Hans Ravn

The Balmorel Model Structure

PRELIMINARY - DO NOT QUOTE OR FORWARD

Version 3.03 (June 2018)

(This document revised 2019.03.11)

PRELIMINARY - DO NOT QUOTE OR FORWARD

;

Contents

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 The original ambitions and limitations have since long been superseded, to that the model is no longer tied to the original focus geography and issues.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 1.4. 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)

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

1.1 Terms of use

The model's source code, written i GAMS, is available under open source conditions. It may be downloaded freely from the Balmorel home page.

Efforts have been made to make a good model. However, most probably the model is incomplete and subject to errors. It is distributed with the idea that it will be usefull anyway, and with the purpose of getting the essential feedback, which in turn will permit the development of improved versions to the benefit of other user. Hopefully it will be applied in that spirit.

More formally, all GAMS code of the Balmorel model is distributed under ICS license.

The license file is located in the base\model folder. The text of the license goes as follows:

AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE."

1.2 This version

The description given here is for version 3.03 (June 2018) (303.20160623) 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 in detail, however you may have a view at the files in the model's base/documentation folder, or contact us for further information, cf. the Balmorel homepage.

The changes from the previous version, 3.02, 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.

Notable feature are addons dealing with time aggregation (addon timeagr), unit commitment (addon UnitComm) and rolling horizon (addon BB4).

The present version 3.03 (303.20160623) is mainly intended for information about changes and for review. Please contact us for any comments or questions.

A new version is expected to be available in August this year, based on comments received.

The model is implemented in GAMS, originally Version 2.25, but some features require a newer version, see further Section 1.4.

The model has been developed on PC/Windows, see further Section 1.4.

1.3 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 13.5 for some specifications (and Sections 3.9 and 4.26 for exceptions).

1.4 A short introduction to GAMS version and syntax

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. The users' guides GAMS User's Guide and McCarl GAMS User Guide, a Tutorial, and other relevant information about the GAMS modeling system may be found in your GAMS system or 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.

1.4.1 GAMS version

The Balmorel model was originally implemented in version 2.25. This version has limitations 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.

When you have executed a GAMS model (like Balmorel) there is at the top of the list file (Section 11.1) text indicating the GAMS version used.

The main limitations following from restriction to version 2.25 relevant for the present description are as observed by us (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 4.27.3 how this is circumvented), (iv) do not use tabulator. These limitations are eliminated in more recent versions.

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.)

These short comments on GAMS version gives a historical view of some of the naming conventions (see later) in Balmorel. Some additions to the Balmorel model do not conform to the 2.25 version of GAMS.

With GAMS 22.7 of May 1, 2008 enhancements that are applied in some parts of the present Balmorel version include that a symbol can have up to 20 dimensions and identifiers and labels can have up to 63 characters.

1.4.2 GAMS syntax

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 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 four countries, this may be specified as

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

where it is seen that slashes (/) are used as delimiters of the definition. Descriptive text may be entered, e.g. as SET CCC "All countries".

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., if only the countries from CCC that have hydro power are interesting, they may be specified as a subset C of CCC,

SET C(CCC) "Hydro 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 /S01*S52/ may be used to indicate the labels S01, S02, ..., 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 3.2.3. Section 11.1 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 (CCC,AllCCC) declares the set CCC and defines it to be identical to the set AllCCC.

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 3.2.3.

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 GAMS' default value, which is zero (observe that the handling of such default zero is for some data items special in Balmorel, cf. EPSconv on page 47).

Acronyms

An acronym is a 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 GAS, COAL, LIGHTOILNOSULPHUR;".

It is possible to assign acronyms to parameters and scalars, like e.g. GDATA('HEATBOILER7','GDTYPE')= GHOB. However, acronyms, which are character string values, can be used in logical conditions only with the 'EQ', '=', 'NE' and '<>' operators. For a parameter element with a value which is an acronym this element cannot participate in arithmetic operations like additions, multiplications, etc.

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 1.5).

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. 1.4.1.)

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, but yet different. For one use in input, see Section 4.28; 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 GAMS User's Guide or McCarl GAMS User 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 3.2.3) 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 of definition. The extended arithmetic symbols INF, NA and EPS (but not UNDF) may be used in input. See e.g. Section 4.28 on the use of EPS. Also see below and Sections 4.25, 4.28 on default data.

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 applied. (But note the special mean of EPS, Section 4.28.)

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

See also Section 4.25.

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 GAMS User's Guide or McCarl GAMS User Guide.

Explicative text with identifiers

It is possible to associate explicative text with identifiers sets, parameters etc., and also with set element. 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 11.

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.

Multi-line comments can be entered between "$ONTEXT" and "$OFFTEXT" (where the $'s must be in the first position of a line).

Comments (inline comments) may also be inserted between "/*" and "*/" (provided it is preceded by "$ONINLINE". Adding also "$ONEOLCOM" makes it possible to start a comment with !!, it stretches to the end-of-line.

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.

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 and some more.

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.

Equations

The constraints in the GAMS model may be on the individual variables by applying .LO, .UP and possibly .FX, cf. above and Section 7 page 75. 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').

Model

In GAMS 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 8 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 12.2.

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 7, and it is in this example to be minimised (the alternative is to specify MAXIMIZING). The problem class is here specified to be LP (linear programming). Other possibilities are e.g. QP (quadratic programming), MIP (mixed integer programming) (which also covers what may in other contexts be called MILP, Mixed Integer Linear Programming), MIQCP (mixed integer quadratic programming) and RMIP (relaxed mixed integer 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 GAMS User's Guide or McCarl GAMS User Guide.

See Section 9 page 79.

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. This division of work between GAMS and the solver illustrates the separation between modeling (which is done in GAMS) and solving (which is done in any suitable solver).

Output, Errors etc.

See Section 11 page 86.

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. See further Section 13.2 page 92.

And more . . .

Many more syntax elements are available and in use in Balmorel, the above serves as an introduction only. See GAMS User's Guide or McCarl GAMS User Guide for more.

1.4.3 Operating system

The model is developed on PC/Windows. GAMS will also run under Unix and Linux and a number of other operating systems. In order to facilitate Unix and Linux applications of the Balmorel model, awareness of the implications for the GAMS code is important. The following illustrates how differences between operating systems may be accounted for.

Apart from consistency in naming, including user of upper/lower case for enhancing readability there is the aspect of the operating system. Balmorel was originally developed on Windows computers. However, with an increasing number of users and use of other operating systems (e.g.. UNIX, Linux) it is necessary to be strict on those issues in the Balmorel code that are operating system dependent. This includes use of upper/lower case letters, use of forwards/backwards slashes and use of commands that involve operating systems. To develop and maintain Balmorel compatibility between Windows NT and UNIX-like systems, at least the points in the following need consistency.

- Usage of lower and upper case letters in folder and file names
- Commands used to move, delete, rename or merge files
- Commands used to make or delete folders
- Folder and file access permissions (for UNIX type systems)
- Usage of forward and backward slash in file and folder paths

The two classes of operating systems, MSNT (Windows) and UNIX (including Linux) are by the operating system in use identified by GAMS and held in the string %system.filesys%.

Here follows some guidance.

Files and folder naming convention:

- All folders are in lower case

- All files are in lower case, except
  - Balmorel.gms (and temporarily Balmorelbb4.inc)
  - All data files are in upper case with extension in lower case, e.g. SSS.inc
  - File LICENSE.txt

Code:

- All items within the GAMS code that relate to folders and/or files have the same upper/lower case convention as the folder and/or file in case.
  E.g., $include '../../base/model/balopt.opt';
  E.g., $include '../../base/data/GKFX.inc';

Usage of forward and backward slash:

- For most cases, the forward slash / can be used for paths, instead of the backslash \
  e.g. $include '../../base/addons/_hooks/setdefine.inc'
  NOT: $include '..\ base\addons\ _hooks\setdefine.inc'
  Note: Sometimes use of quotes may be expedient. E.g. you may declare a file by using
  file testfile / ..\output\testfile.txt /;
  However, this does not conform to the conventions to be applied, you may try this
  file testfile / ../output/testfile.txt /;
  But this mixes up the '/' delimiting the file name specification and the '/' used as separator in file paths, therefore use
  file testfile / "../output/testfile.txt" /;
- Some commands in the MSNT system require the path to be defined by using the backslash, particularly when using the option to manipulate all files of certain type in a folder:
  Using GDXMERGE
  Using commands which are sent to the Windows system for execution: i.e 'del'
  The relative model paths should be defined according to the operation system (forward slash for UNIX, backslash for MSNT)

Commands used to move, delete or rename files:

- To make a directory, use 'mkdir', not 'md'
  E.g. $ifi not dexist "../../simex" execute 'mkdir "../../simex"';
- To delete a file, use 'rm' for UNIX and 'del' for Windows (called MSNT in GAMS):
  $ifi %system.filesys%==UNIX
  $ifi %SAVEPOINTVALUE%==1 execute "rm %relpathoutput%temp/*.gdx";
  $ifi %system.filesys%==MSNT
  $ifi %SAVEPOINTVALUE%==1 execute "del %relpathoutput%temp/*.gdx";
- When system script files need to be used, the structure is different for MSNT and UNIX systems
  $ifi %system.filesys%==MSNT
  $ifi %bb1%==yes putclose fileMERGESAVEPOINTRESULTSbat 'move ' '"%relpathModel%"BALBASE1_p.gdx" ' '"%relpathoutput%temp/'Y.tl:0:0 '.gdx"';
  $ifi %bb1%==yes execute "%batfileMERGESAVEPOINTRESULTS%";
  $ifi %system.filesys%==UNIX
  $ifi %bb1%==yes putclose fileMERGESAVEPOINTRESULTSbash '♯!/bin/bash' / 'mv "%relpathModel%"BALBASE1_p.gdx" "%relpathoutput%temp/'Y.tl:0:0 '.gdx"';
  $ifi %bb1%==yes execute 'chmod +x ./"%bashfileMERGESAVEPOINTRESULTS%"';
  $ifi %bb1%==yes execute './"%bashfileMERGESAVEPOINTRESULTS%"';

Folder and file access permissions (for UNIX systems):

- Because of the difference in permission systems of UNIX type systems compared to MSNT systems, there is a need to provide permissions to access certain files and folder. This is generally achieved by the command in the beginning of the model:

  $ifi %system.filesys%==UNIX execute 'chmod -R ug+rw "../.."';

- However, if there is a need to use special system script files, the execution permission is needed to give execution rights for that specific script, for example:

  $ifi %bb2%==yes execute 'chmod +x ./"%bashfileMERGESAVEPOINTRESULTS%"'

1.5 Naming conventions

We have tried to select identifier names for the various sets, parameters etc. to facilitate the recognition of the meaning from the name. Observe that most names are limited to ten characters which was the limit when Balmorel was originally developed. This explains some perhaps unfortunately short identifier names, however, most names remain unchanged to facilitate recognition and backwards compatibility. Some newer additions do not adhere to this limitation. 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

K: capacity

M: emission

N: new

O: related to output from simulations

X: electricity transmission

Prefixes:

I: internal (set, scalar or parameter)

V: variable (see also VQ)

Q: equation

Suffixes:

_Y or Y: the Year, annual

_S or S: the first division of the Year is into Seasons (e.g. summer, winter; or Jan, ... , Dec; or Week1, ..., Week52; or Day1, ..., Day365) (and implicitly or explicitly also contains year index), cf. Section 3.2.2

_T or T: the division of Seasons is into Terms (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

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 (GHYRR: run-of-river, GHYRS: with storage (reservoir))

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

SOLH: solar, producing heat

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

WAVE: wave, producing 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
gdx: GAMS data exchange 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 11.3 concerning output files.

Index sequence

The sequence of indexes in declarations 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.

Note that in GAMS later indexes run faster. For efficiency reasons this should be taken into account in loops, sums, and other indexed operations. The above harmonization supports such efficient execution.

1.6 User interfaces

The GAMS program is basically 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. Possibilities other than ascii files exist, see further Section 11.2 page 87.

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 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 more Balmorel specific user interface BUI (Balmorel User Interface) is under development, see further Section 14.

2 File Structure

The model is distributed over a number of files. In this section we give an overview.

The files are ascii files, cf. Section 1.6.

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

- addons
- auxils
- bui See further Section 14 page 118.
- data
- documentation
- model
- logerror
- output

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 e.g. Unix and Linux. Cf. Section 1.2.

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.

In order to handle a number of cases (differing with respect to e.g. fuel prices but otherwise identical) and to compare the results between them, the file structure described page 24 is used.

Subdirectory addons

Addons in the Balmorel model are pieces of code that provide extensions to the functionality for the core model version. By default and addon's code are is included as part of the model being executed. They may be included when found relevant. See further Section 13.2.

Subdirectory auxils

Auxils are meant to support the Balmorel in some way, e.g. by pre-processing input data or post-processing output data. In contrast to addons, auxil code is not executed as part of a model run but executed independently.

Subdirectory model

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

- Balmorel.gms is the main model file (running this means running the model)
- bb123.sim contains i.a. the SOLVE statements
- balopt.opt contains options related to overall characteristics of Balmorel functionality, some of such code is found in files in folder addons
- balgams.opt contains options related to details of GAMS
- balcomline.opt contains command line options related to another group of details of GAMS

These files together contain the model. The Balmorel.gms file is the main file, while the others are include files (Section 1.4) which during compilation of the model automatically are inserted into the appropriate places in the Balmorel.gms file or elsewhere. The file bb123.sim holds code that contain the SOLVE statements, typically situated within loops over time (years Y and/or seasons S).

All user specified labels and numerical data are found in the include files in the data folder. The distribution of the different input data (sets, parameters etc.) over the files is done according to the idea that each identifier is contained in a separate file, having the same name as the identifier, and having the extension .inc (e.g. SET CCC is contained in file ccc.inc). All files are to be found in the data folder.

See Section 15 page 118 for an overview.

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 11 page 86 for more on this.

Subdirectory output

This directory has i.a. a subdirectory printinc holding 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 6 page 75 and Section 11 page 86 .

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 11 page 86.

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 11 page 86 and Section 12 page 86.

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 11 page 86.

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.

Project and Cases

Balmorel works with Projects. A Project is as collection of folders and files that together define one or more versions of Balmorel with data.

The Project folder has a name, e.g. Balmorel303MyDownload. The Project has Cases, e.g. base, Highfuelprice and Tester1. The base must always be present. You may think of a Case as a scenario, as a set of consistent input data for a specific model code setup, or similar; there are various terminology terms around, in Balmorel the term is Case.

The folder structures are similar between the Cases, however, the base Case is special by having more folders. The reason is that in the base Case there is Gams code for the central Balmorel model, and the additional code pieces for error checking, printing and more.

Make sure that the Project is located on the computer at a place where you have read and write permission.

Folder structure for a Case
Any case has at least the following subfolders:

- data
- model
- logerror
- output (the output subfolder has additional subfolders)

Folder content
The subfolders in the base Case may hold code, data and/or output. In particular, the base Case represents a full Balmorel model with all required folders, code, data and, after execution, output.

Most non-base subfolders should hold no code. The exception is the model folder which should hold the three files balopt.opt, balgams.opt and balcomline.opt. Note in particular that a non-base Case should not hold the Balmorel.gms file. A non-base Case should hold only the data files that differ in content from those of the base Case. The output from execution will be placed in appropriate folders of the non-base Case.

How it works
It is assumed that GAMS is run through the GAMS IDE and that a GAMS Project is created with the project file (extension: gpr) located in the base/model folder.

Base Case
Running the base case implies that the option files balopt.opt, balgams.opt and balcomline.opt are read, and dependent on chosen options the relevant code and data files will be included. All data files are located in the base/data folder. Output files go to subfolders of the base Case.

Non-base Case
A non-base Case is more involved. Consider for specificity a Case called case3 (this is then also the name of the Case folder). Then following happens:

- The option files balopt.opt, balgams.opt and balcomline.opt are read from case3/model. As minimum the balopt.opt should differ from that of the base Case by having option caseid set as $setglobal CASEID case3
- The data files in the case3/data folder are read. If the needed data files are not in that folder, they will be read form the base/data folder. This ensures that all input data are identical between base and case3, except for the files available in the case3/data folder
- Output will go to appropriate folders of case3
- However, special care is needed to ensure that the GAMS output files Balmorel.lst and Balmorel.log are sent to folder case3/model, see next.

When running GAMS from the IDE the 'home' folder will be the one that holds the project file (extension: gpr), with Balmorel it is the base/model folder. By default, the GAMS output files Balmorel.lst and Balmorel.log are sent to the home folder. All the output files generated by the Balmorel code will go to folders that are specified relatively to the model folder. If it is attempted to open and run file case3/model/Balmorel.gms, the IDE will issue a warning, but it will run, and the output files generated by the Balmorel code will go to the appropriate subfolders under case3. However, Balmorel.lst and Balmorel.log are sent to the base/model folder. This is inconvenient for documentation, it should better go to the case3/model folder. It may be achieved by placing the following GAMS options in the command line of the IDE before running the model:
curdir ../../case3/model, workdir ../../case3/model

Why this way
Advantages of the above handling are that there is a clear distinction between the different

Cases, everything relevant and specific for a Case is in the Case folder. More specifically there are for any Case

- Documentation of the options used (found in balopt.opt, balgams.opt and balcomline.opt in the model folder of any Case)
- Documentation of the data (only files in case3/data differ from the base Case)
- Documentation of the output (Balmorel.lst, Balmorel.log plus any chosen output like error file, printfiles, gdx files, xls files. etc.)
- The Balmorel code files exist only in the base Case hence exactly the same code is used for all Cases (except balopt.opt, balgams.opt andbalcomline.opt)
- Since the code files exist only in the base Case code maintenance is easier and safer.

3 Sets

In this section we describe the sets in the model. According to the distinctions in Section 1.3 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 individual data files, 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 the elements in 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 further on.

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.

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.

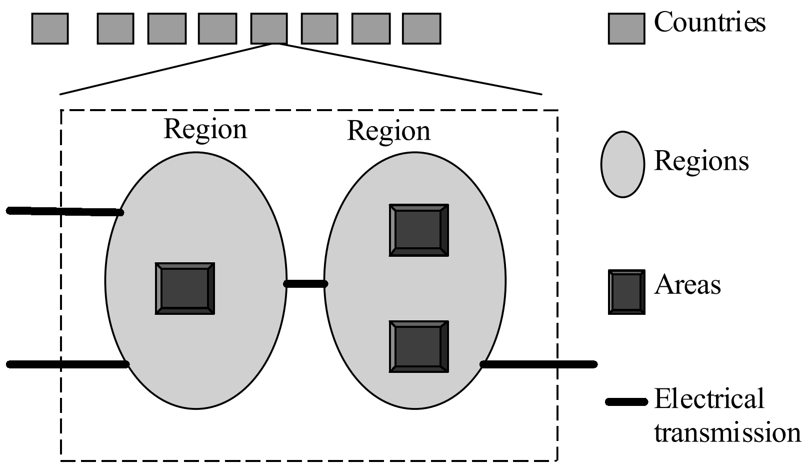

Figure 1: The geographical entities.

Here are some examples of how entities are related to geographical level. 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 All geographical entities: CCCRRRAAA

SET CCCRRRAAA contain all geographical entities, cf. CCC, RRR and AAA below.

3.1.2 Countries: C, CCC

SET CCC(CCCRRRAAA) is the subset of CCCRRRAAA that contains the countries in the data structure (cf. Section 1.3), e.g.:

SET CCC /DENMARK, FINLAND, 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 3.1.3 (and similarly with the areas not in regions in the countries in C, see Section 3.1.4). (An obvious implication of exclusion of a region is that electricity exchange with that region is not possible (therefore variables VX_T (Section 7) relative to the excluded region will not be included in the model).)

3.1.3 Regions: RRR

SET RRR(CCCRRRAAA) 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 13.

The simulated subset IR of RRR is described in Section 3.8.4 along with other variants of RRR.

3.1.4 Areas: AAA

SET AAA(CCCRRRAAA) 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 13.

The simulated subset of AAA is described in Section 3.8.2.

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 possibility to use parentheses.

The internal set ICA(C,AAA) specifies the relation between AAA and C, see Section 3.8.3.

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 13.3.8.

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 and the relations between them. Here are some examples.

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 (*Addon)
- 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)).

Since the labels 1995, 1996, etc. are text strings with no numerical values, the parameter YVALUE is used to provide the years' numerical values, cf. Section 4.2.1 page 48.

The sets YYY and Y are ordered, cf. the comments in Section 3.2.3 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 3.2.3.

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 naming conventions: For chronological specifications of time segments the naming of the individual seasons will typically start from 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 3.2.3.

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 / S01 * S04 /; or
SET SSS / S12_01 * S12_12 /; or
SET SSS / S01 * S52 /; SET SSS / S001 * S365 /;

This gives the possibilities of representing the year with 1, 4, 12, 52 or 365 (366 when needed) Seasons, respectively.

