prices for price dependent electricity exchange with third countries, Section 12.10, the currently simulated year.

Variables VX3VIM_T, VX3VEX_T

> VX3VIM_T(RRR,X3VPLACE0,X3VSTEP0,S,T) "Imported third country price dependent electricity (MW)"
>
> VX3VEX_T(RRR,X3VPLACE0,X3VSTEP0,S,T) "Exported third country price dependent electricity (MW)"

Equations

The above variables are entered in equations VOBJ and QEEQ.

12.11 Reserve generation capacity

[This feature is not properly tested.]

It is common to model an electricity system with a constraint expressing that a certain generation capacity must be available. One version of this may be achieved as follows. It is assumed that the requirement is that in every region the available capacity in each year must exceed the maximum over (S,T) of nominal electricity demand DE by a certain percentage.

1. Make a SET GKRESSET(G) holding those generation technologies that may count as providing reserve. E.g., it could be identical to the already defined set IGDISPATCH(G).
2. Declare a PARAMETER GKRES(RRR) to hold the desired capacities.
3. Assign the appropriate values to GKRES. Thus, if e.g. 20% reserve capacity is desired in all regions, the following statement may be made:
   GKRES(IR)= DE(Y,IR)* (SMAX((S,T),DE_VAR_T(IR,S,T))
   /(IDE_SUMST(IR)))* 1.2;
4. Declare an EQUATION QGKRES(RRR).
5. Define the EQUATION as e.g.
   QGKRES(IR)..
   SUM(IA$(RRRAAA(IR,IA)), SUM(GKRESSET,
   IGKFX_Y(GKRESSET,IA) + GKVACCCO_Y(GKRESSET,IA)
   + VGKN(GKRESSET,IA))) =G= GKRES(IR);
   (An alternative version could multiply the capacities by GKDERATE.)
6. Include QGKRES in the list of equations specifying the MODEL, see Section 8.

12.12 Finer subdivision of the year

The subdivision of time within the year is given by the sets SSS, S, TTT and T. A number of parameters depend on this subdivision, see Section 4.19 and Section 4.20. All these parameters are found in the file VAR.INC, cf. Section 2.

¿From a given subdivision it is easy to aggregate such that only one time segment per year is used, cf. Section 3.2.2.

If another subdivision is desired (with more than one segment per year), then the mentioned sets and all the parameter values in Section 4.19 and Section 4.20 must be changed accordingly.

As a consequence of such revised subdivision it should be expected that also the calibration must be repeated, cf. Section 9.

A version for use with hourly values over one year is due to be made available in 2005.

12.13 Making more parameters depend on the year

Some of the parameters have been made dependent on the year, e.g., demands (parameters DE and DH) and fuel prices (parameter FUELPRICE). It might for certain analyses be desirable to have more parameters depending on the year, e.g. distribution costs and losses, generating units' efficiencies, etc.

This may be done by copying the ideas from those parameters that presently do depend on the year, e.g. DE (see Section 4.29.32).

12.14 Making more parameters depend on geography

Description to be entered.

12.15 Market power

A version deling with incomplete competition and market power in the electricity sector is will be made available in 2005, see the report on www.Balmorel.com.

12.16 Markets for emission, renewable energy and JI

Markets for tradable CO2 emission permits (TEP), markets for tradable green certificats, and joint implementation are relevant issues for modelling using Balmorel. See "Co-existence of electricity, TEP and TGC markets in the Baltic Sea Region" at the Balmorel home page for an example.

12.17 Investment costs depend on technical life time

[This feature is not properly tested.]

Investment decisions related to generation technology depend among other things on the product of the investments cost, given in GDATA(G,'GDINVCOST'), and ANNUITYC. This choice of representation is motivated in part by the expectation that the economical life time of the generation units is not longer than the technical life time, and therefore the technical life time need not be represented. However, in some studies this may be inappropriate. The following describes the introduction of this dependency.

1. Declare in the file GEOGR.INC the PARAMETER ANNUITYGC(GGG,CCC) and enter the appropriate values in a TABLE, e.g. as

   TABLE ANNUITYGC(GGG,CCC)

	Denmark	Estonia	Finland	
CC-Con00-G	0.1627	0.1993	0.1627	
GHydro-res	INF	0.0806	0.0651	;

   ANNUITYGC(G,C)$(ANNUITYGC(G,C) EQ 0)=0.1315;

   (As seen, also a default value of 0.1315 has been entered (corresponding to a 10% interest rate with 15 years of economic life time, cf. Table 5 and Section 4.5.1).)

2. In the file BALMOREL.GMS all instances of ANNUITYC(C) should be replaced by ANNUITYGC(G,C), except where it is multiplied by XINVCOST.

In the above modification the investment on transmission capacity will be evaluated using the values in ANNUITYC.

If the modification desired is not directed towards differentiation of the life time within the set of generation technologies but rather towards the differentiation between life times for (all) generation technologies and transmission, then the following modification could be used.

1. Declare in the file GEOGR.INC the PARAMETER ANNUITYXC(CCC) and enter the appropriate values.

2. In the file BALMOREL.GMS all instances of ANNUITYC(C) that are multiplied by XINVCOST should be replaced by ANNUITYXC(C).

It is possible to introduce the two types of modifications simultaneously.

12.18 Version numbering and referencing

The Balmorel model exists in various versions. We shall here clarify the naming of these.

A distinction will be made between model structure and data (cf. also Sections 1.2 and 14). By model structure is mainly meant identifiers (i.e., the names) of the SCALARS, PARAMETERS, VARIABLES, SETS and EQUATIONS in the model, plus additional information like limitations on variables (declarations as POSITIVE or FREE, specifications of bounds (.UP, LO, or .FX)), and the structure related to the dynamics. By the data is meant the actual labels (i.e., members) in the SETS and the actual numerical values assigned to SCALARS and PARAMETERS. Thus, the present document deals with model structure and not with data.

The identification of different versions should distinguish between model structure versions and data versions.

