WORKING PAPER

USER'S GUIDE FOR THE MATRIX GENERATOR OF MESSAGE II
PART I:
MODEL DESCRIPTION AND IMPLEMENTATION GUIDE

S. Messner

September 1984
WP-84-71a

IIASA
International Institute
for Applied Systems Analysis

NOT FOR QUOTATION
WITHOUT PERMISSION
OF THE AUTHOR

USER'S GUIDE FOR THE MATRIX GENERATOR OF MESSAGE II

PART 1:

MODEL DESCRIPTION AND IMPLEMENTATION GUIDE

S. Messner

September 1984
WP-84-71a

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

Preface

During recent years mathematical modeling has been extensively deployed within the IIASA energy research activities. The shift in the Energy Project's activities from predominantly global energy demand and supply considerations toward more regional and local ones called for greater detail in the representation of the energy system compared to the models used within the scope of "Energy in a Finite World" (W. Häfele et al., Ballinger Publishing Company, 1981). Furthermore, the statistical data base has improved considerably over recent years which allowed for a comprehensive representation of the entire energy chain from resource extraction to end-use conversion.

The change in the scope of the Energy Group's activities and the improved statistical data availability motivated the development of the MESSAGE-II energy model. The point of departure was the original MESSAGE model of IIASA's global energy study. Like its predecessor, MESSAGE-II was conceptualized for the transformation of dynamic linear programming problems into matrix structures. But apart from primary to secondary energy conversion, the main feature of the original model, this new code also depicts energy extraction, transport, distribution and end-use conversion. In addition, MESSAGE-II hosts features such as integer programming, non-linear objectives and the possibility of multi-objective optimization. The optimization of a MESSGAE-II coded problem requires a linear program solving package, e.g. MINOS, APEX, MPSX, etc.

Part I of this report provides the descriptions of both, the mathematical formulation and a guide for the code implementation and specification of input files necessary for the potential user to understand and apply this code. Part II contains a complete example of a MESSAGE-II code application including input files, etc. The actual use of MESSAGE-II may benefit from another publication that is closely related to this report. The "User's Guide for the Post-Processor of MESSAGE-II (WP-84-72) entails convenient pre- and post- calculating procedures, report writing and plotting routines, etc. tailored particularly to interact with MESSAGE-II. The examples used in the post-processor guide correspond to the examples given in the user's guide for the matrix generator.

Hans-Holger Rogner
Leader
International Gas Study

TABLE OF CONTENTS

Page

1. Introduction

MESSAGE II is a dynamic linear programming model that has the option to use mixed integer programming. In connection with *MINOS* [1] it can also handle nonlinear objective functions. The technique of linear programming (LP) has been chosen because practically all types of computers are equipped with commercial solving packages, that are well tested and applicable for large-scale problems. *MESSAGE II* has been developed at IIASA on the basis of *MESSAGE*, a *M*odel for *E*nergy *S*upply *S*trategy *A*lternatives and their *G*eneral *E*nvironmental impact [2]. The underlying principle of both models is the optimization of an objective function under a set of constraints defining the feasible region where all possible solutions of the problem lie. The objective helps to choose the solution considered best according to the criteria specified.

Due to the fact that LP problems are usually solved by commercial LP packages the whole software package of *MESSAGE II* consists of two blocks. The application of block 1 provides the user with the matrix to be further processed by the LP package and with a printed report of the inputs and assumptions.

Block 2 is then used to process the output of the LP package further, e.g. extract growth rates, fuel mixes or elasticities from the solution to a given problem. For *MESSAGE II* the post-processing is done by *CAP* (*CA*lculator *P*rogram) [3], which is also able to handle data from *MEDEE-2* [4].

The reader of this paper is assumed to be familiar with the theory of linear and mixed integer programming; if he wants to apply the nonlinear options, some knowledge about *MINOS* and access to this code is essential. This User's Guide contains the mathematical formulation of *MESSAGE II* and a guide to use the computer codes of the matrix generator and report writer. It is thus intended to be an aid to implement and run the software of *MESSAGE II*.

2. General Remarks

2.1 The Software

The software of *MESSAGE II* consists of two logical blocks (see figure 1). In block 1

-- *CHIN* converts the inputs that are given in a free format to the format needed by the matrix generator (*MXG*),

-- *MXG* generates the matrix corresponding to these inputs and also produces dump files containing the complete information on the input data and

-- *REPO* produces a printable control output from the data on the dump files.

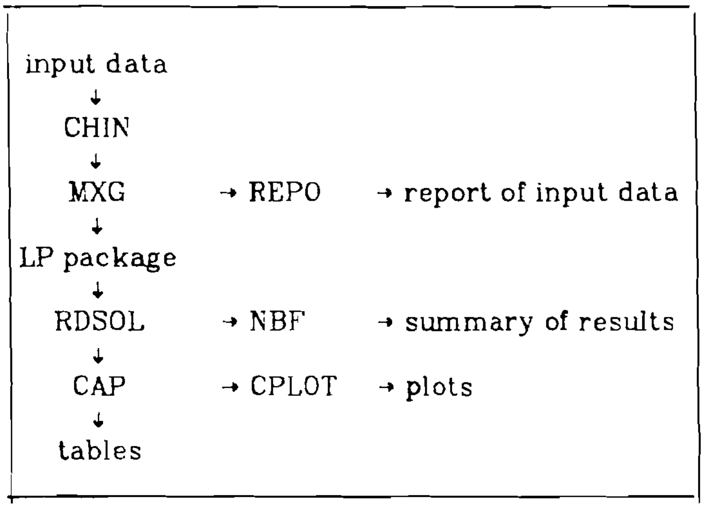


Figure 1: Basic Set-up of the Software of *MESSAGE II*.

In chapter 5 the linkage of the single codes, the units used and the files handled are explained more explicitly.

Block 2 contains five computer codes, namely

-- *RDSOL*, which converts the printable solution file produced by *MINOS* into an unformatted and sorted FORTRAN file; the adaptation to commercial LP packages (e.g., APEX, MPSX) can be done easily,

-- *NBF*, that yields information on the structure of the solution, i.e., which columns and rows are basic and which limits are reached,

-- *CAP*, that is used to produce tables from the solution obtained (according to external instructions), and

- *CPLOT*, that uses information given from *CAP* to produce plots and *SPLOT*, that can produce an output on the line printer or screen which simulates plotting.

The description on the use of the computer codes of block 2 can be found in [3].

In order to be able to adapt the size of the software to the model size an additional code is available:

- *CHDIM*, which changes the dimensions of all arrays in the other codes to the appropriate sizes.

2.2 The Basic Structure

Although this paper is not intended to be a description of the model *MESSAGE II*, but a users guide to block 1 of the software, an outline of the model formulation is given below. This is to help the user to understand the way the code works and to indicate for which types of models the matrix generator can be applied.

In first approximation *MESSAGE II* can be called a physical flow model. Given a vector of demands for specified goods or services, it assures sufficient supply, utilizing the technologies and resources considered. In its usual application the model is used to evaluate energy systems, but any other problem dealing with systems where specified demands can be met by a number of interrelated supply options can be modeled as well.

The backbone of *MESSAGE II* is the technical description of the modeled system. This includes the definition of the categories of energy forms considered, like, e.g., primary energy, final energy, useful energy (see figure 2), and the energy forms actually used, e.g., coal or district heat, but also the tons of steel or useful space heat provided by the use of energy. The technologies are defined by their inputs and outputs, the efficiency and the degree of variability if more than one input or output exists, e.g., the possible production patterns of a refinery. By all these definitions of energy carriers and technologies a so-called energy chain is structured, where the energy flows from the supply side to the demand side. The supplying energy forms can belong to all categories except useful energy, they have to be chosen in light of the actual problem. Maximal amounts available inside the modeled region and import possibilities have to be specified. Together with the demands, that are exogenous to the model, the technical system provides the basic set of constraints: The demands have to be met by the energy flowing from domestic resources and imports through the modeled energy chain.

Level	*Identifier*
Resources	*R*
↓	
Primary Energy	*A*
↓	
Secondary Energy	*X*
↓	
Final Energy (after transmission)	*T*
↓	
Final Energy (after distribution)	*F*
↓	
Useful Energy	*U*

Figure 2: Identification of the Energy Form Levels in *MESSAGE II*.

The amount and quality of obtainable information can be increased considerably by accounting for existing installations and the need to construct new capacities of the technologies. By knowing which types of and how many installations are required to build up a desired system one can assess the effects on the economy.

The investive requirements can be distributed over the construction time of the plant and they can be subdivided into different categories to allow accounting for the requirements from some important industrial and commercial sectors. But also the needs for basic materials during construction of a technology as well as the utilization of non-energetic inputs during the operation of a plant can be accounted for, keeping track of the industrial branches they originate from in monetary terms or just accumulating the needs in physical units.

Minimization of the total system costs can be used as objective to choose a solution (actually this is the default implemented into the system). In this case for all costs occurring at later points in time the present value is calculated by discounting them to the first year of the calculation, the sum of the discounted costs represents the objective function value. Discounting makes the costs occurring in different points in time comparable, the discount rate chosen defines the weights different periods get in the optimization. In principle it should be equal to the long-term real interest rate, i.e. excluding inflation or any other alternative opportunity. A high discount rate gives more weight or importance to present expenditures than to future ones, while a low discount rate reduces these differences and thus favors investments decreasing the run-time expenditures for a technology.

The time horizon of a model application has to be chosen with regard to the problem; it could be long as well as short term. Even the use for a single point it time could give valuable results for complex problems. For the calculations this time horizon is split into periods of optional length, each of which is represented by a sample year in the model.

The development of the modeled system over time can be more or less predefined if relative or absolute limits for certain energy carriers or technologies are given. But additionally *MESSAGE II* gives the possibility to introduce maximal and also minimal growth or decline rates for the installation of new technologies and for the use of domestic and imported fuels. This allows to predefine a range of variability of the system in time, within that the model will dynamically choose an optimal strategy.

Other features of *MESSAGE II* are dealing with

-- energy storage including consideration of decay of contents (e.g. heat storage),

-- load variations using semi-ordered, i.e. not completely ordered load curves,

-- demand and supply elasticities,

-- stock-piling of fuels over the time horizon,

-- inventories and last cores like they are necessary for nuclear reactors,

-- the built-in possibility to model energy density areas,

-- unit sizes of new installations, and

-- nonlinear objective functions, if *MINOS* is used to solve the problem.

As far as these special features are not included in the mathematical formulation (see chapter 3.2) they are explained in chapter 3.3.

2.3 The Sample Input File

Figure 3 gives an example of a simple energy chain that is defined based on the levels of energy forms as shown in figure 2 and used for implementing and testing the codes. This example, called *STIM*, which stands for *S*mall *T*est and *I*mplementation *M*odel, will be used as illustration in the following chapters. The domestic resource in *STIM*--coal--can only be used in a co-generation plant. The imported crude oil has to go through a refinery, outcoming residual fuel oil can be cracked again or used in heat plants or power plants. The lighter fractions can either be consumed for heating or to produce peak electricity. Gasoline (either produced in the refinery or in crackers or imported) goes to the transport sector, only. The second domestic energy source is hydraulic power. The demands in *STIM* are for specific applications of electricity and gasoline (in terms of final energy) and for useful thermal energy.

In order to run this model with *MESSAGE II* the user has to give 6 categories of information on this system:

1) general definitions like, e.g., time horizon, lengths of periods, first year of calculation, energy form levels used out of the set showed in figure 2 or new definition of these levels, energy forms on each level, which of these are to be modeled taking into consideration load variations, etc.,

2) demands and their evolution over time, distribution to the load regions and energy density areas and definition of demand elasticities,

3) additional constraints and relations, accounting rows to be included,

4) objective function to be used,

5) definition of technologies in the energy chain by technical--efficiencies, technical plant life, etc.--and economic--investment costs, operation costs, etc.--parameters and

6) availability and costs of resources, imports and exports, supply elasticities, etc.

Appendix 1 contains the input file for the matrix generator corresponding to the sample system *STIM*. Based on this input file *MXG* would generate a matrix containing at least one activity variable for each technology (representing the annual energy input to that technology, see chapter 3.1.1) per period. If one of the in- or outputs of the technology is defined to have load regions, one activity variable is defined for each load region (the relative production per load region is chosen during optimization, then). This can also be fixed to a predefined pattern,

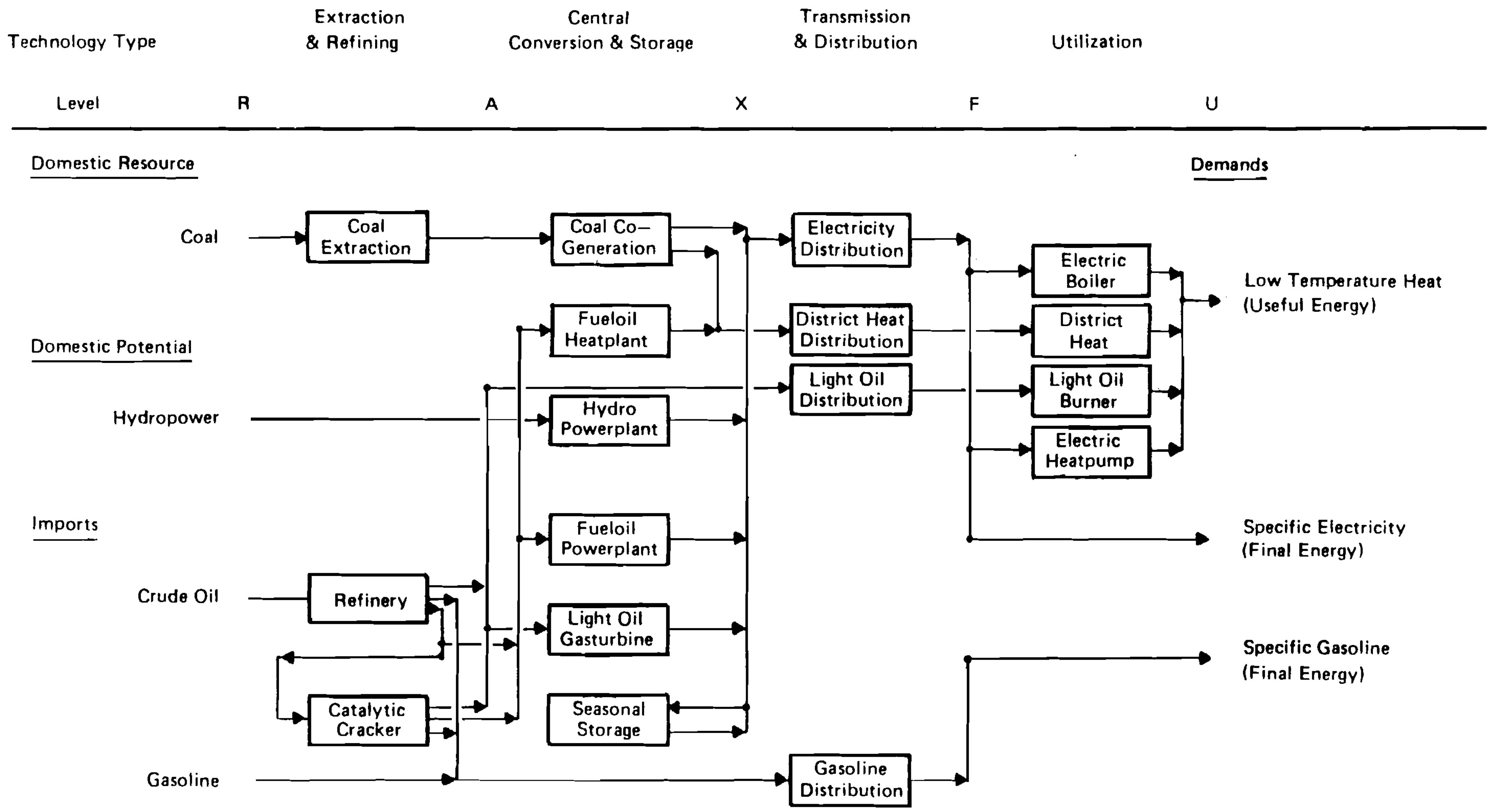


Figure 3: The Energy Conversion Chain of *STIM*.

if desired--in *STIM* the hydroelectric power plants are fixed to the known pattern in that hydraulic power is available. A capacity variable per period (see chapter 3.1.3) is also generated, if it is not switched off explicitly, like for light liquid transportation in *STIM* (because of the short plant life of tank trucks). This variable represents the annual new installations of the system per period. For each technology the two types of activities are linked together by a capacity constraint (see chapter 3.2.23), which ensures that not more capacity is utilized than is available.

For all energy forms defined in the energy chain energy flow constraints are constructed, see for instance chapter 3.2.3. All technologies producing an energy carrier are included with a positive coefficient, while the ones consuming this energy form are included with a negative coefficient. The relation between the coefficients of a technology in different energy flow constraints are defined by the efficiency of that technology. The balances for the energy forms on the demand level contain the demand as right hand side (chapter 3.2.1). Domestic resources and imports are included as suppliers and exports as additional consumers (chapter 3.2.6). In *STIM* gasoline import is included in the balance for gasoline on level *A*.

For heat pumps a dynamic constraint on the build up of new capacity is included. The model can start to use heat pumps from a certain year on with an initial size (given by the start-up parameter, see chapter 3.2.26) and then increase the annual new installations by a given percentage per year. Hydroelectric power plants have a lower dynamic constraint on the activity ensuring a production that does not decline over the time horizon. Analogous constraints can be used for resource extraction, imports and exports.

In addition to this fixed structure the user can define relations himself. The first entries in block 3 guarantee an overcapacity for district heat and electricity production plants of 20% and 30%, respectively. The others ensure that the use of district heat does not decline in relation to the demand, count the production of SO_2 and NO_x from central systems, force peak power plants into the system, and limit the co-generation plant to operate more than 4000 and less than 6000 hours per year.

3. The Mathematical Formulation of MESSAGE II

During model development, which was, as indicated above, initially directed towards an energy supply model, the matrix generator was changed several times in order to allow for a more general applicability. Now *MESSAGE II* is also used for the formalization of dynamic input/output models and progress is made in defining a consumer demand model. For applications at institutes with small-scale computers a decomposition algorithm is being developed, that allows a problem to be solved iteratively in several steps [5].

This chapter describes the system of linear equations that can be generated with the matrix generator of *MESSAGE II*. They are explained here in view of the energy supply model. The first subsection contains a definition of the variables (COLUMNS), the second one the equations (ROWS). The notation of the variable and equation names is the same as in the produced matrix, upper case letters give fix identifiers, while small letters are defined by the user or varied over a set of characters.

The variables and equations of *MESSAGE II* will be explained for the default values of the model with the level definition given in figure 2. This can, of course, be changed by the user. The energy form level '*U*' has special features that are simply related to the identifier and can be avoided by renaming this level.

In order to keep the notation simple and the mathematical description as short as possible some complicated features are omitted from the following formulation and described in an extra section, chapter 3.3.

3.1 Definition of the Variables (COLUMNS)

The variables of *MESSAGE II* can be grouped into three categories:

1) Energy flow variables representing an amount of energy. The unit is usually MWyr for small regions and GWyr for bigger areas,

2) Power variables that stand for the production capacity of a certain technology (usual unit: MW or GW), and

3) Stock-piles representing the amount of a fuel being cumulated at a certain point in time (usual unit: MWyr or GWyr).

3.1.1 Activities of Energy Conversion Technologies

Usvdue.t, and
Zsvdu.lt,

where

U identifies the end-use level and *Z* the other levels (i.e. $Z \in \{F, T, X, A\}$),
s is the main energy input of the technology (supply), if none exists *s* = '.' (e.g., solar technologies),
v identifies the conversion technology,
d is the main energy output of the technology (demand),
u identifies the energy density area; if none are defined or the technology is centralized, *u* = '.',
e is the level of reduction of demand due to own-price elasticities of demands (does only occur on the demand level, otherwise or if this demand has no elasticities *e* = '.'),
l identifies the load region, $l \in \{1,2,3,...\}$ or *l* = '.', if the technology is not modeled with load regions, and
t identifies the period, $t \in \{a, b, c, ...\}$.

The activity variable of an energy conversion technology is an energy flow variable. It contains the annual consumption of the main input per period. If a technology has no input, the variable is related to the main output. It can exist for several energy density areas (*u*) if the output of the technology exists for energy density areas and load regions (*l*) if the main energy input or output is modeled with load regions and the production pattern of the technology is not fixed--see chapter 3.3.4. For technologies serving a demand category, it can also exist for the elasticity classes (*e*) (see chapter 3.3.12).

3.1.2 Activities of Energy Storage Technologies

a) Input to Energy Storage Technologies

SIZsvult,

where

SI identifies the storage input variables,
s is the energy form to be stored,
Z identifies the level on that the energy form is defined (i.e. $Z \in \{F, T, X, A\}$),
v identifies the storage technology,
u identifies the energy density area,
l identifies the load region in that the energy is stored, and
t is the period identifier.

The storage input variables are energy flow variables and contain the amount of fuel s that is stored in the technology v in load region l and period t.

b) Output from Energy Storage Technologies

$OZsvulmt$,

where

O identifies the storage output variables,
s is the energy form stored in the technology,
Z identifies the level on that the energy form is defined (i.e. $Z \in \{F,T,X,A\}$),
v identifies the storage technology,
u identifies the energy density area,
l is the load region in that the energy was stored,
m is the load region in that the energy is retrieved, and
t is the period identifier.

The storage output variables are energy flow variables and contain the amount of fuel s that was stored in load region l and is retrieved in load region m.

3.1.3 Capacities of Energy Conversion Technologies

$YZsvdu.t$,

where

Y is the identifier for capacity variables,
Z identifies the level on that the main energy output of the technology is defined,
s is the main energy input of the technology,
v identifies the conversion technology,
d is the main energy output of the technology,
u is the energy density area, and
t is the period in that this capacity is built.

The capacity variables are power variables. In the current version of *MESSAGE II* they are the only ones that can be defined as integer variables (see chapter 3.3.14), which turned out to be sufficient in all applications so far.

If they are continuous the capacity variables contain the annual new installations of the technology in period t, if they are integer they contain either the annual number of installations of a certain size or the number of installations of $1/\Delta t$ times the unit size (Δt being the length of period t in years). The capacity is measured in units of main output of the technology.

3.1.4 Capacities of Energy Storage Technologies

a) Input/Output Capacity of Storage Technologies

$$YZGsvu.t,$$

where

Y identifies the capacity variables,
G identifies the I/O capacity variables for storage technologies (generation capacity),
s is the energy form stored in this technology,
Z is the level on that this energy form is defined,
u is the energy density area, and
t is the period in that the new capacity is built.

The storage I/O capacity variables are power variables and contain the annual construction of capacity to fill and empty the storage. They can be continuous or integer like the conversion technology capacity variables.

b) Volume Capacity of Storage Technologies

$$YZVsvu.t,$$

where

V identifies the volume capacity variables for storage technologies, and
the other identifiers are the same as under a).

The storage volume capacity variables are stock-pile variables and have an energy- and not power-related unit. They contain the annual new installation of the 'container'. They can be continuous or integer like the conversion technology capacity variables.

3.1.5 Use of Domestic Resources

$$RZrgp..t\,,$$

where

R identifies the resource extraction variables,
Z is the level on that the resource is defined (usually = R),
r is the resource being extracted,
g is the grade (also called cost category), dependent on r,
p is the class of supply elasticity, dependent on $g(r)$ (see chapter 3.3.13), and
t identifies the period.

The resource variables are energy flow variables and contain the annual rate of extraction of resource r, grade g and elasticity class p.

3.1.6 Imports

$$IZscp.lt\,,$$

where

I identifies the import variables,
Z is the level on that the imported energy form is defined (usually A for primary energy and X for secondary energy),
s identifies the imported energy form,
c is the country the imports come from, dependent on s,
p identifies the price category of import, dependent on $c(s)$, and
l is the load region identifier if s is modeled with load regions, otherwise '.'.

The import variables are energy flow variables and contain the annual import of the identified fuel per country and elasticity class and, if the fuel has load regions, per load region.

3.1.7 Exports

$EZrcp.lt$,

where

E is the identifier for export variables, and
all the other identifiers are the same as for imports.

The export variables contain the annual rate of export of the identified fuel per country and elasticity class and, if r is modeled with load regions, load region.

3.1.8 Stock-pile of Fuels

$Qfb....t$,

where

Q identifies stock-pile variables,
f identifies the fuel with stock-pile,
b distinguishes the variable from the equation, and
t is the period identifier.

The stock-pile variables are, as the name says, stock-pile variables and contain the amount of fuel f that is available in period t. Note that these variables do not refer to the years in the period, but to the period as a whole.

3.1.9 Base Load Relocation Variables

$ZPsu...t$,

where

Z is the level on that the energy form is defined,
P identifies base load relocation variables,
s is the identifier of the fuel,
u is the energy density area or '.', if energy density areas are not modeled or the energy forms on level Z do not have energy density areas, and
t is the period.