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

SET TTT / T1 /; or
SET TTT / T073 * T096 /; or
SET TTT / T001 * T168 /;

giving the possibilities to represent the subdivision of the season with 1, 24 or 168 Terms, 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) /S01/ " and "SET T(TTT) /T001/". It is also easy to use other subsets S and T. E.g., if SSS is defined as "SET /S01 * S12/" to represent the twelve months of the year, then specifying SET S(SSS) / S01, S07 / means that only January and July will be used in the simulations to represent the whole year. With e.g.

```
SET SSS / S01 * S12/;
SET TTT / T001 * T012/;
```

a simulation with all 144 time segments will be specified as

```
SET S(SSS) / S01 * S12/;
SET T(TTT) / T001 *T012/;
```

and a simulation with the four time segments (S02,T001), (S02,T005), (S10,T001), (S10,T005), per year will be specified as

```
SET S(SSS) / S02, S10/;
SET T(TTT) / T001, T005/;
```

Although simple to aggregate time this way, it is questionable if it gives a good representation of the full year. In particular note that the above easy aggregation may have a fair representation of input data with respect to annual energy content of i.a. demand and renewable energy (due to the internal scaling done in Balmorel) while other elements are worse represented. See further Section 13.3.12 page 110.

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.

Comment on input data: In relation to acquisition of data relative to time segments within the year attention has to be paid to winter and summer time (sometimes referred to as daylight saving time) conventions for the geographical area in question. Some places around the world use this, others not. The application of it will typically involve advancing the clock from 2:00 to 3:00 some day in spring implying a day with only 23 hours. In autumn the clock is some day set back from 3:00 to 2:00 implying 25 hours in that day. For the European Union countries the two days have since 2002 have been the last Sunday in March and the last Sunday in October. This influences human related activities, implying that e.g. electricity demand will not for these two days display the same time pattern as for other days. On the other hand, natural phenomena like e.g. wind speed and temperature will not be influenced (but it may look this way if evaluated against the time shown on the clock on the wall). For data related to geographical areas where it is relevant it is for the Balmorel model common to change original measurements (or whatever source is used) such that each day has 24 hours, i.e., adding one hour in spring (typically by copying the demand from the hour from 2:00 to 3:00 to the newly added hour) and deleting one hour in autumn (typically be replacing demand in the remaining hour by the average of the two hours in question). Note that the source for the data may already have done some manipulation in that direction, possibly making further adjustment unnecessary.

Comment on input data: The data for Balmorel may typically for time segments within the year be given in its most detailed form based on set SSS with 52 elements and set TTT with 168 elements. This totals 8736 time segments per year. As a non-leap year has 365 day and 8760 hours each time segment in this representation of Balmorel will then have the average length of approximately 1.0027 hours (3609.72 seconds). This value will be calculated in IHOURINST, Section 4.27.6.

	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.

Comment on input data: To harmonize time series it may also be expedient to use standards for week numbering for the case when SSS has 52 elements and TTT has 168, where each element in SSS could represent a week. The ISO 8601 stipulates week numbers 01 through 53 and the weekday number, from 1 through 7, beginning with Monday and ending with Sunday. Further, the following are mutually equivalent and compatible descriptions of week 01: the week with the year's first Thursday in it (the formal ISO definition); the week with 4 January in it; the first week with the majority (four or more) of its days in the starting year; the week starting with the Monday in the period 29 December - 4 January. The following rules for harmonizing given time series (possibly from different sources) for a given year may be suggested for use with Balmorel:

- Assure that each day has 24 hours, make correction as above if needed
- Harmonize time zones as needed
- Delete or insert days at the beginning of the time series so that the time series starts with the first hour (immediately after midnight) of week 1
- Insert or delete days at the end of the time series so that the time series has 52 full weeks
- Check: the time series now has 8736 hours
- Insert the numbering of the hours using (S01.T001) for the first hour, (S01.T002) for the second and so on until (S52,T168) for the last hour.

3.2.3 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". Similarly, "-" denotes "previous" with similar cyclical verrsion "–".

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, HYRSDATA

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 3.8.7.

The set GDATASET is the set of attributes of generation technologies:

SET GDATASET / GDTYPE, GDFUEL, GDCB, GDCV, GDFE, GDESO2, GDNOX, GDCH4, GDAUXIL, GDINVCOST0, GDOMVCOST0, GDOMFCOST0, GDFROMYEAR, GDLIFETIME, GDKVARIABL, GDSTOHLOAD, GDSTOHUNLD, GDCOMB, GDCOMBSK, GDCOMBSLO, GDCOMBSUP, GDLOADLOSS, GDSTOLOSS / ;

These elements are obligatory. Descriptions are given in Section 3.8.7 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 model 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 4.19.1) of a particular generation unit must therefore be specified with indexes reflecting this, i.e. GKFX(*,AAA,GGG), where AAA represents geography and G represents technology.

The idea behind this is that for a geographical area it is not possible to get, nor sensibly to use, precise information about all generation units. Therefore a limited number of technologies have initially been specified in the set GGG. This moreover facilitates the aggregation of existing units into fewer but larger ones.

The GAMS syntax (Section 1.4) permits an explanatory text associated with (some or all) units in GGG. This identification is a possibility only, and therefore to use it systematically a convention is needed, and one such will now be described.

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, recent versions more, 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 this text need not be used. 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,GGG) only in that area. If new investments are permitted the user must similarly make sure that the set AGKN(AAA,GGG) (Section 3.5) specifies that new capacity of the technology in question can only be established in the relevant area.

PARAMETER HYRSDATA(AAA,HYRSDATASET,SSS) holds data for hydro with storage (to be described). The set HYRSDATASET has obligatory members HYRSVOLMIN 'Minimum reservoir filling (fraction) at start of season', HYRSVOLMAX 'Maximum reservoir filling (fraction) at start of season', and HYRSGMIN 'Minimum hydro production (fraction of nominal production capacity)'.

3.3.1 Acronyms

Acronyms are used to identify technology types, defined as follows: ACRONYMS GCND, GBPR, GEXT, GHOB, GETOH, GHSTO, GESTO, GHYRS, GHYRR, GWND, GSOLE, GSOLH, GWAVE, GESTOS, GHSTOS;

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 (page 14 in Section 1.4.2).

Thus for example, the internal set IGCND(G), which holds all technologies of the extraction type (cf. Section 3.8.7) 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 42) 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 39, they are user defined.

3.3.2 Storages

There are two major kinds of storages: short term for electricity and heat, respectively, and hydro with reservoir (storage).

The hydro with reservoir 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, such that storages both serve to distribute the available hydro inflow for use over seasons (to account for imbalance between seasonal inflow and generation) and to some extent between dry, normal and wet years.

The other type of storage is of a shorter term nature, intended for handling daily or weekly cycles. There are electricity and heat storages. Note that the characterization 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 in relation to pumped storage or as compressed air in relation to a compressed air energy storage (CAES).

The two types are handled differently as will be described briefly in the sequel.

Hydro storages

Hydro storages with technology type GHYRS differ in important ways from the other type of storage, both in nature and in modeling. In particular, there is exogenous inflow to the hydro storages. Although the inflow is physically water, it is for all practical issues considered as electricity in the model.

Dynamic equation

The dynamic linking of storage volumes, loading and unloading is given in equation QHYRSSEQ(AAA,SSS), which is the main relation for hydro storages. Essentially the equation states that the reservoir level in the beginning of season $(S+1)$ equals the reservoir level in the beginning of season S plus inflow minus release of water for electricity generation.

The inflow is proportional to the product of WTRRSVARS(AAA,SSS), WTRRSFLH(AAA) and generation capacity. This ensures that over a full year there is balance between inflow and release.

The seasonal balance is cyclical in the sense that the last season in the year in the same way links to the first season of the year as if the the first year (through use of the $S++$ operator).

Volume capacity (MWh), upper and lower bounds

The available volume in the storage (MWh) in any area is given indirectly as the product of three terms: (the sum of GHYRS hydro capacities (MW) in the area) $*$ (the WTRRSFLH (h) for the area) $*$ (the hydro data element HYRSMAXVOL (share) in HYRSDATA). This is handled in equation QHYRSMAXVOL(AAA,SSS). There is similarly a time dependent minimum volume (MWh) QHYRSMINVOL(AAA,SSS).

Data for upper and lower reservoir bounds are given in Set HYRSDATASET and Parameter HYRSDATA(AAA,HYRSDATASET,SSS).

Unloading (MW)

Unloading is specified on technologies GHYRS, using variables VGE_T(IA,IGHYRS,SSS,TTT) and VGEN_T(AAA,IGHYRS,SSS,TTT). Unloading capacity is specified by the capacities of GHYRS technologies, with capacities given in GKFX for exogenous values (possibly updated w.r.t. investments) and VGKN for endogenous capacities. Unloading reduces the content in the storage and contributes to the electricity generation in the region in which AAA is located.

Note that there is also a limit on the total hydro power generation from the sum of hydro with and without storage (GHYRS and GHYRR), equation QHYMAXG(AAA,SSS,TTT).

Loading (MW)

Loading is specified exogenously by time series WTRRSVAR_S(AAA,SSS) ($\sim$ MW) as a profile. This profile is scaled by WTRRSFLH (h) and the capacities GKFX(YYY,AAA,HYRS) (MW).

Some references for this:

QHYRSSEQ(AAA,SSS), Section 8 page 78. QHYRSMAXVOL(AAA,SSS) Section 8 page 78. QHYRSMINVOL(AAA,SSS) Section 8 page 78. QHYMAXG(AAA,SSS,TTT) Section 8 page 78. WTRRSVARS(AAA,SSS) Section 4.16.1 page 58. WTRRSFLH(AAA) Section 4.7.9 page 52. HYRSDATASET Section 3.3 page 35. HYRSDATA(AAA,HYRSDATASET,SSS) Section 3.3 page 35. IHYINF_S(AAA,SSS) Section 4.27.13 page 69. GKFX Section 4.19.1 page 62. VGKN Section 7 page 76.

Short term storages

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

This type of storage are named "short term" storages because they are mainly motivated by their ability to receive, store and release energy on a seasonal (typically weekly) basis. The associated technology types GHSTO and GESTO refer to heat and electricity, respectively. Possibilities to link the storage volume between seasons are available, associated with technology types GHSTOS and GESTOS (Section 3.3.1 page 35) and further controlled by option stolinks (file balopt.opt). The mechanisms of this type of storage are similar for heat and electricity, the only difference being that heat storages interact with the area heat balance while electricity storages interact with the regional electricity balance. Here only electricity will be handled.

Volume capacity (MWh)

The maximum available volume of the storage (MWh) in an area is not given directly as a number at area level but indirectly as the sum of the storage technologies in the area. There is none or a single storage in any area. The available volume (MWh) in the storage in any area is given as the sum of capacities of the GESTO technologies in that area. The variable VESTOVOLT(AAA,S,T) specifies the content (MWh) of the storage at the beginning of any time segment (S,T). Note that the volume variable (and thus also the volume itself) is specified at area level, and it is not as for most other technologies specified at a combination of area and technology level. Also note that VESTOVOLT applies to exogenous (existing) capacity as well as to endogenous (new) capacities.

Unloading (MW)

Unloading reduces the content in the storage and contributes to the electricity balance (equation QEEQ) in the region in which the area is located. Unloading is specified on technologies GESTO, using variables VGE_T(AAA,IGESTO,S,T) (MW). Unloading capacity is given indirectly, being derived from the unloading capacities and the technology data element GDSTOHUNLD (h). This value specifies the number of hours that it takes for the technology to unload a full storage to being empty given that it is unloading at full capacity all the time. The unloading capacity is derived basically as: GKFX/GDATA(GESTO,"GDSTOHUNLD"); as seen, the units on the right are (MWh/h) or together MW. For exogenous capacities the limit is specified through VGE_T.UP, for endogenous capacities see equation QGEKNT (Section 8).

Loading (MW)

Loading increases the content in the storage and reduces the electricity balance (equation QEEQ) by the amount it consumes from the in the region in which the area is located. Loading is specified by variable VESTOLOADT (AAA,S,T) (MW). Loading capacity is given indirectly, being derived from the unloading capacities and the technology data element GDSTOHLOAD (h). Note that the 'H' in GDSTOHLOAD indicates "hour", not heat, and GDSTOHLOAD applies to both electricity and heat storage. This value specifies the number of hours that it takes for the technology to load an empty storage to being full given that it is loading at full capacity all the time. The loading capacity is derived basically as: GKFX/GDATA(GESTO,"GDSTOHLOAD"); as seen, the units on the right are (MWh/h) or together MW. For exogenous capacities the limit is specified through VESTOLOADT.UP, for endogenous capacities see equation QESTOLOADTLIM (Section 8).

Dynamic equation

The dynamic linking of storage volumes, loading and unloading is given in equation QESTOVOLT. Essentially this equation expresses that the storage content at the beginning of the next time segment equals the storage content at the beginning of this next time segment plus loading minus unloading during this time segment.

The dynamic equation for the short term storage is given as indicated here

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

The circular operator "++" is applied to ensure no net loading or unloading during one season. Thus, 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 3.2.3). 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.

In particular, if S is a proper subset of SSS, with e.g., SSS representing the 52 weeks of the year and S has four weeks representing spring, summer, autumn and winter, it may make little sense to link storage volume between seasons.

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 with SSS representing e.g., the 52 weeks of the year and S has four weeks representing spring, summer, autumn and winter. 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. As mentioned above there are flexible possibilities (Section 3.3.1 page 35).

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.

Section 13.3.12 page 110 has more details on this.

New technology

New storage capacities may be introduced endogenously, using new capacity variable VGKN (MWh) and new unloading variables VGEN_T (MW).

Electricity vs heat storages

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 (cf. Addon H2, Section 13.3.7 page 98). 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.

Discussion

The seemingly complicated handling of storages is motivated by considerations related to endogenous investments. Investment in a particular technology in a particular area and year is specified by only one variable, VGKN. This variable specifies the size (capacity) of the new technology, and all other variables that pertain to this technology (such as maximum generation) have they size scaled to the capacity size. For the back pressure technology GBPR for instance, the capacity variable indicates the electricity generation capacity (MW). The capacity for heat generation (MW) is implicitly given by the electricity capacity and the GDCB value. For short term storages the size (capacity) indicates the volume capacity (MWh). The capacity for loading is implicitly given from the unloading capacities and the technology data element GDSTOHLOAD, and the capacity for unloading similarly using the technology data element GDSTOHUNLD. The difference between the two major storage types is motivated by their loading characteristics. While short term storages have their loading specified as endogenous variables, hydro storages have their loading specified through an exogenous time series for inflow. Hydro storages therefore use the full load hours (FLH) idea also applied for other renewable energy sources like wind and solar, see i.a. Sections 4.7.4 and 4.17.4.

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 be 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, FDACRONYM, FDCO2, FDSO2 , FDN2O / ;

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

(Note: FDACRONYM is meant to replace FDNB, however, FDNB is presently kept for compatibility reasons.)

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 34.

Acronyms for fuels

Acronyms are used to handle fuel names. These acronyms are user defined (in contrast to acronyms for technology types (page 14 in Section 1.4.2).

The user should ensure that the acronyms for fuels precisely match the elements in set FFF. Thus, with e.g. SET FFF /Coal, hydro, Wind, GAS /; the acronyms for fuel should be COAL, HYDRO, WIND and GAS.

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.

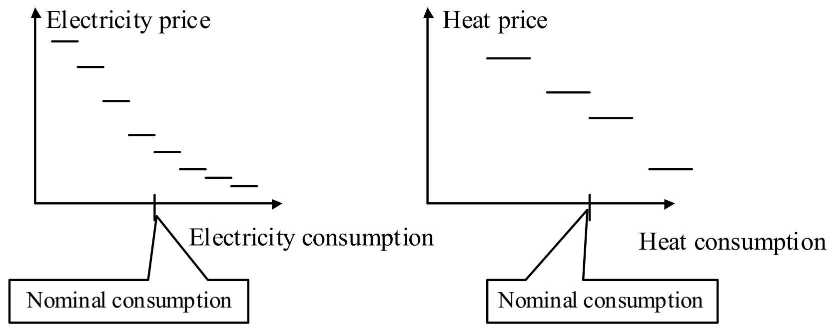

Figure 2: Elastic demand, illustration of price elasticities.

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 FKPOT and GDATA(G,"GDKVARIABL"), Sections 4.15.1, 4.13.1, will automatically be used (through the derived set IAGKN, Section 3.8.8).

See Section 3.8.7 concerning specification of capacity according to heat or electricity side.

Other possibilities are described in Section 13.

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 4.18.1 and 4.17.1.
- 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, 4.17.7, 4.18.2, 4.27.33, 4.27.34, while the related sets are specified in the following.

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. : The flexible demand is now turned into an addon, see Section 13.3.4

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.:

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

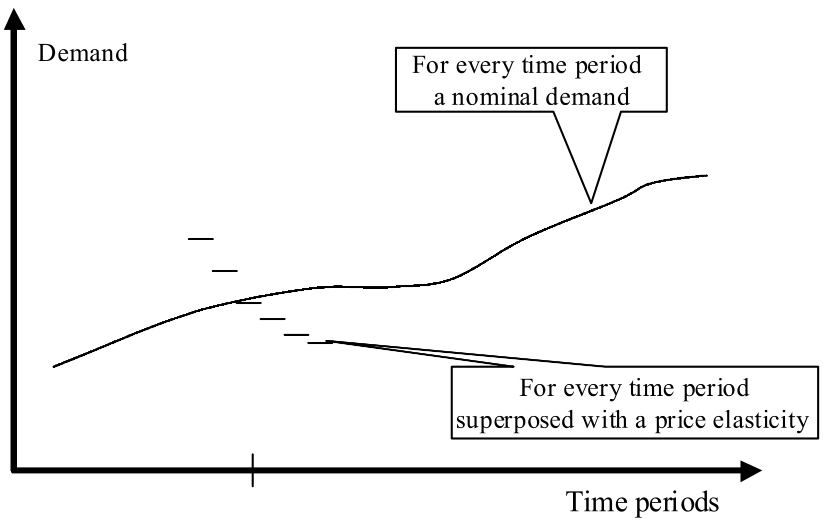

Figure 3: Elastic demand, illustration of development over time.

This example shows 7 steps.

The entry order (Section 3.2.3) of the labels in DEF is important, cf. Section 4.22.1.

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 4.22.1.

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 4.22.1.

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 (default/0/, eps for 0) (t CO2/year)"
LIM_SO2 "Annual SO2 limit (default/0/, eps for 0) (t SO2/year)"
LIM_NOx "Annual NOx limit (default/0/, eps for 0) (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 4.21.1. 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 data sets. The names of these sets start with I, Section 1.5 page 20. The internal sets are dynamic sets in the sense explained in Section 3.8.1 page 42.

The acronyms holding technology types, cf. Section 3.3.1 page 35 are internal (although not sets) in the sense that they shall not be handled by the user as input. Also a number of parameters are internal, Section 4 page 46.