If therefore an analysis is performed where the model used consists of e.g. a model structure called Balmorel version 2.17, modified to exclude transmission, and using data that mostly consisted of data called Balmorel version 2.10, this may be referred to as " ... the analysis used the Balmorel model structure version 2.17, modified to exclude transmission. The data used in the analysis was based on Balmorel data set version 2.10, modified as follows: .... ".

13 User interface

THE FOLLOWING IS NEW FOR VERSION 3.01.

The GAMS IDE (Integrated Development Environtment) is suitable for developing and handling the GAMS code, but less suited for sustained model application, or data management.

A user interface is under development under the name BUI (Balmorel User Interface). Details to come.

14 Overview of model structure components

The model structure consists of the sets, parameters and variables in the model and the relations between them, as described in the preceding pages. Here an overview of the components will be given.

The table identifies in alphabetical order all sets, obligatory set members, scalars, parameters, variables, and equations in the Balmorel model, with specification of the units in which they are given (where relevant, see also page 44), in which file the component is declared and at which page in this document the component is described.

Another way to get an overview over the model components is the use the compiler directives $ONSYMXREF, $ONSYMLIST, $ONUELLIST and $ONUELXREF (the $ in the first position of the line) which produce maps in the LST file.

Not identified in the table are the following aspects:

- Upper bounds and lower bounds on variables

- The internal working of the linking between technology and fuel use, cf. Section 4.8.1.

- The sequence of the statements. A particular case of this is the annual updating parts (linking the individual years).

- The constants 8760, 365, 24 and 3.6.

- Entities related to output, see Section 5.

Name	Domain	Type	Unit	Defined in	Page
1995, 1996,..	-	obl. set member	(none)	SETS.INC	28
AAA	-	set	-	SETS.INC	26
AAARURH	(AAA)	set	-	SETS.INC	26
AAAURBH	(AAA)	set	-	SETS.INC	26
AGKN	(AAA,GGG)	set	-	SETS.INC	36
ANNUITYC	(CCC)	parameter	(none)	GEOGR.INC	46
C	(CCC)	set	-	SETS.INC	25
CCC	-	set	-	SETS.INC	25
CCCRRR	-	set	-	SETS.INC	26
DAYTYPE	-	set	-	SETS.INC	30
DE	(YYY,RRR)	parameter	MWh	DE.INC	50
DE_VAR_T	(RRR,SSS,TTT)	parameter	(none∼MW)	VAR.INC	55
DEF	-	set	-	SETS.INC	37
DEF_D1	-	set	-	SETS.INC	37
DEF_D2	-	set	-	SETS.INC	37
DEF_STEPS	(RRR,SSS,TTT,..				
	..,DF_QP,DEF)	parameter	(none), Money/MWh	GEOGR.INC	57
DEF_U1	-	set	-	SETS.INC	37
DEF_U2	-	set	-	SETS.INC	37
DEFP_BASE	(RRR)	parameter	Money/MWh	GEOGR.INC	47
DEFP_CALIB	(RRR,SSS,TTT)	parameter	Money/MWh	GEOGR.INC	56
DF_QP	-	set	-	SETS.INC	36
DF_PRICE	-	obl. set member	Money/MWh	GEOGR.INC	36
DF_QUANT	-	obl. set member	MW	GEOGR.INC	36
DH	(YYY,AAA)	parameter	MWh	DH.INC	50
DH_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	VAR.INC	54
DHF	-	set	-	SETS.INC	37
DHF_STEPS	(AAA,SSS,TTT,..				
	..,DF_QP,DEF)	parameter	(none), Money/MWh	GEOGR.INC	58
DHF_D1	-	set	-	SETS.INC	37
DHF_D2	-	set	-	SETS.INC	37
DHF_U1	-	set	-	SETS.INC	37
DHF_U2	-	set	-	SETS.INC	37
DHFP_BASE	(AAA)	parameter	Money/MWh	GEOGR.INC	47
DHFP_CALIB	(AAA,SSS,TTT)	parameter	Money/MWh	GEOGR.INC	55