The base load relocation variables are energy flow variables and contain the amount of base load energy form *s* that is relocated from the normal vertical load regions to the horizontal base load load region (see chapters 3.2.4, 3.2.5 and 3.3.6)

3.2 Description of the Equations (ROWS)

As already mentioned, this description makes full use of all built-in level identifiers. Levels can be omitted or their default meaning changed by changing the identifier of the according level.

3.2.1 Demand Constraints

Udu....t

Out of the predefined levels (see figure 2) each one can be chosen as demand level. However, level '*U*' has a special feature. This is related to the fact that useful energy is usually produced on-site, e.g., space heat is produced by a central heating system, and the load variations over the year are all covered by this one system. Thus, an allocation of production technologies to the different areas of the load curve, like the model would set it up according to the relation between investment and operating costs would ignore the fact that these systems are not located in the same place and are not connected to each other. *MESSAGE II* represents the end-use technologies by one variable per period that produces the required useful energy in the load pattern needed and requires the inputs in the same pattern. For special technologies like, e.g., night storage heating systems, this pattern can be changed to represent the internal storage capability of the system.

This representation of end-use technologies has the advantage of reducing the size of the model, because the demand constraints, the activity variables and the capacity constraints of the end-use technologies do not have to be generated for each load region.

If another level is chosen as demand level or the levels are renamed (see chapter 3.2.2), all demand constraints for energy carriers that are modeled with load regions are generated for each load region. The demand constraints are always produced for each energy density area. The general form of the demand constraints is

$$\sum_{svd} \varepsilon_{svd} \times \sum_{e=0}^{e_d} k_e \times Usvdue.t + \sum_{sv\delta} \beta^d_{sv\delta} \times \sum_{e=0}^{e_\delta} k_e \times Usv\delta ue.t \geq Udut \quad ,$$

where

Udut is the annual demand for d in energy density area u and period t,
Usvdue.t is the activity of end-use technology v in period t, energy density area u, elasticity class e and period t (see chapter 3.1.1),
ε_{svd} is the efficiency of end-use technology v in converting s to d,
$\beta^d_{sv\delta}$ is the efficiency of end-use technology v in producing by-product d from s (δ is the main output of the technology),
e_d is the number of steps of demand reduction modeled for own-price elasticities of demand d, and
k_e is the factor giving the relation of total demand for d to the demand reduced to level e due to the demand elasticity.
($k_e \times Usvdue.t = Usvdu0.t$, $k_0 = 1$, k_e is increasing monotonously.)

3.2.2 Distribution Balance

Fsu...lt

This constraint, the final energy balance, matches the use of final energy needed in the end-use technologies and the deliveries of the distribution systems. It is generated for each energy density area, if the energy forms on level F are defined with energy density areas and for each load region, if energy form s is modeled with load regions.

$$\sum_{svs} \varepsilon_{svs} \times Fsvsu.lt - \sum_{svd} \eta_{d.l} \times \sum_{e=0}^{e_d} Usvdue.t -$$

$$\sum_{\sigma vd} \beta^s_{\sigma vd} \times \eta_{d.l} \times \sum_{e=0}^{e_d} U\sigma vdue.t \geq 0 \quad ,$$

where

Fsvsu.lt is the activity of the distribution technology in energy density area u, load region l and period t (see chapter 3.1.1),
ε_{svs} is the efficiency of technology v in distributing s,
Usvdue.t is the activity of end-use technology v in period t, energy density area

u, and elasticity class e,
$\beta^{s}_{\sigma v d}$ is the use of fuel s relative to fuel σ (the main input) by technology v, and
$\eta_{d,l}$ is the fraction of demand for d occurring in load region l.

3.2.3 Transmission or Transportation Balance

Tsu...lt

This constraint gives the simplest form of an energy balance equation of ***MESSAGE II***. It matches the output of transmission to the requirements of distribution systems. The difference to other levels (F,X,A) is not built-in, but emerges from the simplicity of energy transportation (i.e., transportation technologies do usually not have by-products and only one input). Storage could, for instance, be located on this level as well. Also big industrial consumers that are directly connected to the transmission system would have to be included in this constraint. Like level F it does usually exist for all energy density areas and load regions if they are defined for the fuel. Level T is omitted in *STIM*.

$$\sum_{svs} \varepsilon_{svs} \times Tsvsu.lt - \sum_{svs} Fsvsu.lt \geq 0 .$$

where

$Tsvsu.lt$ is the activity of the transportation technology v (see chapter 3.1.1), and all the other entries to the equation are the same as in chapter 3.2.2.

3.2.4 Central Conversion and Storage Balance

Xs....lt

In principle the secondary energy balance is built up in the same way as the two previous ones (chapters 3.2.2 and 3.2.3). It matches the production of central conversion technologies to the requirements of the transmission systems and is given here mainly to explain the introduction of storage. Additionally secondary energy imports and exports of secondary energy are usually assigned to level X.

An addition can be made for energy forms with load regions: the reassignment of base load energy to a special base load load region, a feature that is helpful if the model is not used from primary, but only from secondary to useful energy (see also chapters 3.1.9, 3.2.5 and 3.3.6). However, this relocation does not necessarily have to be assigned to level X and should not be used if production and central conversion of energy is included in the model.

$$\sum_{rvs} \varepsilon_{rvs} \times Xrvs..lt + \sum_{rv\sigma} \beta^{s}_{rv\sigma} \times Xrv\sigma..lt - \sum_{u} \times \sum_{svs} Tsvsu.lt + \sum_{c,p} IXscp.lt -$$

$$\sum_{c,p} EXscp.lt + \sum_{sv} \times \left[\sum_{m=l-m_{sv}}^{l-1} \varepsilon_{sv} \times OXsv.mlt - SIXsv.lt \right]$$

$$\left\{ \begin{matrix} + \lambda_l \times (1-\pi_s) \times XPs....t \\ - \qquad (1-\pi_s) \times XPs....t \end{matrix} \right\} \geq 0, \quad \begin{matrix} l = peak \\ l = base\ load \end{matrix} ,$$

where

$Xrvs..lt$ is the activity of central conversion technology v in load region l and period t (see chapter 3.1.1); if the secondary energy form s is not defined with load regions (i.e. l = '.') and the activity of technology v exists for each load region, this equation will contain the sum of the activity variables of technology v over the load regions.

ε_{rvs} is the efficiency of technology v in converting energy carrier r into secondary energy form s,

$\beta^{s}_{rv\sigma}$ is the efficiency of technology v in converting energy carrier r into the by-product s of technology v,

$Tsvsu.lt$ is explained in chapter 3.2.3, and

$IXscp.lt$ and $EXscp.lt$ are the import and export variables explained in chapters 3.1.6 and 3.1.7, respectively.

The following entries can only exist, if energy carrier s is modeled with load regions:

$OXsv.mlt$ and $SIXsv.lt$ are the activity variables for storage technology v as described in chapter 3.1.2,

ε_{sv} is the I/O efficiency of storage technology v,

λ_l is the length of load region l (as fraction of the year),

π_s is the minimum fraction of demand for s that remains peak,

m_{sv} is the number of load regions that storage technology v can keep the content (see chapter 3.3.5), and

$XPs....t$ is the base load relocation variable, giving the amount of base load energy that is redistributed from peak load regions to the base load load region and therefore base load supply (see also chapters 3.1.9, 3.2.5 and 3.3.6).

3.2.5 Base Load Balance for Imports

$XPs...lt$

These balance equations establish the relation between peak and base load energy use if this option is applied. It will be generated for all energy forms that have load regions on the specified level (usually X).

$$\lambda_l \times XPs....t + \sum_{sv} \times \left[\sum_{m=l-m_{sv}}^{l-1} \varepsilon_{sv} \times 0Xsv.mlt - SIXsv.lt \right] -$$

$$\sum_{u} \times \sum_{svs} Tsvsu.lt \leq 0$$

All entries to this equation are described in chapter 3.2.4.

3.2.6 Resource Extraction, Export and Import Balance

$Ar.....t$

This equation matches production and import of primary energy to the requirements of central conversion, transport and for export. In the general case primary energy does have neither energy density areas nor load regions. Some technologies, like, e.g., nuclear reactors need inventories of primary energy and also leave a last core that is available at the end of the lifetime. It may be necessary to model by-products of extraction technologies, for instance the availability of associated gas at oil production sites.

$$\sum_{rvr} \varepsilon_{rvr} \times Arvr\ldots t - \sum_{l} \times \left[\sum_{rvs} Xrvs..lt + \sum_{pvs} \beta^{r}_{pvs} \times Xpvs..lt \right] + \sum_{c,p} IArcp..t -$$

$$\sum_{c,p} EArcp..t + \sum_{fvs} \times \left[\frac{\Delta(t-\tau_{fvs})}{\Delta t} \times \rho(fvs,r) \times YXfvs..(t-\tau_{fvs}) - \right.$$

$$\left. \frac{\Delta(t+1)}{\Delta t} \times \iota(fvs,r) \times YXfvs..(t+1) \right] \geq 0 \, ,$$

where

$Arvr\ldots t$ is the activity of technology v extracting resource r,

ε_{rvr} is the efficiency of technology v in extracting fuel r (this is usually 1.),

β^{r}_{pvs} is the efficiency of technology v in producing secondary energy form s from the by-input p,

$IArcp..t$ and $EArcp..t$ are the import and export variables described in chapter 3.1.6 and 3.1.7, respectively,

τ_{fvs} is the plant life of technology v in periods (depending on the lengths of the periods covered),

$YXfvs..t$ is the annual new installation of technology v in period t (see chapter 3.1.3),

$\iota(fvs,r)$ is the amount of fuel r that is needed when technology v goes into operation (usually this is the first core of a reactor). It has to be available in the period before technology v goes into operation, the normal unit is kWyr/kW,

$\rho(fvs,r)$ is the amount of fuel r that becomes available after technology v goes out of operation (for a reactor this is the last core that goes to reprocessing). The unit is the same as for $\iota(fvs,r)$, and

Δt is the length of period t in years.

3.2.7 Resource Consumption

Rr.....t

The resources produced by the extraction technologies in a period can come from different cost categories (also called grades), which can, e.g., represent the different effort to reach certain resources. Short-term variations in price due to steeply increasing demand can be represented by an elasticity approach (see chapter 3.3.13).

$$\sum_{g,p} RRrgp..t - \sum_{rvr} Arvr...t \geq 0 \ ,$$

where

RRrgp..t is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t (see chapter 3.1.5), and
Arvr...t is the activity of extraction technology v in period t (as described in chapter 3.1.1).

3.2.8 Resource Availability per Grade

RRrg.g..

Limits the domestic resources available from one cost category (grade) over the whole time horizon.

$$\sum_{p} \times \sum_{t} \Delta t \times RRrgp..t \leq Rrg \ .$$

where

Rrg is the total amount of resource r, cost category g, that is available for extraction,
RRrgp..t is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t, and
Δt is the length of period t.

3.2.9 Maximum Annual Resource Extraction

RRr....t

Limits the domestic resources available annually per period over all cost categories.

$$\sum_{g} \times \sum_{p} RRrgp..t \le Rrt \ ,$$

where

$Rrgt$ is the maximum amount of resource r, grade g, that can be extracted per year of period t, and

$RRrgp..t$ is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t.

3.2.10 Resource Depletion Constraints

RRrg.d.t

The extraction of a resource in a period can be constrained in relation to the total amount still existing in that period. For reasons of computerization these constraints can also be generated for imports and exports, although they do not have any relevance there (they could, e.g., be used for specific scenarios in order to stabilize the solution).

$$\Delta t \sum_{p} RRrgp..t \le \delta_{rg}^{t} \left(Rrg - \sum_{\tau=1}^{t-1} \Delta\tau \times RRrgp..\tau \right),$$

where

Rrg is the total amount of resource r, cost category g, that is available for extraction,

$RRrgp..t$ is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t,

δ_{rg}^{t} is the maximum fraction of resource r, cost category g, that can be extracted in period t,

Rrg is the total amount available in the base year, and

Δt is the length of period t in years.

3.2.11 Maximum Annual Resource Extraction per Grade

$RRrg.a.t$

Limits the domestic resources available from one cost category per year.

$$\sum_{p} RRrgp..t \le Rrgt \ .$$

where

Rrg is the total amount of resource r, cost category g, that is available for extraction, and

$RRrgp..t$ is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t.

3.2.12 Upper Dynamic Resource Extraction Constraints

$MRRr...t$

The annual extraction level of a resource in a period can be related to the previous one by a growth parameter and an increment of extraction capacity resulting in upper dynamic extraction constraints. For the first period the extraction is related to the activity in the base year.

$$\sum_{g,p} RRrgp..t - \gamma_{rt}^{o} \times \sum_{g,p} RRrgp..(t-1) \le g_{rt}^{o} \ .$$

where

γ_{rt}^{o} is the maximum growth of extraction of resource r between period $t-1$ and t,
g_{rt}^{o} is the initial size (increment) of extraction of resource r in period t, and
$RRrgp..t$ is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t.

3.2.13 Lower Dynamic Extraction Constraints

$LRRr...t$

The annual extraction level of a resource in a period can also be related to the previous one by a decrease parameter and a decrement resulting in lower dynamic extraction constraints. For the first period the extraction is related to the activity in the base year.

$$\sum_{g,p} RRrgp..t - \gamma_{rt}^{u} \times \sum_{g,p} RRrgp..(t-1) \geq -g_{rt}^{u} \ ,$$

where

γ_{rt}^{u} is the maximum decrease of extraction of resource r between period $t-1$ and t,
g_{rt}^{u} is the 'last' size (decrement) of extraction of resource r in period t, and
$RRrgp..t$ is the annual extraction of resource r, cost category (grade) g and elasticity class p in period t.

3.2.14 Dynamic Extraction Constraints per Grade

$MRRrg..t$, and
$LRRrg..t$

The same kind of relations as described in chapters 3.2.12 and 3.2.13 can be defined per grade of the resource.

3.2.15 Imports per Country

$IArc.g..$

Limits the imports of a fuel from a specific country c over the whole horizon.

$$\sum_{p} \times \sum_{t} \Delta t \times IArcp..t \leq Irc \ ,$$

where

Irc is the total import limit for r from country c,
$IArcp..t$ is the annual import of r from country c, elasticity class p in period t, and
Δt is the length of period t in years.

3.2.16 Maximum Annual Imports

$IAr....t$

Limits the annual imports of a fuel from all countries per period.

$$\sum_{c} \times \sum_{p} IArcp..t \leq Irt \ ,$$

where

Irt is the annual import limit for r in period t, and
$IArcp..t$ is the annual import of r from country c, elasticity class p in period t.

3.2.17 Maximum Annual Imports per Country

$IArc.a.t$

Limits the imports from one country per year.

$$\sum_{p} IArcp..t \leq Irct \ ,$$

where

$Irct$ is the limit on the annual imports from country c, period t of fuel r, and
$IArcp..t$ is the annual import of r from country c, elasticity class p in period t.

3.2.18 Upper Dynamic Import Constraints

$MIAr\ldots t$

The annual import level of a fuel in a period can, like the resource extraction, be related to the previous one by a growth parameter and an increment resulting in upper dynamic constraints.

$$\sum_{c,p} IArcp..t - \gamma^{o}_{rt} \times \sum_{c,p} IArcp..(t-1) \le g^{o}_{rt} ,$$

where

$IArcp..t$ is the annual import of r from country c, elasticity class p in period t,
γ^{o}_{rt} is the maximum increase of import of r between period $t-1$ and t, and
g^{o}_{rt} is the initial size (increment) of import of r in period t.

3.2.19 Lower Dynamic Import Constraints

$LIAr\ldots t$

The annual import level of a fuel in a period can also be related to the previous one by a decrease parameter and a decrement resulting in lower dynamic import constraints.

$$\sum_{c,p} IArcp..t - \gamma^{u}_{rt} \times \sum_{c,p} IArcp..(t-1) \ge -g^{u}_{rt} ,$$

where

$IArcp..t$ is the annual import of r from country c, elasticity class p in period t,
γ^{u}_{rt} is the maximum decrease of import of r between period $t-1$ and t, and
g^{u}_{rt} is the 'last' size (decrement) of import of r in period t.

3.2.20 Dynamic Import Constraints per Country

MIArc..t and
LIArc..t

The same kind of relations can be defined per country from that the fuel is imported.

3.2.21 Constraints on Exports

The exports of fuels can principally be limited in the same way as the imports. In the identifiers of the variables and constraints the '*I*' is substituted by an '*E*'.

3.2.22 Storage Balance

SXsv..lt

Chapter 3.3.5 describes the background of the implementation of energy storage in *MESSAGE II*. In the storage balances the energy flows into and out of the storage technologies are balanced. *MESSAGE II* does keep track of the time that a certain amount is stored by using a separate storage output variable for each pair of input and output load regions (see also chapter 3.1.2). In the following two examples are given; the equations differ for different kinds of storage (e.g., daily, weekly, seasonal).

a) daily storage

$$\varepsilon_{sv} \times SIXsv.lt - \sum_{m=l+1}^{l+m_{sv}^{l}} \frac{1}{\zeta_{l,m}} \times OXsv.lmt \geq 0 \ ,$$

b) seasonal storage

$$\varepsilon_{sv} \times SIXsv.lt - \sum_{m=l+1}^{l+m_{sv}} f_{l,m}^{1} \times \frac{1}{\zeta_{l,m}} \times OXsv.lmt -$$

$$\sum_{m=l+1}^{l+m_{sv}} f_{l,m}^{2} \times \frac{1}{\zeta_{l,m}} \times OXsv.lm(t+1) \geq 0\,,$$

where

$f_{l,m}$ forwards the appropriate amount of fuel to the next period (this is important for small time steps, for instance $\Delta t = 1$),

$$f_{l,m}^{1} = \begin{cases} 1 & \text{for } l < m \\ \dfrac{\Delta t - 1}{\Delta t} & \text{for } l > m\,, \end{cases}$$

$$f_{l,m}^{2} = \begin{cases} 0 & \text{for } l < m \\ \dfrac{1}{\Delta(t+1)} & \text{for } l > m\,, \end{cases}$$

$SIXsv.lt$ is the amount of fuel s put into storage v in load region l,

$OXsv.lmt$ is the amount of fuel s taken out of storage v in load region m, which was put into storage in load region l,

ε_{sv} is the efficiency of putting fuel s into storage v (e.g. the pumping losses in pumped hydro storage plants can be accounted for this way),

m_{sv}^{l} is the number of load regions that the fuel can be stored. It depends on the kind of storage (for daily storage it is the number of load regions that represent one day, for seasonal storage the whole year, therefore all load regions) and if there is an explicit limit given (e.g., the temperature inside a heat storage can fall below the level where it still can be retrieved after a certain time),

$\zeta_{l,m}$ is the decrease of storage contents from load region l to load region m, used for heat storage (exponential decay), and

Δt is the length of period t in years.

3.2.23 Capacity of Conversion Technologies

CZsvd.lt

For all conversion technologies the capacity constraints will be generated for as many load regions as the activity variables are generated for (see chapters 3.1.1 and 3.3.4). If a technology is defined to exist in several energy density areas, the capacity constraints will be generated for each of these energy density areas. If the technology is an end-use technology the sum over the elasticity classes will be included in the capacity constraint.

Additionally different types of activity variables can be linked to the same capacity variable, resulting in a variable production pattern, which leave the choice of the operation mode open for the model (see 3.2.23 d and 3.3.9)).

a) Technologies without load regions

For technologies without load regions the installed capacity is only related to the production by the plant factor, i.e. the time the technology runs per year. All end-use technologies (technologies on level '*U*') are included in this manner. Thus for these technologies the plant factor has to give the fraction they actually operate per year.

$$\varepsilon_{svd} \times Zsvd...t - \sum_{\tau=t-\tau_{svd}}^{min(t,\kappa_{svd})} \Delta\tau \times \pi_{svd} \times f_i \times YZsvd..\tau \le hc_{svd}^t \times \pi_{svd} .$$

b) Technologies with load regions and "free" production pattern

Here the installed capacity is related to the production in each load region and therefore defined by the highest capacity utilization. The plant factor gives the fraction of operating time in peak operation mode (in general this is the availability factor). Maintenance times can be included by using user defined relations, if necessary; also a minimum operation time can be given this way (see chapter 3.2.30).

$$\frac{\varepsilon_{svd}}{\lambda_l} \times Zsvd..lt - \sum_{\tau=t-\tau_{svd}}^{min(t,\kappa_{svd})} \Delta\tau \times \pi_{svd} \times f_i \times YZsvd..\tau \le hc_{svd}^t \times \pi_{svd} .$$

c) Technologies with load regions and "fixed" production pattern

The production pattern of a technology that has load regions can be fixed (e.g. nuclear and solar technologies) to a certain shape. The plant factor has the same meaning as in case b), but the activity of the technology and thus the capacity constraint does only exist once per period.

$$\frac{\varepsilon_{svd} \times \pi(l_m, svd)}{\lambda_{l_m}} \times Zsvd\ldots t \; -$$

$$\sum_{\tau=t-\tau_{svd}}^{min(t,\kappa_{svd})} \Delta\tau \times \pi_{svd} \times f_i \times YZsvd..\tau \le hc^{t}_{svd} \times \pi_{svd} \; .$$

d) Technologies with Varying Inputs and Outputs

Many types of energy conversion technologies do not have fix relations between their inputs and outputs. Therefore *MESSAGE II* foresees the option of linking several activity variables of conversion technologies together in one capacity constraint. Here this constraint is only described for technologies without load regions; the other types are constructed in an analogous way (see also chapter 3.3.9).

$$\sum_{\sigma v'\delta} rel^{svd}_{\sigma v'\delta} \times \varepsilon_{\sigma v'\delta} \times Z\sigma v'\delta\ldots t \; -$$

$$\sum_{\tau=t-\tau_{svd}}^{min(t,\kappa_{svd})} \Delta\tau \times \pi_{svd} \times f_i \times YZsvd..\tau \le hc^{t}_{svd} \times \pi_{svd} \; .$$

where

$Zsvd..lt$ is the activity of conversion technology v in period t and, if defined so, load region l, see chapter 3.1.1,

$YZsvd..t$ is the capacity variable of conversion technology v (see chapter 3.1.3),

ε_{svd} is the efficiency of technology v in converting the main energy input, s, into the main energy output, d,

κ_{svd} is the last period in that technology v can be constructed,

π_{svd} is the "plant factor" of technology v, having different meaning depending on the type of capacity equation applied (this is described in the input description, chapter 4.2.3),

$\Delta\tau$ is the length of period τ in years,
τ_{svd} is the plant life of technology v in periods,
hc^{t}_{svd} represents the installations built before the time horizon under consideration, that are still in operation in period t. If installations go out of operation within a period, their operation capacity is reduced to the share of capacity that still operates on the average in that period,
f_i is 1. if the capacity variable is continuous, and contains the minimum installed capacity per year (unit size) if the variable is integer,
l_m is the load region with maximum capacity use if the production over the year is fixed,
$\pi(l_m, svd)$ is the share of output in the load region with maximum production,
$rel^{svd}_{\sigma v'\delta}$ is the relative capacity of main output of technology (or operation mode) svd to the capacity of main output of technology (or operation mode) $\sigma v'\delta$,
λ_l is the length of load region l as fraction of the year, and
λ_{l_m} is the length of load region l_m, the load region with maximum production, as fraction of the year.

3.2.24 Input/Output Capacity of Storage

CXGsv.lt

This equation defines the capacity of storing or releasing energy per unit of time in a certain storage technology.

$$\frac{\varepsilon_{sv}}{\lambda_l} \times \left[SIXsv.lt + \sum_{m=l-m_{sv}}^{l-1} OXsv.mlt \right] -$$

$$\sum_{\tau=t-\tau_{sv}}^{\min(t,\kappa_{sv})} \pi_{sv} \times \Delta\tau \times f_i \times YXGsv..\tau \leq hc^{t}_{sv,G} \times \pi_{sv} \ .$$

where

SIXsv.lt and *OXsv.mlt* are the flows into and out of the storage technology v, as described in chapters 3.1.2 and 3.2.22.
YXGsv..τ is the generation capacity of storage v as described in chapter 3.1.4,
ε_{sv} is the efficiency of storage technology v,
λ_l is the length of load region l as fraction of the year,
κ_{sv} is the last period in that technology v can be constructed,

π_{sv} is the plant factor of technology v,
$\Delta\tau$ is the length of period τ in periods,
τ_{sv} is the plant life of technology v in years,
$hc^{t}_{sv,G}$ represents the installations built before the time horizon under consideration, that are still in operation in period t. If installations go out of operation within a period, their operation capacity is reduced to the share of capacity still operating on the average in that period,
f_i is 1. if the capacity variable is continuous, and equal to the minimum installed capacity per year (unit size) if the variable is integer.