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 3.2.3. 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 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))));
```

3.8.4 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)));

3.8.5 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.

3.8.6 Season and time duplication: ISALIAS, ITALIAS

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

3.8.7 Generation technology types

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

A number of convenient subsets of generation technology types have been defined. These are the 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), Solar heat (producing heat), Wave power. Some technologies may be combined into one, see Section 13.3.3.

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 characterized as a "back pressure" technology type and so may a Diesel motor that produces electricity (by driving a generator) and heat (using the surplus heat). From a functional point of view the characterisation below, as e.g. in Table 2, 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), IGHSTO(G), IGESTO(G), IGHYRS(G), IGHYRR(G), IGWND(G), IGSOLE(G), IGSOLH(G), IGWAVE(G), GHSTO_S.

Each technology is automatically allocated to one and only one of these sets according to the acronym (cf. Section 3.3.1 page 35) 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);

ACRONYM =>	GCND	GBPR	GEXT	GHOB	GETOH	GWND	GSOLE	GSOLH	GHYRS	GHYRR	GESTO	GHSTO	GESTOS	GHSTOS	GWAVE
Electricity output	x	x	x			x	x		x	x	x		x		x
Electricity input					x						(x)		(x)		
Heat output		x	x	x	x			x			(x)	x	(x)	x	
Heat input												(x)		(x)	
Storage									x		x	x	x	x	
Dispatchable	x	x	x	x	x				x		x	x	x	x	
Fixed electricity to heat relation		x			x						(x)		(x)		

Table 2: Functionality of the technology types.

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

Acronyms are used because they are easy to remember and recognize. Under circumstances where acronyms can not be used (e.g. in relation to certain non-GAMS data handling systems) integer numbers might be used in the GAMS code, this is why the code may actually be e.g. (assuming consistency between integer 4 and acronym GHOB)

IGHOB(G) = YES$((GDATA(G,'GDTYPE') EQ 4) OR (GDATA(G,'GDTYPE') EQ GHOB));

If acronyms can not be used, use the numbers consistently as seen in the Balmorel.gms code.

Combination of technologies is possible, see Section 13.3.3.

Observe that the sets are defined as subsets of G, not GGG, so only the technologies in the present model instance may enter the above sets.

The technologies in each set has specific properties in the model. The relative characteristics of each of the technology types are described in Table 2. 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 sufficient understanding of the model (and this is the reason why these acronyms are internal). The 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, Table 3.8.7. 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 convention here is: capacity is specified relative to output side, and relative to electricity, whenever this is possible).

The identification of these technology sets are:

IGHH(G): producing only heat

ACRONYM: Internal set:	GCND IGCND	GBPR IGBPR	GEXT IGEXT	GHOB IGHOB	GETOH IGETOH	GHSTO IGHSTO	GESTO IGESTO	GHSTOS IGHSTS	GESTOS IGESTOS
IGE	yes	yes	yes		yes		yes		yes
IGH		yes	yes	yes	yes	yes		yes	
IGEH		yes	yes		yes				
IGEE	yes						yes		yes
IGHH				yes		yes		yes	
IGEENOSTO	yes								
IGHHNOSTO				yes					
IGKE	yes	yes	yes				yes		yes
IGKH				yes	yes	yes		yes	
IGKENOSTO	yes	yes	yes		yes				
IGKHNOSTO				yes					
IGNOTETOH	yes	yes	yes	yes		yes	yes	yes	yes
IGDISPATCH	yes	yes	yes	yes	yes	yes	yes	yes	yes
ACRONYM: Internal set:	GHYRS IGHYRS	GHYRR IGHYRR	GWND IGWND	GSOLE IGSOLE	GSOLH IGSOLH	GWAVE IGWAVE			
IGE	yes	yes	yes	yes		yes			
IGH					yes				
IGEH									
IGEE	yes	yes	yes	yes		yes			
IGHH					yes				
IGEENOSTO	yes	yes	yes	yes		yes			
IGHHNOSTO					yes				
IGKE	yes	yes	yes	yes		yes			
IGKH					yes				
IGKENOSTO	yes	yes	yes	yes		yes			
IGKHNOSTO					yes				
IGNOTETOH	yes	yes	yes	yes	yes	yes			
IGDISPATCH	yes								

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

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 3.8.7 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.

Further technology related internal sets that may be in use are e.g. in Section 13.3.3 page 94.

The set IGGGALIAS is a duplication of the set GGG.

3.8.8 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.

3.8.9 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 7. 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:

- 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 acronyms (which are not sets) describing technology types should should not be changed without proper understanding of the functioning of these acronyms, cf. Section 3.3.1 page 35, cf. also e.g. Table 3.8.7 page 45.

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 3.8.7. 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 34 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, cf. also YVALUE Section 4.2.1 page 48).

Finally, as previously noted, the user should not change the internal sets, cf. Section 3.8 without proper understanding.

4 Parameters and scalars

In this section we describe parameters and scalars that must be specified by the user while in Section 4.27 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 1.4 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.

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 4.26.

4.0.1 Units

The units used in parameters are: MW (megawatt), MWh (megawatthour), GJ, hours, days, kg (kilogram), t (metric tonne), 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 4.18.1.

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.

4.0.2 Data entry

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 4.25.

Note the following on input format and use of EPS. GAMS operates with default values; default values are not stored. In GAMS the default value for parameters and scalars is 0. In some cases, however, it would be nice to be able to distinguish between a 0 which means 'this entry takes the value 0' and a 0 which means 'this entry is not relevant'. This is indeed possible. Consider for example FKPOT(CCCRRRAAA,FFF). If input for an index combination ('NatGas','Stockholm') specifies 0, or nothing at all, the numerical value will be 0. Therefore a condition like "$FKPOT(CCCRRRAAA,FFF)" will for ('Stockholm','NatGas') evaluate to "false". However, if FKPOT('Stockholm','NatGas') is given the value EPS then "$FKPOT(CCCRRRAAA,FFF)" 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,FFF)$FKPOT(IA,FFF)..

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

Application of such refinement may potentially save a lot of input data entering, with positive consequences for memory use, code execution and file sizes. Some care has to be taken in coding when applying this idea.

In the description of individual identifiers and equations that use this convention this will be marked by "EPSconv" or just "EPS".

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 7, Section 3.8.9.

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.

Note in particular for models with integer variables that introduction of feasibility ensuring variables VQ may result in longer solution time. Additionally a solution that is not optimal may result.

The reason for introducing the variables VQ... is that it may be very hard to find out what to do in case of infeasibility.

If an infeasibility exists in a variable due to upper bound being lower than lower bound GAMS will report this in the list file; this is the easy case.

If an infeasibility exists in a single equation (in combination with lower and/or upper bounds on the variables entering the equation) it is often 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 villain 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 12, or the User can supply a display line like e.g. ”display VQEEQ.L;”.

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 particular, if the penalty on the VQ variables for electricity and heat are equal this might easily occur. Making the penalties differ between electricity and heat (e.g., by multiplying one with 100) may apparently shift the infeasibility between the heat and electricity equations.

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

4.1.2 OMONEY

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, which is specified by the user, 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, therefore add spaces in the text string as needed, e.g. ”DKR ”.

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 Section 4.27.4.

(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(S) 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.

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 parameter IWEIGHSUMT, see Section 4.27.5.

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.

Comment on input data: The set T is ordered, cf. Section 3.2.3. 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

	3%	4%	5%	6%	7%	8%	9%	10%	15%	20%
5	0.2184	0.2246	0.2310	0.2374	0.2439	0.2505	0.2571	0.2638	0.2983	0.3344
10	0.1172	0.1233	0.1295	0.1359	0.1424	0.1490	0.1558	0.1627	0.1993	0.2385
15	0.0838	0.0899	0.0963	0.1030	0.1098	0.1168	0.1241	0.1315	0.1710	0.2139
20	0.0672	0.0736	0.0802	0.0872	0.0944	0.1019	0.1095	0.1175	0.1598	0.2054
25	0.0574	0.0640	0.0710	0.0782	0.0858	0.0937	0.1018	0.1102	0.1547	0.2021
30	0.0510	0.0578	0.0651	0.0726	0.0806	0.0888	0.0973	0.1061	0.1523	0.2008

Table 3: 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).

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 3.03. PARAMETER ANNUITYC indicates the transformation of an investment to an annual payment. Unit: (none).

Thus, for instance, an investment of 100 Money 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 Money, 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%.

One expression for calculating the annuity is

$$A = \frac{r}{1-(1+r)^{-n}}$$

where r is the interest rate (example: 0.05), n (example: 20) is the number of years and A (example: 0.0802) is the annuity. Alternative expressions for A are

$$r + \frac{r}{(1+r)^n - 1} \quad \text{and} \quad \frac{r(1+r)^n}{(1+r)^n - 1}$$

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

Table 3 illustrates the dependence of the annuity on various combinations of interest rates, 5, 6, 7, 8, 9, 10, 15, 20% and number of years, 3, 4, 5, 10, 15, 20, 25, 30 years. See also Section 13.

Apart from the above interpretation of ANNUITYC, it may more generally be taken to represent the investors' perception of financing costs, alternatives, expectations to profit, risks, etc., providing ample possibilities to the User. A lower bound on the value of ANNUITYC would with respect to r be one divided by the lifetime of the investment, this could represent the situation where there is no risk nor financing costs associated with the investment.

4.5.2 TAX_DE

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. 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 3.03. 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

PARAMETER DISLOSS_E holds the loss in electricity distribution, as a fraction of the electricity entering the distribution network. Unit: (none).

The idea behind taking it as a fraction of electricity entering the distribution network is that sometimes electricity consumption is measured at the plant, while consumption is the derived from that and losses.

4.6.2 DISCOST_E

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 3.03. 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 4.22.1.

4.7 Parameters on the set AAA

4.7.1 DISLOSS_H

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).

The idea behind taking it as a fraction of heat entering the distribution network is that sometimes heat consumption is measured at the plant, while consumption is the derived from that and losses.

4.7.2 DISCOST_H

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 WNDFLH

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

There are other ways of expressing the relationships between the size of a production unit and the energy output over a period of time (but only the FLH is used in Balmorel). A common term is utilization. The (dimension-less) capacity factor is another term. It expresses the relationship between the actual energy production (MWh) during the interval (typically a year) divided by the maximum possible production (MWh). It may be verified that the capacity factor value is equal to the FLH value times the number of hours during the interval. Thus, let τ be the duration (hours) of the interval in question, K the capacity (MW), C the capacity factor (dimension-less) and E the energy production during the interval. Then the maximum possible production is $K\tau$. By definition

$$FLH \equiv \frac{E}{K} \tag{1}$$

and therefore

$$C \equiv \frac{E}{K\tau} = \frac{E}{K}\frac{1}{\tau} = \frac{FLH}{\tau} \tag{2}$$

and

$$FLH = C\tau \tag{3}$$

4.7.5 SOLEFLH

PARAMETER SOLEFLH holds the full load hours for solar power. Unit: hours. Cf. WNDFLH in Section 4.7.4.

4.7.6 SOLHFLH

PARAMETER SOLHFLH holds the full load hours for solar heat. Unit: hours. Cf. WNDFLH in Section 4.7.4.

4.7.7 WAVEFLH

PARAMETER WAVEFLH holds the full load hours for wave power. Unit: hours. Cf. WNDFLH in Section 4.7.4.

4.7.8 WTRRRFLH

PARAMETER WTRRRFLH holds the full load hours for hydro run of river power. Unit: hours. Cf. WNDFLH in Section 4.7.4.

4.7.9 WTRRSFLH

PARAMETER WTRRSFLH holds the full load hours for hydro with storage power. Unit: hours. Cf. WNDFLH in Section 4.7.4.

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.

There are two values for coupling between generation technology and fuel, FDACRONYM and FDNB. This is a deliberate redundancy, as they have relative merits.

FDACRONYM should be used in most cases because it is safer (and maybe also easier to remember). The User provides the relevant acronyms in data file FFFACRONYM.inc, cf. 5.1. For FDNB the elements GDFUEL (Section 4.13.1) for each technology contains a positive integer that points to the FDNB for the fuel that the technology uses (and therefore two different fuels should not have identical FDNB). In cases where FDACRONYM can be applied without problem outside GAMS FDNB values should be left empty.

However, certain data tools may not be able to handle acronyms, therefore FDNB is also provided. In this case FDNB values have to be given consistently. FDACRONYM values may be given but are not used in this case, because they will be overwritten in Balmorel.gms by the statement FDATA(FFF,'FDACRONYM')$(NOT FDATA(FFF,'FDACRONYM')) = FDATA(FFF,'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, MWh for Storage), Section 4.19.1.

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 (MWh for Storage, MW), Section 4.19.1.

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

Substituted by GEFFRATE, see next.

4.10.5 GEFFRATE

PARAMETER GEFFRATE 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 GEFFRATE. The value of GEFFRATE will be strictly positive and usually close to 1. Default is 1 (so this value should not be entered in the data file).

See also Section 10.1.2.

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 4.18.3.

Observe that this exchange (intended to be positive for export, negative for import, but from Section 4.18.3 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.

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 7.

Electricity exchange with regions not in IR, depending on import-export prices, is described in Section 13.3.8.

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. 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 3.03. GDTYPE: This is an integer or an acronym. According to the value, the technology is uniquely placed in one of the internal sets specified in Section 3.8.7. 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 3.8.7.
>
> THE FOLLOWING IS NEW FOR VERSION 3.03. An acronym will take the place of GDTYPE.
>
> GDFUEL: This is an Acronym (3.4) indicating which fuel the technology uses, corresponding to the fuel acronym 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. In general terms this is the relation between input energy and utilized (output) energy (electricity and/or heat). For thermal technologies the GDFE value will be 1 or less if the higher heating value is used in input data, while it may be slightly higher than 1 if using the lower heating value. The choice between use of higher and lower heating values is made by the User. Values close to 1 are typically obtained for a heat boiler (technology type GHOB). Note some special cases:
>
>> For technology type GEXT typically the GDFE value will be around 0.4.
>>
>> For technology type GBPR the utilized energy is the sum of electricity and heat output; typically the GDFE value will be around 0.8.
>>
>> For technology type GETOH the input energy is the electricity and the utilized energy is the heat output.
>>
>> If the technology represents an electric boiler the GDFE will be 1 or slightly less, if it represents a heat pump the value will typically be in the range 3 to 6. For heat pumps the value of GDFE will equal the value of the coefficient of performance, or COP, a frequently used term.
>>
>> For technology types GHYRR, GHYRS, GSOLE, GSOLH and GWAVE the value will typically be 1 since the energy input is expressed in terms of energy output equivalents.
>
> GDESO2: Degree of desulphuring, value between 0 and 1.

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 3.8.7

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 3.8.7 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 (both el and heat) from empty

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

GCOMB: the technology is a combination technology, see Section 13.3.3.

GDCOMBSK: relevant for a combination technology, see Section 13.3.3.

GDCOMBSLO: relevant for a combination technology, see Section 13.3.3.

GDCOMBSUP: relevant for a combination technology, see Section 13.3.3.

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 code:

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 empty).

4.14 Parameters on the set product (IRRRE,IRRRI)

Between any pair of regions there may be electricity transmission if there is a transmission line between them.

The way Balmorel models the electricity transmission is known as ATC (Available Transfer Capability).

The ATC modeling is a compromise between physical and commercial accuracies. Better consistency with the physical reality may be obtained by using AC or DC models. However, such more complicated models may be less transparent and more demanding with respect to calculation time and data. The ATC modeling is quite common, the Nord Pool Spot is an example of an electricity market that is based on ATC representation of the electrical flows.

In Balmorel there are three characteristic of each of the electricity network connections: capacity, loss and cost (XKINI, XKRATE, XLOSS, XCOST). The electricity will flow in the network in the cheapest way (loss also has implications for cost), and within the transmission capacity limits.

A consequence is that when there are alternative routes where all transmission link have the same cost and losses, the best route will be preferred (here 'best' is with relation to the total objective function). If in particular the cost and losses are identical along alternative routes (e.g. because all losses and costs are zero), then any of the alternative routes may be chosen by the solver, or any mixture, like sending half along each of two alternative routes.

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 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)). It is assumed in the model that the investment cost is symmetric.

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

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.

4.14.4 XLOSS

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).

4.15 Parameters on the set product (CCCRRRAAA,FFF)

4.15.1 FKPOT

PARAMETER FKPOT holds the fuel potentials specified at geographical levels. Unit: MW. *EPSconv* (Section 4.0.2).

The potential is specified as an upper limit on geographical levels' installed generation capacities relative to the individual fuel type.

Note that the declaration is on domain (CCCRRRAAA,FFF), hence values can be entered for geographical levels CCC, RRR and AAA, cf. Section 3.1.1.

Comment on input data: The specification of the fuel potential (MWh) as an upper limit on generation capacity is obviously a simplification. However, in many cases the fuel potential, defined in energy terms (MWh), is not known with precision and this motivates the simplification. See Section 13.

4.15.2 FGEMIN

PARAMETER FGEMIN(CCCRRRAAA,FFF) 'Minimum electricity generation by fuel (MWh)'.

To be described; follows the same idea as FKPOT above.

4.15.3 FGEMAX

PARAMETER FGEMAX(CCCRRRAAA,FFF) 'Maximum electricity generation by fuel (MWh)'.

To be described; follows the same idea as FKPOT above.

4.16 Parameters on the set product (AAA,SSS)

4.16.1 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.

In some situations the surface of the reservoir is large relative to the volume. If some season has no inflow and if in addition there is a considerable evaporation from the reservoir the net

	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
2008	45.84	38.54	29.6	37.86	25.8	40.46	44.43	54.62	67.47	56.48	51.27	44.52
2009	41.41	38.21	35.06	34.04	32.67	35.37	32.81	32.41	28.61	33.76	36.38	39.6
2010	437.6	558.3	458.7	372.9	338.7	354.1	365.1	340.2	390.7	402.4	446.3	646.5
2011	544.3	505.2	503.2	420.5	427.5	379.1	301.9	313.3	223.1	216.2	320.4	261.5
2012	285.5	372.0	219.5	240.1	215.9	189.1	102.5	172.5	187.6	257.5	251.2	316.3
2013	305.7	294.6	335.6	345.6	278.9	258.3	266.6	280.8	306.5	311.4	301.2	274.9
2014	282.1	253.3	221.9	210.7	214.8	206.9	239.4	264.9	285.7	253.7	253.7	284.9

Table 4: Monthly average prices (NOK/MWh) (Euro/MWh for 2008-2009) for the system spot price in the Nordpool area (source: www.nordpool.com).

inflow may in that season be negative. (See in Section 4.27.35 how this might lead to division by zero.)

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

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.

4.16.2 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 HYPPROFILS 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 8.

As example, Table 4 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 HYPPROFILS. 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 HYPPROFILS (it will depend more distinctly on the amount of water available during the year).

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

4.17.1 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 4.18.1).

4.17.2 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 4.18.1).

4.17.3 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 4.18.1).

4.17.4 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 4.18.1).

4.17.5 WAVE_VAR_T

contains the description of seasonal and daily variation of the wave power generation. Unit: (none~MW) (see description in relation to DE_VAR_T, Section 4.18.1).

4.17.6 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 4.18.1).

4.17.7 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. Implemented as Addon.

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 4.27.33 and Section 10.

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

4.18.1 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 4.27.7) and DE_VAR_T and IDE_SUMST in combination with DE are used to calculate IDE_T_Y (Section 4.27.31).

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 4.27.31.

Comment on input data: The calculation of parameter IDE_T_Y, Section 4.27.31, 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 for most of them).

Observe that parameters DE_VAR_T, DH_VAR_T, WND_VAR_T, SOLE_VAR_T, X3FX_VAR_T and similar ones will be handled in similar ways as DE_VAR_T (special care should be taken with X3FX_VAR_T which may contain negative values, see Section 4.18.3).

4.18.2 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. Implemented as Addon.

See the explanation in relation to DHFP_CALIB, Section 4.17.7.

The parameter is used in the calculation of IDEFP_T, Section 4.27.34 and Section 10.

4.18.3 X3FX_VAR_T

PARAMETER X3FX_VAR_T contains the description of seasonal and daily variation in the fixed exchange with third regions, Section 4.11.1. Unit: (none∼MW).

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). The values are seen from the country in the set CCC, and any losses are disregarded.

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. also Section 4.27.16. This can de detailed as follows. By convention, a positive X3FX is meant to mean that the net annual export is positive. This annual amount of energy is distributed over time segments according to X3FX_VAR_T. For any time segment, the sign of the value in X3FX_VAR_T should indicate whether there is export (positive number) or import (negative number). - Calculation of the net export for any time segment is done using the following term
$(X3FX(Y,IR) * X3FX\_VAR\_T(IR,S,T))/IX3FXSUMST(IR)$.
The direction (import or export) will therefore only be correct if the signs of the annual values X3FX and IX3FXSUMST are the same.

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

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

Descriptions to be made for:

PARAMETER GKNMAX(YYY,AAA,GGG) 'Maximum capacity at new technologies (MW)'

PARAMETER GKVACCDECOM(Y,AAA,G) 'Investments in generation technology by BB2 up to and including year Y with subtraction of decommissioning';

PARAMETER GKVACC(Y,AAA,G) 'Investments in generation technology by BB2 up to and including year Y without subtraction of decommissioning';

PARAMETER GVKGN(YYY,AAA,G) 'Investments in generation technology by BB2 in year Y';

PARAMETER TAX_GH(YYY,AAA,G) 'Heat taxes on generation units';

PARAMETER TAX_GE(YYY,AAA,G) 'Electricity taxes(+) and subsidies(-) on generation units (Money/MWh)';

PARAMETER TAX_GF(YYY,AAA,G) 'Fuel taxes on heat-only units';

PARAMETER TAX_KN(YYY,AAA,G) 'Taxes(+) and subsidies(-) for investment in electricity generation (mMoney/MW)';

4.19.1 GKFX

PARAMETER GKFX holds the exogenously specified generation capacities. Unit: MW, except MWh for short terms storages.

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 GKRATE, see Section 4.24.2.

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. So the rule is, "If there is any electricity generation specify electricity capacity, else specify heat capacity". See Section 3.8.7, in particular the sets IGKH and IGKE.

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

4.20.1 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).

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

4.21.1 M_POL

PARAMETER M_POL contains emissions policy data. Unit: See Section 3.7. *EPSconv* (Section 4.0.2).

For each year in the simulation the data is transferred to internal parameters, see Sections 4.27.25 ff.

Observe the following

	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 5: Illustration of TABLE DEF_STEPS

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

4.22.1 DEF_STEPS

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. : The flexible demand is now turned into an addon, see Section 13.3.4

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 5. 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 substitution) type. IS-
CALAR1 is the elasticity, where e.g. -0.1 (must be negative) means that an increase 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 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$(ORD(DEF_D1) EQ CARD(DEF_D1)),
DEF_STEPS(IR,S,T,'DF_PRICE',DEF_D1)=1+ISCALAR2/2;
DEF_STEPS(IR,S,T,'DF_QUANT',DEF_D1)=1+ISCALAR1*ISCALAR2/2;);
LOOP(IANYSET$((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));
```

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

4.23.1 DHF_STEPS

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. : The flexible demand is now turned into an addon, see Section 13.3.4

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 4.22.1.

4.24 Domain overloaded input

Except for scalars and simple sets input values are declared on a domain (a list of indexes), e.g. SET GDATA(GGG,GDATASET) is declared on the domain (GGG,GDATASET). For some input it is not obvious what the appropriate domain should be, this is particularly true for any time related part of the domain, viz., related to YYY, SSS and/or TTT. On the one hand it is desirable to keep input data as simple as possible, on the other hand it should also be sufficiently detailed. The appropriate balance will depend on the analysis.

To provide some flexibility on this, Balmorel applies domain overloading. For instance, GKRATE may be declared as GKRATE(AAA,GGG,SSS) or as GKRATE(AAA,GGG,SSS,TTT).

The mechanism for this is the options, cf. Section 13.2. Here is an example for GKRATE:

In balopt.opt the control statement $setglobal XKRATE_DOL IRRRE_IRRRI will specify that XKRATE is declared as XKRATE(IRRRE,IRRRI) while $setglobal XKRATE_DOL IRRRE_IRRRI_SSS_TTT will specify that XKRATE is declared as XKRATE(IRRRE,IRRRI,SSS,TTT).