Name	Domain	Type	Unit	Defined in	Page
DISCOST_E	(RRR)	parameter	Money/MWh	GEOGR.INC	47
DISCOST_H	(AAA)	parameter	Money/MWh	GEOGR.INC	47
DISLOSS_E	(RRR)	parameter	(none)	GEOGR.INC	47
DISLOSS_H	(AAA)	parameter	(none)	GEOGR.INC	47
FFF	-	set	-	SETS.INC	35
FDCO2	-	obl. set member	kg/GJ	FUELS.INC	35
FDATASET	-	set	-	SETS.INC	35
FDATA	(FFF,FDATASET)	set	-	SETS.INC	48
FDNB	-	obl. set member	(none)	FUELS.INC	35
FKPOTA	(FKPOTSETA,AAA)	parameter	MW	GEOGR.INC	53
FKPOTC	(FKPOTSETC,CCC)	parameter	MW	GEOGR.INC	53
FKPOTR	(FKPOTSETR,RRR)	parameter	MW	GEOGR.INC	53
FKPOTSETA	(FFF)	set	-	SETS.INC	35
FKPOTSETC	(FFF)	set	-	SETS.INC	35
FKPOTSETR	(FFF)	set	-	SETS.INC	35
FDSO2	-	obl. set member	kg/GJ	FUELS.INC	35
FUELPRICE	(YYY,AAA,FFF)	parameter	Money/GJ	FUELP.INC	57
G	(GGG)	set	-	SETS.INC	31
GGCOMB	(GGG,IGGGALIAS)	set	-	SETS.INC	93
GGG	-	set	-	SETS.INC	31
GDCB	-	obl. set member	-	SETS.INC	50
GDCV	-	obl. set member	-	SETS.INC	50
GDATASET	-	set	-	SETS.INC	31
GDESO2	-	obl. set member	-	SETS.INC	50
GEFFDERATE	(GGG,AAA)	parameter	(none)	GEOGR.INC	49
GDATA	(GGG,GDATASET)	parameter	-	SETS.INC	50
GDFE	-	obl. set member	-	SETS.INC	50
GDFROMYEAR	-	obl. set member	-	SETS.INC	51
GDFUEL	-	obl. set member	-	SETS.INC	50
GDINVCOST0	-	obl. set member	-	SETS.INC	51
GINVCOST	(GGG,AAA)	parameter	MMoney/MWh	GEOGR.INC	49
GDKVARIABL	-	obl. set member	-	SETS.INC	51
GDNOX	-	obl. set member	-	SETS.INC	50
GDOMFCOST0	-	obl. set member	-	SETS.INC	51
GDOMVCOST0	-	obl. set member	-	SETS.INC	51
GOMVCOST	(GGG,AAA)	parameter	Money/MWh	GEOGR.INC	49
GOMFCOST	(GGG,AAA)	parameter	MMoney/MW	GEOGR.INC	49
GDAUXIL	-	obl. set member	-	SETS.INC	50
GDSTOHLOAD	-	obl. set member	hours	SETS.INC	51
GDSTOHUNLD	-	obl. set member	hours	SETS.INC	51
GDTYPE	-	obl. set member	-	SETS.INC	50
GKDERATE	(GGG,AAA,SSS)	parameter	(none)	GEOGR.INC	56
GKFX	(YYY,AAA,GGG)	parameter	MW	GKFX.INC	56
HYPPROFILS	(AAA,SSS)	parameter	Money/MWh	VAR.INC	54
IA	(AAA)	int. set	-	BALMOREL.GMS	38
IAGK_Y	(AAA,G)	int. set	-	BALMOREL.GMS	63
IAGKN	(AAA,G)	int. set	-	BALMOREL.GMS	41
IANYSET	-	int. set	-	BALMOREL.GMS	60
ICA	(XYZ)	int. set	-	BALMOREL.GMS	39
IARURH	(AAA)	int. set	-	BALMOREL.GMS	38
IAURBH	(AAA)	int. set	-	BALMOREL.GMS	38
IBALVERSN		int. scalar	-	BALMOREL.GMS	59
IDAYSIN_S	(S)	int. parameter	(none)	BALMOREL.GMS	61
IDE_SUMST	(RRR)	int. parameter	(none∼MWh)	BALMOREL.GMS	61
IDE_T_Y	-	int. parameter	MW	BALMOREL.GMS	64
IDEFP_T	-	int. parameter	Money/MWh	BALMOREL.GMS	65
IDH_SUMST	(AAA)	int. parameter	(none∼MWh)	BALMOREL.GMS	61
IDH_T_Y	-	int. parameter	MW	BALMOREL.GMS	65
IDHFP_T	-	int. parameter	Money/MWh	BALMOREL.GMS	65

Name	Domain	Type	Unit	Defined in	Page
IFUELP_Y	(AAA,FFF)	int. parameter	Money/GJ	BALMOREL.GMS	63
IG2LEVEL	(G)	int. set	-	BALMOREL.GMS	40
IGBPR	(G)	int. set	-	BALMOREL.GMS	39
IGCND	(G)	int. set	-	BALMOREL.GMS	39
IGDISPATCH	(G)	int. set	-	BALMOREL.GMS	41
IGETOH	(G)	int. set	-	BALMOREL.GMS	39
IGEOREH	(G)	int. set	-	BALMOREL.GMS	41
IGEXT	(G)	int. set	-	BALMOREL.GMS	39
IGGGALIAS	alias (GGG)	int. set	-	BALMOREL.GMS	41
IGHOB	(G)	int. set	-	BALMOREL.GMS	39
IGHORHERUR	(G)	int. set	-	BALMOREL.GMS	41
IGHYRS	(G)	int. set	-	BALMOREL.GMS	40
IGKFX_Y	(GGG,AAA)	int. parameter	MW	BALMOREL.GMS	63
IGKE	(G)	int. set	-	BALMOREL.GMS	41
IGKH	(G)	int. set	-	BALMOREL.GMS	41
IGKVACCTOY	(G,AAA)	int. parameter	MW	BALMOREL.GMS	63
IGNOTETOH	(G)	int. set	-	BALMOREL.GMS	41
IGNUC	(G)	int. set	-	BALMOREL.GMS	40
IGSOLE	(G)	int. set	-	BALMOREL.GMS	40
IGWND	(G)	int. set	-	BALMOREL.GMS	40
IHOURSIN24	(T)	int. parameter	(none)	BALMOREL.GMS	61
IHOURSINST	(S,T)	int. parameter	hours	BALMOREL.GMS	60
ILIM_CO2_Y	(C)	int. parameter	t	BALMOREL.GMS	64
ILIM_NOX_Y	(C)	int. parameter	kg	BALMOREL.GMS	64
ILIM_SO2_Y	(C)	int. parameter	t	BALMOREL.GMS	64
IM_CO2	(G)	int. parameter	kg/GJ	BALMOREL.GMS	62
IM_SO2	(G)	int. parameter	kg/GJ	BALMOREL.GMS	62
IOFxyz	-	int. scalar	-	BALMOREL.GMS	60
IPLUSMINUS	-	int. set	-	BALMOREL.GMS	43
IR	(RRR)	int. set	-	BALMOREL.GMS	39
ISALIAS	alias (S)	int. set	-	BALMOREL.GMS	39
ISCALAR1	-	int. scalar	-	BALMOREL.GMS	60
ISOLESUMST	(AAA)	int. parameter	(none∼MWh)	BALMOREL.GMS	62
IST	(S,T)	int. set	-	BALMOREL.GMS	39
ISTS	(S)	int. set	-	BALMOREL.GMS	39
ISTT	(T)	int. set	-	BALMOREL.GMS	39
ITALIAS	alias (T)	int. set	-	BALMOREL.GMS	39
ITAX_CO2_Y	(YYY,CCC)	int. parameter	Money/t	BALMOREL.GMS	63
ITAX_NOX_Y	(YYY,CCC)	int. parameter	Money/kg	BALMOREL.GMS	64
ITAX_SO2_Y	(YYY,CCC)	int. parameter	Money/t	BALMOREL.GMS	64
IWEIGHSUMS	(S)	int. parameter	(none)	BALMOREL.GMS	60
IWEIGHSUMT	(T)	int. parameter	(none)	BALMOREL.GMS	60
IWND_SUMST	(AAA)	int. parameter	(none∼MWh)	BALMOREL.GMS	61
IWTRRRSUM	(AAA)	int. parameter	(none∼MWh)	BALMOREL.GMS	62
IWTRRSSUM	(AAA)	int. parameter	(none∼MWh)	BALMOREL.GMS	62
IX3VPIM_Y	(RRR,X3VPLACE0,...				
	...,X3VSTEP0,S,T)	parameter	Money/MWh	BALMOREL.GMS	97
IX3VPEX_Y	(RRR,X3VPLACE0,...				
	...,X3VSTEP0,S,T)	parameter	Money/MWh	BALMOREL.GMS	97
IXKINI_Y	(IRRRE,IRRRI)	int. parameter	MW	BALMOREL.GMS	63
IXKN	(IRRRE,IRRRI)	int. set	-	BALMOREL.GMS	63
IX3FX_T_Y	(RRR,S,T)	int. parameter	MW	BALMOREL.GMS	65
IX3FXSUMST	(RRR)	int. parameter	(none∼MWh)	BALMOREL.GMS	62
LIM_CO2	(YYY,CCC)	obl. set member	t	MPOL.INC	38
LIM_NOX	(YYY,CCC)	obl. set member	kg	MPOL.INC	38
LIM_SO2	(YYY,CCC)	obl. set member	t	MPOL.INC	38
MPOLSET	-	set	-	SETS.INC	38
PENALTYQ	-	scalar	-	SETS.GMS	44