3.2.25 Volume Capacity of Storage

CXVsv.lt

The amount of energy that can be stored (the maximum content at a time) can either be linked to the I/O capacity or evaluated endogenously in the model. Thus either a predefined storage technology like batteries can be modeled or the model can have the choice to optimize the relation between I/O capacity and storage volume.

$$\sum_{m=l-m^{l}_{sv}}^{l} \zeta_{m,l} \times \left[\varepsilon_{sv} \times SIXsv.mt - \sum_{n=m_{sv}+1}^{l} \frac{1}{\zeta_{m,n}} \times OXsv.mnt \right] \times$$

$$\frac{1}{nl \times \lambda_l} - \left\{ \begin{matrix} \sum\limits_{\tau=l-\tau_{sv}}^{min(l,\kappa_{sv})} \Delta\tau \times \pi_{sv} \times f_i \times YXVsv..\tau \\ \sum\limits_{\tau=l-\tau_{sv}}^{min(l,\kappa_{sv})} \Delta\tau \times \pi_{sv} \times f_{gv} \times f_i \times YXGsv..\tau \end{matrix} \right\} \leq hc^{l}_{sv,V} \times \pi_{sv} \ ,$$

where

SIXsv.lt and *OXsv.mlt* are the flows into and out of the storage technology v, as described in chapters 3.1.2 and 3.2.22,
$YXGsv..\tau$ is the generation capacity of storage v as described in chapter 3.1.4,
f_{gv} is the relation of I/O to volume capacity,
$YXVsv..\tau$ is the volume capacity variable as described in chapter 3.1.4,
nl is the number of occurrences per year (1 for seasonal, 365 for daily, etc.),
$hc^{t}_{sv,V}$ represents the installations built before the time horizon under consideration, that are still in operation in period t. If installations go out of operation within a period, their operation capacity is reduced to the

share of capacity still operating on the average in that period,

f_i is 1. if the capacity variable is continuous, equal to the minimum installed capacity per year (unit size) if the variable is integer,

$\zeta_{m,l}$ is the decrease parameter as described in chapter 3.2.22, and

m^l_{sv} is described in chapter 3.2.22.

3.2.26 Upper Dynamic Constraints on New Built Capacities

MYZsvdut

The dynamic capacity constraints relate the amount of annual new installations of a technology in a period to the ones built annually during the previous period.

$$YZsvdu.t - \gamma y^{o}_{svd,u,t} \times YZsvdu.(t-1) \leq gy^{o}_{svd,u,t},$$

where

$\gamma y^{o}_{svd,u,t}$ is the maximum growth rate for the construction of technology v,

$gy^{o}_{svd,u,t}$ is the initial size (increment) that can be given for the introduction of new technologies,

$YZsvdu.t$ is the annual new installation of technology v in energy density area u and period t.

3.2.27 Lower Dynamic Constraints on New Built Capacities

LYZsvdut

$$YZsvdu.t - \gamma y^{u}_{svd,u,t} \times YZsvdu.(t-1) \geq -gy^{u}_{svd,u,t},$$

where

$\gamma y^{u}_{svd,u,t}$ is the minimum growth rate for the construction of technology v,

$gy^{u}_{svd,u,t}$ is the 'last' size (decrement) allowing technologies to go out of the market, and

$YZsvdu.t$ is the annual new installation of technology v in energy density area u and period t.

3.2.28 Upper Dynamic Constraints on Production

$MZsvdu.t$

The dynamic production constraints relate the activity of a technology in a period to the one in the previous period.

$$\sum_{l} \varepsilon_{svd} \times \left[Zsvdu.lt - \gamma a^{o}_{svd,u,t} \times Zsvdu.l(t-1) \right] \leq ga^{o}_{svd,u,t},$$

where

$\gamma a^{o}_{svd,u,t}$ and $ga^{o}_{svd,u,t}$ are the maximum growth rate and increment as described in chapter 3.2.26, and
$Zsvdu.lt$ is the activity of technology v in load region l, energy density area u.

If demand elasticities are modeled, the according summations are included for end-use technologies.

3.2.29 Lower Dynamic Constraints on Production

$LZsvdu.t$

$$\sum_{l} \varepsilon_{svd} \times \left[Zsvdu.lt - \gamma a^{u}_{svd,u,t} \times Zsvdu.l(t-1) \right] \geq -ga^{u}_{svd,u,t},$$

where

$\gamma a^{u}_{svd,u,t}$ and $ga^{u}_{svd,u,t}$ are the maximum growth rate and increment as described in chapter 3.2.26, and
$Zsvdu.lt$ is the activity of technology v in load region l, energy density area u.

3.2.30 User-Defined Relations

$Nm....lt$ or $Pm....lt$

These relations allow the user to construct equations that are not included in the basic set of constraints. For each technology the user can specify coefficients with that either the production variables (see chapter 3.1.1), the annual new installation variables (see chapter 3.1.3) or the total capacity in a year (like it is used in the capacity constraints, see chapter 3.2.23) can be included into that relation. The relations can be defined with and without load regions, have a lower, upper or fix right hand side and be related to an entry in the objective function, i.e., all entries to this relation are also entered to the objective function with the appropriate discount factor. There are two types of user/defined constraints, for which the entries to the objective function--without discounting--are summed up under the cost accounting rows *CAR*1 and *CAR*2.

The user defined relations can be defined with load regions. Then all entries of activities of technologies with load regions are divided by the length of the according load region resulting in a representation of the utilized power.

Relation without load regions

$$\sum_{u} \times \left\{ \sum_{svd} \times \left[ro_{svd}^{mt} \times \sum_{e=0}^{e_d} Usvdue.t \times \varepsilon_{svd} + \sum_{\tau=t-ip}^{t} rc_{svd}^{mt} \times YUsvdu.\tau \right] + \right.$$

$$\sum_{rvs} \times \left[ro_{rvs}^{mlt} \times \sum_{l} Zrvsu.lt \times \varepsilon_{rvs} + ro_{rvs}^{mt} \times Zrvsu..t \times \varepsilon_{rvs} + \right.$$

$$\left. \sum_{\tau=t-ip}^{t} rc_{rvs}^{mt} \times YZrvsu.\tau \right] + \sum_{sv} \times \sum_{\tau=t-ip}^{t} \left[rg_{sv}^{mt} \times tl \times YZGsv..\tau + \right.$$

$$\left. rv_{sv}^{mt} \times YZVsv..\tau \right] \Bigg\} \begin{cases} free \\ \geq rhs_m^t \\ = rhs_m^t \\ \leq rhs_m^t \end{cases},$$

where

$Usvdue.t$ and $YUsvdu.t$ are the activity and capacity variables of the end-use technologies,

$Zrvsu.lt$, $Zrvsu..t$ and $YZrvsu.t$ are the activity variables of technologies with and without load regions and the capacity variables of the technologies,

ε_{rvs} and ε_{svd} are the efficiencies of the technologies; they are included by the code,

ro_{svd}^{mt} is the relative factor per unit of output of technology v (coefficient) for relational constraint m,

rc_{svd}^{mt} is the same per unit of new built capacity,

ro_{rvs}^{mlt} is the relative factor per unit of output of technology v (coefficient) for relational constraint m, load region l,

rc_{rvs}^{mlt} is the same per unit of new built capacity,

rg_{sv}^{mt} is the same for storage technologies per unit of new built generation capacity,

rv_{sv}^{mt} is the same for storage technologies per unit of new built volume capacity,

tl is 1 for relations to construction and $\Delta\tau$ for relations to total capacity,

ip $\begin{cases} \text{is 1 for accounting during construction and} \\ \text{the plant life in periods for accounting of total capacity ,and} \end{cases}$

rhs_m^t is the right hand side of the constraint.

Relation with load regions

$$\sum_u \times \left\{ \sum_{svd} \times \left[ro_{svd}^{mlt} \times \sum_{e=0}^{e_d} Usvdue.t \times \varepsilon_{svd} + \sum_{\tau=t-ip}^{t} rc_{svd}^{mlt} \times YUsvdu.\tau \right] + \right.$$

$$\sum_{rvs} \times \left[\frac{ro_{rvs}^{mlt}}{\lambda_l} \times Zrvsu.lt \times \varepsilon_{rvs} + ro_{rvs}^{mt} \times Zrvsu..t \times \varepsilon_{rvs} + \right.$$

$$\left. \sum_{\tau=t-ip}^{t} rc_{rvs}^{mlt} \times tl \times YZrvsu.\tau \right] + \sum_{sv} \left[rg_{sv}^{mlt} \times YZGsv..\tau + \right.$$

$$\left. rv^{mlt}_{sv} \times YZVsv..\tau \right] \Bigg\} \left\{ \begin{array}{l} free \\ \geq rhs^{t}_{ml} \\ = rhs^{t}_{ml} \\ < rhs^{t}_{ml} \end{array} \right. ,$$

where

$Usvdue.t$ and $YUsvdu.t$ are the activity and capacity variables of the end-use technologies,

$Zrvsu.lt$, $Zrvsu..t$ and $YZrvsu.t$ are the activity variables of technologies with and without load regions and the capacity variables of the technologies,

ε_{rvs} and ε_{svd} are the efficiencies of the technologies; they are included by the code,

ro^{mlt}_{svd} is the relative factor per unit of utilized capacity of technology v (coefficient) for relational constraint m in load region l, period t (this constraint is adapted to represent the utilized power, as stated above),

rc^{mlt}_{svd} is the same per unit of new built or installed capacity,

ro^{mlt}_{rvs} is the relative factor per unit of output of technology v (coefficient) for relational constraint m, load region l,

rc^{mlt}_{rvs} is the same per unit of new built capacity,

rg^{mlt}_{sv} is the same for storage technologies per unit of new built or installed generation capacity,

rv^{mlt}_{sv} is the same for storage technologies per unit of new built or installed volume capacity,

tl is 1 for relations to construction and $\Delta\tau$ for relations to total capacity,

$$ip \left\{ \begin{array}{l} \text{is 1 for accounting during construction} \\ \text{the plant life in periods for accounting of total capacity ,and} \end{array} \right.$$

rhs^{t}_{ml} and is the right hand side of the constraint.

Construction of relations between periods

$$\sum_{u} \times \left\{ \sum_{svd} \times \left[ro^{ml}_{svd} \times \sum_{e=0}^{e_d} Usvdue.t \times \varepsilon_{svd} - ro^{m(l-1)}_{svd} \times \right.\right.$$

$$\sum_{e=0}^{e_d} Usvdue.(t-1)\times\varepsilon_{svd}\Big] + \sum_{rvs}\times\Big[ro_{rvs}^{mt}\times\sum_{l} Zrvsu..t\times\varepsilon_{rvs} - ro_{rvs}^{m(t-1)}\times$$

$$\sum_{l} Zrvsu..(t-1)\times\varepsilon_{rvs}\Big] + \sum_{rvs}\times\Big[ro_{rvs}^{mlt}\times\sum_{l} Zrvsu.lt\times\varepsilon_{rvs} - ro_{rvs}^{ml(t-1)}\times$$

$$\sum_{l} Zrvsu.l(t-1)\times\varepsilon_{rvs}\Big]\Big\}\begin{cases} free \\ \geq rhs_m^t \\ = rhs_m^t \\ < rhs_m^t \end{cases},$$

where

Usvdue.t and *YUsvdu.t* are the activity and capacity variables of the end-use technologies,

Zrvsu.lt and *Zrvsu..t* are the activity variables of technologies with and without load regions,

ε_{rvs} and ε_{svd} are the efficiencies of the technologies; they are included by the code,

ro_{svd}^{mt} is the relative factor per unit of output of technology v (coefficient) for relational constraint m, period t,

ro_{rvs}^{mlt} is the relative factor per unit of output of technology v (coefficient) for relational constraint m, load region l, and

rhs_m^t and is the right hand side of the constraint.

For this type of constraints only the ro-coefficients have to be supplied by the user, the rest is included by the model. It can be defined with and without load regions.

The second type of user defined relations differs from the first one in the fact that the activity of the end-use technologies is multiplied by k_e and therefore represents the production without reduction by demand elasticities.

Thus this constraint can be applied to force a certain reduction level due to the elasticities reached in one period to be also reached in the following period, allowing the interpretation of the reduction as investments in saving. The coefficient of the technologies supplying a demand have to be the inverse of this demand in the current period, then (this can be done by a switch--see input description, chapter 4.2.3.). This constraint has the following form:

$$\sum_{sv} \times \sum_{e=0}^{e_d} Usvdue.t \times \varepsilon_{svd} \times \frac{\kappa_e}{Udut} -$$

$$\sum_{sv} \times \sum_{e=0}^{e_d} Usvdue.(t-1) \times \varepsilon_{svd} \times \frac{\kappa_e}{Udu(t-1)} \le 0,$$

where

the coefficients are supplied by *MESSAGE II*. The user can additionally define multiplicative factors for these coefficients.

3.2.31 Stock-Piling of Fuels

$Qf.....t$

Q is a special level on that energy forms can be defined that are accumulated over time and consumed in later periods. One example is the accumulation of plutonium and later use in fast breeder reactors. Another example of using an equation like this is to model re-building of old hydroelectric power stations (this option is included in *STIM*, the sample input file, see appendix 1): a resource can contain all hydraulic potential of a country that can be built up at a certain cost (grades can be used to differentiate), the cost of building the dam is assigned to this resource. By a transfer technology having no cost and an efficiency of one this resource can be transferred to a stock-pile. The technology hydroelectric power plant in turn, needs some input of this stock-pile during construction and releases it back to the pile after the end of its operation life. The investment costs other than for the dam are assigned to the capacity of the power plant. This construction can be done for all technologies that consist of a part with a very long life and one with a lifetime that lies in the considered time horizon.

The general form of this constraint is:

$$Qfb....t - Qfb....(t-1) + \sum_{v} \times \left\{ \sum_{u} \times \left[\sum_{l} \Delta t \times (Zfvdu.lt + \beta^{f}_{\varphi vd} \times Z\varphi vdu.lt - \right.\right.$$

$$\varepsilon_{svf} \times Zsvfu.lt - \beta^{f}_{sv\varphi} \times Zsv\varphi u.lt \;) + \Delta(t+1) \times \iota(svd.f) \times YZsvdu.(t+1) -$$

$$\Delta(t-\tau_{svd}) \times \rho(svd.f) \times \sum_{u} YZsvdu.(t-\tau_{svd}) \Big] \Big\} = 0 \,,$$

where

f is the identifier of the man-made fuel (e.g. plutonium, U_{233}),
τ_{svd} is the plant life of technology v in periods,
$\iota(svd.f)$ is the 'first inventory' of technology v of f (relative to capacity of main output),
$\rho(svd.f)$ is the 'last core' of f in technology v, see alse chapter 3.2.6,
Δt is the length of period t in years,
$Zfvdu.lt$ is the annual input of technology v of fuel f in load region l and period t (l is '.' if v does not have load regions), and
$YZfvdu.t$ is the annual new installation of technology v in period t.

3.2.32 Cost Accounting Rows

The different types of costs (i.e. entries for the objective function) can be accumulated over all technologies in built-in accounting rows. These rows can be generated per period or for the whole horizon and contain the sum of the undiscounted costs. They can also be limited. The implemented types are:

CCUR -- fix (related to the installed capacity) and variable (related to the production) operation and maintenance costs,
CCAP -- investment costs; if the investments of a technology are distributed over the previous periods, also the entries to this accounting rows are distributed (if the capital costs are levelized, the total payments in a period can be taken from *CINV*; *CCAP* shows the share of investments in the according period, then),
CRES -- domestic fuel costs,
CAR1 -- costs related to the user defined relations of type 1 (see chapter 3.2.30),
CAR2 -- costs related to the user defined relations of type 2 (see chapter 3.2.30),
CRED -- costs for reducing demands due to demand elasticities, only related to technologies supplying the demands directly,
CIMP -- import costs,
CEXP -- gains for export, and
CINV -- total investments (in case of levelized investment costs, see *CCAP*)

3.2.33 The Objective Function

FUNC

In its usual form the objective function contains the sum of all discounted costs, i.e. all kinds of costs that can be accounted for. All costs related to operation (i.e. resource use, operation costs, costs of demand elasticities,...) are discounted from the middle of the current period to the first year. Costs related to construction are by default discounted from the beginning of the current period to the first year. By using the facility of distributing the investments or accounting during construction these costs can be distributed over some periods before or equal to the current one (see chapter 3.3.2). This distribution can also be performed for user defined relations.

Another easy change can be made by putting additive and multiplicative weights to the cost coefficients; the additive weights will be discounted. Such coefficients can be put on the different kinds of costs defined in the accounting rows. The minimization of primary energy use can for instance be achieved by setting all weights except the additive ones on resource and import costs to zero, the latter ones are set to 1. Thus the costs of primary energy are ignored, only the energy content is counted.

Another possibility to change the objective of *MESSAGE II* is provided by two fortran functions in the code. One of them is additive and one multiplicative, they are called at the calculation of each cost coefficients. These functions can be changed by the user if he wants to implement special objectives. Currently they yield 0. as additive and 1. as multiplicative weight.

The two functions are

weight$(i,i1,i2,i3,i4)$	-	additive, and
cweigh$(i,i1,i2,i3,i4)$	-	multiplicative,

where

i is the number of accounting row this kind of cost belongs to,

$i1$ is 0 for conversion technology related costs,
1 for storage technology related costs,
0 else (resources, imports, exports),

$i2$ is number of conversion/storage technology;
0 for resources, imports, exports, and

$i3,i4$ are not yet used and set to 0 in all calls.

The objective function has the following general form:

$$\sum_{t} \beta_m^t \,\Delta t \left\{ \sum_{svd}\sum_{u}\sum_{l} Zsvdu.lt \times \varepsilon_{svd} \times \left[ccur(svd,t) + \sum_{i}\sum_{m} ro_{svd}^{mlt} \times cari(ml,t) \right] + \right.$$

$$\sum_{svd}\sum_{u} \varepsilon_{svd} \times \sum_{e=0}^{e_d} Usvdue.t \times \varepsilon_{svd} \times \left[\kappa_e \times (ccur(svd,t) + \sum_{m} ro_{svd}^{mt} \times car2(m,t)) + \right.$$

$$\left. cred(d,e) + \sum_{m} ro_{svd}^{mt} \times car1(m,t) \right] + \sum_{sv}\sum_{l} \varepsilon_{sv} \left(SIZsv.lt + \sum_{m=l+1}^{l+m_{sv}^{l}} OZsv.lmt \right) \times$$

$$ccur(sv,t) + \sum_{svd}\sum_{u}\sum_{\tau=t-\tau_{svd}}^{t} \Delta\tau \times YZsvdu.\tau \times cfix(svd,\tau) +$$

$$\sum_{sv}\sum_{\tau=t-\tau_{sv}}^{t} \Delta\tau \times YZFsv..\tau \times cfix(sv,\tau) + \sum_{r} \left[\sum_{g}\sum_{l}\sum_{p} RZrgp.lt \times cres(rgpl,t) + \right.$$

$$\left. \sum_{c}\sum_{l}\sum_{p} IZrcp.lt \times cimp(rcpl,t) - \sum_{c}\sum_{l}\sum_{p} EZrcp.lt \times cexp(rcpl,t) \right] \Bigg] \Bigg\} +$$

$$\beta_b^t \times \left\{ \sum_{svd}\sum_{u}\sum_{\tau=t}^{t+t_d} \Delta t \times YZsvdu.\tau \times \left[ccap(svd,\tau) \times fri_{svd}^{t_d-\tau} + \right. \right.$$

$$\left.\sum_i \sum_m rc_{svd}^{mt} \times cari(m,t) \times fra_{svd,m}^{t_d-\tau}\right] + \sum_{sv} \Big[YZGsv..t \times (ccap(svG,t) +$$

$$\left.\left.\sum_i \sum_m rc_{sv}^{mt} \times cari(m,t)) + YZVsv..t \times ccap(svV,t) \right]\right\},$$

where

Δt is the length of period t in years,

$$\beta_b^t = \prod_{i=1}^{t-1} \left[\frac{1}{1+\frac{dr(i)}{100}} \right]^{\Delta i},$$

$$\beta_m^t = \beta_b^t \times \left[\frac{1}{1+\frac{dr(t)}{100}} \right]^{\frac{\Delta t}{2}},$$

$dr(i)$ is the discount rate in period i in percent,

$Zsvdu.lt$ is the annual consumption of technology v of fuel s in energy density area u, load region l and period t; if v has no load regions, l = '.', if it has no energy density areas, u='.',

ε_{svd} is the efficiency of technology v in converting s to d,

$ccur(svd,t)$ are the variable operation and maintenance costs of technology v (per unit of main output) in period t,

ro_{svd}^{mlt} is the relative factor per unit of output of technology v for relational constraint m in period t, load region l,

$car1(m,t)$ and $car2(m,t)$ are the coefficients for the objective function, that are related to the user defined relation m in period t,

$car1(ml,t)$ and $car2(ml,t)$ are the same for load region l, if relation m has load regions,

$Usvdue.t$ is the annual consumption of fuel s of end-use technology v in energy density area u, period t and elasticity class e,

κ_e is the factor giving the relation of total demand for d to the demand reduced due to the elasticity to level e,

ro_{svd}^{mt} is the relative factor per unit of output of technology v for relational constraint m in period t,

$cred(d,e)$ is the cost associated with reducing the demand for d to elasticity level e,

$SIZsv.lt$ is the annual input to storage technology v in period t, load region l,

ε_{sv} is the efficiency of the I/O device of storage technology v,

$OZsv.lmt$ is the annual output of storage technology v of fuel stored in load region l to load region m in period t,

m^l_{sv} is the number of load regions the content can be kept (see chapter 3.2.22)

$ccur(sv,t)$ are the variable operation and maintenance costs of storage technology v in period t,

$YZsvdu.t$ is the annual new built capacity of technology v in period t,

$cfix(svd,t)$ are the fix operation and maintenance cost of technology v that was built in period t,

$YZGsv..\tau$ is the annual new installed capacity for I/O of storage technology v in period τ,

$YZVsv..t$ is the annual new installed volume capacity of storage technology v in period t,

$cfix(sv,\tau)$ are the fix operation and maintenance cost of storage technology v that was built in period t,

$ccap(svd,t)$ is the specific investment cost of technology v in period t (given per unit of main output),

fri^n_{svd} is the share of this investment that has to be paid n periods before the first year of operation,

rc^{mt}_{svd} is the relative factor per unit of new built capacity of technology v for user defined relation m in period t,

$fra^n_{svd,m}$ is the share of the relative amount of the user defined relation m that occurs n periods before the first year of operation (this can, e.g., be used to account for the use of steel in the construction of solar towers over the time of construction),

$ccap(svG,t)$ is the specific investment cost of storage technology v in period t for the generation part,

$ccap(svV,t)$ is the specific investment cost of storage technology v in period t for the volume part,

rc^{mt}_{sv} is the relative factor per unit of new built capacity of storage technology v for user defined relation m in period t,

$RZrgp.lt$ is the annual consumption of resource r, grade g, elasticity class p in load region l and period t,

$cres(rgpl,t)$ is the cost of extracting resource r, grade g, elasticity class p in period t and load region l (this should only be given, if the extraction is not modeled explicitly),

$IZrcp.lt$ is the annual import of fuel r from country c in load region l, period t and elasticity class p; if r has no load regions l='.',

$cimp(rcpl,t)$ is the cost of importing r in period t from country c in load region l and elasticity class p,

$EZrcp.lt$ is the annual export of fuel r to country c in load region l, period t and elasticity class p; if r has no load regions l='.', and

$cexp(rcpl,t)$ is the gain for exporting r in period t to country c in load region l and elasticity class p.

3.3 Special Features of the Matrix Generator

The mathematical formulation of *MESSAGE II* as presented in chapter 3.2 shows the structure of all constraints as the matrix generator builds them up. The background of the more complicated features is given here for a better understanding.

3.3.1 The Time Horizon--Discounting of the Costs

The whole time horizon of the calculations is divided into periods of optional length. All variables of *MESSAGE II* are represented as average over the period they represent, resulting in a step-function. All entries in the objective function are discounted from the middle of the respective period to the first year, if they relate to energy flow variables and from the beginning of that period if they represent power variables. The function to discount the costs has the following form:

$$c_t = \frac{C_t^T}{\prod_{k=1}^{t-1} (1 + \frac{dr_k}{100})^{\Delta k} \times f_i} ,$$

where

C_t^T is the cost figure to be discounted,

c_t is the objective function coefficient in period t,

$$f_i = \begin{cases} 1 & \text{for costs connected to investments,} \\ (1 + \frac{dr_t}{100})^{\frac{\Delta t}{2}} & \text{else, and} \end{cases}$$

dr_k is the discount rate in period k.