The appropriate data file will be included according to the option value specified.

Two special options are used. One for specifying domain-less sets (i.e., simple sets) and parameters (i.e., SCALARS), indicated by NODOM. Another to indicate that no data file should be used, indicated by NOFILE.

Domain overloaded input identifiers are described in the following sections, while the associated internal identifiers are described under Section 4.27.

4.24.1 GKRATE

This ID is from version 3.03 replaced by GKRATE below.

4.24.2 GKRATE

PARAMETER GKRATE represents a reduction in capacity. Unit: (none). Domain overloaded: (AAA,GGG,SSS) and (AAA,GGG,SSS,TTT). Default value 1, use eps for 0. (Section 4.28)

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 GKRATE has implication for capacity of both electricity and heat (if both are relevant).

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

To represent an average of 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 10.1.1.

Some of the domain overloaded parameter are listed next.

4.24.3 XKDERATE

This ID is from version 3.03 replaced by XKRATE, see below.

4.24.4 XKRATE

Domain overloaded.

XKRATE is transmission capacity rating, typically close to 1. Default value 1, use eps for 0. (Section 4.28, 4.25)

4.24.5 GMAXF

Domain overloaded. *EPSconv* (Section 4.0.2, 4.25). To be described, 4.25.

4.24.6 GMINF

Domain overloaded. To be described, 4.25.

4.24.7 GEQF

Domain overloaded. To be described.

4.24.8 FGEMAXF

EPSconv (Section 4.0.2). To be described, 4.25.

4.24.9 FGEMINF

EPSconv (Section 4.0.2) (Section 4.28)). To be described, 4.25.

4.25 Default values - GAMS and Balmorel

Default values - GAMS

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 GEFFRATE(GGG,AAA)
              DK_E_Urban   LT_R_Urban
 CC-Cond1        0.94          0.99
 ST-BP-2-C                     0.87        ;

GEFFRATE(G,A)$(GEFFRATE(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 value EPS has a special meaning in GAMS, and this is exploited in Balmorel in relation to default values, see further Section 4.28.

Default values - Balmorel

Apart from applying values EPS in a specific way for some data input, also a 0 in some data input may be interpreted in a specific way in Balmorel, viz., as a 1. This is when the value 1 appears to be a reasonable value for lack of better information, cf. Section 4.28.

4.26 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 4.27.31, 4.27.32, 4.18.3.

- 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.

4.27 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 6. 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.

4.27.1 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.

4.27.2 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/.

4.27.3 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 1.4.1, 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)),
...
));
```

4.27.4 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.

4.27.5 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.

4.27.6 IHOURSINST

The internal parameter IHOURSINST(S,T) holds the length of the time segment (S,T) measured in hours. Unit: (hours).

4.27.7 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 4.27.31.

4.27.8 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 4.27.7, Section 4.18.1 and Section 4.27.31. The use is described in Section 4.27.32.

4.27.9 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 4.18.1 and Section 4.27.31.

4.27.10 IWAVESUMST

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

IWAVESUMST(IA) = SUM((S,T), IHOURSINST(S,T)*WAVE_VAR_T(IA,S,T))

Unit: (none∼MWh). See also Section 4.18.1 and Section 4.27.31.

4.27.11 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 4.18.1 and Section 4.27.31.

4.27.12 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 4.18.1 and Section 4.27.31.

4.27.13 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 3.03.

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

4.27.14 IWTRRRSUM

The internal PARAMETER IWTRRRSUM holds the annual amount of hydro run-of-river generated electricity, similarly as IWND_SUMST in Section 4.27.9.

Unit: (none∼MWh).

4.27.15 IWTRRSSUM

The internal PARAMETER IWTRRSSUM holds the annual amount of hydro run-of-river generated electricity, similarly as IWND_SUMST in Section 4.27.9.

Unit: (none∼MWh).

4.27.16 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,

IX3FXSUMST(IR) = SUM((S,T), IHOURSINST(S,T)*X3FX_VAR_T(IR,S,T))

Unit: (none∼MWh). See also Section 4.18.3.

4.27.17 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.

4.27.18 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.

4.27.19 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 4.27.20. 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 4.27.21.

4.27.20 IGKFX_Y

The internal PARAMETER IGKFX_Y holds the externally given (parameter GKFX, see Section4.19.1) generation capacity at the beginning of the currently simulated year. Unit: MW, MWh for Storage.

Total capacity at the beginning of the currently simulated year and throughout this year are described in Section 4.27.19.

4.27.21 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 4.27.19.

4.27.22 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.

4.27.23 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 4.19.1) and accumulated endogeneously found capacity IGKVACCTOY (Section 4.27.19) up to the beginning of the year.

4.27.24 IFUELP_Y

The internal parameter IFUELP_Y holds the fuel price in the year simulated, transferred from parameter FUELPRICE, Section 4.20.1. Unit: Money/GJ.

4.27.25 ITAX_CO2_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. 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 4.21.1, will be transferred to this internal parameter.

4.27.26 ITAX_NOX_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. 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 4.21.1, will be transferred to this internal parameter.

4.27.27 ITAX_SO2_Y

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. 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 4.21.1, will be transferred to this internal parameter.

4.27.28 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 4.21.1, will be transferred to this internal parameter.

4.27.29 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 4.21.1, will be transferred to this internal parameter.

4.27.30 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 4.21.1, will be transferred to this internal parameter.

4.27.31 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 4.18.1 and Section 4.27.7 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 4.27.7.

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 and others will be handled in similar ways as DE_VAR_T (special care should be taken with X3FX_VAR_T which may contain negative values, see Section 4.18.3).

4.27.32 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 4.27.31.

4.27.33 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 9.

4.27.34 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 9.

4.27.35 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.

4.27.36 IGKRATE

For domain overloading, cf. Section 4.24.2 and Section 4.24.

4.27.37 IXKRATE

For domain overloading, cf. Section 4.24.4 and Section 4.24.

4.27.38 IGMAXF

For domain overloading, cf. Section 4.24.5 and Section 4.24.

4.27.39 IGMINF

For domain overloading, cf. Section 4.24.6 and Section 4.24.

4.27.40 IGEQF

For domain overloading, cf. Section 4.24.7 and Section 4.24.

4.28 EPS in input - convention

EPS is one of the possible values in relation to extended value arithmetic (page 14 in Section 1.4.2). EPS is numerically equal to zero, but yet different, as the following example illustrates.

```
scalar sc1 /9/, sc2 /0/;
```

sc1 = ((1+2+3)/sc2)$(sc2 ne 0) ; /* sc1 not assigned here, value 9 remains */

sc1 = ((1+2+3)/sc2)$sc2; /* sc1 not assigned here, value 9 remains */

sc2=eps;

sc1 = ((1+2+3)/sc2)$(sc2 ne 0) ; /* sc1 not assigned here, value 9 remains */

sc1 = ((1+2+3)/sc2)$sc2; /* Error: Division by zero */

This property of EPS is exploited in Balmorel data input entry and associated code.

In GAMS the default value for parameters and scalars is 0, which is convenient for data entry. Moreover, it saves storage space. The value EPS is numerically 0, but EPS is stored anyway.

In some cases it would be nice to be able to distinguish between a 0 which means 'this entry is 0 but not relevant' and a 0 which means 'this entry is 0 and relevant'. This is indeed possible, if some conventions are applied. Consider for example FKPOTA(FKPOTSETA,IA) (Section 3.4). If input for an index combination ('NatGas','Stockholm') specifies 0, or nothing at all, the numerical value will be 0. Therefore a condition like "$FKPOTA(FKPOTSETA,IA)" will for ('NatGas','Stockholm') evaluate to "false", cf. the example above. However, if FKPOTA('NatGas','Stockholm') is given the value EPS then "$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.

For some other cases it would be nice to achieve that entering nothing would be interpreted to mean that the value is 1, i.e. where the numerical value 1 seems a natural default value. For the data item GKRATE, for instance, not assigning a value to one of the elements would (without further specific coding) mean that that GKRATE value is 0. This may indeed be the intended value, however, a default value of 1 seems more appropriate for GKRATE (Section 4.24.2). Therefore the Balmorel code has been designed such that in this case the numerical value 0 may be attained by entering EPS, while entering the value 0 or nothing will imply that 1 is used in the code.

Application of such refinement may potentially save a lot of input data entering, with positive consequences for memory use, code execution and file sizes. However, some care has to be taken in data entry and coding when applying this idea.

Here are the IDs that interpret input value 0 as 'this entry is not relevant': FGEMAXF (4.24.8), FGEMINF (4.24.9), FKPOT (4.15.1), FGEMINF (4.24.9), FGEMAXF (4.24.8), GMAXF (4.24.5), M_POL (4.21.1). (Note: need checking here and in code.)

Here are the IDs that interpret input value 0 as 1: GKRATE (4.24.2), GEFFRATE (4.10.5), XKRATE (4.24.4)

5 Acronyms

See Section 1.4.2 on acronyms.

5.1 FFFACRONYM

Note: with Balmorel version 3.03 file FFFACRONYM.inc is not used, the acronyms are in file FFF.inc.

The input data file FFFACRONYM.inc should contain acronyms related to fuels. The acronyms must match exactly the name of the elements in SET FFF "Fuels" (Section 3.4 page 39). Thus, with FFF given as e.g. SET FFF "Fuels" /BIO, GAS, OIL/; the content of FFFACRONYM.inc should be ACRONYMS BIO, GAS, OIL; These acronyms will mainly be used in GDATA (Section 4.13.1 page 55) as value for element GDFUEL.

In contrast to acronyms for technology types (Section 3.3.1 page 35) the acronyms for fuel are User defined. Therefore with Balmorel version 3.03 file FFFACRONYM.inc is not used, the acronyms are in file FFF.inc to make it easier to secure concistency.

5.2 Internal acronyms

The acronyms for technology types are internal, cf. Section 3.3.1 page 35.

6 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 4.27 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. See Section 4.1.2 page 48.
- 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.
- PARAMETERS OCARDSETA, OCARDSETR; SCALARS OTLW, OTLW1, OTLG, OTL1 OSCALAR1, OSCALAR2, OSCALAR3; sets given as ALIAS(S,OALIASS), ALIAS(T,OALIAST): used for handling various details in the layout and similar of print files. See file print1.inc for details.

PRINTSEPARATOR in file Balopt.opt is for permitting insertion of separators like ';' in output for use with e.g. copy-paste to Excel.

Error and logging are described in Section 12.2, and model output is described in Section 11.3.

7 Variables

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"
VGF_T(AAA,G,S,T) "Fuel consumption rate (MW), existing units"
VGFN_T(AAA,G,S,T) "Fuel consumption rate (MW), new units"
VX_T(IRRRE,IRRRI,S,T) "Electricity export from region IRRRE to IRRRI (MW)"
VXKN(IRRRE,IRRRI) "New electricity transmission capacity (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)"
VDECOM(AAA,G) 'Decommissioned 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(RRR,S,T,IPLUSMINUS) "Feasibility in electricity balance equation QEEQ (MW)"
VQHEQ(AAA,S,T,IPLUSMINUS) "Feasibility in heat balance equantion QHEQ (MW)";
VQESTOVOLT(AAA,S,T,IPLUSMINUS) "Feasibility in electricity storage equation QESTOVOLT (MWh)"
VQHSTOVOLT(AAA,S,T,IPLUSMINUS) "Feasibility in heat storage equation VQHSTOVOLT (MWh)"
VQHYRSSEQ(AAA,S,IPLUSMINUS) "Feasibility of QHYRSSEQ (MWh)"
VQGEQCF(C,FFF,IPLUSMINUS) "Feasibility in Requered fuel usage per country constraint (MWh)"
VQGMINCF(C,FFF) "Feasibility in Minimum fuel usage per country constraint (MWh)"
VQGMAXCF(C,FFF) "Feasibility in Maximum fuel usage per country constraint (MWh)"
VQGEQRF(RRR,FFF,IPLUSMINUS) "Feasibility in Requered fuel usage per region constraint (MWh)"
VQGMAXRF(RRR,FFF) "Feasibility in Minimum fuel usage per region constraint (MWh)"
VQGMINRF(RRR,FFF) "Feasibility in Maximum fuel usage per region constraint (MWh)"
VQGEQAF(AAA,FFF,IPLUSMINUS) "Feasibility in Requered fuel usage per area constraint (MWh)"
VQGMAXAF(AAA,FFF) "Feasibility in Minimum fuel usage per area constraint (MWh)"
VQGMINAF(AAA,FFF) "Feasibility in Maximum fuel usage per area constraint (MWh)"

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 8) 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 12.

Note in particular for models with integer variables that introduction of feasibility ensuring variables VQ may result in longer solution time. Additionally a solution that is not optimal may result.

The variables with names starting with VQ are declared also on IPLUSMINUS, see Section 3.8.9, 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 11.1.

8 Equations and constraints

The constraints in the GAMS model may be on the individual variables, cf. Section 7. 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"
QGFEQ(AAA,G,S,T) "Calculate fuel consumption, existing units (MW)"
QGFNEQ(AAA,G,S,T) "Calculate fuel consumption, new units (MW)"
QGKNSOLH(AAA,G,S,T) "Generation on new solarheat limited by capacity and sun (MW)"
QGKNWAVE(AAA,G,S,T) "Generation on new wavepower limited by cap and waves (MW)"
QFGEMINC(C,FFF) "Minimum electricity generation by fuel per country (MWh)"
QFGEMAXC(C,FFF) "Maximum electricity generation by fuel per country (MWh)"
QFGEMINR(RRR,FFF) "Minimum electricity generation by fuel per region (MWh)"
QFGEMAXR(RRR,FFF) "Maximum electricity generation by fuel per region (MWh)"
QFGEMINA(AAA,FFF) "Minimum electricity generation by fuel per area (MWh)"
QFGEMAXA(AAA,FFF) "Maximum electricity generation by fuel per area (MWh)"
QGMINCF(C,FFF) "Minimum fuel usage per country constraint (MWh)"
QGMAXCF(C,FFF) "Maximum fuel usage per country constraint (MWh)"
QGEQCF(C,FFF) "Required fuel usage per country constraint (MWh)"
QGMINRF(RRR,FFF) "Minimum fuel usage per region constraint (MWh)"
QGMAXRF(RRR,FFF) "Maximum fuel usage per region constraint (MWh)"
QGEQRF(RRR,FFF) "Required fuel usage per region constraint (MWh)"
QGMINAF(AAA,FFF) "Minimum fuel usage per area constraint (MWh)"
QGMAXAF(AAA,FFF) "Maximum fuel usage per area constraint (MWh)"
QGEQAF(AAA,FFF) "Required fuel usage per area constraint (MWh)"
QXMAXINV(IRRRE,IRRRI) "Limit of new transmission capacity (MW)"
QFMAXINVEST(C,FFF) "Limit on investment in capacity defined per fuel"
QGMAXINVEST2(C,G) "Maximum model generated capacity increase from one year to the next (MW)"
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"
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"
QHYRSMINVOL(AAA,SSS) "Hydropower reservoir - minimum level"
QHYRSMAXVOL(AAA,SSS) "Hydropower reservoir - maximum level"
QHYMAXG(AAA,SSS,TTT) "Hydropower - maximum generation"
QESTOVOLT(AAA,S,T) "Electricty storage dynamic equation (MWh)"
QESTOLOADTLIM(AAA,S,T) "Upper limit to electridity storage loading (model Balbase2 only) (MW)"
QHSTOVOLT(AAA,S,T) "Heat storage dynamic equation (MWh)"
QHSTOLOADTLIM(AAA,S,T) "Upper limit to heat storage loading (model Balbase2 only) (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"

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 GAMS User's Guide or McCarl GAMS User 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 7.

8.1 The Welfare Criterion

The default objective function of the Balmorel model is maximization of the welfare function. This is expressed in equation QOBJ.

Here a technical statement of a simplified version of this objective is indicated for a electricity-only system. In particular also the relation between welfare and actors' surplus is explained.

The Welfare Criterion - one electricity region

The criterion (to be maximized) for one electricity region a is consumers' utility minus producers' costs, i.e., the expression

$$D^a_{utility} - S^a_{cost}$$

This may be formulated as

$$= \qquad \int_0^{d^a} D^a(x)\mathrm{d}x - \int_0^{s^a} S^a(y)\mathrm{d}y$$

This expresses the difference between consumers' utility (function D^a related to demand) and producers' costs (function S^a related to supply). The quantity d^a bought by consumers is identical to the quantity s^a sold by producers, because with only one region no import or export is possible.

Introducing a market clearing price p^a this may be reformulated as

$$= \quad (\int_0^{d^a} D^a(x)\mathrm{d}x - p^a d^a) + (p^a s^a - \int_0^{s^a} S^a(y)\mathrm{d}y)$$

The two expressions are called consumers' surplus and producers' surplus (or producers' profit in this one-region case, although not in a multi-region case), i.e., the above expresses the criterion

$$= \quad \text{Consumers' surplus + Producers' surplus}$$

The Welfare Criterion - two or more electricity regions

With two regions a and b the criterion (to be maximized) is again consumers' utility minus producers' costs, i.e., the expression

$$D^a_{utility} - S^a_{cost} + D^b_{utility} - S^b_{cost}$$

However, the formulation in terms of surplus will be different, as shown here.

The net export from region a is $(s^a - d^a)$ which is equal to net import $(d^b - s^b)$ to region b.

Introducing a market clearing price in each region, p^a and p^b, and TSO's (bottle neck) surplus $(p^b - p^a)(s^a - d^a) = (p^b - p^a)(d^b - s^b)$, the above criterion may be rewritten as follows, indicating the contributions to each of the now tree types of actors,

$$\begin{aligned}
= \quad & \int_0^{d^a} D^a(x)\mathrm{d}x - \int_0^{s^a} S^a(y)\mathrm{d}y + \int_0^{d^b} D^b(x)\mathrm{d}x - \int_0^{s^b} S^b(y)\mathrm{d}y \\
= \quad & \{\int_0^{d^a} D^a(x)\mathrm{d}x - p^a d^a\} + \{p^a s^a - \int_0^{s^a} S^a(y)\mathrm{d}y\} + \{p^a(d^a - s^a)\} \\
+ \quad & \{\int_0^{d^b} D^b(x)\mathrm{d}x - p^b d^b\} + \{p^b s^b - \int_0^{s^b} S^b(y)\mathrm{d}y\} + \{p^b(d^b - s^b)\} \\
= \quad & D^a_{surplus} + S^a_{surplus} + D^b_{surplus} + S^b_{surplus} + (p^b - p^a)(s^a - d^a) \\
= \quad & D^a_{surplus} + D^b_{surplus} + S^a_{surplus} + S^b_{surplus} + (p^b - p^a)(s^a - d^a) \\
= \quad & \text{Consumers' surplus + Producers' surplus + TSOs' surplus}
\end{aligned}$$

Similar interpretations may be made with three or more regions.

The derivations above assume no taxes, subsidies, tariffs or similar, with such elements represented also the state (as responsible for taxes and subsidies), DSOs (distribution grid tariffs) and possibly others would be actors.

9 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 8 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 12.2.

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 7, 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 GAMS User's Guide or McCarl GAMS User Guide.

Choice of solver may be specified for each types of model (LP, MIP, etc.). Thus for instance the statement OPTION LP=CBC will specify to use solver CBC for LP problems. Other possibilities with open source and other solvers that come free with GAMS are OPTION MILP=CBC; OPTION RMIP=BDMLP; OPTION QCP=IPOPT; MIQCP=COUENNE; Consult GAMS solver manuals. Be aware that solution time and other elements of solvers and algorithms are highly dependent on the problem solved, hence no single solver or algorithm is superior for all problems, nor on all problems formulated with Balmorel.

An option file may be provided to some solvers to specify various solver specific details. Consult GAMS solver manuals. The solver option file must be located in the model folder. A non-zero number for optfile will specify that an option file shall be used. The solver option file shall with optfile set at 1 have name 'solver'.opt, where 'solver' is the name of the solver specified above. With a value higher than 1 another option file will be used, consult GAMS manuals.

Choice of solver and solve details are specified in balgams.opt.

9.1 Base models in Balmorel

Balmorel has defined a number of Base models, presently these:

1. BALBASE1: solving one whole year simultaneous for all the year's time segments
2. BALBASE2: solving one whole year with endogenous investments simultaneous for all the year's time segments
3. BALBASE3: solving one season with known capacities simultaneous for all time the season's time segments
4. BALBASE4: solving a sequence of years with endogenous investments simultaneous for all the years' time segments

Each of these models use some or all of the previously defined sets, parameters, variables and equations.

10 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.

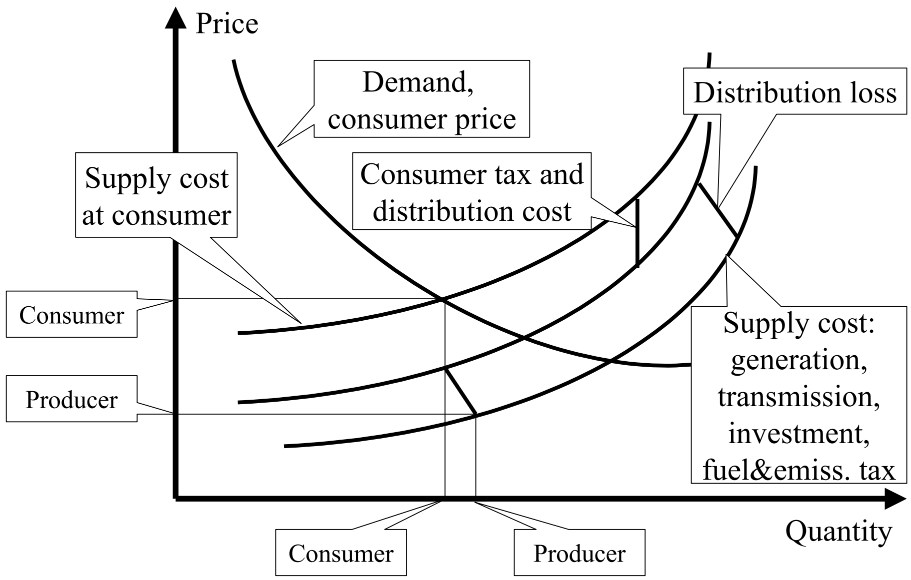

Figure 4: Illustration of elements in the calibration of demand functions

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: GKRATE, GEFFRATE, DEFP_CALIB and DHFP_CALIB. The basic information about these parameters is given in their respective sections, 4.24.2, 4.10.5, 4.18.2 and 4.17.7.

The first two are used to attempt consistency between fuel consumption as determined in the model and as specified in energy statistics, respectively. GKRATE 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. GEFFRATE 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.)

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 GKRATE then GEFFRATE and finally DEFP_CALIB and DHFP_CALIB.

10.1 Calibration of fuel consumption

10.1.1 Calibration of GKRATE

The calibration of GKRATE 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 GKRATE 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 GKRATE.

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 6: 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 6 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 7.

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

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 7: Results for Model 1

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 7.

Model 2:

The similar table for model 2 is shown in Table 8. Observe, that as the model is not actually stochastic the terms "probability" and "expected" are somewhat misleading.

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 8: Results for Model 2

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 9.

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 9: 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 $i \leq i_1$.

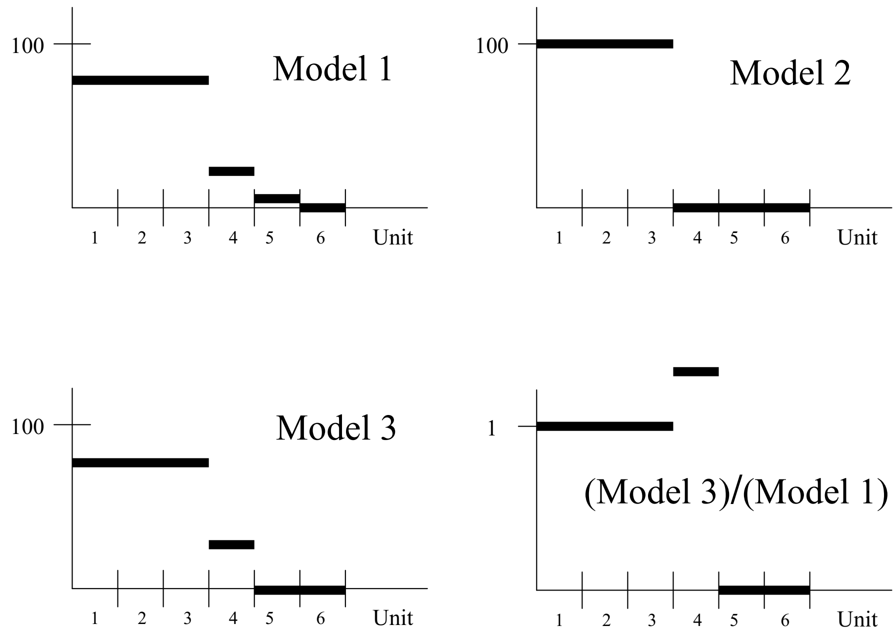

Figure 5: Expected generation (Models 1, 2 and 3) and comparison of results for Model 1 and Model 3

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, details may be found in Jørgensen and Ravn: Incorporation of Thermal Stochastic Elements Into a Hydro-Thermal Model, pp. 251-258 in Broch, Lysne, Flatabø and Helland-Hansen (eds.): Proceedings, Hydropower ’97, Trondheim 1997.

10.1.2 Calibration of GEFFRATE

Once GKRATE has been given its values, GEFFRATE 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 GKRATE, but note that also for wind power there is inter-dependency with GEFFRATE 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 GEFFRATE 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.
7. Exclude the possibility of new investments.
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 GEFFRATE 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 GEFFRATE 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 GEFFRATE 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.

11 Output

There are two types of output, that generated automatically by the GAMS system, and that generated by auxiliary parts of the Balmorel model.

11.1 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 GAMS User's Guide or McCarl GAMS User 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 12.1.

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 */, and ONEOLCOM makes it possible to add end-of-line comments, starting with !!.

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 GAMS User's Guide or McCarl GAMS User 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).

Some of the options related to printing to the lst file are i Balmorel handled in file balgams.opt.

The layout of the lst file may be controlled to some extent. For instance,

profile Get information on statement execution time and associated memory usage. Use command line with profile=1 (giving minimum) or profile=2 (maximum)

cerr Control the compile time maximum number of errors. Use command line with e.g. cerr=10 to stop compilation after 10 errors

pw Control the lst file page width. Use command line with e.g. pw=72 (minimum) or pw=32767 (maximum)

display Control the layout of sets and parameters display and the number of digits for parameters. Use e.g. in Balmorel.gms a line "OPTION GKFX:2:0:1"; DISPLAY GKFX;" to have the display of GKFX appear in list format with 2 decimals

See details in GAMS User's Guide or McCarl GAMS User Guide.

11.2 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').

A default format is applied but may be overwritten by the OPTION statement. One form of this is "OPTION DE:d:r:c;" to be placed before "DISPLAY DE;". In this notation: d is the number of decimals (neglected for sets), d∈[0..8]; r is the number of rows, r∈[0..(dimension-1)]; c indicates the number of columns across page. If r=0 the list format is used, and c is the maximum number of columns across page; if r ∈[1..(dimension-1)] r is the number of labels each row, and c must be ∈[1..(dimension-rows)]. See further GAMS User's Guide or McCarl GAMS User Guide.

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.

GDX facilities may also be used for exchanging data between GAMS and e.g. Microsoft Excel and Access, see further GAMS User's Guide or McCarl GAMS User Guide.

11.3 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 12.2.

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!

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.

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 10: Print files

A list of available print files is given in Table 10.

Declarations of the print files are given in file print1.inc. Additionally a number of identifiers are declared and defined for use in the print files: SCALAR OCASEID; PARAMETERS OCARDSETA, OCARDSETR; PARAMETERS OSCALAR1, OSCALAR2, OSCALAR3; SCALARS OTLW, OTLW1, OTLG, OTL1; ALIAS(S,OALIASS), ALIAS(T,OALIAST). See the file for details.

PRINTSEPARATOR in file Balopt.opt is for permitting insertion of separators like ';' in output for use with e.g. copy-paste to Excel.

12 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.

12.1 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 GAMS User's Guide or McCarl GAMS User Guide for further information.

12.2 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 11.3).

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 12.1.

12.3 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 12.1. The information in the output files described in Section 12.2 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 12.1.

If the execution of GAMS was successful, further information may be acquired by using the facilities provided in the auxiliary parts described in Section 12.2. 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 12.2. 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 12.2.

If the solution is not successful then study the error messages in the LOGFILE.OUT and in the Balmorel.lst files. See GAMS User's Guide or McCarl GAMS User 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.

13 Model variants and addons

The above description refers to the "standard" version of the Balmorel model. In the following, a number of obvious modifications will be described. Additionally documentation is given in Section 13.2 for the included addons.

13.1 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.

13.2 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 12.2, 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 3.8.7 - 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.

- 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 15). 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, in particular in relation to the addons. Section 13.2 next.

13.3 Addons in Balmorel

13.3.1 Introduction

Addons in the Balmorel model are pieces of code that provide extensions to the functionality for the core model version. They are by default excluded from being applied. However, the code is in the model, but is active only when decided to be so.

Code for the addons are found under folder addons in the base folder. The code is typically held in a number of files, and consists of declarations, definitions and code snippets. Each file is to be included at specific places in the main Balmorel code using the $include statement. The files are included according to the value of an associated option given in file balopt.opt.

Thus, to each addons there is an option with an option name as for instance heattrans for the heat transmission option. This option may be given a value using e.g.