Name	Domain	Type	Unit	Defined in	Page
QECOMBGKL	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QECOMBSLO	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QECOMBSUP	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QHCOMBGKL	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QHCOMBSLO	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QHCOMBSUP	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNECOMBGKL	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNECOMBSLO	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNECOMBSUP	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNHCOMBGKL	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNHCOMBSLO	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QNHCOMBSUP	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	94
QEEQ	(RRR,S,T)	equation	MW	BALMOREL.GMS	68
QESTOVOLT	(AAA,S,T)	equation	MW	BALMOREL.GMS	68
QESTOLOADT	(AAA,S,T)	equation	MW	BALMOREL.GMS	68
QGCBGBPR	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGCBGEXT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGCVGEXT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGGETOH	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGNCBGBPR	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGNCBGEXT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGNCVGEXT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGNGETOH	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGEKNT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGE2LEVEL	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	69
QGHKNT	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGKNWND	(RRR,AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGKNSOLE	(RRR,AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QGKNHYRR	(AAA,G,S,T)	equation	MMoney	BALMOREL.GMS	68
QHSTOVOLT	(AAA,S,T)	equation	MW	BALMOREL.GMS	68
QHSTOLOADT	(AAA,S,T)	equation	MW	BALMOREL.GMS	69
QHYRSSEQ	(AAA,S)	equation	MMoney	BALMOREL.GMS	68
QHNRURPROP	(AAA,G,S,T)	equation	MW	BALMOREL.GMS	68
QKFUELC	(C,FKPOTSETC)	equation	MW	BALMOREL.GMS	69
QKFUELR	(RRR,FKPOTSETR)	equation	MW	BALMOREL.GMS	69
QKFUELA	(AAA,FKPOTSETA)	equation	MW	BALMOREL.GMS	69
QLIMCO2	(C)	equation	ton	BALMOREL.GMS	69
QLIMSO2	(C)	equation	ton	BALMOREL.GMS	69
QLIMNOX	(C)	equation	kg	BALMOREL.GMS	69
QOBJ	-	equation	MMoney	BALMOREL.GMS	68
QXK	(IRRRE,IRRRI,S,T)	equation	MW	BALMOREL.GMS	69