3.3.2 Distribution of Investments

In order to support short term applications of *MESSAGE II* the possibility to distribute the investments for a new built technology over several periods was implemented. The same type of distributions can be applied to entries in user defined relations if they relate to construction. The distribution of investments can be performed in several ways. There is one common parameter that is needed for all of these possibilities, the construction time of the technology [ct].

The implemented possibilities are:

1. Explicit definition of the different shares of investments for the years of construction. The input are ct figures that will be normalized to 1 internally.

2. The investment distribution is given as a polynomial function of 2nd degree, the input consists of the three coefficients:

$$y = a + bx + cx^2 \quad , x = 1(1)ct ,$$

where

ct is the construction time.

The values of the function are internally normalized to 1, taking into account the construction time.

3. Equal distribution of the investments over the construction period.

4. A distribution function based on a logistic function of the type

$$f = \frac{100}{1+e^{-a(x-x_0)}} ,$$

where

$$x_0 = \frac{ct}{2} ,$$

and

$$\alpha = \frac{2}{ct} \ln\left(\frac{100}{\varepsilon} - 1\right) .$$

This function is expanded to a normalized distribution function of the following type:

$$g = \left[\frac{100}{1 + e^{-\frac{\ln(\frac{100}{\varepsilon} - 1)(x - 50)}{50}}} - \varepsilon \right] \times \frac{1}{1 - \frac{\varepsilon}{50}} .$$

g gives the accumulated investment at the time x, x is given in percent of the construction time. The parameter ε describes the difference of the investment in the different years. ε near to 50 results nearly in equal distribution, an ε close to 0 indicates high concentration of the expenditures in the middle of the construction period.

In order to shift the peak of costs away from the middle of the construction period the function is transformed by a polynomial function:

$$x = az^2 + bz \quad , 0 < z < 100 ,$$

where

$$b = \frac{5000 - d^2}{100d - d^2} \quad , 0 < d < 100 ,$$

and

$$a = \frac{1 - b}{100} .$$

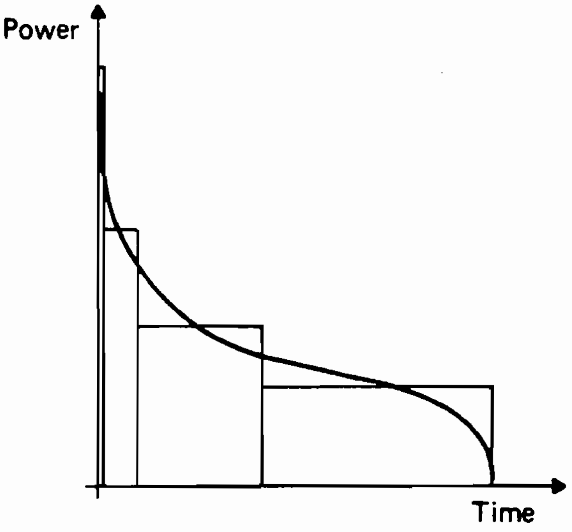
d denotes the time at that the peak of expenditures occurs in percent of ct. This kind of investment function was taken from [6].

The distribution of these yearly shares of investments is done starting in the first period of operation with a one years share, the expenditures of the remaining $ct-1$ years are distributed to the previous periods.

The coefficients of the capacity variables of a technology in a relational constraint can be distributed like the investments.

3.3.3 The Load Curve

The years representing a period can be subdivided into so-called load regions. This can be done by either ordering the whole year according to the power requirements for the most important energy carriers like, e.g., electricity, or by grouping the year into load regions with similar characteristics (hereafter called characteristic loads), like, e.g., winter days and nights and summer days and nights. The first option results in an interpolation of the usual representation of the load curve by a step function (see figure 4a), the second one in a step-function where the time is still ordered in a historic way (see figure 4b). Thus *MESSAGE II* can keep track of the in- and output of storage technologies and their contents at any point in time.



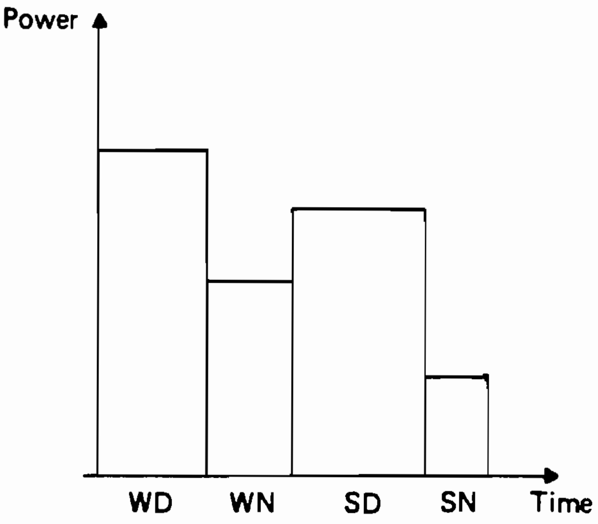


Figure 4a: Example of an ordered load curve.

Figure 4b: Example of a semi-ordered load curve.

(WD stands for winter day, WN for winter night, SD for summer day and SN for summer night.)

3.3.4 Consideration of Load Variations in Conversion Technologies

The activity of a conversion technology is generated for each load region, if the main in- or output energy form is defined to have load regions. In this case the relation of these activities between the load regions is freely chosen by the model. The relations can be fixed by the user to reflect a certain fixed production pattern. In this case the activity will only be generated once and written to the energy flow balances with coefficients reflecting the chosen pattern. A power plant operating in base load mode would for instance have the shares of the load regions in the year as coefficients in the balances of energy forms with load regions.

For end-use technologies (output level '*U*') the production is assumed to meet the demand pattern, the input of the technology is fixed to reflect the according demand variations. This can also be changed into to a different pattern. This would, e.g., model night storage heating systems that meet the heat demand of a household, but generate a final electricity demand with a different load distribution, namely at night.

3.3.5 The Implementation of Energy Storage

MESSAGE II contains a quite complex model of energy storage. Chapters 3.2.22, 3.2.24 and 3.2.25 contain the mathematical formulation. In order to allow for different types of storage like daily and seasonal the distribution of demands over the year has to be depicted in a semi ordered load curve: The user has to define the load regions in a physical order. Daily storage would for instance need the definition of several parts of the day that are ordered like in an actual day. The model can then store energy in one part of the day and release it in one of the following load regions, keeping track of the storage contents in each load region. This loop of storage is closed for all but seasonal storage, where an appropriate part of the energy stored in the last load region is delivered to the next period.

The length of time that the content of a storage can be held can also be limited to some fraction of the time it is dedicated to. An example would be a daily heat storage that can only keep the heat 80% of the day, after that time it could have too low a temperature to be used. The loss of energy in the case of heat storage can be modeled by a decay function:

$$co_{l+1} = co_l \times e^{\zeta \times \delta l},$$

where

co_l is the content of storage in load region l,

ζ is the decay constant of storage [unit: $\frac{1}{k}$],

k is 1 day for daily storage, 1 year for seasonal storage, etc. and

δl is the fraction of k that lies between load regions l and $l+1$ [unit: k].

The amount of energy available from storage is reduced over time according to this function.

If several types of load regions are defined, e.g. weekly and seasonal, (it should rather be named yearly for reasons of consistency) they are ordered according to the length of the time period they span. The 'bigger' one (the seasonal) can then work like the smaller one (weekly), too (see figure 5). The decay of content and limitation on time is only applied to the biggest type of load the storage works in.

Figure 5: Flows of energy in daily and seasonal storage.

The two basic parts of a storing device, namely the input/output part (for a pumped hydro storage the generator/turbine/pump part) and the real storage (dam and reservoir) can be handled in two different ways. One of them is to link them in size, i.e. to fix the content (in MWyrs or GWyrs) in relation to the generation capacity (in MW or GW), as it is usually the case with batteries.

The other possibility, which could, e.g., be useful for pumped hydroelectric power storage plants, is to keep them separate with their own costs and leave the relation of the two open for the optimization process.

3.3.6 Relocation of Base Load Demand

For special purposes *MESSAGE II* can be used to optimize only the part of the energy system from secondary or final to useful energy, i.e. the utilization of energy. In this case or if, e.g., electricity imports have to be included in the model of a small region, it can be essential to differentiate the production or import cost (price) of certain energy carriers according to their location in the load curve.

Figure 4 shows a simple example of a semi ordered load curve interpolated by a step function. It could for instance represent the electric load curve in a country with some direct and much night storage electric heating. For each of the load regions (winter day, winter night, summer day and summer night) the import price can be given separately; but still--according to the amount of night storage heating and warm water production systems--this load curve could change and have an either bigger or smaller base load fraction, thus changing the import costs. *MESSAGE II* gives the option to include an additional load region (called base load), which is located horizontally and can take away some share (depending on the load variations in the lowest load region, in this case summer night) of the minimum power requirements in all load regions. Since base load electricity is cheaper the model could reshape the load curve in order to have a maximum base load fraction. For the formulation see chapters 3.2.4 and 3.2.5.)

3.3.7 Energy Density Areas

Usually energy density areas are defined by giving the W/m^2 that are needed on the average in a certain area. The use of this word in the context of *MESSAGE II* is related to this definition, but differs slightly. Energy density areas are distinguished by different possibilities to meet the same kind of demand and by different costs and efficiencies to deliver final energy.

An example for the first difference is the demand for space heating that can be supplied with district heat in urban areas, but not in the countryside. The second difference can also be seen in district heating for which distribution is more efficient and cheaper in urban areas than in suburbs.

In defining the energy chain for *MESSAGE II* the user can define a level from which on up to the demand all energy forms and technologies are created for each energy density area (if not stated differently in the input file). This is--in light of the previous paragraph--usually level F, because the characteristics of distribution technologies do already differ for different energy density areas.

3.3.8 Lag Times Between Input and Output of a Technology

Since *MESSAGE II* can be used for very short time steps, even for steps of 1 year per period, the implementation of lag-times between input and output of a conversion technology seemed to be appropriate. One possible application are the reprocessing units for nuclear fuels, which usually keep the fuels for several years.

The lag time for a technology is given in years and the period in which the output is available is calculated beginning from the middle of the period when the input is required.

3.3.9 Variable Inputs and Outputs

A lot of power plants can use different fuels for electricity generation, the highest variability occurs between oil products and natural gas as fuel. This can be modeled by having two or even more energy conversion variables with different inputs, efficiencies and variable operation and maintenance costs linked to one capacity in one capacity equation (see also chapter 3.2.23).

The same link of different conversion activities can be used to model co-generation of electricity and heat with a variable output pattern. In this case one of the conversions would be to electricity (with an efficiency ε_e) and the other one producing a mix of electricity and the maximal possible share of district heat (producing ε_c electricity and δ_c district heat from one unit of input). In the latter case the efficiency to electricity (ε_c) is lower than in the first case (ε_e), but the overall efficiency is naturally much higher. The two conversion variables have to be related to the same capacity by a factor giving the relative production of the main product possible with one unit of installed capacity, which is always related to the first operation mode. In the terms used above this would mean that the plant can produce ε_e electricity in the first operation mode, while it can produce ε_c with the same capacity in the second operation mode. For the model this means that the electric capacity is not utilized fully in the second mode, the relation has to be defined by the user. (It would be ε_e / ε_c in the described case, but could also be independent from the efficiencies for other technologies.)

The cracker as included in *STIM* (see appendix 1) is another example of a technology with a variable production pattern.

3.3.10 The Contribution of Capacities Existing in the Base Year

The possible contribution of an installation that exists in the base year is kept track of over time. There are two possibilities to give the necessary information to *MESSAGE II*.

1. Define the capacities that were built in the years $iyr, \ldots, iyr-\tau+1$, with iyr = base year and τ = plant life in years explicitly. These capacities are then distributed to historic periods of the length ν.

2. Define the total capacity, c_0, that exists in iyr and the rate at that it grew in the last τ years, γ. This information is then converted to one similar to 1. by using the function:

$$y_0 = c_0 \frac{\gamma^{-\nu}-1}{\nu(\gamma^{-\tau}-1)} ,$$

$$y_t = y_0 \gamma^{-t \times \nu} \quad , t = 1(1)\frac{\tau}{\nu},$$

where

y_t is the annual construction in period $-t$, (0 = base year),

γ is the annual growth of new installations before the base year,

c_0 is the total capacity in the base year,

τ is the plant life, and

ν is the length of the periods in that the time before the base year is divided.

The right hand sides in the capacity constraints are derived by summing up all the old capacities that still exist in a certain period (according to the plant life). If the life of a technology expires within a period, *MESSAGE II* takes the average production capacity in this period as installed capacity (this represents a linear interpolation between the starting points of this and the following period).

3.3.11 Capacities which Operate Longer than the Time Horizon

If a capacity of a technology is built in one of the last periods its life time can exceed the calculation horizon. This fact is taken care of by reducing the investment costs by the following formula:

$$C_t^T = C_t \times \frac{\sum_{k=1}^{\tau_p - \nu} \prod_{\tau=t}^{t+k-1} \frac{1}{1+dr_\tau}}{\sum_{k=1}^{\tau_p} \prod_{\tau=t}^{t+k-1} \frac{1}{1+dr_\tau}}$$

where

ν is the number of years the technology exists after the end of the calculation horizon,

dr_τ is the discount rate for year τ,

τ_p is the plant life in years,

C_t is the investment cost in year t, and

C_t^T is the reduced investment.

3.3.12 Own-Price Elasticities of Demand

Own-price elasticities of demand can be interpreted either as short-term elasticities resulting in reduced demand due to sharp price increases (they have to relate to a reference price- and demand level and represent renunciation of services) or as long-term elasticities reached by substituting capital for energy. In the latter case the user has to assure that the relatively decreased demand level is maintained over the calculation horizon by applying user-defined relations (see chapter 3.2.30). The costs and levels of demand reduction can be derived from the investments and savings that are associated to certain additional installations, like, e.g., three-glass windows to save in space heating.

The form of the own price elasticity function of demand is

$$\frac{Q}{Q_r} = \left[\frac{P}{P_r}\right]^{\varepsilon} ,$$

where

Q_r is the reference demand level,

P_r is the reference price level, and

ε is the elasticity, (assumed to be < 0).

It says that the demand will decrease by a factor of x^{ε} if the price rises by x. This function is approximated by a step-function of the following form:

The demands (Q) and prices (P) are normalized to the reference levels:

$$Q = q \times Q_r \quad ,$$

and

$$P = p \times P_r \quad ,$$

the normalized values follow the function

$$q = p^{\varepsilon} \quad ,$$

or

$$p(q) = q^{\frac{1}{\varepsilon}} .$$

To reduce the demand to the level q_i the supply has to have the cost

$$c(q_i) = \int_{q_i}^{1} q^{\frac{1}{\varepsilon}} \, dq = \frac{1}{1+\frac{1}{\varepsilon}} \times \left[1 - q_i^{\,1+\frac{1}{\varepsilon}}\right] ,$$

a function increasing monotonously with decreasing q_i (see also figure 6). In absolute terms this means that the cost would be higher by an absolute value of

$$R(Q_i) = c\left(\frac{Q_i}{Q_r}\right) \times Q_r \times P_r$$

compared to the cost at the reference level.

The step-function is then defined by choosing certain levels of demands and prices $(Q_i, P_i), i=1(1)n$ with $Q_i < Q_r$, that fulfill the elasticity function. The code can choose, which demand level it supplies, but if it supplies a level $Q_i < Q_r$ it has to pay additionally $R(Q_i)$, the cost of reducing the demand to level i.

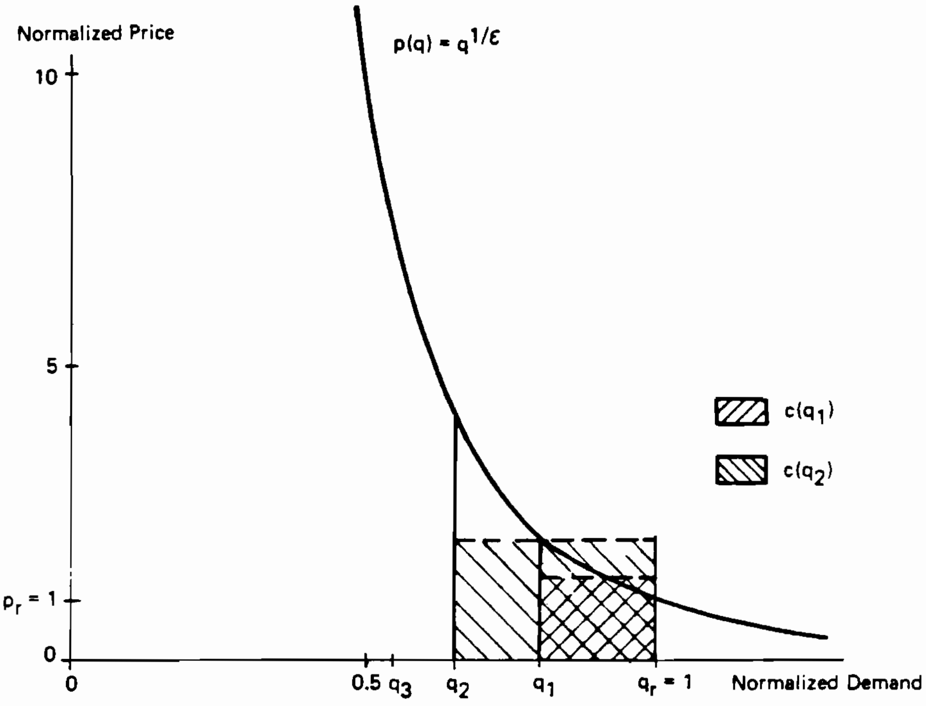


Figure 6: Representation of Demand Elasticities.

3.3.13 Supply Elasticities

The reaction of the market prices to changes in demand can be expressed as elasticities:

$$\frac{P}{P_r} = \left[\frac{S}{S_r}\right]^{\alpha},$$

where

P_r is the reference price level,

S_r the reference supply level, and

α the elasticity.

The normalized form of this equation is

$$c = s^{\alpha},$$

where

$c = \frac{P}{P_r}$, and

$s = \frac{S}{S_r}$.

The relationship is converted to a step-function with n steps, which is shown in figure 7. $f(s_1)$ is the cost of supplying amount s_1 relative to supplying s_r, while $f(s_1) + (s_2)$ ist the relative cost of supplying the amount s_2. The marginal costs are then defined as

$$\mu(s) = \frac{\int_{s_{i-1}}^{s_i} \sigma^{\alpha} d\sigma}{s_i - s_{i-1}},$$

where

$s_{i-1} < s \leq s_i$.

According to the normalized function the total price of buying the amount s is then

$$tc(s) = \sum_{j=1}^{i-1} \mu(s_j) \times (s_j - s_{j-1}) + \mu(s_i) \times (s - s_{i-1}).$$

The price of the amount S, $S_{i-1} < S < S_i$, is defined as

$$TC(S) = P_r \times S_r \times tc\left(\frac{S}{S_r}\right).$$

In the matrix this function is implemented as $n+1$ additive elasticity classes for resources and imports $(R_0 = S_r, R_i = S_i - S_{i-1}, i=1(1)n)$, which have increasing costs. The code takes these classes as supply one after the other and has to pay increasing prices, then.

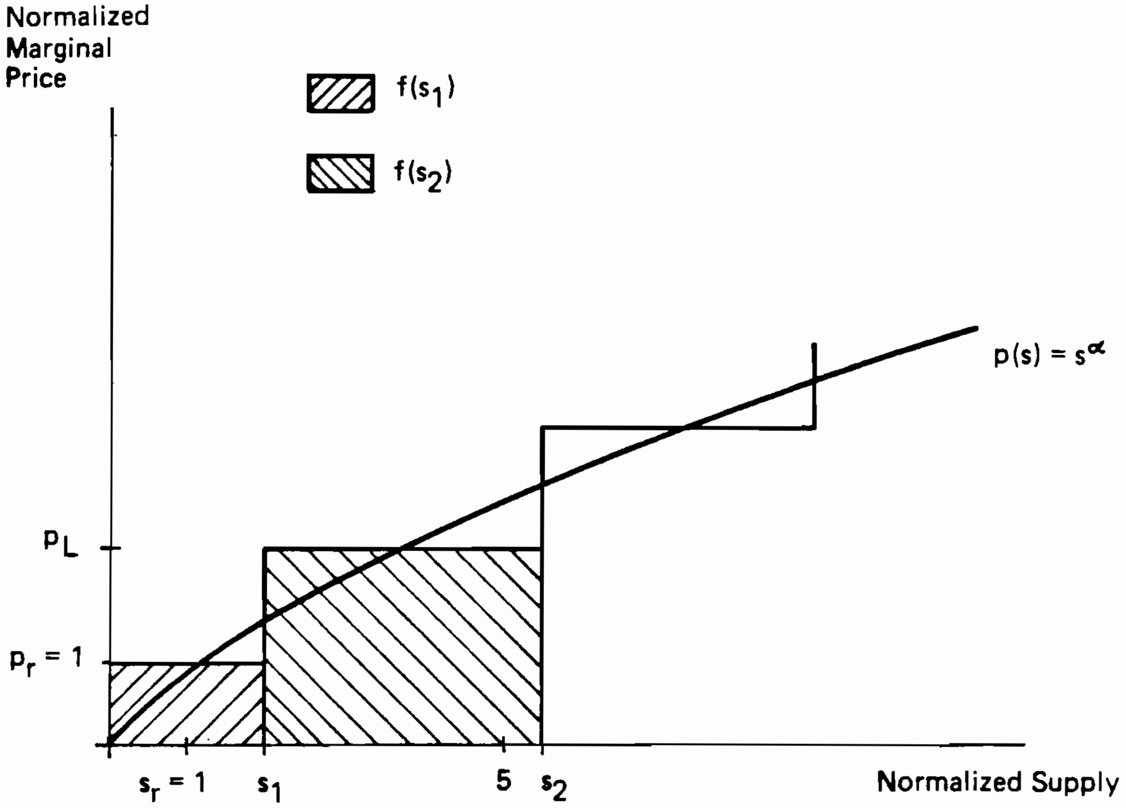


Figure 7: Representation of Supply Elasticities.

3.3.14 Application of the Mixed Integer Option of MESSAGE II

If the LP-package used to solve a problem formulated by *MESSAGE II* has the capability to solve mixed integer problems, this can be used to improve the quality of the formulated problems, especially for applications to small regions.

The improvement consists in a definition of unit sizes for certain technologies that can only be built in large units. This avoids for instance the installation of a 10 kW nuclear reactor in the model of the energy system of a city or small region (it can only be built in units of ,e.g., 700 MW). Additionally this option allows to take care of the 'economies of scale' of certain technologies.

This option is implemented for a technology by simply defining the unit size chosen for this technology. The according capacity variable is then generated as integer in the matrix, its value is the installation of one power plant of unit size.

If a problem is formulated as mixed integer it can be applied without this option by changing just one switch in the control input. All capacity variables are generated as real variables, then (see chapter 4.3.2).

3.3.15 The Nonlinear Objective Function

In combination with *MINOS* [1] *MESSAGE II* can be applied to problems with a partly nonlinear objective function. The requirements are that the function is differenciable and convex with respect to the solution space.

In order to use a nonlinear objective the user has to identify the variables that are to be included with nonlinear coefficients in the input file (they will be written to the matrix as first entries in the columns section–as required by *MINOS*) and to supply *MINOS* with an additional subroutine (Calcfg), which yields the nonlinear part of the objective and the first derivatives to all variables with nonlinear coefficients depending on the values of these variables.

In order to start a nonlinear problem it can be solved as linear problem in the beginning. The nonlinear variables can be fixed to user-defined estimates by specifications in the bounds section--this can also be done using the normal *MESSAGE II* input file (see input description).

The order in that the nonlinear variables appear in the input file is essential, because the same order is used for identifying them in Calcfg. *MESSAGE II* generates the activity variables first, then the capacity variables (both of them in the order in that the technologies appear in the input file). The loops in producing the columns are nested in the following order:

-- energy density areas,

-- load regions,

-- demand elasticity classes, and

-- time periods.

4. Setting up a Model with MESSAGE II

4.1 Data Requirements

The input data requirements of *MESSAGE II* can, as already mentioned in chapter 2.3, be grouped into 6 logical blocks. The information to be given in the different blocks is:

Block 1 General Definitions
- name of the model,
- reference (or base) year, time horizon, period lengths,
- special switches for further processing (on mixed integer programming, format for the matrix, identifiers for the levels, handling of investments),
- discount rate and interest rate,
- limits on expenditures,
- distribution function for investments,
- definition of energy density areas and load regions, and
- definition of the energy chain modeled.

Block 2 Demand Related Data
- demands,
- distribution of demands to energy density areas,
- distribution of demands to load regions, and
- definition of own-price elasticities of demand.

Block 3 User Defined Relations
- definition of additional relations,
- definition of additional constraints and the according limits,
- definition of counters, and
- definition of rows to which costs are assigned and these costs.