```
$setglobal heattrans yes
```

in file balopt.opt. Anything else than *yes* will (for this addon and option) deactivate the addon, typically the empty value is used, like

```
$setglobal heattrans
```

Other addons may have more option values. (In the heattrans example above the option values are *yes* (for activating the addon) and *any other value* for no activation.)

Any data files for addons are to be placed in the data folder, like for all other data.

In Balmorel version 3.02 and earlier (illustrated above) the activation statement is placed in the main code (Balmorel.gms and bb123.sim), and it is clearly identified which addon the statement relates to (in the illustration above it is H2). Addons are produced regularly, and with and increasing number the need for more structured and tidy coding is required.

A new addon inclusion style is introduced with Balmorel version 3.03, it is, however, not implemented for all addons yet. The following is to be implemented in the coming Balmorel version 3.03. The addon folder get a folder called _hooks with files as illustrated here,

```
base
    addons
        _hooks
            balbase1.inc
            balbase2.inc
            balbase3.inc
            eqndecdef.inc
            error2.inc
            qeeq.inc
            qobj.inc
            setdeclare.inc
```

The content of the any file is like e.g. for error2.inc:

$ifi %COMBTECH%==yes $include "..\..\base\addons\combtech\combtech_error2.inc"

For e.g. setdeclare.inc:

$ifi %COMBTECH%==yes $include "..\..\base\addons\combtech\combtech_setdeclare.inc";

In the Balmorel.gms code i.a. the following lines are added at an appropriate places

$include "..\..\base\addons_hooks\setdeclare.inc"
$include "..\..\base\addons_hooks\setdeclare.inc"
$include "..\..\base\addons_hooks\qeeq.inc"

Note for the last one in particular that this line is placed as part of the definition of equation QEEQ. So the idea and is to make the main files Balmorel.gms and bb123.sim display less statements related to addons in order to achieve a more tidy code in these files. Moreover, any change to an addon, or addition of a new one, will only influence the content of the addons folder. The cost is that one level more of navigation is required to go from the model folder to the real addon code.

Presently not all the addons described here are available as part of version 3.03 (June 2018) of Balmorel. You may ask if there is a particular addon that you would like to use.

In any case, the available addons demonstrate a proven way of handling valuable extension to the main parts of the Balmorel model.

13.3.2 Heat transmission (Addon heattrans)

In the standard version of Balmorel the heat consumption and generation are specified at area level, and there is no possibility of transmitting heat from one area to another. The present section describes how such heat transmission may be introduced. The aim of the modification is to permit a modelling of district heating areas in more detail by e.g. sub-division of existing areas or linking of existing areas by transmission.

It will be done in a way which is similar to the way electricity transmission between regions is modelled in the standard version. Indeed, the main formal difference essentially is that electricity transmission code is duplicated where pairs of regions are replaced by pairs of areas, areas taking the place of regions, and the names of identifiers related to heat transmission contain the letter combination XH.

Therefore any issues related to the application of heat transmission and to interpretation of the results will be closely related to similar issues with electricity transmission.

The introduction of heat transmission has here been implemented using option heattrans.

The heat transmission is based on the identifiers shown in Table 11.

Variables and equation are

> POSITIVE VARIABLE VXH_T(IAAAE,IAAAI,S,T) "Heat export from area IAAAE to IAAAI (MW)"
> POSITIVE VARIABLE VXHKN(IAAAE,IAAAI) "New heat transmission capacity (MW)"
> EQUATION QXHK(IAAAE,IAAAI,S,T) "Heat transmission capacity constraint (MW)"

The various declarations and code pieces are $included at the relevant places in the main Balmorel code. Search "%heattrans%" in Balmorel.gms and bb123.inc.

Name	Domain	Type	Unit
IAAAE	alias AAA	set	-
IAAAI	alias AAA	set	-
IAE	alias IA	set	-
IAI	alias IA	set	-
IXHKINI_Y	(IAAAE,IAAAI)	parameter	MW
IXHKN	(IAAAE,IAAAI)	set	-
VXH_T	(IAAAE,IAAAI,S,T)	variable	MW
VXHKN	(IAAAE,IAAAI)	variable	MW
QXHK	(IAAAE,IAAAI,S,T)	equation	MW
XHCOST	(IAAAE,IAAAI)	parameter	Money/MWh
XHINVCOST	(IAAAE,IAAAI)	parameter	Money/MWh
XHKINI	(IAAAE,IAAAI)	parameter	MW
XHLOSS	(IAAAE,IAAAI)	parameter	(none)

Table 11: The identifiers used in representing heat transmission. Declaration and possibly code are found in the heat transmission addon subfolder. The last four identifiers are for data, the corresponding data files are to be found in the data folder.

Data files, cf. Table 11, are as usual to be found in the data folder.

The addon is activated by using

```
$setglobal heattrans yes
```

in file balopt.opt. Anything else than *yes* will deactivate the addon.

Note The possibility of heat transmission was estimated to be too specialized for inclusion in the original Balmorel version. With time, such refinements have been proved to be relevant, leading to the development of the addon idea. Had this been thought of originally, the names of the electricity transmission identifiers would have had an 'E', e.g. VXE_T rather than the present VX_T.

13.3.3 Combination technologies (Addon 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. Various maximum and minimum shares on the proportion attained by each unit in a combtech group may be specified.

There are two broad classes of application, distinguished by the way the primal technology (see below) is working. (I) The primal technology represents a traditional technology with generation (electricity and/or heat) based on fuel input with associated short and long term costs, efficiency, etc. (II) The primal technology represents a common constraint on secondary technologies. This may be on fuel input, e.g., capacity on a common coal mill or gas pipe. And/or it may be on output, e.g., limited capacity on a common electricity transformer or heat exchanger. The primal technology's long term costs (possibly zero) are related to the common restriction.

One example (A) 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 (B) is that one technology using coal is combined with one technology using gas and a third using straw. 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 straw can be at most 0.5, Further examples could specify minimum or maximum shares of total production from each of the combination technologies. In a third example (C) the shares of the individual technologies' fuel consumption out of total fuel consumption is limited by upper and/or lower bounds. In further examples, now illustrating class (II) above, the total fuel input (D) and/or generated output (E) may be limited as a share of rated primal capacity.

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. GDATASET should include the elements,

GDCOMB "Combination technology, primal (1) or secondary (2)"
GDCOMBGUP "Combination technology group's sum-of-generation's (MW) maximum share of primal rated capacity (default 1) ([0;1])"
GDCOMBGSHAREK1 "Combination technology, maximum generation (MW) share of primal rated capacity (default 1) ([0;1])"
GDCOMBFUP "Combination technology group's sum-of-fuel-use's (GJ) maximum share of primal rated capacity ([0;1])"
GDCOMBFSHAREK1 "Combination technology, maximum fuel (GJ) share of primal rated capacity ([0;1])"
GDCOMBGSHARELO "Combination technology, minimum generation (MW) share of total generation ([0;1])"
GDCOMBGSHAREUP "Combination technology, maximum generation (MW) share of total generation ([0;1])"
GDCOMBFSHARELO "Combination technology, minimum fuel (GJ) share of total fuel use ([0;1])"
GDCOMBFSHAREUP "Combination technology, maximum fuel (GJ) share of total fuel use ([0;1])"

data is to be entered in GDATA, with the following meaning (indicated for some only).

The following from the input in GDATA is used for identification of combination technologies, and combtech groups,

- 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,GGG) where the pair (GGG,GGG) constitutes a combination technology combination, GGG (index 1) must be a primary technology and GGG (index 2) must be a secondary technology. GGG and GGG 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, GGG)/ 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 (primal or secondary) can be in only one combtech group.
- 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 following from the input in GDATA is used for definition of equations,

- The value GDATA(GGG,'GDCOMBGSHAREK1') specifies the maximum fraction (values in (0;1]) of rated primal installed capacity of a combination technology that this technology can produce any (S,T). The fraction is defined with respect to VGE_T if GGG is in IGKE, VGH_T if GGG is in IGHE (Example A, class (I) above). Similarly the value GDATA(GGG,'GDCOMBFSHAREK1') is used, now relating to the fuel input (Example

C, class (I) above). The value zero (or no entry) for GDATA(GGG,'GDCOMBGSHAREK1') and GDATA(GGG, 'GDCOMBFSHAREK1') will be interpreted to mean that this constraint is irrelevant (see EPS below).

- The value GDATA(GGG,'GDCOMBGSHARELO') specifies the minimum fraction (value between 0 and 1) that the production from technology GGG must any (S,T) 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 (Example B, class (I) above). Similarly the value GDATA(GGG,'GDCOMBFSHARELO') is used, now relating to the fuel input. GDATA(GGG,'GDCOMBGSHAREUP') and GDATA(GGG, 'GDCOMBFSHAREUP') are used with analogue meaning (Examples B, class (I) above).
- If a combtech value (other than GDCOMB) in GDATA is 0 (or the entry is empty) it is here interpreted to mean that the corresponding equation (cf. below) is irrelevant. If is EPS the value will be interpreted to mean that the corresponding equation is relevant and the value is zero. If this is applied to e.g. GDATA(GGG,'GDCOMBGSHARELO') it implies that it cannot generate any output. In particular for GDATA(GGG,'GDCOMBGSHAREK1') with a primal technology this means that there will be no generation, however, be there will a capacity, and it will be possible to limit total input or output for all secondary technologies (Examples D and E, class (II) above).

Here are some details on the internal mechanisms.

- SET IGCOMB1(G): The set of primary combination technologies in G. Defined as IGCOMB1(G)=YES$(GDATA(G,'GDCOMB') EQ 1).
- SET IGCOMB2(G): The set of secondary combination technologies in G. Defined as IGCOMB2(G) = YES$(GDATA(G,'GDCOMB') EQ 2).
- The equation QECOMBGSHAREK1(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 rated capacity. Similarly, the equation QHCOMBGSHAREK1(AAA,IGKH,S,T) applies to total heat production. For the total fuel input QCOMBFSHAREK1 is applied similarly. As noted above, the input value zero implies that no constraint is generated while the value EPS implies that there will be a constraint limiting fuel input and/or heat and electricity output to zero for the primal technology.
- The equation QECOMBGSHARELO(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,'GDCOMBGSHARELO'). Similarly for equations QHCOMBGSHARELO, QNECOMBGSHARELO and QNHCOMBGSHARELO.
- The equation QECOMBGSHAREUP(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,'GDCOMBGSHARELO').
- As usual, specification of possibility of investment are given in GDATA and AGKN for primary and secondary technologies individually,

Similarly for other equations.

Here are equations for existing technologies, illustrated for fuel and electricity only,

QCOMBFUP(AAA,G,S,T) "Existing combtech, sum of fuel (GJ) use any (S,T) limited by rated capacity (MW) on primal (0 means irrelevant) (GJ)"
QECOMBGUP(AAA,G,S,T) "Existing combtech, sum of production (MW) any (S,T) limited by rated capacity on primal, electricity (0 means irrelevant) (MW)"
QECOMBGSHARELO(AAA,G,S,T) "Existing combtech in IGKE, maximum share any (S,T) of total generation for this combtech group, electricity (0 means irrelevant) ([0;1])"
QECOMBFSHARELO(AAA,G,S,T) "Existing combtech in IGKE, minimum share any (S,T) of total fuel use for this combtech group, electricity (0 means irrelevant) ([0;1])"
QECOMBGSHAREUP(AAA,G,S,T) "Existing combtech in IGKE, maximum share any (S,T) of total generation for this combtech group, electricity (0 means irrelevant) ([0;1])"

Where possible, equations are substituted b assignments of lower and/or upper bounds.

Observations, possible errors:

- Observation: If there are only two technologies in a pair it seems redundant to specify both GDATA(*,'GDCOMBGSHARELO') and GDATA(*, 'GDCOMBGSHAREUP') for both technologies, since one can be derived from the other. In this case it is possible to either specify GDATA(*,'GDCOMBGSHARELO') as 0 for both technologies and the relevant values for GDATA(*,'GDCOMBGSHAREUP'), or specify GDATA(*, 'GDCOMBGSHAREUP') as 1 for both technologies and the relevant values for GDATA(*,'GDCOMBGSHARELO'). This will be beneficial for computation and interpretation of results.

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".

13.3.4 Price elastic demand quantities (Addon DFLEXQUANT)

The functionality in this addon has been in Balmorel all the time as an element in the base model.

It has now been decided to handle it as an addon. This is implemented for Balbase4, however, the present documentation has not been updated accordingly. Also, this documentation does not presently reflect that demand flexibility now is specified for demand groups.

Please search this documentation for words DEF, DEF_D1, DEF_U1, DF_QP DEF_STEPS, and similar for heat side DHF, DHF_D1,DHF_U1, DF_QP DHF_STEPS. There is also relevant material in Section 13.3.5.

13.3.5 Calibrating price side of electricity demand (Addon DEFPCALIB)

In case of price elastic (flexible) electricity demand it may be convenient to be able to calibrate the price level. This may be accomplished by using addon DEFPCALIB.

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 for electricity done by modifying the base price IDEFP_T by DEFP_CALIB; for heat it may similarly be done by modifying the base price IDHFP_T by DHFP_CALIB, Section 13.3.6.

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.

There are some other mechanisms for calibration as well, cf. Section 10.1.2 and Section 10.1.1, see also Figure 4 page 81.

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. Exclude the possibility of transmission
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.

13.3.6 Calibrating price side of heat demand (Addon DHFPCALIB)

In case of price elastic (flexible) heat demand it may be convenient to be able to calibrate the price level. This may be accomplished by using addon DHFPCALIB. It is done in line with similar calibration for electricity, therefore see Section 13.3.5.

13.3.7 Hydrogen (Addon H2)

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. : addon H2

(Note that there is no interference between this addon and the mentioning of hydrogen in relation to electricity storage and equation QHEQ in Section 3.3.2).

13.3.8 Price dependent electricity exchange with third countries (Addon X3V)

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, Section 4.11.1, or without it. The price-quantity relationships are given as a piecewise step function. There are card(X3VSTEP) 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 of each export step. The associated prices are X3VIMP and X3VIMP, respectively. The prices are given on a yearly basis, the value for the currently simulated year are held in IX3VPIM_Y, and IX3VPEX_Y. 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. The simulated price dependent electricity exchange transmission connections are specified in the set X3VX: The set RX3VSUBSTI may be used to reduce the risk of user errors.

Associated variables are VX3VIM_T and VX3VEX_T.

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	Page
IX3VPIM_Y	(RRR,X3VPLACE0,X3VSTEP0,S,T)	parameter	Money/MWh	100
IX3VPEX_Y	(RRR,X3VPLACE0,X3VSTEP0,S,T)	parameter	Money/MWh	100
RX3VSUBSTI	(RRR,X3VPLACE0)	set	-	99
X3VPEX	(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	100
X3VPIM	(YYY,RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	100
X3VPLACE0	-	set	-	99
X3VQEX	(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	X3.INC	100
X3VQIM	(RRR,X3VPLACE0,X3VSTEP0,SSS,TTT)	parameter	X3.INC	100
X3VSTEP0	-	set	-	99
X3VSTEP	(X3VSTEP0)	set	-	99
X3VX	(RRR,X3VPLACE0)	set	-	99
VX3VIM_T	(RRR,X3VPLACE0,X3VSTEP0,S,T)	variable	MW	100
VX3VEX_T	(RRR,X3VPLACE0,X3VSTEP0,S,T)	variable	MW	100

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 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 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).

X3VPLACE0

The set of regions with which there can be price dependent electricity exchange 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.

X3VX

The combinations of RRR and X3VPLACE0 that are to be simulated for price dependent electricity exchange 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. It indicates (by assigning YES) if elements in X3VPLACE0 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 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,X3VPLAC hold the prices for price dependent electricity exchange with third countries 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.

A X3V-specific equation QX3VBAL(X3VPLACE) is applied according to sub-control X3VfxQ value.

Options

The addon is controlled by option variable X3V. The addon is activated by $setglobal X3V yes.

There are sub-controls X3VfxQ, X3VfxP, X3VPRICECASE. If X3VfxQ is set at yes the equation QX3VBAL is active.

13.3.9 Discrete Size Investments in Technology (Addon AGKNDISC)

Introduction

In the base version of Balmorel, endogenous investment in technology capacity is specified by a continuous variable VGKN. One consequence of the continuity is that unrealistically sized (e.g., very small) investments may be found. Although this may be acceptable in some contexts, it may be undesirable in others.

The addon AGKNDISC permits specification that for any technology the investment must be in one out of a pre-specified number of sizes, e.g. either 200, 500 or 700 MW (or none at all). This condition may apply to some or all of the technologies.

Input data

The technologies have the same data (given i GDATA) irrespective if they are to be invested with discrete or continuous sizes. Those that are to be invested with discrete sizes have additional data as described in the sequel.

Set AGKNDISCAG

SET AGKNDISCAG(AAA,GGG) 'Areas for possible location of discrete capacity investments in technologies'. Input data.

Set AGKNDISCAG is supposed to be a subset of AGKN.

If a particular technology 'Gx' has been defined as discrete size investments in an area 'Ay', it will automatically be impossible to have continuous size investments in this combination ('Ay','Gx') of area and technology.

Note that the condition of discrete investment is on the combination (AAA,G). This means that a particular technology 'Gz' may be permitted continuous size endogenous investments in some area 'Aw' while being conditioned to discrete size investments in others.

Set AGKNDISCGSIZESET

SET AGKNDISCGSIZESET 'Set of possible sizes for discrete capacity investments in technologies'. Input data.

Define for instance "SET AGKNDISCGSIZESET / AGKNDISCGSIZE1, AGKNDISCGSIZE2, AGKNDISCGSIZE3 /;" to permit up to three different sizes (in addition to the no investment possibility).

Set AGKNDISCGDATASET

SET AGKNDISCGDATASET 'Technology investment data types for discrete capacity size investments'. Input data.

This set specifies three obligatory elements:

AGKNDISCSIZE 'Size (MW)'
AGKNDISCINVCOST 'Investment cost (MMoney)'
AGKNDISCOMFCOST 'Annual operating and maintenance costs (MMoney)'

Note that the units of the two latter elements are MMoney (million money), not Money/MW or similar.

Note that the data is independent of the area. If this is unsatisfactory, define two or more technologies that differ only in this respect and use AGKNDISCAG to specify the possible locations of each of them.

Parameter AGKNDISCGDATA

PARAMETER AGKNDISCGDATA(GGG,AGKNDISCGSIZESET,AGKNDISCGDATASET) 'Technology investment data for discrete capacity size investments'. Input data.

This parameter specifies for each technology and each element in AGKNDISCGSIZESET the values for each element in AGKNDISCGDATASET.

If 0 (or nothing) is specified as value for 'AGKNDISCSIZE' for a certain element in AGKNDISCGSIZESET this means that this element in AGKNDISCGSIZESET is not considered an option (thus, the number of elements in AGKNDISCGSIZESET indicates the maximum possible number of discrete sizes (not counting 0 MW), the actual number used depends on the individual technology).

Internals

Set IAGKNDISCAG

SET IAGKNDISCAG(AAA,G) 'Area, technology for discrete size investment, where technology may be invested based on AGKN and implicit constraints'. Internal.

This internal set is very much similar to IAGKN. And similarly to the AGKNDISCAG/AGKN pair, it is supposed to be a subset of IAGKN.

Parameter IAGKNDISCDIFFCOST

PARAMETER IAGKNDISCDIFFCOST(AAA,G,AGKNDISCGSIZESET, AGKNDISCGDATASET) 'Investment costs in relation to discrete capacity size investments, as difference for each size (Money) (Addon AGKNDISC)'. Internal.

One characteristic of the economy related to endogenous investments in the base version of Balmorel is that the cost associated with an investments is proportional to the size of the invested capacity (it is a linear model). In contrast, the cost related to discrete size investments need not be proportional to the size, since the cost is specified individually for each element in AGKNDISCGSIZESET.

The AGKNDISC addon uses the same variables VGKN to represent the endogenous investment as does the base version of Balmorel, and there is therefore a proportional cost term in the objective function QOBJ. This term is eliminated through the construction of IAGKNDISCDIFFCOST:

```
IAGKNDISCDIFFCOST(IA,G,AGKNDISCGSIZESET, AGKNDISCGDATASET)
$AGKNDISCAG(IA,G) =
AGKNDISCGDATA(G,AGKNDISCGSIZESET,'AGKNDISCINVCOST')
- AGKNDISCGDATA(G,AGKNDISCGSIZESET,'AGKNDISCSIZE')*GOMFCOST(IA,G)
- AGKNDISCGDATA(G,AGKNDISCGSIZESET,'AGKNDISCSIZE')*GINVCOST(IA,G);
```

As seen, in IAGKNDISCDIFFCOST the proportional terms are subtracted from the discrete size related terms in AGKNDISCGDATA. It is noted that VGKN can take only values AGKNDISCGDATA(G,AGKNDISCGSIZESET,'AGKNDISCSIZE') (or zero), see equation QAGKNDISCCONT.

IAGKNDISCDIFFCOST (multiplied by VGKNDISC) is added to the objective function.

Variables

VGKNDISC

BINARY VARIABLE VGKNDISC(AAA,G,AGKNDISCGSIZESET) 'New generation capacity in relation to discrete capacity size investments (binary)'.

Note that the variables VGKNDISC may be considered as being internal, they will not normally be of interest once the solution is found. The endogenously found technology capacity is given by VGKN.L also in case of discrete size endogenous investments.

Equations, model and solve

Equation QAGKNDISC01

QAGKNDISC01(AAA,G) 'At most one of the specified discrete capacity size investments is chosen (0/1)'

This equation ensures that at most one of the binary values in VGKNDISC(*,*,AGKNDISCGSIZESET) can take the value 1.

Equation QAGKNDISCCONT

QAGKNDISCCONT(AAA,G) 'The invested capacity must be one of the specified sizes or zero (MW)'

This equation ensures that VGKN can take only values specified in AGKNDISCGDATA(*,*,'AGKNDISCSIZE').

Model and solve

The equations are to be included in model Balbase2.

The model is of type MIP (mixed integer programming), therefore the solve statement must specify "using MIP". This is ensured by the control setting "$Setglobal SOLVEMIP yes". It is of course assumed that the users has a solve capable of solving MIP problems.

Error, printing and similar

Some output is printed to file /printout/gkn_ag.out.

File, folder and include structure

The new code input is held in a number of files located in folder /addon/AGKNDISC/. Data input is found in files agkndiscag.inc, agkndiscgdataset.inc, agkndiscgdata.inc, agkndiscgsizeset.inc in the data folder.

Output may be printed to file /printout/gkn_ay.out from file /printinc/gkn_ay.inc.

Variables, equations and other internal items are in folder /addon/agkndisc/.

The files are included into the model at appropriate places by statements of the form "$ifi %AGKNDISC%==yes $include '...';".

Addon controls

The application of the addon is controlled by $Setglobal AGKNDISC, where a yes specifies that the addon be applied.

Relevant only for BB2; applicastion to BB4 should be possible, but is not implemented presently.

See the comments on model type MIP above.

Some observations and interpretations

Calculation time

The problem to be solved is of the MIP type, it is well known that in general this may imply long calculation time.

Prices

Key concepts to be considered about this (and any other) investment model are the short run marginal costs (SRMC) and the long run marginal costs (LRMC). The point to be made here is that standard solvers for MIP problems for derivation of the electricity price will provide only values similar to the SRMC. Further, there seems to be no other formulation that is suitable for standard solvers and that will indicate the LRMC.

Consider an example given in relation to investment in an electricity only unit, however, the same considerations apply for heat only and chp types of units.

Assumptions are as follows. SRMC: 100 Money/MWh; investment size: 100 MW; investment annuity cost for 100 MW: 5400000 Money; to be distributed over 1000 hours during the year of investment; demand to be covered by this unit each of the 1000 hours: 90 MW. It is further assumed that the unit is marginal (viz., price setting) in the hours considered.

This implies that each hour the consumption has to 'pay' 5400 Money in order to cover the investment annuity cost, or 5400/90 = 60 Money/MWh. Thus, in order to get exactly covered the costs, the electricity price should be SRMC+60 = 160 Money/MWh. However, a standard solver will usually give the value 60, i.e. the SRMC.

A tiny GAMS model that may verify this example is given here.

```
* Begin model

$Title Discrete size investments - illustration of prices

SCALAR SRMC "Variable cost (Money/MWh)" /100/;

SCALAR INVCOSTANNUITY "Annual invest. cost each relevant hour (Money)" /5400/;

SCALAR CAPACITY "Capacity considered invested (MW)" /100/;

SCALAR DEMAND "Fixed demand (MW)" /90/;

FREE VARIABLE VOBJ "Objective function variable: costs (Money)";

POSITIVE VARIABLE VGEN_T "Generation (MW)";

BINARY VARIABLE VGKNDISC "Investment decision (0/1)";

EQUATION QOBJ "Objective function equation: costs (Money)";

EQUATION QEEQ "Electricity balance (MW)";

EQUATIONS QCAPACITY "Investment must take place if production is positive";

QOBJ.. VOBJ =E= VGEN_T*SRMC + VGKNDISC*INVCOSTANNUITY;