Name	Domain	Type	Unit	Defined in	Page
RRR	-	set	-	SETS.INC	26
RRRAAA	-	set	-	SETS.INC	26
RX3VSUBSTI	(RRR,X3VPLACE0)	set	-	SETS.INC	97
S	(SSS)	set	-	SETS.INC	28
SOLEFLH	(AAA)	parameter	hours	GEOGR.INC	48
SOLE_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	VAR.INC	54
SSS	-	set	-	SETS.INC	28
T	(TTT)	set	-	SETS.INC	28
TAX_CO2	-	obl. set member	Money/t	MPOL.INC	38
TAX_DE	(CCC)	parameter	Money/MWh	GEOGR.INC	47
TAX_F	(FFF,CCC)	parameter	Money/MWh	GEOGR.INC	48
TAX_DH	(CCC)	parameter	Money/MWh	GEOGR.INC	47
TAX_NOX	-	obl. set member	Money/kg	MPOL.INC	38
TAX_SO2	-	obl. set member	Money/t	MPOL.INC	38
TTT	-	set	-	SETS.INC	28
TWEEKEND	-	set	-	SETS.INC	30
TWORKDAY	-	set	-	SETS.INC	30
VDEF_T	(RRR,S,T,DET_STEPS)	variable	MW	BALMOREL.GMS	66
VDHF_T	(AAA,S,T,DHFSTEPS)	variable	MW	BALMOREL.GMS	66
VGE2LEVEL	(AAA,G,S,DAYTYPE)	variable	MW	BALMOREL.GMS	67
VESTOLOADT	(AAA,S,T)	variable	MW	BALMOREL.GMS	66
VESTOVOLT	(AAA,S,T)	variable	MWh	BALMOREL.GMS	66
VGKN	(AAA,G)	variable	MW	BALMOREL.GMS	66
VGE_T	(AAA,G,S,T)	variable	MW	BALMOREL.GMS	66
VGEN_T	(AAA,G,S,T)	variable	MW	BALMOREL.GMS	66
VGH_T	(AAA,G,S,T)	variable	MW	BALMOREL.GMS	66
VGHN_T	(AAA,G,S,T)	variable	MW	BALMOREL.GMS	66
VHSTOLOADT	(AAA,S,T)	variable	MW	BALMOREL.GMS	66
VHSTOVOLT	(AAA,S,T)	variable	MWh	BALMOREL.GMS	67
VOBJ	-	variable	MMoney	BALMOREL.GMS	66
VQEEQ	(RRR,S,T,IPLUSMINUS)	variable	MW	BALMOREL.GMS	67
VQESTOVOLT	(AAA,S,T,IPLUSMINUS)	variable	MWh	BALMOREL.GMS	67
VQHEQURBAN	(AAA,S,T,IPLUSMINUS)	variable	MW	BALMOREL.GMS	67
VQHSTOVOLT	(AAA,S,T,IPLUSMINUS)	variable	MWh	BALMOREL.GMS	67
VQHYRSSEQ	(AAA,S)	variable	MW	BALMOREL.GMS	67
VX3VIM_T	(RRR,X3VPLACE0,...				
	...0,X3VSTEP0,S,T)	variable	MW	BALMOREL.GMS	97
VX3VEX_T	(RRR,X3VPLACE0,...				
	...,X3VSTEP0,S,T)	variable	MW	BALMOREL.GMS	97
VXKN	(IRRRE,IRRRI)	variable	MW	BALMOREL.GMS	66
VX_T	(IRRRE,IRRRI,S,T)	variable	MW	BALMOREL.GMS	66
WEIGHT_S	(SSS)	parameter	(none)	VAR.INC	45
WEIGHT_T	(TTT)	parameter	(none)	VAR.INC	45
WND_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	VAR.INC	55
WNDFLH	(AAA)	parameter	hours	GEOGR.INC	48
WTRRSVARS	(AAA,SSS)	parameter	(none∼MW)	VAR.INC	53
WTRRRVAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	VAR.INC	55
WTRRRFLH	(AAA)	parameter	hours	GEOGR.INC	48
WTRRSFLH	(AAA)	parameter	hours	GEOGR.INC	48
X3FX	(YYY,RRR)	parameter	MWh	X3FX.INC	49
X3FX_VAR_T	(RRR,SSS,TTT)	parameter	(none∼MW)	VAR.INC	56
X3VPEX	(YYY,RRR,X3VPLACE0,...				
	...,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	X3.INC	97
X3VPIM	(YYY,RRR,X3VPLACE0,...				
	...,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	X3.INC	97
X3VPLACE0	-	set	-	SETS.INC	96
X3VQEX	(RRR,X3VPLACE0,...				
	...,X3VSTEP0,SSS,TTT)	parameter	MW	X3.INC	97
X3VQIM	(RRR,X3VPLACE0,...				
	...,X3VSTEP0,SSS,TTT)	parameter	MW	X3.INC	97
X3VSTEP0	-	set	-	SETS.INC	96
X3VSTEP	(X3VSTEP0)	set	-	SETS.INC	96
X3VX	(RRR,X3VPLACE0)	set	-	SETS.INC	96
XCOST	(IRRRE,IRRRI)	parameter	Money/MWh	TRANS.INC	52
XINVCOST	(IRRRE,IRRRI)	parameter	Money/MWh	TRANS.INC	52
XKINI	(IRRRE,IRRRI)	parameter	MW	TRANS.INC	51
XLOSS	(IRRRE,IRRRI)	parameter	(none)	TRANS.INC	52
Y	(YYY)	set	-	SETS.INC	28
YVALUE	(YYY)	parameter	(none)	SETS.INC	45
YYY	-	set	-	SETS.INC	28

Index

∼, 44
' (quotes), 13
(none), 44
****, 82
*, comment, 16
*, multiplication, 15
*, set intersection, 38
*, with labels, 13
+, ++, next, 30
+, addition, 15
+, set union, 38
+, table continuation, 16
-, set difference, 38
-, subtraction, 15
. (dot), 27
.., delimiter with equations, 69
/**/, comment, 16
/, division, 15
/, slash, delimiter, 13
/, slash, in path names, 12
=, 15, 16
=E=, 15, 70
=G=, 15, 70
=L=, 15, 70
$USE225, 12, 86
$if, $ifi, 17
1995,...,2030, 28
3.6, 44
365-366-365.24, 28, 44, 45, 61
8760-8784, 28, 44, 45

24, 44, 61

AAA, 23, 26
AAARURH, 26
AAAURBH, 26
ACRONYM, 14
Acronym, 32, 35
ADDINVEST, 92
addon, 86
AGKN, 36
ALIAS, 13, 39
ALL, 70
AND, 15
ANNUITYC, 46
ascii, 19
assign, 13, 14
assignment, 13, 15, 16
assignment, sets, 38
asterisk notation, 13
attribute, 67, 77
auxiliary, 22, 59, 65, 85

back pressure unit, 39
BALBASE1, 83
balbase1, 20, 22, 83, 95
balbase2, 20, 95
Balbase3, 34
BALMOREL.GMS, 20
base load, 70
bb1, 91
bb2, 91
bb3, 91
BSR, Baltic Sea Region, 8
BUI, 9, 19, 101