Block 4 The Objective

- definition of the weights on the components of the objective function.

Block 5 Technology Data

- definition of conversion technologies by
  - energy inputs, energy outputs and efficiencies,
  - first inventories and last retirements of energy fuels,
  - coefficients in the relations set in block 3,
  - plant factor,
  - plant life,
  - investment, fix and variable operation and maintenance costs,
  - bounds on production and new installations, dynamic constraints on production and new installation,
  - age structure and production in the base year, and
  - special switches concerning lag-times, distribution of investments, declaration of nonlinear variables, integer variables, fixing the operation pattern to the load regions.
- definition of storage technologies by
  - energy form stored, efficiency of I/O and losses over time,
  - coefficients in the relations set in block 3,
  - plant factor,
  - plant life,
  - investment, fix and variable operation and maintenance costs, both for I/O and volume capacity,
  - bounds on new installations and dynamic constraints on new installation for I/O and volume capacity, and
  - age structure and amount stored in the base year.

Block 6 Resources - Imports - Exports

- total availability and cost of domestic resources and imports
- maximum extraction and depletion rates,
- dynamic constraints,
- supply elasticities, and
- potential and gains for exports.

4.2 The Formalized Input Description

4.2.1 The Format Used

The following chapters give a formalized description of the input requirements. The physical order in that the inputs are given is mandatory.

The format used for the explanation consists of 3 columns giving

- the format in that the entry is read,
- the name of the variable to that the value is assigned and some control of input flows, and
- a description of the kind of data to be supplied.

a) The input format
Only a limited number of input formats are applied, most of which can be easily understood since they resemble FORTRAN formats (e.g., 2a4, i5, a1, a4, g12.5, 4a1, ...). By a # in the first columns of the data files those lines are identified that are included in the report on the input data. This is indicated by a # in column 1 of the input format. a1-* means that an identifier can be appended by a - and a description, which will also be included in the report.

_*ts_ indicates that a time series has to be read. Then an input routine is used that can read time series data in several formats. It reads a switch and then the data according to this switch:

isw = -2 => no equation is generated (must not occur for prices; internally the first value is set to 1.e7)
isw = -1 => read no time series
for costs: multipliers are set to 1., costs are set to 0.
for limits: eqn exists and is not limited.
in the other cases the default is zero or listed in the input description below.
isw = 0 => one value is read and used for the whole time horizon.
isw = 1 => single values per period are read
in the case of limits: if a value is set to 1.e7 the according equation is not limited
isw = 2 => read first value and annual growth for the whole horizon (2 entries), and
isw = 3 => read first value and annual growth per period (1+(nto-1) values).

*tst denotes a similar input routine that is used for time-dependent entries of conversion technologies (if they are switched with a '_v_'), _*its_ means a time series in integer format and _&ts_, _&its_ indicate that a switch for time series is read, but it is not jet fully implemented; only switch 0 is accepted, then.

b) Input flow control
Column 2 contains the flow control for the input as well as the variable names that the data are assigned to if they are not converted immediately.

Loops are indicated by an initial assignment of the index variable, a counting command, an end condition and a label where execution is resumed after exit from the loop. The depth of nesting is indicated by a '|' on the left side of column 2.

Also conditional statements do occur. They start with a condition. The following lines starting with '|' are only performed if the condition is true.

c) Description
The third column gives a description of the entries to be read and also some additional information on the flow control

The following example shows the setup of the input description:

```
Format   | Variable Name                    | Description
         | Flow Control                     |
------------------------------------------------------------------
         |                                  |
#        |                                  | general information
i5       | int1                             | integer value 1
         | if(int1.eq.1)                    |
g12.5    | | var1                           | real value 1
i5       | int2                             | integer value 2
```

The first line of input file read according to this description has to be a comment line starting with a #, the next entry is an integer value that is stored in variable int1 followed by a real value that is only read if int1 is 1 and a second integer value that is read in any case. The example in Appendix 2 starts with a sequence that would be rescribed like this.

4.2.2 General Input

BLOCK 1

```
Format  | Variable Name              | Description
        | Flow Control               |
--------+----------------------------+-----------------------
2a4     | pname(i),i=1,2             | name of problem,
        |                            | (up to 8 characters)
#       |                            | general information on
#       |                            | the run, is written below
#       |                            | the header of the report
        |                            |
i5      | ntu                        | number of first period
        |                            | in the matrix
        |                            |
i5      | ntrun                      | number of last period in the
        |                            | matrix
        |                            |
i5      | isw                        | =0 => no mixed integer
        |                            |      applied, all
        |                            |      variables are
        |                            |      generated as real
        |                            | =1 => mixed integer
        |                            |      applied if
        |                            |      specified in the
        |                            |      technology input
        |                            | see chapter 3.3.14
        |                            |
        | isw                        | =0 => matrix is written
        |                            |      with format f12.5
        |                            | =1 => format for matrix has
        |                            |      to be next entry and
        | if(isw.eq.1)               |      supplied with a closing
 a8     | | format                   |      bracket
        | |                          |      (e.g. 'e12.6)  ')
        |                            |
 i5     | isw                        | =0 => no level identifiers
        |                            |      are changed
        |                            | >0 => level no isw
        |                            |      has identifier Z
        | if(isw.lt.0)               | <0 => all level ids are
 a1     | | zh(i),i=1,7              | read beginning with
```

		resources, the last one is for man-made fuels. If level 6 has a different identifier it looses its special features. The default values for level identifiers are:
		no id description ilev zh(ilev) 6 U energy utilization 5 F distribution 4 T transmission 3 X central conversion 2 A preparation, ... 1 R resources Q fuels with stock-pile see also figure 2
i5	isw	switch to relate the coefficients for additional relations to: =0 => the main output =1 => the main input =2 => exactly as they are given in the input the mathematical formulation in chapter 3 is given for isw=0
i5	ncap	=0 => capital costs are paid at once, discounted from the beginning of the that they are built (if this is not changed by distributing them, see ch. 3.3.2) =1 => capital costs are levellized (i.e. the capital costs are distributed to the whole plant life using the interest rate--air--read below) the mathematical formulation in chapter 3 is given for ncap=0
i5	imapo	= 0 => no check on repeated

		technology identifiers and missing declarations of relations = 1 => check if all relation identifiers used are declared = 2 => performs checks 1 and 3 = 3 => check on repetition of technology identifiers (the repetition of identifiers of user defined relations is checked in any case)
i5	idum x 4	dummy switches, not used
i5	iyr	reference year, defines the labels of the periods, sump gives the number of years between iyr and the 1st calculation year the existing structure has to be given for the year iyr+sump-lpo ntu=1 => sump=lpo, ntu=2 => sump=lpo+lp(1), ntu=3 => sump=lpo+lp(1)+lp(2) etc
i5	nto	number of periods in the input file, has to be greater or equal ntrun
i5	lpo	length of periods before iyr
i5	lp(i),i=1,nto	lengths of periods in years, the labels of the years are then: lbl(ntu) = iyr+lpo+ $\sum_{i=1}^{ntu-1} lp(i)$ lbl(i+1) = lbl(i) + lp(i)
*ts	dr	discount rate (default for time series input is 6.)
	if(ncap.ne.0)	

```
*ts      | | air                                | interest rate used to
         |                                      | levellize capital costs
         |                                      | (in %), default is 6.
         |                                      |
i5       | imat                                 | switch for generation of
         |                                      | cost accounting rows
         |                                      | as showed in chapter 3.2.32
         |                                      | =0 => no accounting rows
         |                                      | =1 => total accounting
         |                                      | =2 => annual accounting
         |                                      | =3 => total + annual
         |                                      |        accounting
         |                                      |
         | if(imat.gt.0)                        |
a4       | | idcst       =* => exit             | id of cost accounting row
         | |                                    | to be limited
         | |                                    | the implemented possibilites
         | |                                    | are: CCAP, CCUR, CRES, CIMP,
         | |                                    | CEXP, CRED, CINV, CAR1, CAR2,
         | |                                    | for an explanation see chapter 3.2
         | | if(imat.eq.1.or.imat.eq.3)         |
g12.5    | | | costb                            | bound on total cost
         | |                                    | accounting row with id idcst
         | |                                    |
         | | if(imat.gt.1)                      |
g12.5    | | | acostb(j,i),i=1,nto              | bounds on annual cost
         | |                                    | accounting row with id idcst
         |                                      |
         |  ------j=0 -------------------------|
a1       | | idis        =* => exit             | switch for kind of
         | |                                    | distribution function
         | |                                    | for investments
         | |                                    | for that parameters
         | |                                    | are to be read
         | |                                    | see chapter 3.3.2
         | |  -----j=j+1 ----------------------|
         | | if(idis.eq.e)                      | equal distribution
         | |                                    |
         | | if(idis.eq.p)                      | distribution according
         | | |                                  | to a polynomial function:
         | | |                                  | a + bx + cxx
g12.5    | | | a                                | coefficients
g12.5    | | | b                                |
g12.5    | | | c                                |
         | |                                    |
```

```
          | | if(idis.eq.d)                    | distribution according
          | | |                                | to specific values (can
          | | |                                | only be used during
          | | |                                | technology input and
          | | |                                | is given here for completeness)
          | |                                  |
          | | if(idis.eq.l)                    | distribution according to
          | | |                                | a logistic distribution
g12.5     | | | eps                            | variation in annual
          | | |                                | investments, (0<eps<50)
g12.5     | | | delta                          | point of time with highest
          | | |                                | annual investments in
          | | |                                | % of the total construction
          | | |                                | time
          | |                                  |
          |  ------exit -------------|  => mdistr = j: total
          |                                    | number of sets of
          |                                    | parameters that exist
          |                                    |
          |  ------i=0 -------------|
a1-*      | | namu(i+1)     =* => exit         | identifiers
          | |        i=i+1                     |
#         | |                                  | explanation of energy
#         | |                                  | density area no i
          |  ------exit -------------|  nu=max(i,1): number of
          |                                    | energy density areas
          |                                    | (see chapter 3.3.7)
          | if(nu.gt.1)                        |
a1        | | nux                              | id of last level with energy
          | |                                  | density areas (counted from
          | |                                  | demand level)
          |                                    |
#         |                                    | the definition of the load
#         |                                    | curve can be preceded by
#         |                                    | a description
          |  ------j=0 -------------|
a1-*      | | idlr(j+1)     =* => exit         | identifier of characteristic
          | |  -----j=j+1 ------------|  load, e.g. d for daily, s for
          | |                                  | seasonal; they have to be
          | |                                  | ordered in increasing
          | |                                  | length. This distinction is
          | |                                  | necessary if storage is
          | |                                  | modeled. (see chapter 3.3.3)
i5        | | islr(j)                          | number of times that
          | |                                  | this load region occurs
```

```
        | |                             | in one year (e.g. 365 for d)
i5      | | nlrp(j)                     | number of load regions
        | |                             | with the same characteristics
        |  ------exit ------------------| => npk=j : number of
        |                               |    load characteristics
        |                               |    nlrm= number of
        |                               |    load regions
        |                               |
g12.5   | flr(j),j=1,nlrm               | fraction of time that
        |                               | load region j occurs
        |                               | in the year
        |                               | (is normalized to 1 after
        |                               | input)
i5      | ibas                          | =1 => extra base load is
        |                               | modeled
        |                               | =0 => no extra base load
        |                               | is modeled
        |                               | Extra base load can be modeled
        |                               | if the central conversion
        |                               | technologies are not included
        |                               | in the model and the cost
        |                               | difference of base- and peak load
        |                               | is given in the resource input
        |                               | (see alse chapter 3.3.6)
        |                               |
        |                               | => nlr=  nlrm + ibas
        |                               |    number of load regions
        |                               |    including base load
        |                               |
        |  ------j=0 -------------------|
4a1     | | ilev(j+1),eq   =* => exit   | number or identifier of level
        | |  ----- j=j+1  -------------| ordered from useful
        | |                             | energy to resources
        | |                             | if it is appended by a -eq
        | |                             | the balance equations on
        | |                             | this level are set to
        | |                             | equal instead of greater
        | |                             | than, which means that no
        | |                             | energy may be left over
        | |                             |
        | |                             | The level ids can be
        | |                             | chosen from the identifiers
        | |                             | described above.
        | |                             | At least two levels
        | |                             | are necessary, first level
```

```
         | |                                  | in the input is the demand,
         | |                                  | the last one the resource level
         | |                                  |
         | | if(zh(j).ne.U.and.nlrm.ne.0)     |
         | |  -----i=0 -----------------------| if the level is not the
         | | |                                | special demand level
         | | |                                | called U and load regions
         | | |                                | exist
a1-*     | | | name(j,i+1) =* => ex-i         | id of i-th energy form
         | | |                                | with load regions on level j
         | | |                                | the description appended to the
         | | |                                | identifier is used as
         | | |                                | identification in the report
#        | | |                                | after each fuel id a
#        | | |                                | description can follow,
#        | | |                                | it will be written after
#        | | |                                | the fuel identification
#        | | |                                | in the report
         | | |------i=i+1 ---------------------|
         | |  ------ex-i ---------------------|
         | |                                  | => ndl(j)=i : number of
         | |                                  | energy forms with load
         | |                                  | regions on level j
         | |  -----i=i------------------------|
a1-*     | | | name(j,i+1) =* => ex-i         | id of (i-ndl(j))th energy
         | | |                                | form without load regions on
         | | |                                | level j that is no nuclear fuel
         | | |                                | the description appended to the
         | | |                                | identifier is used as
         | | |                                | identification in the report
#        | | |                                | after each fuel id a
#        | | |                                | description can follow,
#        | | |                                | it will be written after
#        | | |                                | the fuel identification
#        | | |                                | in the report
         | | |------i=i+1 ---------------------|
         | |  ------ex-i ---------------------|
         | |                                  | => nucf(j)=i : number of last
         | |                                  | no-nuclear fuel on level j
         | |  -----i=i------------------------|
a1-*     | | | name(j,i+1) =* => ex-i         | id of (i-nucf(j))th nuclear
         | | |                                | fuel on level j
         | | |                                | the description appended to the
         | | |                                | identifier is used as
         | | |                                | identification in the report
```

```
#       | | |                               | after each fuel id a
#       | | |                               | description can follow,
#       | | |                               | it will be written after
#       | | |                               | the fuel identification
#       | | |                               | in the report
        | | |-----i=i+1 -------------------|
        | |  -----ex-i --------------------|
        | |                                 | => nef(j)=i : number of
        | |                                 | energy forms on level j
        |  ------exit-j -------------------|
        |                                   | => nlev=j : number of
        |                                   | levels in the modeled energy
        |                                   | chain
        |                                   |
        |  ------j=0 ----------------------|
#       | |                                 | fuel description like
#       | |                                 | above
a1-*    | | id        =* => exit            | id of fuel with stock-pile
        | |                                 | (see chapters 3.1.8 and 3.2.31)
        | |                                 | the description appended to the
        | |                                 | identifier is used as
        | |                                 | identification in the report
        | |  -----j=j+1 -------------------|
g12.5   | | qex(j)                          | existing stock of that
        | |                                 | fuel in period ntu
#       | |                                 | after each fuel id a
#       | |                                 | description can follow,
#       | |                                 | it will be written after
#       | |                                 | the fuel identification
#       | |                                 | in the report
        |  ------exit ---------------------|
        |                                   | => nman=j : number of
        |                                   | man-made fuels
```

BLOCK 2

```
Format   | Variable Name                    | Description
         | Flow Control                     |
---------+----------------------------------+-------------------------
         |                                  |
         |  --------------------------------|
a1       | | iddem      =* =>exit           | id of fuel on demand level
         | |                                | having demands (has number i
         | |                                | in the input sequence)
         | |  -----j=1,nto------------------|
g12.5    | | | d(i,j)                       | annual demand for i in
         | | |                              | period j
         | |  ------------------------------|
         |  ------exit ---------------------|
         |                                  |
         | if(nu.gt.1)                      |
         |  --------------------------------|
a1       | | iddem      =* =>exit           | id of fuel on demand level
         | |                                | having the demands distributed
         | |                                | to the energy density areas
         | |  -----j=1,nu ------------------|
g12.5    | | | fdu(i,j)                     | fraction of demand for
         | | |                              | i that occurs in energy
         | | |                              | density area j, default is
         | | |                              | equal distribution
         | |  ------------------------------|
         |  -------exit --------------------|
         |                                  |
         | if(nlrm gt.0.and ndl(1).gt.0)    | if this energy chain is modeled
         | |                                | with load regions and some
         | |                                | energy forms on the demand
         | |                                | level have load regions
a1       | | iddem      =* =>exit           | id of fuel on demand level
         | |                                | having the demands distributed
         | |                                | to the load regions (the others
         | |                                | are assumed base load)
         | |  -----j=1,nlrm-----------------|
g12.5    | | | fdl(i,j)                     | fraction of demand for
         | | |                              | i that occurs in load
         | | |                              | region j
         | |  ------------------------------|
         |  -------exit --------------------|
         |                                  |
```

```
        |  -------------------------|
#       | |                         | for the report each
#       | |                         | elasticity function can be
#       | |                         | described before its definition
2a1     | | iddem-idu   =* =>exit   | id of fuel on demand level
        | |                         | having a demand elasticity,
        | |                         | can be followed by the id of
        | |                         | the energy density area (idu),
        | |                         | else all energy density areas are
        | |                         | set equal (see chapter 3.3.12)
        | |  -----l=0 --------------|
g12.5   | | | rred(i,j,l+1)         | fraction of demand
        | | |                       | being actually suplied
        | | |                       | on elasticity level l,
        | | |        0 => exit-l    | j is set according to idu
        | | |  ----l=l+1 -----------|
g12.5   | | | cred(i,j,l)           | cost of reducing the demand
        | | |                       | to that level
        | |  -----exit-l -----------|
        | |                         | => nred (i,j)=l : number
        | |                         | of reduction levels for
        | |                         | energy form  i in energy
        | |                         | density area j
        |  ------exit --------------|
```

BLOCK 3

```
Format  | Variable Name               | Description
        | Flow Control                |
--------+-----------------------------+------------------------
        |                             |
        |  ------i=0 -----------------|
6a1-*   | | namat(i):x  =* => exit-i  | identifier of user defined
        | |                           | relation number i of type 1
        | |                           | The id can be appended by a
        | |                           | :x, where x is the identifier
        | |                           | of the load region for that it
        | |                           | should be generated.
        | |                           | Specifications of the same
        | |                           | relation with different load
        | |                           | regions have to come in a
        | |                           | sequence. If x is set to '+'
```

```
         | |                                  | the relation is established
         | |                                  | between periods (see chapter
         | |                                  | 3.2.30). This whole
         | |                                  | identification can be
         | |                                  | appended by a -xx before
         | |                                  | the description, then the
         | |                                  | according relations
         | |                                  | are set to
         | |                                  |   - equal for xx=fx,
         | |                                  |   - greater than for xx=lo, and
         | |                                  |   - less than for xx=up
         | |                                  | (up is default).
         | |  -----i=i+1 ---------------------|
*ts      | | pr                               | price of user defined
         | |                                  | relation of type 1
         | |                                  | per period, eventually
         | |                                  | load region
*ts      | | max                              | limit on user defined
         | |                                  | relation of type 1
         | |                                  | per period, eventually
         | |                                  | load region
#        | |                                  | after its definition each user
#        | |                                  | defined relation can be
#        | |                                  | described for the report
         |  ------exit-i ---------------------|
         |                                    | => nmat = i : number of
         |                                    | user defined relations
         |                                    | of type 1
         |                                    |
         |  ------i=0 ------------------------|
6a1-*    | | pol(i):x    =* => exit-i         | identifier of user defined
         | |                                  | relation number i of type 2
         | |                                  | The id can be appended by a
         | |                                  | :x, where x is the identifier
         | |                                  | of the load region for that it
         | |                                  | should be generated.
         | |                                  | Specifications of the same
         | |                                  | relation with different load
         | |                                  | regions have to come in a
         | |                                  | sequence. If x is set to '+'
         | |                                  | the relation is established
         | |                                  | between periods (see chapter
         | |                                  | 3.2.30). This whole
         | |                                  | identification can be
         | |                                  | appended by a -xx before
```

```
           | |                                  | the description, then the
           | |                                  | according relations
           | |                                  | are set to
           | |                                  |   - equal for xx=fx,
           | |                                  |   - greater than for xx=lo, and
           | |                                  |   - less than for xx=up
           | |                                  | (up is default).
           | |  -----i=i+1 ---------------------|
*ts        | | pr                               | price of user defined
           | |                                  | relation of type 2
           | |                                  | per period, eventually
           | |                                  | load region
*ts        | | max                              | limit on user defined
           | |                                  | relation of type 2
           | |                                  | per period, eventually
           | |                                  | load region
#          | |                                  | after its definition each user
#          | |                                  | defined relation can be
#          | |                                  | described for the report
           |  ------exit-i ---------------------|
           |                                    | => npol = i : number of
           |                                    | user defined relations
           |                                    | of type 2
           |                                    |
           | if(ibas.eq.1)                      |
a1         | | idll                             | if ibas=1 base load demand
           | |                                  | is relocated on level idll
           | |                                  | leaving a fraction of peak
           | |                                  | see chapter 3.3.6)
           | | if(idll.ne.*)                    |
           | |  -----i=1,ndl(nll) --------------| nll is the level with id idll
g12.5      | | | frpk(i)                        | fraction of energy form
           | | |                                | i that has to remain peak
           | |  --------------------------------|
```

BLOCK 4

```
Format  | Variable Name                          | Description
        | Flow Control                           |
--------+----------------------------------------+------------------------------
        |                                        |
        |  ------while(isw.ne.*) ------          |
a4      | | isw        =* => exit                | id of type of cost getting a
        | |                                      | weight (see chapter 3.2.32
        | |                                      | for the definition)
g12.5   | | amul                                 | multiplicative weight
g12.5   | | add                                  | additive weight
        |  ------exit --------------             |
        |                                        |
```

BLOCK 5
(beginning)

```
Format  | Variable Name                          | Description
        | Flow Control                           |
--------+----------------------------------------+------------------------------
        |                                        | ndev=0
        |                                        | ivar=0
        |                                        | nsto=0
        |                                        |
        | while(its.ne.*)                        |
        | |                                      |
4a1     | | its,eqn                              | its:
        | |                                      | =c => read conversion
        | |                                      |         technology
        | |                                      | =s => read storing
        | |                                      |         technology
        | |                                      | =v => read conversion
        | |                                      |         technology with
        | |                                      |         time series
        | |                                      | =* => stop technology
        | |                                      |         input
        | |                                      | (can be appended by a
        | |                                      | -fx for its=c or v;
        | |                                      | then the capacity
        | |                                      | equations are set to
        | |                                      | equal )
```

#		description of the
#		previous technology
#		for the report

4.2.3 Conversion Technologies

If the current technology is linked to the capacity of the preceding one (see chapters 3.2.23 d and 3.3.9) only the entries signed with an asterisk ('*') after the format are read. In this case there is also an additional entry after the last year of construction.

A *tst indicates that, if the technology was switched with its=v, this entry is read as time series according to routine 'tst' described in chapter 4.3.1, a &tst means that the time-series routine is used, but only constant parameters can be used in the current version. For entries in integer format the according routines are called *its and &its.