QEEQ.. VGEN_T =E= DEMAND;
```

QCAPACITY.. VGEN_T =L= CAPACITY*VGKNDISC;

MODEL DISCRETESIZEINVESTMENT /ALL/;

SOLVE DISCRETESIZEINVESTMENT MINIMIZING VOBJ USING MIP;

DISPLAY VGEN_T.L, VGKNDISC.L;

DISPLAY "Intuitively this would be the electricity price (Money/MWh):", QEEQ.M;

* End model

A number of alternatives may be considered.

Alternative 1. Do not use the agkndisc addon. This will give the LRMC in the LP model, but will not in general give a size of new investment that comply with any specific discrete size.

In relation to the above example observe the following for a LP formulation . The quantity invested will be 90 MW (not 100). The investment related cost to be covered by each relevant hour is 5400*(90/100)/90 = 54 Money/MWh. The average cost is then 154 Money/MWh, and this is also the marginal cost, i.e., the LRMC. A standard solver will for this LP model usually return the expected 154. This may be verified by substituting "using MIP" by "using RMIP" (this will then result in a LP model) in the solve statement in the above model and then resolving.

Alternative 2. As Alternative 1, but then round down the found investment sizes to the nearest acceptable discrete size. This may resolve the problem (i.e., it will give LRMC and discrete sizes), however, it will not in general ensure that the capacities provide a feasible solution (due to shortage of capacity).

Alternative 3. As Alternative 1, but then round up the found investment sizes to the nearest acceptable discrete size (which has to be assumed to exist). This will give the LRMC and discrete sizes. But if the full capacity is not exploited in the peak load hour the units will not have their investments' annuity costs covered at this value of LRMC. Moreover, it may be argued (if the investments are assumed to be distributed over efficiently competing companies) that it may only be possible to achieve the SRMC due to competition, because there is over-capacity.

So, in general it seems that either quantities (viz., discrete ones) or prices (viz., LRMC) will be intuitive and appropriate, but not both in the same model.

13.3.10 Seasonal hydro amounts in BB3 (HYRSBB123)

Balmorel's standard model BALBASE1 and BALBASE2 permit handling of limitation on available annual amounts of hydro power. Model BALBASE3 works on a number of seasonal models, and there is no direct way of handling annual amounts of hydro power in BALBASE3. Addon HYRSBB123 provides a mechanism for transferring seasonal information obtained in BALBASE1 or BALBASE2 for use in BALBASE3.

In the optimization literature this would typically be described as decomposition: a larger model is attempted solved by solving a sequence of smaller linked models (BALBASE3). Further, the two main types of decomposition are primal and dual decomposition, the former referring to use of quantities, the latter referring to use of prices. Both primal and dual decompositions are implemented in Balmorel as well as a combination.

The control variable is HYRSBB123 with options:

> *!option quant "Use fixed seasonal quantities of hydro generation (from WATERVOL.gdx) in BB3"
>
> *!option price "Use fixed seasonal prices for hydro generation (from WATERVAL.gdx) in BB3"
>
> *!option quantprice "Use seasonal quantities and prices for hydro generation (from WATERVOL.gdx and WATERVAL.gdx) in BB3"

The quant option aims at ensuring that the quantities found in BALBASE1 or BALBASE2 will be reproduced in the BALBASE3 solution. Often this will if fact happen, however in a number of cases infeasibilities occur in BALBASE3 due to the finer time resolution within the year (which seem to require more hydro power in certain seasons).

The price option aims at ensuring that the prices for hydro power found in BALBASE1 or BALBASE2 will be reproduced in the BALBASE3 solution. As has been observed so far this also happens. A drawback of this option as that the annual amount of hydro power used in BALBASE3 may exceed that of BALBASE1. This may also happen relative to the BALBASE2 solution, however, here it may also happen that less hydro power will be used, because the water values (hydro power prices) obtained from BALBASE2 reflect also marginal investment costs, while investments are not included in BALBASE3.

The quantprice option combines elements from the quant and price options. Hydro power quantities aim at being the same in BALBASE3 as in BALBASE1 or BALBASE2, but if this results in infeasibilities more water is allocated that season; the cost of this water is somewhat higher than the water values found from BALBASE1 or BALBASE2. Thus, infeasibilities are avoided. A drawback of this option as that the annual amount of hydro power used in BALBASE3 may exceed that of BALBASE1 or BALBASE2.

13.3.11 Bypass of electricity production of CHP units (Addon bypass)

This document describes modelling of options for energy generation technologies of types back-pressure and extraction. The first section on backpressure is described in OptiWaste terms, while the second section on extraction is described in Balmorel terms. The notation applied is inconsistent.

Implementation was made by Amalia Rosa Pizarro Alonso and Hans Ravn 2015-2016.

Bypass of electricity for Backpressure type units

For the TopWaste project's OptiWaste model the possibility of making bypass of the turbine, i.e. the electricity generation, on backpressure type (BPR) units is implemented. Energy technologies in OptiWaste are modelled in detail and understanding as in Balmorel but within the OptiWaste framework.

Operation region

Heat generation is V_h, electricity generation is V_e. Relevant slopes are given by constants CB (positive, specifying back-pressure line slope) and BYPC (positive, specifying iso-fuel slope), implemented in GDATASET and GDATA.

Standard relation between V_h and V_e, specifying the operating region is, cf. the solid line segment in Figure 6,

$$V_e = CB * V_h \tag{4}$$

The operating region with electricity bypass is the triangle limited by the H-axis, the CB-line and the BYPC-line. Hence, (4) is replaced by

$$V_e \leq CB * V_h \tag{5}$$

and the BYPC-line limits is

$$V_e \leq K_{BYPC} - BYPC * V_h \tag{6}$$

Derivation of K_{BYPC}, cf. Figure 6,

$$K_{BYPC} = K_e(1 + \frac{BYPC}{CB}) \tag{7}$$

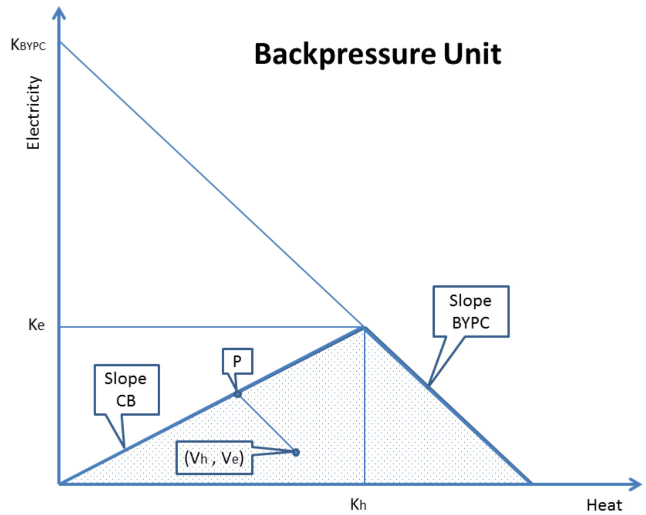

Figure 6: The backpressure unit, operating region without (line segment with slope CB) and with bypass option (dotted triangle).

Fuel consumption

For point (V_h, V_e) on Figure 6 the fuel consumption is found by first translating the point to the CB-line along the iso-fuel line with slope BYPC to point $P = (h, e)$ and then using the standard relation for fuel consumption applied to point P.

Equations for identification of $P = (h, e)$ are (4) and

$$(V_h - h)BYPC + V_e = e \tag{8}$$

Combining (4) and (8) yields after a little manipulation

$$h = \frac{V_h BYPC + V_e}{CB + BYPC} \tag{9}$$

$$e = CB \frac{V_h BYPC + V_e}{CB + BYPC} \tag{10}$$

The fuel consumption for any point (V_h, V_e) is found from the standard relation for fuel consumption applied to point $P = (h, e)$ which depends on (V_h, V_e) through relations (9) - (10).

Bypass of electricity for Extraction type units

For the Extraction type unit the electricity bypass option is modelled as follows, cf. Figure 7.

The following uses positive coefficients c_b, c_v, η and K_e, where the former two are the slopes of the non-vertical part of the feasible region, the third one is the energy efficiency, and K_e is the capacity (specified as maximum electricity output), to specify the following relations between the non-negative variables e (electricity generation), h (heat generation) and F (fuel input):

$$e \geq hc_b \tag{11}$$

$$e \leq K_e - hc_v \tag{12}$$

$$F = \frac{e + hc_v}{\eta} \tag{13}$$

$$e \leq K_e \tag{14}$$

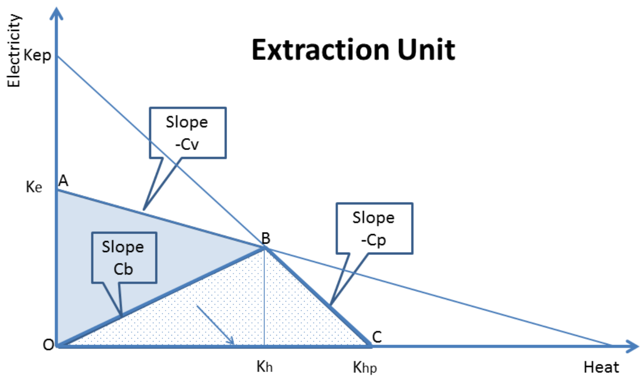

Figure 7: The Extraction unit, operating region without (shaded triangle OAB) and with bypass option (solid plus dotted triangles, i.e. OABC).

Figure 7 illustrates the feasible region of combinations of e and h based on (11) - (12), combined with the assumption that e, h and F are all non-negative, without and with en electricity bypass option.

(Note: To model this Balmorel system in OptiWaste some manipulation is needed. Combining (12) and (13) yields

$$F \leq \frac{K_e}{\eta} \tag{15}$$

This provides the OptiWaste capacity bound that substitutes the Balmorel capacity bound (14). Note that while the capacity size in Balmorel is relative to an outFlow it is in OptiWaste relative to the inFlow.)

Allowing bypass of electricity increases the feasible region by a triangle according to the line with the bypass slope $-c_p$ (c_p is positive, typically 1), implying the relation

$$e \leq K_{ep} - hc_p \tag{16}$$

for some K_{ep} (cf. (21)). This extends the heat capacity from K_h to K_{hp}, cf. Figure 7. The derivation of K_{hp} is as follows. First observe that the maximum heat production K_h and the corresponding electricity production e_{kh} are derived as

$$K_h = \frac{K_e}{c_b + c_v} \tag{17}$$

$$e_{Kh} = \frac{K_e c_b}{c_b + c_v} \tag{18}$$

Then heat capacity K_{hp} with bypass option is found by following the line with slope $-c_p$ from the found point (17) - (18) to intersection with the heat axis, this gives

$$K_{hp} = \frac{K_e(1 + c_b)}{c_b + c_v} \tag{19}$$

The line intersecting the heat axis at K_{hp} intersects the electricity axis at $K_{ep}c_p$, i.e., K_{ep} in (16) is $K_{hp}c_p$ (cf. also (21)).

The feasible regions of operation for the Extraction technology with electricity bypass option now consists of two separate regions. Region R_1 is the original no-bypass regions given by (11) - (12) while the bypass region R_2 is given by

$$e \leq hc_b \tag{20}$$

$$e \leq K_{ep}c_p - hc_p \tag{21}$$

The line segment along $e = hc_b$, cf. (11) and (20), is common between the two regions.

To derive the fuel use it is noted that it is linear in e and h in the two separate regions, but non-linear on the whole region. The fuel use F_1 in region R_1 is given in (13).

The fuel use F_2 in region R_2 may be derived as follows. First note that the fuel use in the point with maximum heat production in the non-bypass situation (cf. (17) - (18)) is K_e/η. The fuel use in the point with maximum heat production in the non-bypass situation is also K_e/η (the two points are on the same isofuel line in R_2). Then for any point (h, e) in Region R_2 the fuel use is the same as for the point $(h + e/c_p, 0)$ on the heat axis (the two points are on the same isofuel line in R_2). Finally observe that due to the linearity in region R_2 the fuel use F_2 in this point $(h + e/c_p, 0)$ is in the same proportion to the fuel use in the bypass maximum heat point (17) - (18) as the heat $h + e/c_p$ in this point $(h + e/c_p, 0)$ is in proportion to the heat $\frac{K_e}{C_b+c_v}$ in the bypass maximum heat point (17) - (18), i.e.,

$$\frac{F_2}{K_e/\eta} = \frac{h + e/c_p}{\frac{K_e(1+c_b)}{c_b+c_v}} \tag{22}$$

implying

$$F_2 = \frac{h + e/c_p}{\eta} \frac{c_b + c_v}{1 + c_b} \tag{23}$$

or in particular with $c_p = 1$

$$F_2 = \frac{h + e}{\eta} \frac{c_b + c_v}{1 + c_b} \tag{24}$$

The fuel consumption F on the whole operating region $(R_1 + R_2)$ is now given as the maximum of the fuel use in any of the two regions as expressed individually by (13) and (23). Hence F is given by the non-linear expression $F = \max(F_1, F_2)$, or

$$F = \max(\frac{e + hc_v}{\eta}, \frac{h + e/c_p}{\eta} \frac{c_b + c_v}{1 + c_b}) \tag{25}$$

This may be exploited in a model that minimizes fuel consumption (either directly or through minimization of e.g. fuel cost); e.g. in Balmorel and OptiWaste. A sketch of such application is given as this:

$$\text{minimize} \quad F \tag{26}$$

$$F \geq \frac{e + hc_v}{\eta} \tag{27}$$

$$F \geq \frac{h + e/c_p}{\eta} \frac{c_b + c_v}{1 + c_b} \tag{28}$$

subject further to conditions (12), (21) and the non-negativity of e and h.

To complete the analysis the following demonstrations should be made, this is left as an exercise for the reader. (a) The combined bypass region $R_1 + R_2$ is convex if and only if $c_v \leq c_p$. (b) The correct fuel use is obtained using $F = \max(F_1, F_2)$ if and only if $c_v \leq c_p$. (c) The function $\max(F_1, F_2)$ is convex (while $\min(F_1, F_2)$ is not, indeed it is concave). (d) The above optimization model is valid for obtaining the desired fuel use.

Implementation in Balmorel

The relevant code is found in files gdataset.inc, gdata.inc, Balmorel.gms and balopt.opt.

In GDATASET.inc: Element GDBYPC "Bypass coefficient (non-negative; 0 means no bypass possibility)". The data for the technology is to be given in GDATA.inc.

In Balmorel.gms:

SET IGBYPASS(G) 'Technologies that may apply turbine bypass mode (subject to option bypass)';

Equations that are are added to relevant models:

QGCBGBPR_BYPASS1(AAA,G,S,T) 'CHP generation (back pressure) with bypass limited by Cb-line (MW)'

QGCBGBPR_BYPASS2(AAA,G,S,T) 'CHP generation (back pressure) with bypass limited by BYPC-line (MW)'

QGNCBGBPR_BYPASS1(AAA,G,S,T)'CHP generation (back pressure) with bypass limited by Cb-line, new (MW)'

QGNCBGBPR_BYPASS2(AAA,G,S,T)'CHP generation (back pressure) with bypass limited by BYPC-line, new (MW)'

The use of bypass mode is subject to addon bypass, with option value "yes" activating the bypass mode possibility.

Note that for each technology GDBYPC in GDATA specifies if bypass may be relevant, and if it is, then the technical value is specified. Also note, that bypass mode is applied subject to the value of option bypass. Hence, bypass mode is only applied for a technology if both the GDBYPC value and the option value permit it.

Status: Implementation for the backpressure technology type is under testing. Implementation for the extraction technology type is considered.

13.3.12 Time aggregation (Addon timeaggr)

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

The addon timeaggr permits handling of time aggregation in a more elaborate way, overcoming some of the shortcomings of simple selection of representative S and T. Presently no documentation is available.

Storages provide a particular challenge in relation to time aggregation. A discussion is provided in the following.

13.3.13 Renewable energy share of electricity (Addon REShareE)

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03. : REShareE

13.3.14 Rolling horizon models (Addon BB4)

Introduction

The purpose of addon BB4 is to permit a model with endogenous investments that (in contrast to Balbase2) treats two or more years in an integrated fashion.

The first implementation of investment horizon functionality for the Balmorel model was made by Jesper Felstedt and Morten Middelboe Pedersen as Master's thesis "Modellering af Investeringer i Elsektoren", at IMM, Technical University of Denmark, 2005, under supervision by Hans Ravn. An enhanced version was presented in "Dynamic Power System Investment Modeling and Analysis" by Hans Ravn at the Risø International Energy Conference 2011. That implementation has been adapted for the present version of Balmorel.

Name	Domain	Type	Unit	Short description	Page
YMODEL	-	SET	-	Set of years in each model (not subset of Y)	111
YMODELDELTA	-	SET	-	Num. years between two year in each model	111
DISCOUNTRATE	-	SCA	frac	Discount rate	112
ANNUITYC	CCC	PAR	frac	BB4: See text	112
IY411	Y	SET	-	The years in current Balbase4 model	
IY410	Y	SET	-	Like IY411 except last year	
IY401	Y	SET	-	Like IY411 except first year	
IY411FIRST	Y	SET	-	First year in the current Balbase4 model	
IYFIRST	Y	SET	-	First year in Y	
IY411PREVY	(Y,Y)	SET	-	Previous Y (idx 2) to Y (idx 1) in IY411	
IY4XINVCOSTWEIGHT	Y	PAR	?	Discounted future annuity trans. payments	
IFIRSTYEARNEXTBB4	-	SCA		First year in the next Balbase4 model	114
IWEIGHTY	(Y)	SET	-	Relative weight of the individual years in Y	112
IORD_IY	YMODEL	PAR		Ord of years Y in present Balbase4 model	
IAGK_HASORPOT	(Y,AAA,G)	SET	-	Tech. with existing or possibly invested cap.	114
VGE_T	(Y,AAA,G,S,T)	VAR	MW	BB4: Elec gen. both existing and new tech.	114
VGH_T	(Y,AAA,G,S,T)	VAR	MW	BB4: Heat gen. both existing and new tech.	114
VGKNACCUMNET	(Y,AAA,G)	VAR	MW	Accumulated new investments	114
VGKNACCUMGROSS	(Y,AAA,G)	VAR	MW	Accumulated new investments this BB4	114
VXKNACCUMNET	(Y,IRRRE,IRRRI)	VAR	MW	Accumulated new transmission investments	114
QGKNACCUMNET	(Y,AAA,G)	EQU	MW	Accumulated new investment in tech.	115
QXKNACCUMNET	(Y,IRRRE,IRRRI)	EQU	MW	Accumulated new investment in trans.	115

Table 12: Identifiers specific for BB4. Some are described in the text at the indicated pages, while others are assumed to be fairly obvious (otherwise consult the code). YMODEL, YMODELDELTA and DISCOUNTRATE are new input data IDs.

Input data

Most input data for Balbase4 are identical to those for Balbase2 as of version 3.03. New are sets YMODEL and YMODELDELTA and scalar DISCOUNTRATE. As will be clear from the following, input data ANNUITYC will have a slightly different meaning. Also some internal mechanisms differ and well as some internal IDs, e.g. the electricity generation VGE_T, therefore output has to be interpreted differently. As a special observation note that both Balbase1 and Balbase2 functionalities may be exactly attained as a special case of Balbase4.

Some internal identifiers are used in BB4. Table 12 holds all IDs that are specific for BB4 or have a different meaning.

To appreciate the following YMODEL and YMODELDELTA, introduce the rolling horizon. A "rolling horizon" approach may be described as follows in relation to investments. Solve an investment model from a certain year Å1 ('now'), looking Å2 years ahead for making an investment decision (cf. YMODEL below). Once found, only the first Å3 years of this is implemented (cf. YMODELDELTA below). Then Å1 ('now') is incremented by Å4 (cf. YMODELDELTA below), and the process is repeated. Å2, Å3 and Å4 are user choices. This is essentially what is achieved in model Balbase4. This does not mean that it is in general recommended to use a rolling horizon approach, see more on this later, but it is possible.

Set YMODEL

In SET YMODEL, CARD(YMODEL) is used to specify the number of years in any model Balbase4. Additionally, YMODEL is used as domain for YMODELDELTA. Input data.

The number CARD(YMODEL) implies the number of years that are included in any model Balbase4, ie., the Å2 from page 111. Thus, with e.g. YMODEL given as "/Jesper, Felstedt, Morten, Middelboe, Pedersen/;" there will be five years in any model Balbase4. The years included are from set Y.

Since only CARD(YMODEL) matters for this, the labels (element names) in YMODEL may be any. It will be unwise to choose labels that may be confused with other labels or identifiers in the Balmorel model.

Parameter YMODELDELTA

Parameter YMODELDELTA specifies the number of years (in set Y, not in set YYY) between a year (in Y) and the preceding year (in Y) in model Balbase4. Input data.

Here, the "number of years" means the difference in ord(Y)'s between two elements in set Y (not YYY)) symbolizing one year and the preceding year. For instance, if set Y is given as "/ 2001, 2005, 2006, 2025, 2027 /;", YMODEL is given as "/BB4simy1, BB4simy2, BB4simy3/;" then there are three elements from Y in each BB4 model; and with YMODELDELTA is given as "/BB4simy1=0, BB4simy2=2, BB4simy3=1 /;" the first Balbase4 model will be constituted by years (2001,2006,2025) and the second Balbase4 model will be constituted by years (2005,2025,2027). There will be no more than two models Balbase4, because further increments of the first year in a Balbase4 model will not be consistent with YMODEL and YMODELDELTA.

The value for the first element in YMODELDELTA may be 0 or positive, and the value for any following element must be a strictly positive integer.

With this, the two extreme, but typical, ways of using Balbase4 are specified as follows. For illustration, set Y given as "/ 2001, 2005, 2006, 2025, 2027 /;" will be used.

To get exactly the same as in Balbase2, use the following: YMODEL is given as "/BB4simy1/;" and YMODELDELTA is given as "/BB4simy1 0 /;".

To get exactly one model Balbase4, use the following: YMODEL is given as "/BB4simy1, BB4simy2, BB4simy3, BB4simy4, BB4simy5 /;" and YMODELDELTA is given as "/BB4simy1 0, BB4simy2 1, BB4simy3 1, BB4simy4 1, BB4simy5 1/;".

More generally, exactly one model Balbase4 will be generated if SUM(YMODEL, YMODELDELTA(YMODEL)) equals CARD(Y)-1. If CARD(YMODEL) is greater than CARD(Y) no model Balbase4 will be generated. Similar negative result is obtained by setting the values in YMODELDELTA such that SUM(YMODEL, YMODELDELTA(YMODEL)) is greater than CARD(Y)-1.

While the first of these examples does not have very must "horizon" about it, the second one does not have much "rolling". More "rolling horizon" is achieved e.g. with Y given as "/ 2001, 2002, 2005, 2010, 2020, 2030 /;", and using YMODEL given as "/BB4simy1, BB4simy2, BB4simy3/;" and YMODELDELTA given as "/BB4simy1 0, BB4simy2 1, BB4simy3 2/;". This will generate three Balbase4 models, first with years (2001,2002,2010), then with years (2002,2005,2020) and finally with years (2005,2010,2030). Technology capacities generated endogenously in the model over years (2001,2002,2010) are taken as exogenous in the model over years (2002,2005,2020), but only for the year 2002 etc., cf. Section 13.3.14. Other examples are given in Table 13.

Further refinement will be added according to the value of the first element in YMODELDELTA. If the value is larger than 1, the distance between two Balbase4 models' first years (measured as distance between YVALUE(Y) of first years) is equal to that value. Otherwise it is 1 as in the first four examples in Table 13. This is the Å4, and additionally this is also the value Å3 of from page 111.

Discounting

Discounting mechanisms on the objective is applied, making distant future years count less in the model that near years. It is based on the scalar DISCOUNTRATE (fraction). The DISCOUNTRATE represents society's perception of how future years' costs and benefits shall be evaluated today.

The ANNUITYC used in model BB2 is still used. It is to represent the actors' (e.g., companies or TSOs) perception of financing costs, alternatives, expectations to profit, risks, etc. It differs from DISCOUNTRATE in that it is to be understood as specific for actors, energy companies (for electricity and heat production and related investments) and TSOs (for electricity transmission). The values in ANNUITYC may be zero.

Weighting of individual years