C, 23, 25
CAES, 92
CAL, 18
calibration, 70
capacity (production), 41, 56
capacity (transmission), 51
CARD, 30
cartesian product, 13, 14
case sensitive, 14
CCC, 23, 25
CCCRRRAAA, 23
CES, 58
chp, 56
chronology, 28, 30, 46
city, 26
cmp (file extension), 18
co-generation, 56
COMB, 18
combination technology, 93
COMBTECH, 91
comment out, 16, 22
comments, 16
commissioning, 95
compare facility, 79
compiler directive, 77, 101
complement (set), 38
compressed air, 33
condensing unit, 39
conditional compilation, 17
conditional expression, 15
constraint, 68
control variable, 17
country, 25
CREATEDTIME, 92
CREATETIME, 92
CYCLESINS, 34, 45

data, 100
data base, 8
data entry, 16
data structure, 11
Data-pre-inc subdirectory, 21

day, 28
DAYTYPE, 30
DE, 50
DE.INC, 20
DE_VAR_T, 55
declaration, 13, 16
DECOMP?, 92
DECOMPEFF, 92
DEF, 37
DEF_D1, 37
DEF_D2, 37
DEF_D3, 37
DEF_STEPS, 57
DEF_U1, 37
DEF_U2, 37
default, 16, 44, 49
default (user specified), 44, 58
definition, 13, 14
DEFP_BASE, 47
DEFP_CALIB, 56
demand, 36
DF_PRICE, 36
DF_QP, 36
DF_QUANT, 36
DFP_CALIB.INC, 76
DH, 50
DH.INC, 20
DH_VAR_T, 54
DHF, 37
DHF_D1, 37
DHF_D2, 37
DHF_D3, 37
DHF_STEPS, 58
DHF_U1, 37
DHF_U2, 37
DHF_U3, 37
DHFP_BASE, 47
DHFP_CALIB, 55
difference (set), 38
dimension (sets), 13
DIS, 18
DISCOST0E, 48
DISCOST0H, 48
DISCOST1E, 48
DISCOST1H, 48
DISCOST_E, 47
DISCOST_H, 47
DISLOSS0E, 48
DISLOSS0H, 48
DISLOSS1E, 48
DISLOSS1H, 48
DISLOSS_E, 47
DISLOSS_H, 47
dispatch, 26, 41, 93
DISPLAY, 13, 77
Documentation subdirectory, 23
DOLLAR, 77
dollar include, 16
dollar on the left, 59
dollar on the right, 59
dollar operator, 15
dot (.), 27
duration, 46
dynamic set, 13, 38
dynamics, 100

E, =E=, 15, 70
economic dispatch, 26, 93
elastic demand, 17, 36
electric heating, 39
electricity storage, 11, 39
element in set, 13, 23, 43
elevated water, 33
ELSEIF, 15
emission policy, 38
EMPTY, 81
endogenous, 13, 14, 27, 66
entry order, 13, 30
EPS, 15, 16, 44, 53, 59
EQ, equal to, 15
EQUATION, 13, 15, 68
ERROR.INC, 83
errors, 83
ERRORS.OUT, 83
Euro, 44
exceptions, 15
execute_unload, 78
EXECUTEUNLOADINPUTDATA, 92
exogenous, 13, 27, 43
extended arithmetic, 15
extraction unit, 39

FDATA, 48
FDATASET, 35
FDCO2, 35
FDN2O, 11, 35
FDNB, 35
FDSO2, 35
FFF, 35
file name, 20
files, 19
FKPOTA, 53
FKPOTC, 53
FKPOTR, 53
FKPOTSETA, 35
FKPOTSETC, 35
FKPOTSETR, 35
flexible demand, 17
FLH, 18
floating point, 16
FOR, 12, 15
FOR-DOWNTO, 12, 15
FOR-TO, 12, 15
FOSSILTARGET, 92

FREE, 14, 67
FREE variable, 67
fuel, 35, 93
FUEL.INC, 20
FUELP.INC, 20
FUELPRICE, 57
full load hours, 18
FX, 14, 18, 67, 77, 100

G, 31
G, =G=, 15, 70
GAMS, 12
GAMS homepage, 12
GAMS User's Guide, 12
gams2prm, 82
gams2txt, 82
GAS, 91
GBPR, 32
GCND, 32
GCOMB, 51
GDATA, 50
GDATASET, 31
GDAUXIL, 31, 50
GDCB, 31, 50
GDCH4, 11, 31
GDCOMB, 31, 93
GDCOMBSK, 31, 51, 93
GDCOMBSLO, 31, 51, 93
GDCOMBSUP, 31, 51, 94
GDCV, 31, 50
GDESO2, 31, 50
GDFE, 31, 50
GDFROMYEAR, 31
GDFROMYEAR , 51
GDFUEL, 31, 50
GDINVCOST0, 31, 51
GDKVARIABL, 31, 51
GDLIFETIME, 11, 31
GDNOX, 31, 50
GDOMFCOST0, 31, 51
GDOMVCOST0, 31, 51
GDSTOHLOAD, 31, 51
GDSTOHUNLD, 31, 51
GDTYPE, 31, 50
GDX, 78
GDXDIFF, 78
GE, greater or equal, 15
GEFF_CAL1.INC, 75
GEFF_CAL2.INC, 75
GEFFDERATE, 49
generation technology, 31
generic term, 44
GEOGR.INC, 20
geography, 23
GESTO, 32
GETOH, 32
GEXT, 32
GGCOMB, 93
GGG, 31
GHOB, 32
GHSTO, 32
GHYRR, 32
GHYRS, 32
Gigajoule, 44
GINVCOST, 49
GJ, 44
GKDERATE, 56
GKFX, 56
GKFX.INC, 20
gms (file extension), 18
GMT, Greenwich Mean Time, 29
GOMFCOST, 49
GOMVCOST, 49
GSOLE, 32
GSOLH, 32
GT, greater than, 15
GTREE, 78
GUI, 101
GWND, 32

H2, 92
heat only, 39
heat storage, 11, 39
heatpump, 39
HEATTRANS, 92
HOLDFIXED, 70
homepage, GAMS, 12
homepage, Balmorel, 8
HY, 18
hydro power, 31, 39
hydro power storage, 11
hydro run-of-river, 39
hydrogen, 33, 35
HYPPROFILS, 54