BLOCK 5a

Format	Variable Name Flow Control	Description
4a1 *	its,eqn (its= c or v)	switch conversion/storage and variable entries if eqn = '-fx' the capacity equations (see chapter 3.2.23) of this technology are set to equal instead of less than
		ndev=ndev+1 ; n=ndev
a1-* *	ida(n)	identifier of conversion technology no n the description after this identifier will be used as header for the technology in the report
a1 *	idx	level of output of

```
          |                                      | technology n (can be
          |                                      | the number or the identifier
          |                                      | as described in chapter 4.2.2)
          |                                      | the main output of the
          |                                      | technology has to lie on
          |                                      | on this level
          |                                      |
5a1  *    | idur(n)                              | energy density area(s) in
          |                                      | which the technology
          |                                      | exists; e.g.:
          |                                      | =* => in all,
          |                                      | =xyz => in areas with the
          |                                      | ids x,y,z
          |                                      | (definition of areas
          |                                      | from general input, max=4)
          |                                      | this switch is only applicable
          |                                      | if at least the main output
          |                                      | lies on a level with energy
          |                                      | density areas as defined in
          |                                      | chapter 4.2.2
          |                                      |
i5   *    | iref                                 | reference year of
          |                                      | technology n, defines, if it
          |                                      | exists or from when on it
          |                                      | can be built in future
          |                                      |
i5   *    | lref                                 | last year in that the
          |                                      | device can be built
          |                                      | (=0 => no last year)
          |                                      |
          | if(ivar.gt.0)                        | (i.e. this technology
          | |                                    | is linked to a previous
          | |                                    | one)
&tst *    | | rel                                | relation between use of
          | |                                    | capacity for main output
          | |                                    | of main technology and
          | |                                    | current technology (if
          | |                                    | the use is lower in the
          | |                                    | linked technology rel
          | |                                    | is greater than 1., see
          | |                                    | chapter 3.2.23d)
          |                                      |
          |  ------j=0 --------------------------|
4a1  *    | | id-lev    id=* => exit             | id: identifier of inputs
          | |                                    | lev: identifier of the
```

```
         | |                           |    according level,
         | |                           | if no level is given,
         | |                           | the code looks for id
         | |                           | as energy form on the
         | |                           | levels in the direction
         | |                           | to the resources from
         | |                           | the next level after
         | |                           | the output level;
         | |                           | a technology can
         | |                           | remain without input
         | |  -----j=j+1 ------------|
         | |if(j.eq.1)                 |
g12.5 *  | | | amount                  | amount of main input
         | |                           |
         | |if(j.gt.1)                 |
*tst  *  | | | amount                  | amount of additional input
         | |                           |
         |  ------exit --------------|
         |                             |
         |  ------j=0 ---------------|
4a1      | | id-lev    id=* => exit    | id: identifier of inventories
         | |                           | lev: identifier of the
         | |                           |    according level,
         | |                           | if no level is given,
         | |                           | the code looks for id
         | |                           | as energy form on the
         | |                           | levels in the direction
         | |                           | to the resources from
         | |                           | the next level after
         | |                           | the output level;
         | |                           | a technology can
         | |                           | remain without inventory
         | |  -----j=j+1 ------------|
*tst     | | amount                    | amount of inventory
         | |                           | per unit of new built
         | |                           | capacity for main output
         |  ------exit --------------|
         |                             |
         |  ------j=0 ---------------|
4a1   *  | | id-lev      =* => exit    | id of output of technology
         | |                           | a technology has to have
         | |                           | at least one output, the
         | |                           | first one (or main output)
         | |                           | has to be on the output
         | |                           | level of the technology
```

```
          | |                                  | (idx), the other identifiers
          | |                                  | are searched form this level
          | |                                  | towards the demand level,
          | |                                  | if lev is not specified
          | |                                  |
          | |  -----j=j+1 ------------|
*tst  *   | | amount                           | amount of output (relative
          | |                                  | to main input in the 1st
          | |                                  | period)
          |  ------exit -------------|
          |                                    |
          |  ------j=0 -------------|
4a1       | | id-lev      =* => exit           | id of man-made or
          | |                                  | nuclear fuel
          | |                                  | lev has the same meaning
          | |                                  | as in the other cases,
          | |                                  | the search is the same
          | |                                  | as for outputs
          | |                                  |
          | |  -----j=j+1 ------------|
*tst      | | amount                           | amount of
          | |                                  | retirement (amount
          | |                                  | going back to the
          | |                                  | energy balance equation
          | |                                  | after last year of
          | |                                  | operation relative to
          | |                                  | the installed capacity)
          |  ------exit -------------|
          |                                    |
          |  ------j=0 -------------|
5a1   *   | | id:x        =* => exit           | id of user defined relation
          | |                                  | of type 1 related to
          | |                                  | operation, can be appended
          | |                                  | by the number of the load
          | |                                  | region in that the following
          | |                                  | amounts occur, specifications
          | |                                  | of the same accounting rows
          | |                                  | have to come in sequence.
          | |  -----j=j+1 ------------|
*tst  *   | | amount                           | coefficient for this
          | |                                  | accounting row relative
          | |                                  | to one unit of main
          | |                                  | output or input according
          | |                                  | to the switch in the main
          | |                                  | input. If the relation
```

```
            | |                                  | is defined to have load
            | |                                  | regions the amounts
            | |                                  | will be related to used
            | |                                  | capacity, not to production
            | |                                  | in this part of the year.
            | |                                  | If isw (see description of
            | |                                  | *ts, chapter 4.3.1) is set
            | |                                  | to 20+isw the coefficients
            | |                                  | are divided by the demand
            | |                                  | supplied by this technology.
            | |                                  | The technology has to lie
            | |                                  | on the demand level and the
            | |                                  | demands have to be nonzero,
            | |                                  | then; see chapter 3.2.30.)
            |  -------exit --------------|
            |                                    | if the id of a row is
            |                                    | not found it will be
            |                                    | skipped together with the
            |                                    | following time series
            |                                    | without warning.
            |                                    | Switch imapo can be set
            |                                    | in the general input to
            |                                    | to prevent this.  Then
            |                                    | input is stopped when
            |                                    | an identifier is not found
            |                                    |
            |  ------j=0 --------------|
6a1         | | id:x      =* => exit            | id of relation of type 1
            | |                                  | related to construction
            | |                                  | or, if x='t' or a load
            | |                                  | region number, related
            | |                                  | to total installed
            | |                                  | capacity as a whole or
            | |                                  | per load region.
            | |  -----j=j+1 -------------|
*tst        | | amount                           | coefficient relative
            | |                                  | to one unit of capacity
            | |                                  | for main output
            |  -------exit --------------|
            |                                    |
            |  ------j=0 --------------|
5a1   *     | | id:x      =* => exit            | id of user defined relation
            | |                                  | of type 2 related to
            | |                                  | operation, can be appended
            | |                                  | by the number of the load
```

```
          | |                                  | region in that the following
          | |                                  | amounts accur, specifications
          | |                                  | of the same accounting rows
          | |                                  | have to come in sequence.
          | |  -----j=j+1 ------------|
*tst *    | | amount                           | coefficient for this
          | |                                  | accounting row relative
          | |                                  | to one unit of main
          | |                                  | output or input according
          | |                                  | to the switch in the main
          | |                                  | input.  If the relation
          | |                                  | is defined to have load
          | |                                  | regions the amounts
          | |                                  | will be related to used
          | |                                  | capacity, not to production
          | |                                  | in this part of the year.
          | |                                  | If isw (see description of
          | |                                  | *ts, chapter 4.3.1) is set
          | |                                  | to 20+isw the coefficients
          | |                                  | are divided by the demand
          | |                                  | supplied by this technology.
          | |                                  | The technology has to lie
          | |                                  | on the demand level and the
          | |                                  | demands have to be nonzero,
          | |                                  | then; see chapter 3.2.30.)
          |  -------exit -------------|
          |                                    | if the id of a row is
          |                                    | not found it will be
          |                                    | skipped together with the
          |                                    | following time series
          |                                    | without warning
          |                                    | switch imapo can be set
          |                                    | in the general input to
          |                                    | to prevent this.  Then
          |                                    | input is stopped when
          |                                    | an identifier is not found
          |                                    |
          |  ------j=0 -------------|
6a1       | | id:x      =* => exit             | id of relation of type 2
          | |                                  | related to construction
          | |                                  | or, if x='t' or a load
          | |                                  | region number, related
          | |                                  | to total installed
          | |                                  | capacity as a whole or
          | |                                  | per load region.
```

```
          | |  -----j=j+1 ------------|
*tst      | | amount                  | coefficient relative
          | |                         | to one unit of capacity
          | |                         | for main output
          |  -------exit -------------|
          |                           |
&its  *   | lag(n)                    | lag-time from input to
          |                           | output [in years] (see
          |                           | chapter 3.3.8)
          |                           |
i5    *   | isw                       | switch for fixing
          | if(isw.eq.1)              | production pattern
          | |                         | of technology over the
          | |                         | load regions: (see
          | |                         | chapter 3.3.4)
          | |                         |  - for conversion
          | |                         |    technologies:
          | |                         |    the production pattern
          | |                         |    of the technology in
          | |                         |    the load regions is
          | |                         |    fixed, the plant
          | |                         |    factor is applied
          | |                         |    to the load region
          | |                         |    with the highest use
          | |                         |    of the capacity
          | |                         |  - for end-use devices
          | |                         |    (with output level U):
          | |                         |    the technology is
          | |                         |    assumed to have storage
          | |                         |    included, the output
          | |                         |    fits to the demand
          | |                         |    pattern, the input is
          | |                         |    changed according to
          | |                         |    a fixed pattern, the
          | |                         |    factor is applied
          | |                         |    to the whole
          | |                         |    production
          | |                         |
&tst  *   | | frout(n,j),j=1,nlrm     | fraction of production
          | |                         | per load region
          | if(isw.eq.2)              | production pattern is fixed
          |                           | to base load
          |                           |
i5        | isw                       | switch for time-distri-
          |                           | bution of investments
```

```
       |                                   | and accounting
       |                                   | during construction
       |                                   | (see chapter 3.3.2)
       |                                   | =0 => no distribution
       | (if isw .ne. 0)                   |
       | |                                 | isw = number of years
       | |                                 | to that accounting
       | |                                 | and investments
       | |                                 | are distributed
a1     | | nfnc                            | >0 => function already
       | |                                 |       known, no nfnc
       | |                                 |       in general input
       | |                                 | =* => investments not
       | |                                 |       distributed
       | |                                 | else=> new kind of funct-
       | |                                 |       ion, input like in
       | |                                 |       general input
       | |                                 |       (nfnc =e,l,p,d),
       | |                                 |       mdistr=mdistr+1
       | |  -----j=0 ----------------------|
a4     | | | id      =* => exit            | id of type 1 relation
       | | |                               | related to construction
       | | |                               | that is distributed over
       | | |  ----j=j+1 -------------------| time
       | | |                               |
i5     | | | nfnc                          | >0 => function already
       | | |                               |       known, no nfnc
       | | |                               |       in general input
       | | |                               | else=> new kind of function
       | | |                               |       input like in
       | | |                               |       general input
       | | |                               |       (nfnc =e,l,p,d),
       | | |                               |       mdistr=mdistr+1
       | |  -----exit ---------------------|
       | |                                 |
       | |  -----j=0 ----------------------|
a4     | | | id      =* => exit            | id of type 2 relation
       | | |                               | related to construction
       | | |                               | that is distributed
       | | |                               | (all input like for type 1
       | | |                               | relations)
       | |  -----exit ---------------------|
       | |                                 |
       |  ------endif ---------------------|
       |                                   |
```

i5	iswint	switch for mixed integer not equal 0 => capacity variables of this technology are integer, they give the number of new installations, see chapter 3.3.14
	if(iswint.ne.0)	
&tst	cmin(n)	size of one installation per period (if iswint = 1, the capacity variable is related to cmin/lp(k)) size of one installation per year (if iswint = 2, the capacity variable is related to cmin)
i5 *	nlinc	switch for alternative sorting in case of nonlinear objective, see also chapter 3.3.15 = 0 => no variable of the technology has nonlinear objective coefficients = 1 => capacity = 2 => activity = 2 => capacity and activity variables have nonlinear objective coefficients
i5 *	idum	dummy switch, not jet used
*tst	pf(n)	plant factor or availability factor if the technology doesn't have load regions or has output level U: maximum use of the capacity if it has load regions: maximum use of the capacity per load region default is 1.

```
           |                                      |
&tst       | pl(n)                                | technical plant life
           |                                      | in years
           |                                      | default is 30.
           |                                      |
*tst       | capc(n)                              | capital costs of device
           |                                      | in $/kW main output,
           |                                      | (as default)
           |                                      | if capc(n)<0 => no
           |                                      | capacity and dynamic
           |                                      | constraint equations
           |                                      | and capacity variables
           |                                      | are generated for this
           |                                      | technology
           |                                      |
*tst       | fix(n)                               | fixed operation and
           |                                      | maintenance costs in
           |                                      | $/kW/yr main output
           |                                      | (as default)
           |                                      |
*tst  *    | curc(n)                              | variable operation and
           |                                      | maintenance costs in
           |                                      | $/kWyr main output
           |                                      | (as default)
           |                                      |
           |                                      | inu=nu
           |                                      | if(idx.gt.nul)inu=1
           |                                      |
           | if(iref.lt.lbl(ntu))                 |
           |  ------j=1,inu ----------------------|
           | | if(idur(n).ne.*.and.               |
           | |   namu(j) is not in idur(n)) exit-3
           | |  ----------------------------------|
g12.5 *    | | | actin(n,j)                       | production in year lbl(ntu)-lpo
*ts        | | | capin(n,j)                       | capacity in the year lbl(ntu)-lpo
           | | |                                  | (only switches 0, 1 and 2
           | | |                                  | can be used here, if switch 1
           | | |                                  | is used here, iref is the lower,
           | | |                                  | lbl(ntu)-lpo the upper loop index
           | | |                                  | for reading the single values, a
           | | |                                  | maximum of pl values is read)
           | | |  --------------------------------|
           | |------exit-3------------------------|
           |                                      |
           |  ------j=0 --------------------------|
```

```
a4   *    | | id       id(2)=* => exit          | id for activity bounds
          | |                                   | (fx, lo, up, fr, init)
          | |                                   | all upper bounds to 0.
          | |                                   | are changed to fixed
          | |                                   | bounds
          | |                                   |
          | | if(id.ne.fr)                      | for free bounds no values
          | | |                                 | are read, init bounds relate
          | | |                                 | to the nonlinearity option,
          | | |                                 | see chapter 3.3.15
          | | |------j=j+1 ---------------------|
          | | |------l=1,nu --------------------| loop over energy
          | | |                                 | density areas if needed
          | | |------i=1,nred ------------------| loop over reduction
          | | |                                 | levels of demand if
          | | |                                 | level is demand level
          | | |------ll=1,nlr-------------------| loop over load regions
          | | |                                 | if one of the handled
          | | |                                 | energy forms has load regions
*ts  *    | | | bnd                             | bound on annual output
          | | |                                 | if(idx=U, -1<=bnd<0 =>
          | | |                                 |   -bnd gives the fraction
          | | |                                 |   of demand that is then
          | | |                                 |   used as bound
          | | |                                 | >= 1.e7 => the bound is
          | | |                                 |   not applied
          |   -------exit ----------------------|
          |                                     |
          |   -------j=0 -----------------------|
a4        | | id       id =* => exit            | id for construction
          | |                                   | (fx, lo, up, fr, init)
          | |                                   | (they are set to li
          | |                                   | and ui for integer
          | |                                   | variables internally,
          | |                                   | integer variables
          | |                                   | have to be bounded, if
          | |                                   | no bounds are given, the
          | |                                   | according variable is set
          | |                                   | to a binary (0/1) one)
          | |                                   | all upper bounds to 0.
          | |                                   | are changed to fixed
          | |                                   | bounds
          | |                                   |
          | | if(id.ne.fr)                      | for free bounds no values
          | | |                                 | are read, init bounds relate
```

```
            | | |                              | to the nonlinearity option,
            | | |                              | see chapter 3.3.15
            | | |------j=j+1 ------------------|
            | | |------l=1,nu -----------------|
*ts         | | | bnd                          | bound on annual
            | | |                              | >= 1.e7 => the bound is
            | | |                              |   not applied
            |   -------exit -------------------|
            |                                  |
            |   -------j=0 --------------------|
2a1   *     | | id      id =* => exit          | id for dynamic constraint
            | |                                | on activity (up or lo)
            | |                                | see chapters 3.2.28 and 3.2.29
            | |  ------j=j+1 ------------------|
*tst  *     | | g                              | increment for dynamic
            | |                                | constraint (in MWyr, etc.)
*tst  *     | | gam                            | growth for dynamic constraint
            |   -------exit -------------------|
            |                                  |
            |   -------j=0 --------------------|
2a1         | | id      id =* => exit          | id for dynamic constraint
            | |                                | on capacity (up or lo)
            | |                                | see chapters 3.2.26 and 3.2.27
            | |  ------j=j+1 ------------------|
*tst        | | g                              | increment for dynamic
            | |                                | constraint (in MW, etc.)
*tst        | | gam                            | growth for dynamic constraint
            |   -------exit -------------------|
            |                                  | if (ivar.ne.0) ivar=ivar-1
            |                                  |
i5          | ivar                             | number of additional
            |                                  | conversion technologies
            |                                  | that are linked to the
            |                                  | same capacity, they have
            |                                  | to follow in the input file
```

4.2.4 Storage Technologies

BLOCK 5b

Format	Variable Name Flow Control	Description
i5	its = s	identifier for storage technology nstor=nstor+1 ; n=nstor
a1	ida	identifier of storage technology no n
a1-*	idx	id of level on that storage n works
a1	idi	load characteristic, for that this storage works, else it is set to the one defined as last one); see chapter 3.3.3.
g12.5	fgv(n)	>0 => fgv(n) = ratio between size of I/O and volume capacity <0 or =0 => I/O and volume capacities are independent see chapters 3.2.25 and 3.3.5
a5	idur	energy density area(s) in which the storage exists =* => in all (used like for conversion technologies)
i5	iref	reference year >= lbl(ntu) => 1st year when it can be built in future < lbl(ntu) => since when it exists
i5	lref	last year in that device can be built (=0 => no last year)
a1	id	id of stored energy form has to be defined with load regions and lie on level idx
g12.5	eff	efficiency of I/O part of

```
          |                                         | storage => eff of total
          |                                         | I/O = eff*eff
g12.5     | exl                                     | parameter for decay of
          |                                         | content (=0 => no decay)
          |                                         | see chapters 3.3.5 and 3.2.22
g12.5     | alf                                     | fraction of time in the modeled
          |                                         | peak (idi) the content can be kept
          |                                         |
          |  ------j=0 -----------------------------|
a4        | | id        =* => exit                  | id of type 1 relation
          | |                                       | for construction or total
          | |                                       | installed capacity of I/O
          | |                                       | part (see conversion
          | |                                       | technologies)
          | |  -----j=j+1 --------------------------|
g12.5     | | amount                                | amount per unit of
          | |                                       | capacity
          |  ------exit ----------------------------|
          |                                         |
          | if(fgv(n).le.0)                         |
          | |  -----j=0 ----------------------------|
a4        | | | id        =* => exit                | id of type 1 relation
          | | |                                     | for construction or total
          | | |                                     | installed capacity of
          | | |                                     | volume part (see conversion
          | | |                                     | technologies)
          | | |  ----j=j+1 -------------------------|
g12.5     | | | amount                              | amount per unit of
          | | |                                     | capacity
          | |  -----exit ---------------------------|
          |                                         |
          |  ------j=0 -----------------------------|
a4        | | id        =* => exit                  | id of type 2 relation
          | |                                       | for construction or total
          | |                                       | installed capacity of I/O
          | |                                       | part (see conversion
          | |                                       | technologies)
          | |  -----j=j+1 --------------------------|
g12.5     | | amount                                | amount per unit of
          | |                                       | capacity
          |  ------exit ----------------------------|
          |                                         |
          | if(fgv(n).le.0)                         |
          | |  -----j=0 ----------------------------|
a4        | | | id        =* => exit                | id of type 2 relation
```

```
             | | |                                    | for construction or total
             | | |                                    | installed capacity of
             | | |                                    | volume part (see conversion
             | | |                                    | technologies)
             | | |-----j=j+1 -------------------------|
g12.5        | | | amount                             | amount per unit of
             | |                                      | capacity
             | |   -----exit -------------------------|
             |                                        |
i5           | isw                                    | switch for time-distri-
             |                                        | bution of investments
             |                                        | and coefficients in relations
             |                                        | related to construction
             |                                        | of I/O part
             |                                        | =0 => no distribution
             | if isw .ne. 0)                         |
             | |                                      | isw = number of years
             | |                                      | to that coefficients
             | |                                      | and investments
             | |                                      | are distributed
a1           | | nfnc                                 | > 0 => function already
             | |                                      |        known, no nfnc
             | |                                      |        in general input
             | |                                      | = * => investments not
             | |                                      |        distributed
             | |                                      | else=> new kind of function,
             | |                                      |        input like in
             | |                                      |        general input
             | |                                      |        (nfnc =e,l,p,d),
             | |                                      |        mdistr=mdistr+1
             | |   -----j=0 --------------------------|
a4           | | | id       =* => exit                | id of type 1 relation
             | | |                                    | related to construction
             | | |                                    | that is distributed over
             | | |  ----j=j+1 ------------------------| time
             | | |                                    |
i5           | | | nfnc                               | > 0 => function already
             | | |                                    |        known, no nfnc
             | | |                                    |        in general input
             | | |                                    | else=> new kind of function,
             | | |                                    |        input like in
             | | |                                    |        general input
             | | |                                    |        mdistr=mdistr+1
             | |   -----exit -------------------------|
             | |                                      |
```

```
          | |  -----j=0 --------------|
a4        | | | id      =* => exit      | id of type 2 relation
          | | |                         | related to construction
          | | |                         | that is distributed
          | | |                         | (all input like for type 1
          | | |                         | relations)
          | |  -----exit --------------|
          | |                           |
          |  ------endif --------------|
          |                             |
          | if(fgv(n).le.0)             |
i5        | | isw                       | switch for time-distri-
          | |                           | bution of investments
          | |                           | and coefficients in
          | |                           | relations related to
          | |                           | construction of volume
          | |                           | part
          | |                           | =0 => no distribution
          | |  -----if isw .ne. 0)------|
          | | |                         | isw = number of years
          | | |                         | to that coefficients in
          | | |                         | relations and investments
          | | |                         | are distributed
a1        | | | nfnc                    | = i => function already
          | | |                         |        known, no nfnc
          | | |                         |        in general input
          | | |                         | = * => investments not
          | | |                         |        distributed
          | | |                         | else=> new kind of function,
          | | |                         |        input like in
          | | |                         |        general input
          | | |                         |        (nfnc =e,l,p,d),
          | | |                         |        mdistr=mdistr+1
          | | |  ----j=0 --------------|
a4        | | | | id      =* => exit    | id of type 1 relation
          | | | |                       | related to construction
          | | | |                       | that is distributed over
          | | | |  ---j=j+1 -----------| time
          | | | |                       |
i5        | | | | nfnc                  | > 0 => function already
          | | | |                       |        known, no nfnc
          | | | |                       |        in general input
          | | | |                       | else=> new kind of function,
          | | | |                       |        input like in
          | | | |                       |        general input
```

```
          | | | |                                   |     mdistr=mdistr+1
          | | |  ----exit --------------------------|
          | | |                                     |
          | | |  ----j=0 ---------------------------|
a4        | | | | id       =* => exit               | id of type 2 relation
          | | | |                                   | related to construction
          | | | |                                   | that is distributed
          | | | |                                   | (all input like for type 1
          | | | |                                   | relations)
          | | |  ----exit --------------------------|
          | | |                                     |
          | |  -----endif --------------------------|
          |                                         |
i5        | idum x 3                                | dummy switches
          |                                         |
          | if(fgv(n).le.0)                         |
i5        | | idum x 3                              | dummy switches
          |                                         |
g12.5     | pfs                                     | availability factor
          |                                         |
g12.5     | pls                                     | technical plant life in
          |                                         | years
          |                                         |
g12.5     | capcs                                   | capital costs of I/O
          |                                         | part of this storage
          |                                         | in $/kW (if no other unit
          |                                         | is used)
          |                                         |
g12.5     | fixs                                    | fixed operation and
          |                                         | maintenance costs in
          |                                         | $/kW for the I/O part
          |                                         | (default)
          |                                         |
g12.5     | curcs                                   | variable operation and
          |                                         | maintenance costs in
          |                                         | $/kWyr for the I/O part
          |                                         | (default)
          |                                         |
          |  ------if fgv(n) .lt. 0) ---------------|
g12.5     | | capcsv                                | capital costs of volume
          | |                                       | part of this storage in
          | |                                       | $/kWyr content (default)
          | |                                       | if capcsv   <0 => no
          | |                                       | capacity and dynamic
          | |                                       | constraints and capacity
```

```
          | |                                  | variables are generated
          | |                                  | for the volume part of
          | |                                  | this storage
          | |                                  |
g12.5     | | fixsv                            | fixed operation and
          | |                                  | maintenance costs in
          | |                                  | $/kWyr for the volume part
          | |                                  |
g12.5     | | curcsv                           | variable operation and
          | |                                  | maintenance costs in
          |                                    | $/kWyr/yr for the volume
          |                                    | part
          |                                    |
          | if(iref<lpl(ntu))                  |
          | |  -----j=1,nu -------------------| according to idurs
*ts       | | | histcaps                       | capacity for I/O existing
          | | |                                | in base year
          | |  -------------------------------|
          | |                                  |
          | | if(fgv(n).lt.0)                  |
          | | |  ----j=1,nu ------------------| according to idurs
*ts       | | | | histcaps                     | capacity for volume
          | | | |                              | existing in base year
          | | |  -----------------------------|
          | |                                  |
          |                                    |
          |  ------j=0 ------------------------|
2a1       | | id       id(2)=* => exit         | id for construction
          | |                                  | bounds (fx, lo, up)
          | |                                  | on I/O capacity
          | |  -----j=j+1 --------------------|
          | |  -----l=1,nu -------------------|
*ts       | | bnds                             | bound on annual
          | |                                  | construction of I/O part
          |  ------exit -----------------------|
          |                                    |
          | if(fgv(n).le.0)                    |
          |  ------j=0 ------------------------|
2a1       | | id       id(2)=* => exit         | id for construction
          | |                                  | bounds (fx, lo, up)
          | |                                  | on volume capacity
          | |  -----j=j+1 --------------------|
          | |  -----l=1,nu -------------------|
*ts       | | bnds                             | bound on annual construction
          | |                                  | construction of volume part
```

```
          |  ------exit -------------|
          |                          |
          |  ------j=0 --------------|
2a1       | | id      id =* => exit  |  id for dynamic constraint
          | |                        |  on I/O capacity (up or lo)
          | |  -----j=j+1 -----------|
*tst      | | g                      |  increment for dynamic
          | |                        |  constraint (in MW, etc.)
*tst      | | gam                    |  growth for dynamic constraint
          |  -------exit ------------|
          |                          |
          | if(fgv(n).le.0)          |
          |  -------j=0 -------------|
2a1       | | id      id =* => exit  |  id for dynamic constraint
          | |                        |  on volume capacity (up or lo)
          | |  -----j=j+1 -----------|
*tst      | | g                      |  increment for dynamic
          | |                        |  constraint (in MWyr, etc.)
*tst      | | gam                    |  growth for dynamic constraint
          |  -------exit ------------|
```