A BB4 model is formulated over a subset of Y, which again is a subset of YYY. This raises the challenge of how to weight the individual years in Y against each other (not in the discounting sense). For instance, with set Y defined as /2015, 2020, 2030, 2040 / an interpretation might be that 2030 'represents' the years from 2026 through 2035. Hence, the weight of 2030 could be 10 (years). Then maybe 2040 could represent the years from 2036 through 2040; however, this assigns only 5 (years) to 2040. So maybe 2040 could represent the years from 2036 through 2045 (assuming including the same number of years after 2040 as before 2040), hence with this idea 2040 is assigned 10 (years). The latter idea would then assign the value 5 (years) (years 2012.5 through 2017.5) to 2015.

IWEIGHTY(Y) holds the weights. * ISIMWEIGHT(YMODEL) holds the weights.

YYY	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Ex 1:																					
Y	10					15					20					25					30
BB4a	10					15					20										
BB4b						15					20					25					
BB4c											20					25					30
Ex 2:																					
Y	10	11	12		14			17			20					25					30
BB4a	10	11	12																		
BB4b		11	12		14																
BB4c			12		14			17													
BB4d					14			17			20										
BB4e								17			20					25					
BB4f											20					25					30
Ex 3:																					
Y	10	11	12	13	14	15	16	17	18	19	20	21	22	23							
BB4a	10					15					20										
BB4b		11					16					21									
BB4c			12					17					22								
BB4d				13					18					23							
Ex 4:																					
Y	10	11	12	13	14	15	16	17	18	19	20	21	22	23							
BB4a	10	11		13			16	17	18	19	20										
BB4b		11	12		14			17	18	19	20	21									
BB4c			12	13		15			18	19	20	21	22								
BB4d				13	14		16			19	20	21	22	23							
Ex 5:																					
Y	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
BB4a	10	11	12			15			18			21									
BB4b				13	14	15			18			21			24						
BB4c							16	17	18			21			24			27			
BB4d										19	20	21			24			27			30

Table 13: Examples of use of YYY, Y, YMODEL and YMODELDELTA in BB4. YYY is the same for all examples: 2010 through 2030. Y is given in the Table for each example. The BB4 models and the years they are formulated over are shown for each example. Example 1: YMODEL: 3 elements, YMODELDELTA: 0;1;1. Example 2: YMODEL: 3 elements, YMODELDELTA: 1;1;1. Example 3: YMODEL: 3 elements, YMODELDELTA: 0;5;5. Example 4: YMODEL: 8 elements, YMODELDELTA: 0;1;2;3;1;1;1. Example 5: YMODEL: 6 elements, YMODELDELTA: 3;1;1;3;3;3.

The option bb4weighty (file balopt.opt) permits selection between two possibilities: (option value same:) identical weights to all years or (option value shareYYY:) years in YYY but not in Y are shared equally between the two nearest years in Y (first and last years double the shared weight contribution).

Parameter scrapvalue

Not implemented yet (i.e., implicitly 0).

Internals

The main loop over years in BB4.sim is over set Y (or a set that is aliased with Y). For each pass of the loop, a new Balbase4 model is generated and solved. The years included in any model Balbase4 are found according to specification of Y, YMODEL and YMODELDELTA, cf. Section 13.3.14.

SET IY411(Y) denotes for any pass of the loop the years in model Balbase4. SET IY401(Y) and SET IY410(Y) are identical to IY411, except that they exclude the first and the last year in IY411, respectively.

Scalar IFIRSTYEARNEXTBB4 holds the first year in the next Balbase4 model.

Investment possibilities are specified by set as in BB2. However, some limitation on the investments in reality will reduce this set. In particular, there are implicit constraints from the available fuel potential held in FKPOT. This is used to define the internal set IAGKN for potential investments. IAGKN is equal to AGKN minus what follows from FKPOT. Further investment constraints follows from the first investment year information in GDATA. This is used to further limit IAGKN and hold the result in set IAGKNY which is indexed also over Y.

The set IAGK_HASORPOT holds technologies with existing or potentially invested capacity for any year.

Variables

The Balbase4 model uses the variables VGE_T and VGH_T to denote generation of electricity and heat. They are declared over a domain containing Y. In this, these variables are the same as in Balbase2, however, in Balbase4 the variables have an additional index indicating the year. In Balbase4 there are no variables VGEN_T and VGHN_T to hold generation from new investments as there is in Balbase2, this is also included in VGE_T and VGH_T in Balbase4. The endogenously invested generation capacity is held in variable VGKN declared over a domain containing Y.

For any model Balbase4, and any year within that model, VGE_T and VGH_T denote generation on technologies. Capacity may originate from four sources, (1) specified exogenously in GKFX (fixed number), (2) found endogenously in one or more Balbase4 models that were solved previous to present model, minus any decommissioning in any of those previous models due to expiration of life time (fixed number represented by variable VGKNACCUMNET.L of previous years), (3) endogenous investment present model Balbase4 (variable VGKN for present year) minus any decommissioning of this investment in present model, and (4) endogenous investment (VGKACCULTNET.L) in a previous year, minus decommissioning in the present model Balbase4.

Note that VGKNACCUMNET is formulated over a domain containing Y. Since any IY411 is a subset of Y, VGKN, VGKNACCUMNET, VGE_T, VGH_T and other variables can accommodate values for solution to Balbase4 models previous to the present model Balbase4 for any IY411. Hence there is no need to save solution values for each Balbase4 model. Similar considerations apply to information related to equations, e.g. for the electricity balance equation QEEQ the dual values (for deriving "electricity prices") are available for all Balbase4 models. Note though that in case of Balbase4 models overlapping for some Y only the latest solution value will be available in these variables.

Similar implementation is used for transmission capacity investments applying VXKN and VXKNACCUMNET.

See the code for details.

Equations, model and solve

Most equations are quite similar between models Balbase2 and Balbase4. They are distinguished by an additional index indicating the year. Also details differ.

Some additional equations are made to represent linkages between years, QGKNACCUMNET and QXKNACCUMNET.

See the code for details.

Model and solve

The equations are to be included in model Balbase4.

File, folder and include structure

Not clear yet. The problem is that really many code lines differ because there is an additional year index in many identifiers.

Present implementation: most code i Balmorel.gms is skipped if option BB4 is set at yes. As substitution, another file BB4.gms is $included. That in turn $includes BB4.sim.

Some supporting code is found in addon BB4.

The files are included into the model at appropriate places by statements of the form "$ifi %BB4%==yes $include ...".

Add-on option

The application of the addon is controlled by $Setglobal BB4, where a yes specifies that the addon be applied.

Can not be used with models Balbase1, Balbase2 or Balbase3. Can be used with addon agkndisc (discrete investments).

Some further observations and interpretations

Terminology

The following terminology is suggested [with examples from Ex. 1 in Table 13]:

Represented years The years in Y [10, 15, 20, 25, 30]
Planning horizon Time span until and including the last year in Y [from 10 to 30]
Intermediate model A model solved as part of the full BB4 run [there are three formulated, over years (10, 15, 20), (15, 20, 25) and (20, 25, 30), respectively]
Current model The intermediate model currently solved [for the second model: the years 15, 20, 25]
Current model horizon Time span until and including the last year in the current intermediate model [for the second model: the year 25]
Base year The first year in Y [10]

Balbase4 versus Balbase2

One might intuitively think that model Balbase4 would be "better" than model Balbase2. The reason is that Balbase4 includes all of model Balbase2 and then some more. The additional component is a better look-ahead mechanism for investment decisions.

The only look-ahead mechanism of Balbase2 is the annuity aspect related to investments. If the future is similar to the present, this may be a fair representation, but if the future is different (e.g. very different fuel prices or environmental goals), then there may be investments now that will appear unfavorable later. In this respect Balbase4 is "better", since it knows the future.

Calculation time

Often, model Balbase4 is larger than Balbase2, because it includes several years. On the other hand, Balbase4 may be solved less times. In many application, supposedly, if Balbase4 is X times larger than Balbase2, then Balbase2 will be solved X times more often than Balbase4.

A common experience from solution of LP (Linear Programming) models is that the solution time with a simplex based algorithm runs in $O(n^2m)$ time, where n is the number of variables and m is the number of equations. Thus, larger problems typically take un-proportionally longer time to solve.

It may translate to using BB4 takes longer time than using a comparable sequence of BB1 or BB2 models. Or for instance that solving a BB4 model with a single intermediate model over five years (e.g., 10, 15, 20, 25, 30) will take more time than solving with five intermediate models (one for each or the years). From the above formula one could estimate that with one single-year model with m equations and n variables taking x seconds to solve then five single-year models takes $5x$ seconds; while solving one five-years model takes $(5^2 5)/5$ as much time, i.e. $25x$.

Prices

For a data set that happens to give the same primal solution for Balbase2 and Balbase4 the objective value VOBJ will be the same (in case of no discounting in Balbase4). This could imply that marginal costs (some of which are interpreted as electricity prices) would also be identical. However, the situation is more complex, because although all input data are identical, and all optimal primal values are identical this does not suffice to make all interpretations (including marginal costs or dual variables) identical.

Note on literature

My attention was recently drawn to a 1982 paper *) that provides a neat classification of rolling horizon approaches. A restatement of the idea in slightly different wording may be given as follows,

- Year-by-year optimization without any idea of the future
- Year-by-year optimization with some look-ahead; Balmorel BB1 and BB2 could be examples of this
- Optimization over the full planning horizon; Balmorel BB4 (without rolling) could be an example of this

*) K. D. Le and J. T. Day, Rolling horizon method: A new optimization technique for generation expansion studies, IEEE Trans. PAS-101, No. 9, September 1982.

13.4 Other option

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

Addons mostly refer to new functionalities related to detailed functionalities in a model, e.g. BALBASE1 (Section 9.1).

Options are also used in relation to possibilities and choices related to other aspects. As an example, consider option semislash.

An input data file holds data for a single identifier. Two or more data files for the same identifier may be prepared by the user, holding different versions of data, e.g. for high, low or medium wind power penetration. Therefore in principle two kinds of information for any data file are relevant: for the type of identifier and for the specific data instance.

The type of identifier is most naturally given by repetition of the declaration part from (in most cases) the file Balmorel.gms, e.g. "PARAMETER FDATA(FFF,FDATASET) 'Fuel specific values' ". If the User handmade the data file as a text file it is unproblematic to write "PARAMETER FDATA(FFF,FDATASET) 'Fuel specific values' /", continue with the data, and end the file with "/;". However, if the data were given from some other source, e.g. a data base, it may or may not be possible for that source to provide the declaration, "/" and "/;" parts.

The Control variable semislash is aimed at this instance. The options are

*!option ;

*!option /

If all data input files include the appropriate declaration, "/" and "/;" parts use option ;. Note that in this case it is possible to use also the Table format. If they all use the list format use option / (acronyms are provided as a list).

The specific data instance information should if possible be provided in the user as a comment in the file. Good information would be i.a. a short description, indication of source and the years of relevance.

13.5 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 1.3 and 15). By *model structure* is mainly meant identifiers (i.e., the names) of the SCALARS, PARAMETERS, VARIABLES, SETS and EQUATIONS in the model, and additional information like limitations on variables (declarations as POSITIVE or FREE, specifications of bounds (.UP, LO, or .FX)),; plus the associated code. By the *data* is meant the actual labels (i.e., members) in the SETS and the actual numerical values given 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: .... ".

14 User interface

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 more directly related to Balmorel is under development under the name BUI (Balmorel User Interface). Details to come.

15 Overview of model structure components

The model structure consists of the sets, parameters,variables, equations and models 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 46), 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 to use the compiler directives $ONSYMXREF, $ONSYMLIST, $ONUELLIST and $ONUELXREF (the $ in the first position of the line) which produce maps in the LST file after a GAMS executions of the model.

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 6.

THE FOLLOWING NEEDS UPDATE FOR VERSION 3.03.

Name	Domain	Type	Unit	Page
1995, 1996,..	-	obl. set member	(none)	30
AAA	-	set	-	29
AGKN	(AAA,GGG)	set	-	40
ANNUITYC	(CCC)	parameter	(none)	50
C	(CCC)	set	-	28
CCC	-	set	-	28
CCCRRR	-	set	-	29
CCCRRRAAA	-	SET	-	28
CYCLESINS	(S)	parameter		49
DE	(YYY,RRR)	parameter	MWh	54
DE_VAR_T	(RRR,SSS,TTT)	parameter	(none∼MW)	60
DEF	-	set	-	40
DEF_D1	-	set	-	41
DEF_D2	-	set	-	41
DEF_D3	-	set	-	41
DEF_STEPS	(RRR,SSS,TTT,..			
	..,DF_QP,DEF)	parameter	(none), Money/MWh	63
DEF_U1	-	set	-	41
DEF_U2	-	set	-	41
DEF_U3	-	set	-	41
DEFP_BASE	(RRR)	parameter	Money/MWh	51
DEFP_CALIB	(RRR,SSS,TTT)	parameter	Money/MWh	61
DF_QP	-	set	-	40
DF_PRICE	-	obl. set member	Money/MWh	40
DF_QUANT	-	obl. set member	MW	40
DH	(YYY,AAA)	parameter	MWh	55
DH_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	59
DHF	-	set	-	41
DHF_STEPS	(AAA,SSS,TTT,..			
	..,DF_QP,DEF)	parameter	(none), Money/MWh	64
DHF_D1	-	set	-	41
DHF_D2	-	set	-	41
DHF_D3	-	set	-	41
DHF_U1	-	set	-	41
DHF_U2	-	set	-	41
DHF_U3	-	set	-	41
DHFP_BASE	(AAA)	parameter	Money/MWh	52
DHFP_CALIB	(AAA,SSS,TTT)	parameter	Money/MWh	60
DISCOST_E	(RRR)	parameter	Money/MWh	51
DISCOST_H	(AAA)	parameter	Money/MWh	51
DISLOSS_E	(RRR)	parameter	(none)	51
DISLOSS_H	(AAA)	parameter	(none)	51

Name	Domain	Type	Unit	Page
FDATASET	-	set	-	39
FDATA	(FFF,FDATASET)	set	-	53
FDCO2	-	obl. set member	kg/GJ	39
FDNB	-	obl. set member	(none)	39
FDSO2	-	obl. set member	kg/GJ	39
FFF	-	set	-	39
FGEMAX	(CCCRRRAAA,FFF)	parameter	(MWh)	58
FGEMIN	(CCCRRRAAA,FFF)	parameter	(MWh)	58
FKPOT	(CCCRRRAAA,FFF)	parameter	MW	58
FUELPRICE	(YYY,AAA,FFF)	parameter	Money/GJ	62
G	(GGG)	set	-	34
GDATA	(GGG,GDATASET)	parameter	-	55
GDATASET	-	set	-	34
GDAUXIL	-	obl. set member	-	56
GDCB	-	obl. set member	-	55
GDCV	-	obl. set member	-	55
GDESO2	-	obl. set member	-	55
GEFFDERATE	Use GEFFRATE	parameter	(none)	54
GEFFRATE	(GGG,AAA)	parameter	(none)	54
GDFE	-	obl. set member	-	55
GDFROMYEAR	-	obl. set member	-	56
GDFUEL	-	obl. set member	-	55
GDINVCOST0	-	obl. set member	-	56
GGCOMB	(GGG,IGGGALIAS)	set	-	95
GGG	-	set	-	34
GINVCOST	(GGG,AAA)	parameter	MMoney/MWh	53
GDKVARIABL	-	obl. set member	-	56
GDNOX	-	obl. set member	-	56
GDOMFCOST0	-	obl. set member	-	56
GDOMVCOST0	-	obl. set member	-	56
GDSTOHLOAD	-	obl. set member	hours	56
GDSTOHUNLD	-	obl. set member	hours	56
GDTYPE	-	obl. set member	-	55
GKDERATE	Use GKDERATE	parameter	(none)	65
GKRATE	(GGG,AAA,SSS)	parameter	(none)	65
GKFX	(YYY,AAA,GGG)	parameter	MW / MWh	62
GKNMAX	(YYY,AAA,GGG)	parameter		61
GKVACC	(Y,AAA,G)	parameter	MW	62
GKVACCDECOM	(Y,AAA,G)	parameter	MW	62
GOMFCOST	(GGG,AAA)	parameter	MMoney/MW	53
GOMVCOST	(GGG,AAA)	parameter	Money/MWh	53
GVKGN	(YYY,AAA,G)	parameter	MW	62
HYPPROFILS	(AAA,SSS)	parameter	Money/MWh	59
HYRSDATA	-	set	-	35
IA	(AAA)	int. set	-	42
IAGK_Y	(AAA,G)	int. set	-	71
IAGKN	(AAA,G)	int. set	-	45
IANYSET	-	int. set	-	67
ICA	(XYZ)	int. set	-	42
IBALVERSN		int. scalar	-	67
IDE_SUMST	(RRR)	int. parameter	(none~MWh)	68
IDE_T_Y	-	int. parameter	MW	72
IDEFP_T	-	int. parameter	Money/MWh	73
IDH_SUMST	(AAA)	int. parameter	(none~MWh)	68
IDH_T_Y	-	int. parameter	MW	72
IDHFP_T	-	int. parameter	Money/MWh	72

Name	Domain	Type	Unit	Page
IFUELP_Y	(AAA,FFF)	int. parameter	Money/GJ	71
IGBPR	(G)	int. set	-	43
IGCND	(G)	int. set	-	43
IGDISPATCH	(G)	int. set	-	45
IGETOH	(G)	int. set	-	43
IGEOREH	(G)	int. set	-	45
IGEXT	(G)	int. set	-	43
IGGGALIAS	alias (GGG)	int. set	-	45
IGHOB	(G)	int. set	-	43
IGHYRS	(G)	int. set	-	43
IGKFX_Y	(GGG,AAA)	int. parameter	MW / MWh	70
IGKE	(G)	int. set	-	45
IGKH	(G)	int. set	-	45
IGKVACCTOY	(G,AAA)	int. parameter	MW / MWh	70
IGNOTETOH	(G)	int. set	-	45
IGSOLE	(G)	int. set	-	43
IGSOLH	(G)	int. set	-	43
IGWAVE	(G)	int. set	-	43
IGWND	(G)	int. set	-	43
IHOURSINST	(S,T)	int. parameter	hours	68
IHYINF_S	(AAA,SSS)	int. parameter	MWh/MW	69
ILIM_CO2_Y	(C)	int. parameter	t	71
ILIM_NOX_Y	(C)	int. parameter	kg	72
ILIM_SO2_Y	(C)	int. parameter	t	71
IM_CO2	(G)	int. parameter	kg/GJ	70
IM_SO2	(G)	int. parameter	kg/GJ	70
IOFxyz	-	int. scalar	-	67
IPLUSMINUS	-	int. set	-	45
IR	(RRR)	int. set	-	43
ISALIAS	alias (S)	int. set	-	43
ISCALAR1	-	int. scalar	-	67
ISOLESUMST	(AAA)	int. parameter	(none∼MWh)	69
ITALIAS	alias (T)	int. set	-	43
ITAX_CO2_Y	(YYY,CCC)	int. parameter	Money/t	71
ITAX_NOX_Y	(YYY,CCC)	int. parameter	Money/kg	71
ITAX_SO2_Y	(YYY,CCC)	int. parameter	Money/t	71
IWEIGHSUMS	(S)	int. parameter	(none)	67
IWEIGHSUMT	(T)	int. parameter	(none)	68
IWND_SUMST	(AAA)	int. parameter	(none∼MWh)	68
IWTRRRSUM	(AAA)	int. parameter	(none∼MWh)	69
IWTRRSSUM	(AAA)	int. parameter	(none∼MWh)	69
IX3VPIM_Y	(RRR,X3VPLACE0,...			
	...,X3VSTEP0,S,T)	parameter	Money/MWh	100
IX3VPEX_Y	(RRR,X3VPLACE0,...			
	...,X3VSTEP0,S,T)	parameter	Money/MWh	100
IXKINI_Y	(IRRRE,IRRRI)	int. parameter	MW	70
IXKN	(IRRRE,IRRRI)	int. set	-	70
IX3FX_T_Y	(RRR,S,T)	int. parameter	MW	73
IX3FXSUMST	(RRR)	int. parameter	(none∼MWh)	70
LIM_CO2	(YYY,CCC)	obl. set member	t	42
LIM_NOX	(YYY,CCC)	obl. set member	kg	42
LIM_SO2	(YYY,CCC)	obl. set member	t	42
M_POL	(YYY,MPOLSET,CCC)	parameter		62
MPOLSET	-	set	-	42
OMONEY		parameter		48
PENALTYQ	-	scalar	-	47

Name	Domain	Type	Unit	Page
QECOMBGLK	(AAA,G,S,T)	equation	MW	??
QECOMBSLO	(AAA,G,S,T)	equation	MW	??
QECOMBSUP	(AAA,G,S,T)	equation	MW	??
QHCOMBGLK	(AAA,G,S,T)	equation	MW	??
QHCOMBSLO	(AAA,G,S,T)	equation	MW	??
QHCOMBSUP	(AAA,G,S,T)	equation	MW	??
QNECOMBGLK	(AAA,G,S,T)	equation	MW	??
QNECOMBSLO	(AAA,G,S,T)	equation	MW	??
QNECOMBSUP	(AAA,G,S,T)	equation	MW	??
QNHCOMBGLK	(AAA,G,S,T)	equation	MW	??
QNHCOMBSLO	(AAA,G,S,T)	equation	MW	??
QNHCOMBSUP	(AAA,G,S,T)	equation	MW	??
QEEQ	(RRR,S,T)	equation	MW	77
QESTOVOLT	(AAA,S,T)	equation	MW	78
QGCBGBPR	(AAA,G,S,T)	equation	MW	77
QGCBGEXT	(AAA,G,S,T)	equation	MW	77
QGCVGEXT	(AAA,G,S,T)	equation	MW	77
QGGETOH	(AAA,G,S,T)	equation	MW	77
QGNCBGBPR	(AAA,G,S,T)	equation	MW	77
QGNCBGEXT	(AAA,G,S,T)	equation	MW	78
QGNCVGEXT	(AAA,G,S,T)	equation	MW	78
QGNGETOH	(AAA,G,S,T)	equation	MW	78
QGEKNT	(AAA,G,S,T)	equation	MW	78
QGHKNT	(AAA,G,S,T)	equation	MW	78
QGKNWND	(RRR,AAA,G,S,T)	equation	MW	78
QGKNSOLE	(RRR,AAA,G,S,T)	equation	MW	78
QGKNHYRR	(AAA,G,S,T)	equation	MMoney	78
QHSTOVOLT	(AAA,S,T)	equation	MW	78
QHYRSSEQ	(AAA,S)	equation	MMoney	78
QKFUELC	(C,FKPOTSETC)	equation	MW	78
QKFUELR	(RRR,FKPOTSETR)	equation	MW	78
QKFUELA	(AAA,FKPOTSETA)	equation	MW	78
QLIMCO2	(C)	equation	ton	78
QLIMSO2	(C)	equation	ton	78
QLIMNOX	(C)	equation	kg	78
QOBJ	-	equation	MMoney	77
QXK	(IRRRE,IRRRI,S,T)	equation	MW	78
RRR	-	set	-	29
RRRAAA	-	set	-	29
RX3VSUBSTI	(RRR,X3VPLACE0)	set	-	99
S	(SSS)	set	-	31
SOLEFLH	(AAA)	parameter	hours	52
SOLE_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	60
SOLHFLH	(AAA)	parameter	hours	52
SOLH_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	60
SSS	-	set	-	31

Name	Domain	Type	Unit	Page
T	(TTT)	set	-	31
TAX_DE	(CCC)	parameter	Money/MWh	50
TAX_DH	(CCC)	parameter	Money/MWh	51
TAX_F	(FFF,CCC)	parameter	Money/MWh	53
TAX_GE	(YYY,AAA,G)	parameter	Money/MWh	62
TAX_GF	(YYY,AAA,G)	parameter	Money/MWh	62
TAX_GH	(YYY,AAA,G)	parameter	Money/MWh	62
TAX_KN	(YYY,AAA,G)	parameter	Money/MWh	62
TAX_CO2	-	obl. set member	Money/t	42
TAX_NOX	-	obl. set member	Money/kg	42
TAX_SO2	-	obl. set member	Money/t	42
TTT	-	set	-	31
VDEF_T	(RRR,S,T,DET_STEPS)	variable	MW	76
VDHF_T	(AAA,S,T,DHFSTEPS)	variable	MW	76
VESTOLOADT	(AAA,S,T)	variable	MW	76
VESTOVOLT	(AAA,S,T)	variable	MWh	76
VGKN	(AAA,G)	variable	MW	76
VGE_T	(AAA,G,S,T)	variable	MW	75
VGEN_T	(AAA,G,S,T)	variable	MW	75
VGH_T	(AAA,G,S,T)	variable	MW	75
VGHN_T	(AAA,G,S,T)	variable	MW	75
VHSTOLOADT	(AAA,S,T)	variable	MW	76
VHSTOVOLT	(AAA,S,T)	variable	MWh	76
VOBJ	-	variable	MMoney	75
VQEEQ	(RRR,S,T,IPLUSMINUS)	variable	MW	76
VQESTOVOLT	(AAA,S,T,IPLUSMINUS)	variable	MWh	76
VQHSTOVOLT	(AAA,S,T,IPLUSMINUS)	variable	MWh	76
VQHYRSSEQ	(AAA,S)	variable	MW	76
VX3VIM_T	(RRR,X3VPLACE0,...			
	...0,X3VSTEP0,S,T)	variable	MW	100
VX3VEX_T	(RRR,X3VPLACE0,...			
	...,X3VSTEP0,S,T)	variable	MW	100
VXKN	(IRRRE,IRRRI)	variable	MW	75
VX_T	(IRRRE,IRRRI,S,T)	variable	MW	75
WAVEFLH	(AAA)	parameter	hours	52
WAVE_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	60
WEIGHT_S	(SSS)	parameter	(none)	49
WEIGHT_T	(TTT)	parameter	(none)	49
WND_VAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	60
WNDFLH	(AAA)	parameter	hours	52
WTRRSVARS	(AAA,SSS)	parameter	(none∼MW)	58
WTRRRVAR_T	(AAA,SSS,TTT)	parameter	(none∼MW)	60
WTRRRFLH	(AAA)	parameter	hours	52
WTRRSFLH	(AAA)	parameter	hours	52

Name	Domain	Type	Unit	Page
X3FX	(YYY,RRR)	parameter	MWh	54
X3FX_VAR_T	(RRR,SSS,TTT)	parameter	(none~MW)	61
X3VPEX	(YYY,RRR,X3VPLACE0,...			
	...,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	100
X3VPIM	(YYY,RRR,X3VPLACE0,...			
	...,X3VSTEP0,SSS,TTT)	parameter	Money/MWh	100
X3VPLACE0	-	set	-	99
X3VQEX	(RRR,X3VPLACE0,...			
	...,X3VSTEP0,SSS,TTT)	parameter	MW	100
X3VQIM	(RRR,X3VPLACE0,...			
	...,X3VSTEP0,SSS,TTT)	parameter	MW	100
X3VSTEP0	-	set	-	99
X3VSTEP	(X3VSTEP0)	set	-	99
X3VX	(RRR,X3VPLACE0)	set	-	99
XCOST	(IRRRE,IRRRI)	parameter	Money/MWh	57
XKDERATE	See XKRATE	parameter		65
XKRATE	(IRRRE,IRRRI)	parameter		65
XINVCOST	(IRRRE,IRRRI)	parameter	Money/MWh	57
XKINI	(IRRRE,IRRRI)	parameter	MW	57
XLOSS	(IRRRE,IRRRI)	parameter	(none)	57
Y	(YYY)	set	-	30
YVALUE	(YYY)	parameter	(none)	48
YYY	-	set	-	30

Index