IA, 38
IAGK_Y, 63
IAGKN, 41
IANYSET, 60
IARURH, 38
IAURBH, 38
IBALVERSN, 59
ICA, 39
IDAYSIN_S, 61
IDE_SUMST, 61
IDE_T_Y, 64
IDEFP_T, 65
identifier, 13, 14, 70, 100
IDH_SUMST, 61
IDH_T_Y, 65
IDHFP_T, 65
IF-ELSE, 15
IFUELP_Y, 63
IG2LEVEL, 40

IGBPR, 39
IGCND, 39
IGCOMB1, 40, 94
IGCOMB2, 40, 94
IGDISPATCH, 41, 93
IGE, 41
IGEE, 41
IGEENOSTO, 41
IGEH, 41
IGEOREH, 41
IGESTO, 40
IGETOH, 39
IGEXT, 39
IGGALIAS, 41
IGH, 41
IGHH, 41
IGHHNOSTO, 41
IGHOB, 39
IGHORHERUR, 41
IGHSTO, 40
IGHYRR, 40
IGHYRS, 40
IGKE, 41, 56
IGKENOSTO, 41
IGKFIND, 41
IGKFX_Y, 63
IGKH, 41, 56
IGKHNOSTO, 41
IGKKNOWN, 41
IGKVACCTOY, 63
IGNOTETOH, 41
IGNUC, 40
IGSOLE, 40
IGSOLH, 40
IGWND, 40
IHOURSIN24, 61
IHOURSINST, 60
IHYINF_S, 62
ILIM_CO2_Y, 64
ILIM_NOX_Y, 64
ILIM_SO2_Y, 64
IM_CO2, 62
IM_SO2, 62
imperfect competition, 99
inc (file extension), 18
INCLUDE, 16
include file, 16
inelastic demand, 37
INF, 15, 16
INI, 18
initbase.cmp, 81
initvar.cmp, 81
INLINE, 16
input parameter, 43
input set, 23, 38
INTEGER, 14
integer, 16
internal parameter, 43, 59
internal set, 23, 32, 38
intersection (set), 38
INV, 18
investment, 36
IOF, 60
IOF1000, 60
IOF24, 60
IOF365, 60
IOF3P6, 60
IOF8760, 60
IOF8784, 60
IPLUSMINUS, 43, 67
IR, 39
IRE, 39
IRI, 39
IRRRE, 39
IRRRI, 39
ISALIAS, 39
ISCALAR1, 60
ISOLESUMST, 62
ISOLHSUMST, 62
IST, 11, 39
ISTS, 11, 39
ISTT, 11, 39
ITALIAS, 39
ITAX_CO2_Y, 63
ITAX_NOX_Y, 64
ITAX_SO2_Y, 64
ITERLIM, 70
IWEIGHSUMS, 60
IWEIGHSUMT, 60
IWND_SUMST, 61
IWTRRRSUM, 62
IWTRRSSUM, 62
IX3FX_T_Y, 65
IX3FXSUMST, 62
IX3VPEX_Y, 97
IX3VPIM_Y, 97
IXKINI_Y, 63
IXKN, 63

JI, joint implementation, 99
joint implementation, JI, 99

k, kg, kilogram, 44
key words, 14
KILL, 81
kilogram, 44

L, 77
L, =L=, 15, 70
L, level, 14, 77
label, 14, 23, 43, 100
LE, less or equal, 15
level, 77
LIM, 18

LIM_CO2, 38
LIM_NOx, 38
LIM_SO2, 38
LIMCOL, 77
LIMROW, 77
linear programming, 14, 70
Linux, 8, 12, 20
list, 44
LISTING, 77
LO, 14, 67, 77, 100
load duration, 46
LOG file, 21, 77
LogError subdirectory, 22
logerror subdirectory, 19
LOGFILE.OUT, 83
logical expression, 15
LOOP, 13, 15
loss, el. distribution, 47
loss, el. transmission, 52
loss, heat distribution, 47
lower case letter, 12
LP, 70
LP, linear programming, 14
LST, 77
LST file, 21, 77
LT, less than, 15

m (milli), 44
M (million), 44
M, MARGINAL, 14, 77
M_POL, 57
MAKEACCESS, 92
MAKEINVEST, 92
MAKEWATER, 92
mapping, 26
marginal unit, 76
marginal value, 77
market power, 99
MAX, 15
MAXIMIZING, 70
med (file extension), 18
medium load, 70
megawatt, 44
member of set, 13, 23, 43
merit order, 70
milli, 44
MIN, 15
MINIMIZING, 70
MODEL, 13, 70
model structure, 8, 100
Model subdirectory, 20
model subdirectory, 19
modelled, 11
Money, 44
MOVERESULTS, 92
MP, 91
MPOL.INC, 20
MPOLSET, 38
mss, 18, 22, 83
MULTI, 81
multi fuel, 93
MW, 44
MWh, 44

NA, 15, 16
naming, 17, 28, 31, 35
NAP, 92
NE, not equal to, 15
net present value, 46
new generation tech., 36
NO, 38
NOELTRANSINV, 92
nominal, 24, 36, 37, 50, 64, 65, 70, 98
normal completion, 83
NOT, 15
NOT, set complement, 38
NPV, 46
numeric, 16

OBJECTIVE VALUE, 77
obligatory set member, 43
obligatory value, 51, 59
OCASEID, 66
OM, 18
OMONEY, 66
ONEMPTY, 37, 95
opt (file extension), 18
OPTION, 70, 77
OR, 15
ORD, 30
ordered set, 13, 30
out (file extension), 18
outage, 56, 72
output, 65, 77
Output subdirectory, 23
overwrite, 16

PARAMETER, 13, 16
parameter, 43, 59
path name, 20
PC-Windows, 8, 12
PENALTYQ, 44, 59, 67
perfect competition, 99
POL, 18
POSITIVE, 14
POSITIVE variable, 67
price setting, 76
Print-inc subdirectory, 21
PRINT.INC, 78
PRINTFILES, 91
printinc subdirectory, 19
printout subdirectory, 19
PRIOR, 77
PROD, 15