4.2.5 Resource Input

BLOCK 6

```
Format  | Variable Name              | Description
        | Flow Control               |
--------+----------------------------+-----------------------
i5      | idum * 10                  | 10 dummy switches,
        |                            | not yet used
*ts     | aresp                      | annual multipliers for
        |                            | costs of resource
        |                            | extraction
*ts     | aimpp                      | annual multipliers for
        |                            | prices for imports
*ts     | aexpp                      | annual multipliers for
        |                            | prices for exports
        |                            | nres=0
        |                            | nimp=0
        |                            | nexp=0
a1      | id          =* => exit-1   | identifier of fuel
```

```
a1      | lev                                  | id of level on that
        |                                      | the fuel is defined
        |                                      | = * => search from resource
        |                                      | level to demand level for id
        |                                      | as energy form
        |  ------------------------------------|
#       | |                                    | explanation on this fuel
a1      | | ideq       =* => exit-2            | identifier for kind of
        | |                                    | equation : r/i/e  =>
        | |                                    | resource/import/export
        | |                                    |
        | |                                    | the following input is
        | |                                    | only explained for
        | |                                    | resources (ideq=r)
        | |                                    | (imports/exports are
        | |                                    | the same)
        | |                                    |
        | |  ----------------------------------| nres=nres+1
*ts     | | spres                              | annual multipliers for
        | |                                    | costs of resource
        | |                                    | extraction of this fuel
*ts     | | aresl                              | limits on annual
        | |                                    | extraction of this fuel
        | |                                    |
        | |                                    | ngr(nres)=0 number of
        | |                                    | grades of this resource
        | |                                    |
a1      | | id         =* => exit-3            | id of grade
        | |  ----------------------------------| ngr(nres)=ngr(nres)+1
g12.5   | | | tres(nres,ngr(nres))             | total availability of
        | | |                                  | this grade (usually in MWyr,
        | | |                                  | etc.); if tres = -1.
        | | |                                  | => equation is not limited
        | | |                                  |
*ts     | | | aresel                           | limits on annual extrac-
        | | |                                  | tion of this grade
        | | |                                  |
        | | |                                  | if the fuel does not have
        | | |                                  | load regions: nl = 1
        | | |                                  | if the fuel has load
        | | |                                  | regions: nl = nlr
        | | |  ----l=1,nl----------------------|
        | | | |  ---ne=0 ----------------------|
a1      | | | | | idela  =* => exit-5          | switch for end of elasticity
        | | | | |                              | classes of this grade
```

```
            | | | | |                          | in load region l
            | | | | | --ne=ne+1----------------|
*ts         | | | | | aprel                    | cost (usually in $/kWyr;
            | | | | |                          | have to be negative
            | | | | |                          | for exports)
a2          | | | | | idlim   =* => exit-6     | id for kind of bound (up/lo/fx)
*ts         | | | | | | arese                  | annual bound
            | | | | | |                        | (usually in MWyr/yr, etc)
            | | | | | --exit-6-----------------|
            | | | | ---exit-5------------------| nela(nres,ngr,l)=ne
            | | | ---------------------------|
            | | |                              |
            | | | | ---nmp=0-------------------|
a2          | | | | idmp     =* => exit-4      | id for kind of dynamic
            | | | |                            | constraint on extraction
            | | | | ---nmp=nmp+1 --------------| per grade (up/lo)
            | | | |                            | see chapter 3.2.14
            | | | |                            |
g12.5       | | | | grg                        | increment (usually in MWyr, etc)
g12.5       | | | | gamrg                      | annual growth
            | | | ----exit-4-------------------|
            | | |                              |
g12.5       | | | resex(nres,ngr)              | annual extraction in
            | | |                              | base year
*ts         | | | resrem(nres,ngr)             | fraction of the resource
            | | |                              | left in that period that
            | | |                              | can be extracted in that
            | | |                              | period (default = 1.)
            | | |                              | see chapter 3.2.10
            | | -----exit-3--------------------|
            | |                                |
            | | | ----nmp=0--------------------|
a2          | | | idmp      =* => exit-4       | id for kind of dynamic
            | | |                              | constraint on total extraction
            | | |                              | (up/lo), see chapters 3.2.12
            | | |                              | and 3.2.13
            | | | ----nmp=nmp+1 ---------------|
g12.5       | | | gr                           | increment (usually MWyr, etc.)
g12.5       | | | gamr                         | annual growth
            | | | ----exit-4-------------------|
            | | -----exit-2--------------------|
            | ------exit-1---------------------|
```

4.3 The Physical Data Files

The matrix generator of *MESSAGE II (MXG)* reads the input data described in chapter 4.2 entry by entry, each of them on a new line. To keep the data files smaller and easy to handle *CHIN* can be used to convert more condensed files into the ones needed by *MXG*. The principle idea is to allow the input files to be written in a format which gives a lot of freedom in grouping the various inputs. The program does also recognize some control characters which are used to identify variables belonging to different scenarios, to read the inputs from different physical data files and others which save some work during preparation of the input files. Additionally it is possible to include comments which are either ignored during generation of the actual input files or are passed on to the matrix generator. In the latter case the matrix generator writes these comments to a file which is used to prepare a report on all the input data used to describe the energy chain modeled (see also chapter 4.4).

4.3.1 Program Description of CHIN

The program reads the input files from UNITS 3 (general and technology input) and 4 (resource input) and writes each string, which is enclosed by an optional number of blanks, to a single line of a new file on UNITS 8 and 9. Repetition of the same string can be indicated by a comma instead of retyping the string. The first comma has to be separated by at least one blank from the string to be repeated. If a string is to be repeated more than once the according commas can also be typed without any blanks between them. A semicolon is interpreted as *END OF LINE*, thus it is possible to use the rest of the line for comments. (The semicolon can be typed immediately after the last string in the line.)

The program recognizes three types of variables:

-- Real Variables
Real variables are recognized by the mandatory decimal point and can consist of a maximum of 12 characters.

-- Integer Variables
A string containing no decimal point and no dash in the first 6 columns is interpreted as integer value or character string and is shifted to the right margin of an I6-format in the new file. An integer variable can consist of a maximum of six characters.

-- Character Variables
Strings containing a dash in one of the first 6 columns are interpreted as identifiers (e.g. fuel name, technology identifier, etc.) followed by an explanatory string after the dash. The identifier (1 to 4 characters that

can be followed by a :x, where x can be a load region number or a 't') is written so that it ends in the sixth column (including the :x, if it exists) followed by the rest of the string. In the matrix generator the identifier is used and the following 20 characters of the string are written to an intermediate file for the report writing program (*REPO*). Before this a string, a 'xx', can be included between dashes, which can identify the kind of equation used (fx, lo, up).

An explanatory string must not contain blanks or commas. Otherwise the whole string is either shifted to the wrong side of the string written to the new file or it is written to two separate lines both resulting in errors during matrix generation. If the explanatory string should contain a blank when written to the report it has to be identified by a tilde ( ). Lines marked with a number sign (#) in the first column are shifted right by ten columns, but still marked by a '#' in the first column. This information is then used by the report writing program.

For convenience it is also possible to use more than the standard input files. This is especially useful when certain parts of the input file are produced by other programs (e.g: demand data or bound data using program *CAP*) or when a simple data base, containing technical coefficients for the technologies included, is used. A diversion to another file is indicated by a 'commercial add' (@) in the first column followed by a two digit number indicating the FORTRAN UNIT assigned to the file required. (UNIT numbers 3, 4, 5, 6, 8, 9 and 10 are used by the program itself and thus reserved).

CHIN can also recognize and select marked lines. This can be useful when for instance alternative technologies are to be selected or different price evolutions for energy imports are to be used for different model runs. The according lines are to be marked by an exclamation mark in the first column followed by a plus sign (indicating inclusion) or a minus sign (indicating exclusion) and two digit numerical identifiers separated by slashes. The lines are selected according to a numerical identifier read from the standard input file (unit 5). Unmarked lines are read in any case. This implies that e.g., a line being marked with !+01/03 would be read if a 01 or a 03 is read from the standard input file.

To allow for more flexibility it is possible to define scenario identifiers that combine various numerical identifiers in separate lines of the input file. This is indicated by two exclamation marks followed by a name (up to 8 characters), a colon, a plus or minus sign (having the same meaning as above) and a list of numerical identifiers separated by slashes. The definition of !!sc1:+01/05/06 would imply that all lines being marked with one of the numerical identifiers are to be included in the input file when the string sc1! is read from the standard input file. These definitions can be extended or reduced from the standard input, e.g., sc1!-01+02 would choose scenario sc1 and exclude lines marked with !+01 and additionally include ones with !+02 from scenario sc1. Another way of defining these scenario identifiers is to type, e.g., sc2?+03/04 to the standard input, what would mean that all lines marked with !+03 or !+04 are to be included while reading the input file, the ones with !-03 or !-04 are to be ignored, then.

The second line of standard input for *CHIN* allows to reset five default values. When the first one is set to 1, an additional control output file, containing only the lines of the input files that are actually chosen, is written to unit 10. The next four switches allow to set the unit numbers that are used for input and output of the data files. Their sequence is:

-- unit number for first input file (usually general and technology input, default is 3),

-- unit number for first output file (default 8),

-- unit number for second input file (usually resource input, default is 4), and

-- unit number for second output file (default 9),

where a '0' as entry refers to the default value and a '-1' can be used to suppress reading from the according file. If an *END OF FILE* is encountered instead of these inputs, all values are set to their defaults.

If there is no exclamation mark found as first character in a line of th input files the variables in that line are interpreted as being scenario independent. The identification for comments to be written in the report (#) and to change the input unit (@) have to follow the scenario switch immediately, so to say in the new column 1.

The combination of the scenario switch and division options allows to keep input files for different scenarios separate, thus it is possible to keep the input files smaller.

The *END OF INFORMATION* is to be indicated by a commercial add (@) followed by a blank or a second commercial add.

4.3.2 The Control Input

The kind of output produced by *MXG* has to be controlled using two additional input files. The two files produced by *CHIN* are read by two different codes of the matrix generator--*ROWS* and *RES*. These two programs do also need information on the type of output they have to produce--either generate a matrix or provide *REPO* with the needed information or both. See table 1 for the switches needed. They are read in format I2.

Table 1: Output control for *ROWS* and *RES*.

Code	Switch	Produced Files
ROWS	0	matrix and information for *REPO*
	1	only information for *REPO*
RES	0	matrix
	1	information for *REPO*
	2	matrix and information for *REPO*

The first line of control input for *ROWS* does also have to contain the name of the LP solver used, if it is not *MINOS*--this information has to start in column 4, can have up to 8 characters and is forwarded to *RDSOL* as a switch for the format in that the solution has to be read.

4.4 The Report

If an input file of *MESSAGE II* is finished, the report writer *(REPO)* can be used to produce a control output (a report) of the input data. *REPO* can either be used interactively or in batch mode. In the first case--interactive use--it expects a technology identifier (*Zsvd* as described in chapter 3.1.1) from UNIT 5 and writes one line of information on this technology to UNIT 6. This one line consists of:

- investment costs per unit of total output,
- fix O+M costs per unit of total output,
- variable O+M costs per unit of total output,
- overall efficiency,
- plant life,
- plant factor, and
- average cost per unit of total output excluding fuel costs.

The formula used for this calculation is:

$$\frac{ccap \times \frac{dr\,(1+dr)^{\tau}}{(1+dr)^{\tau}-1} + cfix}{\pi} + ccur\ ,$$

where

$ccap$ is the investment per unit of total output,
$cfix$ is the fix O+M cost per unit of total output,
$ccur$ is the variable O+M cost excluding fuel cost per unit of total output,
dr is the discount rate,
τ is the technical plant life, and
π is the plant factor.

In batch mode *REPO* reads a control file from UNIT 8 that contains switches in format I5. They control the amount of information produced, and the page control for printing (see table 2 for a detailed description of this control file). The report, a FORTRAN output file with printer control characters in the first column, is written to UNIT 1 and a table of contents to UNIT 3.

The report produced in batch mode can readily be used as Technical Report for publications. After the first header and before the list of technologies additional information can be added, which will be included in the page control. All pages besides the definition of the energy chain have less than 80 characters per line; therefore the report can easily be copied. Appendix 3 contains the report for *STIM* as an example.

Table 2: List and Description of Control Input for *REPO*

no	no/yes	description
1.	0/1	general information on energy forms, load curve, demands, additional relations, etc.
2.	0/1	information on prices and limits on resources, imports and exports
3.	0/1/2	no/short/extensive technology information
4.	0/n	1: read info from UNIT n and write after the header DESCRIPTION OF TECHNOLOGIES (7 is a free UNIT)
5.	0/1	1: include cost tables for the technologies
6.	0/1	1: include short identifiers of energy forms, additional relations and technology
7.	0/1	1: include user-defined relations
8.	0/1	1: include annually built historic capacities
9.	0/1	1: include bounds
10.	0/n	n = number of page with 1. technology
11.	0/1	1: new page for each technology
12.	0/1	checks for technology identifier repetition
13.	0/1	requires input after switches : (a3,f10.4)currency name, exchange rate to $80
14.	0/1	0: costs are given relative to total output, 1: relative to main output
15.	nn	maximum number of lines per page (default 60)

line switched by no 13, e.g.:

```
DM     2.463    a3,f10.4: currency name, multiplier with $80
```

4.5 The Dimensioning Program CHDIM

This program is common to all codes related to the *MESSAGE II* model and is used to adapt the array dimensions of the codes to the specific requirements of an explicit application. All codes handling the same type of information have to contain the same data statements and common blocks with arrays being of equal size for a complete model run. *CHDIM* works on blocks of information in the FORTRAN codes. The beginning of such a block is identified by three COMMENT lines, where the second line contains a four character block name. The end of the block is signaled by one more empty COMMENT line. Thus such a block has the following pattern:

```
c
c name
c
            Block of information to be changed, e.g.; dimension statements,
            common blocks or data statements.
c
```

Program *CHDIM* reads three switches from the standard input (unit 5). The first switch tells *CHDIM* under which operating system the programs are running. Currently there are two options, i.e.; U for UNIX or C for Control Data operating systems NOS or NOS/BE, but these options can easily be extended to other operating systems if necessary. The next two switches control the performance of the program where the first one is related to dimensions of arrays common to all programs and the second one to dimensions of arrays contained in *CAP* only.

If the according switch is set to zero, the program tries to read the required dimension from files assigned to unit 4 (in the case of dimensions relevant to all programs) and unit 3 (in the case of dimensions relevant to *CAP* only). If no input file is found, the program uses default values when writing the new dimension statements. Here only the first input file is described in detail, the description of the first file can be found in [3]. The input file read from unit 4 contains the following variables (the default values are also shown):

NAME	DESCRIPTION	DEFAULT
levdat	- maximum number of levels (changes the data statement of the identifiers of the levels, the data statement containing the identifiers has to be changed then, too).	[7]
levmax	- maximum number of levels used (including fuels with stockpiles). This data statement changes the sizes of the arrays concerning energy forms.	[7]
ndmax	- maximum number of demands.	[25]
nefmax	- maximum number of energy forms per level (has to be greater or equal to the previous entry).	[25]
ntmax	- maximum number of periods (counted from 1, not from ntu)	[8]
nredm	- maximum number of reduction levels for demand elasticities.	[1]
nbym	- maximum number of by-inputs and -outputs (for all conversion technologies together).	[240]
nhoutm	- maximum number of conversion technologies with fixed production pattern.	[11]
npkmax	- maximum number of load characteristics for the load regions	[1]
nlrmax	- maximum number of load regions (including base load)	[5]
numax	- maximum number of energy density areas	[1]
nendm	- relates to a feature not active any more	
nrhsm	- dummy has to be = 1	[1]

ncost	- number of accounting rows for capital costs, etc. (see chapter xx) (relates to data statement with names of rows)	[9]
ndevm	- maximum number of conversion technologies	[100]
nstom	- maximum number of storage technologies	[1]
nmatm	- maximum number of type 1 relational constraints	[16]
npolm	- maximum number of type 2 relational constraints	[6]
ncap	- maximum number of periods for historic capacities (=plant life / length of historic periods).	[12]
nucmax	- maximum number of nuclear fuels (including fuels with stock-piles).	[3]
nmanm	- maximum number of fuels with stock-pile.	[2]
mdism	- maximum number of functions for distribution of investments and additional relations.	[1]
nfacm	- maximum number of parameters for these functions	[1]
nentrm	- maximum number of entries in form of a time series per conversion technology	[13]
ngamm	- maximum number of lower or upper dynamic constraints for all technologies	[100]
nresm	- maximum number of resources with extraction	[16]
ngrm	- maximum number of grades per resource	[5]
nelrm	- maximum number of elasticity classes per grade	[1]
mlimr	- maximum number of limits on resources	[256]
mprr	- maximum number of costs for resources	[170]
ndeprm	- maximum number of depletion limits for resources	[50]
nimpm	- maximum number of fuels with imports	[16]
ncim	- maximum number of countries per import	[2]
nelim	- maximum number of elasticity classes per country	[1]
mlimi	- maximum number of limits on imports	[256]
mpri	- maximum number of prices for imports	[128]
ndepem	- maximum number of depletion limits for imports	[1]
nexpm	- maximum number of fuels with exports	[16]
ncem	- maximum number of countries per import	[1]
nelem	- maximum number of elasticity classes per country	[1]
mlime	- maximum number of limits on exports	[256]
mpre	- maximum number of prices for exports	[128]
ndepim	- maximum number of depletion limits for exports	[1]

If the switch is set to 1, the program writes, depending on the setting of the first switch, either

include 'com*name*'

in the case of U(NIX), or

*call com*name*

in the case of C(DC) instead of complete blocks of code, containing dimension and common statements. These statements are used to indicate that a file called 'com*name*' is stored outside the program and has to be included during compilation (in the case of UNIX) or to be included when the program is extracted from an UPDATE library in the case of CDC operating systems. In case of other operating systems the write statement can be adapted easily to the specific requirements. In any of these cases (switch set to zero or one) the program to be changed is read from unit 1 and the new code is written to unit 2.

Setting the switch to -4 results in creating the blocks of information on separate files having the required names and format. This is required when using the UNIX operating system.

If the switch is set to -5, the program creates one file containing all blocks of information according to the format specifications required by CDC UPDATE libraries (i.e.; each block is preceded by *cd com*name*).

5. The Implementation on the Computer

The following chapters give the information on how to implement and run the matrix generator (*MXG*) and the reports writer (*REPO*), and the two supplementary programs, *CHIN* and *CHDIM*.

The next section will describe the files needed (which can be received on a tape from IIASA) for compiling the programs. The second and third sections contain information on how to run the programs and adapt the dimensions.

5.1 Description of the Files

All file names will be written in capital letters, FORTRAN source codes end with a '.f', object (compiled) codes with '.obj', the other files are data and execution control files. None of the codes requires subroutines from a public library.

CHDIM
is used to produce the files containing the common blocks or to change the dimensions of the codes for *MESSAGE II*, e.g., if the current compilation does not fit any more due to an increased number of technologies. It has to be applied to *ROWS*, *RES*, *COLD*, *COLS*, *RHS* and *REPO*, because they all use the same common blocks.

The main program and subroutines of *CHDIM* are listed below by name followed by the names of the included common block and data files needed.

NAME		COMMON	DATA
chdim.f	(MAIN)	cdcomdim	cdcomdat
cdrdcp.f		cdcomdim	
cdrdm2.f		cdcomdim	
wrcapc.f		cdcomdim	
wrms2c.f		cdcomdim	
stopup.f			
wrincl.f			

CHIN
rewrites the input files of the matrix generator in order to allow the user to use a free format (see chapter 4.3.1), whereas the matrix generator needs the inputs as one entry per line and shifted to a certain column. Additionally CHIN can combine several files to one or select scenarios if several are defined in the input file.

Again the main program and subroutines needed are listed below.

NAME	
chin.f	(MAIN)
scena.f	
scens.f	
scan.f	

MXG,
the matrix generator consists of four programs producing different parts of the matrix:

ROWS
reads the general input and the technology definitions (UNIT 9), produces a technology related dump file (on UNIT 10, unformatted), one on the bounds (on UNIT 12, unformatted), one on time series (on UNIT 14, unformatted) and one for the additional relations (UNIT 17, unformatted, direct access, opened in the program), which can be read by the other codes using the same common blocks. Additionally it writes all row definitions except the ones on resources, imports and exports (MAT0, UNIT 8, formatted), and creates a file containing the actual dimensions of the current problem (UNIT 7, formatted, which is read by *RES* and supplemented with the resource-relevant dimensions) and one containing information for postprocessing (UNIT 3, formatted). UNIT 4 is used for unformatted internal I/O, UNITS 2 and 3 contain additional information for *REPO* (formatted). The names of the routines of *ROWS* are listed below, the main program is given in *italic* letters.

balr.f
capgr.f
capr.f
chkeda.f
chkin.f
cklper.f
costr.f
dataro.f
demr.f
disfnc.f
error.f
fcap.f
fillmp.f
getlr.f
iin.f
in.f
iout.f
itstec.f
lin.f
lkefin.f
look.f
look1.f
lout.f
mpdis.f
mpin.f
mpr.f
mulob.f
pkr.f
pres.f
presi.f
presv.f
r3.f
rchain.f
rdcap.f
rddemd.f
rddev.f
rddir.f
rdeden.f
rdela.f
rdems.f
rdenf.f
rdlr.f
rdmp.f
rdswi.f
rend.f
reptec.f
rin.f
rout.f
rows.f
rowsm.f
setcst.f
stocr.f
sump.f
tsmp.f
tstec.f
wrdim.f
wrdmp.f

COLD

reads the general, additional relation and time series dump files (UNITS 10, 14 and 17) and generates the columns on conversion technologies (nonlinear variables: MAT2, UNIT 16 and linear variables: MAT3, UNIT 8, both formatted). It uses UNIT 1 for unformatted internal I/O. The names of the routines of *COLD* are listed below.

capc.f
chkeda.f
cmpol.f
coldm.f
costc.f
cweigh.f
devc.f
disfun.f
fcap.f
getlr.f
getper.f
getvar.f
iin.f
init.f
lin.f
objc.f
pres.f
rddmp.f
rin.f
rlogis.f
setbet.f
setvar.f
stock.f
sump.f
weight.f

COLS

reads the same dumps as *COLD* and generates the columns on storage technologies (MAT4, UNIT 8, formatted). The names of the routines of *COLS* are listed below.

chkeda.f
cmpol.f
colsm.f
colsto.f
costc.f
cweigh.f
dec.f
disfun.f
fcap.f
fuelc.f
getper.f
getvar.f
iin.f
init.f
lin.f
objc.f
pcol.f
pres.f
rddmp.f
rin.f
rlogis.f
setbet.f
setvar.f
stocc.f
sump.f
weight.f

RHS

reads the same dumps as *COLD* and additionally the bounds dump (UNIT 12) and creates the general and technology related right hand sides (MAT6, UNIT 8) and bounds (MAT8, UNIT 7). It does also use UNIT 1 for unformatted internal I/O. The names of the routines of *RHS* are listed below.

accrhs.f caprhs.f chkeda.f conbnd.f demrhs.f
fcap.f getlr.f getper.f getvar.f iin.f
lin.f mpcrhs.f mprhs.f oprhs.f penrhs.f
pres.f rddmp.f rhs.f *rhsm.f* rin.f
setbet.f setvar.f stckr.f sump.f

RES

reads the general dump file (on UNIT 10), the dimension file created by *ROWS*. (UNIT 7) and the resource input and produces a file containing all dimensions (DIM, UNIT 8) and optionally the matrix parts relevant for resources, imports and exports (MAT1, MAT5, MAT7 and MAT9 on UNITS 1,2,3 and 4) and/or a resource dump files (UNITS 11 and 13). The names of the routines of *RES* are listed below.

costc.f cweigh.f datare.f error.f fcap.f
iins.f iout.f lin.f mkdump.f mpdef.f
mpres.f objc.f rddump.f *resm.f* ress.f
riein.f rieout.f rins.f rout.f setbet.f
ts.f weight.f

The following list gives all routines of *MXG* in alphabetic order with the common blocks they need.