QECOMBGKL, 69, 94
QECOMBSLO, 69, 94
QECOMBSUP, 69, 94
QEEQ, 68
QESTOLOADT, 68
QESTOVOLT, 68
QGCBGBPR, 68
QGCBGEXT, 68
QGCVGEXT, 68
QGE2LEVEL, 69
QGEKNT, 68
QGGETOH, 68
QGHKNT, 68
QGKNHYRR, 68
QGKNSOLE, 68
QGKNWND, 68
QGNCBGBPR, 68
QGNCBGEXT, 68
QGNCVGEXT, 68
QGNGETOH, 68
QHCOMBGKL, 69, 94
QHCOMBSLO, 69, 94
QHCOMBSUP, 69, 94
QHEQ, 68
QHNRURPROP, 68
QHSTOLOADT, 69
QHSTOVOLT, 68
QHYMAXRS, 68
QHYMING, 68
QHYMINRS, 68
QHYRSSEQ, 54, 68
QKFUELA, 69
QKFUELC, 69
QKFUELR, 69
QLIMCO2, 69
QLIMNOX, 69
QLIMSO2, 69
QNECOMBGKL, 69, 94
QNECOMBSLO, 69, 94
QNECOMBSUP, 69, 94
QNHCOMBGKL, 69, 94
QNHCOMBSLO, 69, 94
QNHCOMBSUP, 69, 94
QOBJ, 68
QUAD, 91
quickfix, 92
quotes, 13, 16
QXK, 69

real number, 16
region, 26
RENEWTARGET, 92
reserved words, 14
reservoir, 30, 39, 46
RESLIM, 70
RRR, 23, 26
RRRAAA, 26
run-of-river, 39
rural area, 26
RX3VSUBSTI, 97

S, 28
savebase.cmp, 81
savevar.cmp, 81
SCALAR, 13, 16
scalar, 43
SCALE, 77
scaling, 16, 77
scenario, 79
scope, 84
season, 28
segment, 28
sequence of statements, 15, 85
SET, 13, 38
set (control variable), 17
set intersection, 38
set operations, 38
set product, 14
set, (un)ordered, 13, 30
set, dynamic, 13, 38
set, static, 13, 30, 38
setglobal, 17
setlocal, 17
SETS.INC, 20
sim (file extension), 18, 20
simulated, 11
slash, 12
SMAX, 15, 98
SMIN, 15
solar power, 39
SOLE, 18
SOLE_VAR_T, 54
SOLEFLH, 48
SOLH_VAR_T, 54
SOLHFLH, 48
SOLPRINT, 77
SOLVE, 70
solver, 17
spreadsheet, 8, 23, 78
SSS, 28
static set, 13, 30, 38
STATUS, 77
stochastic outage, 72
storage, 30, 32, 39, 46
storage, short term, 32
structure, 8
sub period, 28
subdirectory, 19
subset, 13
suffix, 18
SUM, 15
SYMLIST, 77, 101
SYMXREF, 77, 101
SYSOUT, 77

T, 28
t (ton), 44
TABLE, 13, 16
tabulator, 12
TAX_CO2, 38
TAX_DE, 47
TAX_DH, 47
TAX_F, 48
TAX_NOx, 38
TAX_SO2, 38
TECH.INC, 20
technology, 31
TEP, tradable emission permits, 99
TEXT, 16
TGC, tradable green certificats, 99
third country, 49, 56, 62, 65, 75, 76, 95–97
third region, 95
time, 28
time period, 28
time segment, 28
time zones, 29
timeaggr, 92
ton, 44
tradable emission permits, TEP, 99
tradable green certificats, TGC, 99
TRANS.INC, 20
transport, 92
trnsbase.cmp, 81
trnsvar.cmp, 81
TTT, 28
TWEEKEND, 30
TWORKDAY, 30

UELLIST, 77, 101
UELXREF, 77, 101
UNDF, 15, 16
union (set), 38
unit (data), 44
Unix, 8, 12, 20
unordered set, 13, 30
UP, 14, 67, 77, 100
urban area, 26
USD, 44
USE225, 12
user interface, 19
User's Guide, 12
USING, 70

VAR, 18
VAR.INC, 20
VARIABLE, 13, 14, 66
variant, 79
VDEF_T, 66
VDHF_T, 66
version, 100
version number, 8
VESTOLOADT, 66
VESTOVOLT, 66
VGE2LEVEL, 67
VGE_T, 66
VGEN_T, 66
VGH_T, 66
VGHN_T, 66
VGHYPMS_T, 66
VGKN, 66
VHSTOLOADT, 66
VHSTOVOLT, 67
VHYRS_S, 66
VOBJ, 66
VQ, 44, 67
VQEEQ, 67
VQESTOVOLT, 67
VQHEQURBAN, 67
VQHSTOVOLT, 67
VQHYRSSEQ, 67
VX3VEX_T, 97
VX3VIM_T, 97
VX_T, 66
VXKN, 66

water, 31
WATERVAL, 92
WEIGHT_S, 45
WEIGHT_T, 45
WELFARE, 92
WHILE, 12, 15
wind power, 39
Windows, 8, 12
WINDTARGET, 92
within the season, 28
within the year, 28
WND, 18
WND_VAR_T, 55
WNDFLH, 48
WNDSHUTDOWN, 92
WTR, 18
WTRRRFLH, 48
WTRRRVAR_T, 55
WTRRSFLH, 48
WTRRSVARS, 53

X3.INC, 20
X3FX, 49
X3FX_VAR_T, 56
X3V, 91
X3VPEX, 97
X3VPIM, 97
X3VPLACE0, 96
X3VQEX, 97
X3VQIM, 97
X3VSTEP, 96
X3VSTEP0, 96
X3VX, 96

XCOST, 52
XCOST1, 52
XINVCOST, 52
XKINI, 51
XLOSS, 52
XLOSS1, 52
XOR, 15

Y, 28
year, 28
YES, 38
YVALUE, 45
YYY, 28