NAME	COMMON BLOCK			
accrhs.f:	combl	comchar		
balr.f:	combl	comsto	comchar	
capc.f:	combl	comdev	comchar	compol
	commat	cominvs	comdim	comieq
capgr.f:	combl			
capr.f:	combl	comsto	comieq	comchar
caprhs.f:	combl	commpio	commprh	comchar
chkin.f:	combl	comdev	comchar	comend
cklper.f:	combl	comdev	comsto	
cmpol.f:	combl	comchar	cominvs	commpio
colh.f:	combl	comchar	commult	combl
	comchar	combl	comchar	cominvs
	cominvs			

```
conbnd.f: combl    comdev   comsto   comchar
costc.f:  combl    comchar
costr.f:  combl    comchar
datare.f: comres   comimp   comexp   comreda
          combone
dataro.f: combl    comdev   comsto   compol
          commat   comend   cominvs  comchar
          commult  comvar   comdim   comieq
          comdata  combone
dec.f:    combl
demr.f:   combl    comend   comieq   comchar
demrhs.f: combl    comchar
devc.f:   combl    comdev   commat   compol
          comchar  comdim   comieq
disfnc.f: cominvs
fcap.f:   combl    comdev   comsto
fillmp.f: combone  combl
fuelc.f:  combl    comend   comchar
getlr.f:  combl    comchar
getper.f: comkeep
getvar.f: comkeep
in.f:     combl    comdev   comsto   compol
          commat   comend   cominvs  comchar
          commult  comvar   comdim   comieq
          combone
init.f:   cominvs
itstec.f: comvar   combl
look.f:   combl
look1.f:  combl    comchar
mkdump.f:          comiore
mpcrhs.f: combl    commpio  commprh
mpdef.f:  combl
mpdis.f:  comchar  commpio  cominvs  combone
mpin.f:   commpio  combl    comchar  combone
mpr.f:    combl    comchar
mpres.f:  combl
mprhs.f:  combl    commpio  commprh
mulob.f:  combl    commult  comdim   combone
objc.f:   commult  combl    comchar
pcol.f:   combl    comchar
penrhs.f: combl    comchar
pkr.f:    combl    comchar
r3.f:     combl    comchar  commpio  combone
rchain.f: combl    comchar  comdim   comieq
          combone
```

rddev.f: comvar combl commat compol comend cominvs comchar combone

rddmp.f: comioda

rddump.f: comdim commult cominbl

rdeden.f: combl comchar comdim combone

rdela.f: combl comchar comdim combone

rdems.f: combone

rdenf.f: comvar combl comchar combone

rdlr.f: combl comchar comdim combone

rdmp.f: combl commpio comchar

rdswi.f: combl comchar cominvs comdim combone

ress.f: combl comchar comdim comres comimp comexp comried combone

rhs.f: combl comdev comsto commat compol comchar comieq comdim

riein.f: combl comchar combone

rieout.f: combl comchar

rows.f: combl comdev comsto compol commat comieq comchar comdim

setbet.f: combl commult

setcst.f: combl compol commat

setvar.f: combl comkeep

stckr.f: combl comchar

stocc.f: combl comsto comchar compol commat cominvs comdim comieq

stock.f: combl comchar

stocr.f: combl comchar

sump.f: combl

ts.f: combl

tsmp.f: combl

tstec.f: comvar combl

wrdim.f: combl comdev comsto compol commat comend cominvs comchar commult

wrdim.f: comvar comdim

wrdmp.f: comioda

REPO

reads all dump files produced by *ROWS* and *RES* (UNITS 2, 10, 11, 12, 13, 14, and 17), a switch on interactive use from the standard input (UNIT 5), the control input (UNIT 8), and writes the report (UNIT 1 or 6) and the table of contents (UNIT 3). The following two lists give the routines of *REPO* and the common blocks they need.

acccom.f acclim.f accprc.f alin.f bounds.f
ciin.f ciout.f clin.f clout.f cmpv.f
crin.f crout.f cvar.f deman.f dumpi.f
dumpr.f dynco.f exdyn.f genin.f getv.f
matorp.f ncheck.f nucfl.f pgcnt.f rdvar.f
repm.f repm.f rescom.f resour.f rimex.f
seelp.f setmpt.f setz.f shortd.f sort.f
text.f variab.f yrchk.f

NAME	COMMON BLOCK			
Acccom.f:	combl	comrepo		
Acclim.f:	combl	comieq	commat	commpio
	compol	comrepo		
Accprc.f:	combl	commat	commpio	compol
	comrepo			
Bounds.f:	combl	comchar	comdev	comrepo
	comsto			
Cmpv.f:	commpio	commpvr		
Cvar.f:	comvar	comvarc		
Deman.f:	combl	comrepo		
Dumpi.f:	comioda			
Dumpr.f:	comiore			
Dynco.f:	combl	comdev		
Genin.f:	combl	comdev	comrepo	comsto
Getv.f:	comvar	comvarc		
Matorp.f:	commpvr	combl	comchar	comdev
	comdim	comend	commat	commpio
	commult	compol	comrepo	
Nucfl.f:	combl	comrepo		
Rdvar.f:	comvar			
Repm.f:	combl	combl	comchar	comchar
	comdev	comdev	comdim	comdim
	commat	commat	compol	compol
	comrepo	comrepo	comsto	comsto
Resour.f:	combl	comdim	comexp	comimp
	comrepo	comres		
Rimex.f:	combl	comrepo		
Seelp.f:	comhash			
Setmpt.f:	combl	commpio	comrepo	
Setz.f:	comrepo			
Shortd.f:	comdev	comrepo		
Variab.f:	commpio	commpvr	combl	comdev
	commat	compol	comvar	comvarc

5.2 Running the Programs

In the following a call of a program will be indicated by writing the name of the object code (name call). The numbers of the UNITS used for the different files are indicated by 'no=file', e.g. '5=INPUT'. All programs contain at the beginning a program statement as used for CDC-FORTRAN as COMMENT cards. These program statements show a consistent set of file names.

The matrix generator of *MESSAGE II* requires at least two data files. The input of the sample model--*STIM*--is contained in 4 files: *STIM.i*, *STIM.d*, *STIM.t* and *STIM.r*, which contain general, demand, technology and resource related data.

```
CHIN.obj     5=input     2=STIM.d     3=STIM.i     4=STIM.r
             6=output    8=STIM.1     9=STIM.2     11=STIM.t
```

The switch determining the number of the scenario to be chosen from the input files has to be given on UNIT 5. In the case of *STIM* '00' is sufficient as switch, but also others are included (see Appendix 1). Another line with switches is needed thereafter. These can be used to change the default UNIT number. Use a blank line to keep the default values (see chapter 4.3.1).

The resulting file is directly fed into *ROWS*, an additional switch determining the kind of output and the name of the LP-solver to be used have to be given on UNIT five. If the switch is 0 all possible output is generated, if it is 1 only the dump files are generated (see also chapter 5.1).

```
ROWS.obj     5=input      4=intm       9=STIM.1     6=output
             8=MATO       7=DIMINT     2=TECID      3=SOLSIZ
             10=TECDMP    12=BNDDMP    14=VARDMP    17=RELDMP
```

UNIT 4 is used for unformatted internal I/O, error messages are written to UNIT 6. They can indicate a wrong use of identifiers, that the common blocks should be increased in size or a repetition of certain identifiers. TECDMP (unformatted) contains the dump file on general and technology data, BNDDMP (unformatted) the one on bounds of technologies, RELDMP (unformatted and direct access, the record length (i.e. the number of bytes per record, other machines like CDC use the number of words per record) is 4 times mprcl, which can be seen on the data file com*ioda* that is produced by *CHDIM*) the one on additional relations and VARDMP (unformatted) the one on time series for technology data. MATO is the first part of the matrix produced. DIMINT is the file containing the necessary dimensions for the common blocks, which is read by *RES*. TECHID contains written information on the conversion technologies and can be used again to write a report by REPO. The information on UNIT 3 is needed for post processing the results and is described in [3].

RES.obj should be run after *ROWS.obj* in order to destroy no information. This is only necessary if no file names are assigned to the UNITS. The package is in principle organized in a way, that this is not necessary as the file produced by one code being input to another one are always identified by the same UNIT number (with the exception of the matrix parts, that have to be assigned to a file).

RES.obj *5=input* *9=STIM.2* *6=output* *10=TECDMP*
1=MAT1 *2=MAT5* *3=MAT7* *4=MAT9*
7=DIMINT *8=MIIDIM* *11=RESDMP* *13=RESID*

Again the code of *RES.obj* reads a switch from UNIT 5 defining the kind and number of files generated. If it is 0, only the matrix parts are generated, if it is 1, only the dump file (RESDMP, unformatted) and the written information on the resources, imports and exports on RESID (unformatted) is written, if the switch is 2, all possible files are generated. The information produced using switch 1 is only necessary to produce a report containing information on resources, imports and exports. On UNIT 6 again error messages on wrong inputs or dimensioning will occur. On UNIT 10 the dump file produced by *ROWS* has to be given, the dimensions writen by *ROWS* have to be on DIMINT, while the complete ones are written to MIIDIM on UNIT 8. MAT1, MAT5, MAT7 and MAT9 are parts of the matrix, the numbering indicates the order in that they have to be catenated together.

The order in that the next programs are executed is optional. All of them just read dump files produced by *ROWS.obj* and produce matrix parts. UNIT 1 is used for unformatted internal I/O.

COLD.obj *10=TECDMP* *14=VARDMP* *17=RELDMP* *8=MAT3*
1=intm *16=MAT2*
COLS.obj *10=TECDMP* *14=VARDMP* *17=RELDMP* *8=MAT4*
1=intm
RHS.obj *10=TECDMP* *14=VARDMP* *17=RELDMP* *12=BNDDMP*
8=MAT6 *7=MAT8* *1=intm*

After running *CHIN.obj* and each of the 5 parts of the matrix generator the parts of the matrix have to be catenated together. On certain computers care has to be taken that one does not include empty matrix parts.

MATRIX = MAT0,MAT1,MAT2,MAT3,MAT4,MAT5,MAT6,MAT7,MAT8,MAT9

REPO.obj can be used now to produce a report on the input data and check them.

REPO.obj *5=input* *1=REPORT* *2=TECID* *3=CONTENTS*
4=intm1 *6=output* *8=CONTROL* *10=TECDMP*
11=RESDMP *12=BNDDMP* *13=RESID* *14=VARDMP*
15=intm2 *17=RELDMP*

It reads a switch from UNIT 5 and, depending on this, more control input either form UNIT 5 or 8. Additionally it uses all dump files and writes a report to UNIT 6 (interactive) or UNIT 1 (batch mode). In the latter case also a table of contents is produced on UNIT 3.

5.3 Recompiling the Programs

If during the setup of a new input file for *MESSAGE II* a need for larger dimensions of the common blocks of the codes of *MESSAGE II* is detected or if the new scenarios require significantly smaller common blocks all the codes should be recompiled after changing their dimension statements. These changes are done using *CHDIM.obj*, which reads the programs and rewrites them according to the new dimension specifications given. In order to produce new common blocks (files with names starting with com) only (switch -4 on UNIT 5), no names have to be given in the call. Then *CHDIM.obj* will open the files MIIDIM and CAPDIM in order to read the dimensions needed for *MXG* and *CAP*, and it will also open the files containing the common blocks on UNIT 2 by name and write them with the demanded dimensions. UNIT 5 is control input on the kind of action to be taken, UNIT 6 is used to ask the user for his wishes. The switches that can be written to UNIT 5 are:

0 => write common blocks into the FORTRAN codes

1 => write include into the FORTRAN codes. This option can be used for all machines, that give the opportunity to include files into a program during compilation at the place where they are called (for instance IBM gives this option, also UNIX, DEC or NORD operating systems). Usually the format of the include statements has to be changed. This has to be done in subroutine WRINCL, format statement 1001.

-1 => write include, stop checking after a 'c endu' and just copy the rest of the file (this option is implemented for CDC UPDATE libraries, that have the declaration of common blocks in the beginning)

-2 => do not check for common blocks (this option can be used to switch off checking of *CAP* common blocks, since this switch is given separately for the dimensions in MIIDIM and CAPDIM.

-4 => write common blocks without checking an input file. The common blocks are written to separate files having the according common block names.

-5 => write all common blocks without checking an input file to one file, each one preceded by '*cd comname'. This file can then be used to create a CDC UPDATE library.

If the common blocks are to be changed in the codes, the old code has to be given on UNIT 1, *CHDIM.obj* will write the new code to UNIT 2.

```
CHDIM.obj   5=cntrl     6=ask       1=oldcode    2=newcode
            3=CAPDIM    4=MIIDIM
```

5.4 Extra Characters Used by the Codes of MESSAGE II

The following section should be a help for users receiving a tape containing *MESSAGE II*, if the ASCII or EBCDIC standard at their computer is not exactly the same as at IIASA.

Extra characters used by *CHIN*:

The extra characters used for *CHIN* occur in the input files for the matrix generator (MXG) and in the control input on UNIT 5, if scenarios are to be switched by name.

'!+nn' identifies lines that should be included in scenario number nn,
'!-nn' identifies lines that should be excluded from scenario number nn,
'@nn' tells the code to continue reading the input on unit nn,
'@@' identifies the end of input,
'-' in one of the first 6 columns of a string results in an output with this '-' (dash) in column 7,
'#' in the first column identifies comment lines that are to be included in the report, and
';' identifies *END OF LINE*, thereafter comments can be written.

Extra characters used by *MXG*:

MXG does not use any special characters, that are not uniquely translated by all the versions of ASCII and EBCDIC. However, the input files often contain a 'non-printable character, '~', which is then translated into blanks by *REPO*. This character is in some mappings reported like a blank and thus translated as blank, resulting in an input error for *MXG*.

List of References:

[1] Murtagh B. A., Saunders M. A.: MINOS - A Large Scale Nonlinear Programming System (User's Guide). Technical Report SOL 77-9, Systems Optimization Laboratory, Stanford University, Stanford, USA, February 1977.

[2] M. Agnew, L. Schrattenholzer, A. Voss, User's Guide for the MESSAGE Computer Program, IIASA, RM-78-21

[3] M. Strubegger, Users Guide for the Post Processor of *MESSAGE II*, IIASA 1984, forthcoming

[4] B. Lapillonne, MEDEE-2: A Model for Long-Term Energy Demand Evaluation, IIASA, RR-78-17

[5] E. Nurminski, Convergence and Numerical Experiments With a Decomposition Algorithm, IIASA, WP-82-8

[6] R. Codoni, B. Fritsch (Eds.), Capital Requirements of Alternative Energy Strategies, ETH Zürich, Institut für Wirtschaftsforschung, 1980.

WORKING PAPER

USER'S GUIDE FOR THE MATRIX GENERATOR OF MESSAGE II
PART II:
APPENDICES

S. Messner

September 1984
WP-84-71b

IIASA
International Institute
for Applied Systems Analysis

USER'S GUIDE FOR THE MATRIX GENERATOR OF MESSAGE II

PART II:
APPENDICES

S. Messner

September 1984
WP-84-71b

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

APPENDIX 1

Input files for *STIM*
(Small Test and Implementation Model)

This appendix lists the input files for *STIM* in the following order:

- *STIM.i*, the general definitions,
- *STIM.d*, the demand data,
- *STIM.t*, the technology data, and
- *STIM.r*, the resource data.

They are input to *CHIN*, which converts them to the format needed by the matrix generator (*MXG*, see chapter 5.2). Appendix 2 gives an example of an input conversion as *CHIN* performs it, and Appendix 3 lists the report for *STIM*, which is produced by *REPO*.

STIM.i

```
;DEFINITION OF SCENARIOS:
!!lowSO2:+01
!!lowNOx:+02
!!lowemission:+01/02
;
;Built-in switches:
;        01: costs on the emissions of SO2 from central systems
;        02: costs on the emissions of NOx from central systems
;
;
TEST.inp     ;name of problem
;=======
#
#All costs are given in $'80 per kW or per kWyr,
#capacities in MW and energy quantities in MWyr
#if not stated differently.
#
;          *****************
;          * INPUT BLOCK 1 *
;          *****************
;
;
;         GENERAL DEFINITIONS
;==============================================================================
;
1                             ;ntu (number of first period in matrix)
4                             ;ntrun (number of last period in matrix)
0                             ;0 => MIP not possible on this machine
                              ;1 => MIP possible on this machine
1 e12.6)                      ;format for matrix
0                             ;no different identifiers for energy form levels
0                             ;activity bounds are given relative to main output
0                             ;ncap (switch for levellizing costs)
1                             ;check on identifiers of user defined constraints
                              ;during technology input is set on
0 0 0 0                       ;dummy switches
;
1975                          ;iyr (refenrence year (the existing structure is
                              ;    given for that year))
4                             ;nto (number of periods in input file)
5                             ;lpo (length of periods before reference year)
5 , 10 ,                      ;lp (period lengths)
;
;                             period     1      2      3        4
;                             years   80-84, 85-89, 90-99, 2000-2009
;
```

```
0 4.                          ;discount rate
2                             ;imat (annual accounting switched on)
*                             ;no bounds on accounting rows
e                             ;1st distribution function is equal distribution
*                             ;no more distribution functions for investments
*                             ;no energy density areas
;-----------------------------------------------------------------------------
;input for load regions
;=============================================================================
#The year is ordered into 4 load regions according to the
#average load per month. The correct peak installations of
#powerplants and heating plants is assured by user defined
#relations (elcp and dhcp).
#
l-load
1                             ;no of times it occurs per year
4                             ;no of regions it is divided into
*                             ;end peaks
.084 .333 .333 .25            ;lengths of load regions (fraction of the year)
0                             ;no extra baseload modelled
;-----------------------------------------------------------------------------
;
;
;INPUT OF ENERGY CHAIN AND ENERGY FORMS ON IT
;
;=============================================================================
u                             ;useful energy level
;-----------------------------------------------------------------------------
e-specific~electricity
#Non-thermal requirements of electricity.
g-gasoline
#Specific uses of gasoline for transportation.
h-thermal~uses
#Useful energy demand for thermal uses of energy.
*                             ;no nuclear fuel on this level
*                             ;end level u
;
;
;=============================================================================
f                             ;final energy level - transportation level
;-----------------------------------------------------------------------------
e-electricity
#Final electricity.
d-district~heat
#Final district heat.
*                             ;end of energy forms with load regions
```

```
l-light~oil
#Light oil products for thermal uses.
g-gasoline
#Gasoline for utilization as motor fuel.
*                             ;no nuclear fuel on this level
*                             ;end level f
;
;
;===============================================================================
x                             ;secondary energy level - central conversion level
;-------------------------------------------------------------------------------
e-electricity
#Secondary electricity.
d-district~heat
#Secondary district heat.
*                             ;end of energy forms with load regions
*                             ;no nuclear fuel on this level
*                             ;end level x
;
;===============================================================================
a                             ;primary energy level - extraction level
;-------------------------------------------------------------------------------
*                             ;no energy forms with load regions
l-light~oil
#Light oil products for thermal uses.
g-gasoline
#Gasoline for utilization as motor fuel.
f-fueloil
#Heavy refinery products that can either be used
#in powerplants or heating plants.
c-hard~coal
x-so2-dummy
*                             ;no nuclear fuel on this level
*                             ;end level a
;
;
;===============================================================================
r-fx                          ;resource/import level (constraint fixed to equal)
;-------------------------------------------------------------------------------
*                             ;no energy forms with load regions
o-crude~oil
c-hard~coal
h-hydrocap
f-fueloil
#Heavy refinery products that oan be cracked again
#or transferred to the level of use.
*                             ;no nuclear fuel on this level
*                             ;end level r
```

```
;
;
*                             ;end levels
;===============================================================================
;
;
;INPUT OF FUELS WITH STOCK-PILING
;
h-hydrocapacity
0.                            ;stock in reference year
#Interlink for the build-up of new hydropowerplants (Unit: MW).
*                             ;end fuels with stock-pile
;===============================================================================
;
;          *****************
;          * INPUT BLOCK 2 *
;          *****************
;
;         DEMAND RELATED DATA
@02                           ;read demands and related information from unit 2
;          *****************
;          * INPUT BLOCK 3 *
;          *****************
;
;  DEFINITION OF USER DEFINED RELATIONS

                              ;user defined relations of type 1:
;===============================================================================
elcp:1 -1 0 0.                ;name noprice uplim
#Constraint to force an overcapacity of 30%
#for electricity generation, load region 1.
;
elcp:2 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 2.
;
elcp:3 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 3.
;
elcp:4 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 4.
;
dhcp:1 -1 0 0.                ;name noprice uplim
#Constraint to force an overcapacity of 20%
#for district heat production, load region 1.
;
```

```
dhcp:2 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 2.
;
dhcp:3 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 3.
;
dhcp:4 -1 0 0.                ;name noprice uplim
#Same constraint to for load region 4.
;
pkel -1 0 0.
#Constraint to force a minimum electricity production
#from peak powerplants of 3%.
*                             ;end of user defined relations of type 1
;------------------------------------------------------------------------
                              ;user defined relations of type 2:
;========================================================================
dhrl:+ -1 0 0.                         ;name noprice uplim
#The share of district heat shall not decline
#in relation to the demand.
;
!-01 SO2-emissions -1 -1               ;name noprice nolim
!+01 SO2-emissions 0 3.2 -1            ;name price nolim
#Counter on the emissions of SO2 from central conversion plants.
;
!-02 NOx-emissions -1 -1               ;name noprice nolim
!+02 NOx-emissions 1 0. 1. 2. 2.5 -1   ;name price nolim
#Counter on the emissions of NOx from central conversion plants.
;
prel-lo-peak~power -1 0 0.              ;name:limtype noprice lolim
#10% of installed capacity has to be peak powerplants.
;
cgup-max~cogen -1 0 0.                  ;name noprice uplim
#Maximum operation time of cogeneration per year: 6000h
;
cglo-lo-min~cogen -1 0 0.               ;name noprice lolim
#Minimum operation time of cogeneration per year: 4000h
;
*                             ;end of user defined relations of type 2
;------------------------------------------------------------------------
;
;
```

```
;          *****************
;          * INPUT BLOCK 4 *
;          *****************
;
;  DEFINITION OF OBJECTIVE FUNCTION
;==============================================================================
;
*                             ;no change in objective function
;------------------------------------------------------------------------------
;
;          *****************
;          * INPUT BLOCK 5 *
;          *****************
;
;                        TECHNOLOGY INPUT
;==============================================================================
;
@11                            ;continue reading from unit 11
*                              ;end technology input
;------------------------------------------------------------------------------
;
@@                            ;end of file (switch to resource file)
```

STIM.d

```
;DEMANDS (annual figures in MWyr/yr)
;===============================================================================
e-electricity                                     ;
   344.000    360.000    375.000    390.000
h-heat                                            ;
  3200.000   3150.000   3100.000
g-gasoline                                        ;
   892.000    940.000    990.000   1060.000
*                                                 ;end of demands
;-------------------------------------------------------------------------------
;
;
;DEMAND DISTRIBUTION TO LOAD REGIONS
;===============================================================================
e-electricity
.107  .394 .301 .2                                ;
h-heat
.152  .477 .299 .096                              ;
*                                                 ;end of demand distribution to load regions
;if the entries do not sum up to one the code normalizes them
;all energy forms on the demand level for that no distribution given
;are distributed equally
;-------------------------------------------------------------------------------
;
;
;DEMAND DISTRIBUTION TO ENERGY DENSITIY AREAS
;===============================================================================
                                                  ;since no energy density areas are defined,
                                                  ;no switch is read here
;-------------------------------------------------------------------------------
;
;
;DEMAND ELASTICITIES
;===============================================================================
*                                                 ;end of demand elasticities
;-------------------------------------------------------------------------------
;
@03                                               ;return to unit 3
```

STIM.t

```
;
;*******************
;* end use devices *
;*******************
;
;
;=========================================================================
;
c                ;its=c    =>conversion technology with fix entries over time
e-elec-appliance ;ida-description (identifier of technology)
u                ;idx      (output level)
*                ;idur     (in all energy density areas)
1980             ;iref     (reference year of technology)
0                ;lref     (last year in that thechnology can be built, =0 => no
                 :          last year)
e                ;id       (identifer of input; no level is given =>
                 :          search on levels F,X,A, and R -- in this order)
1.               ;amount-in
*                ;id=*     => no more inputs for this technology
*                ;id=*     => no inventories for this technology
e                ;id       (identifier of output; has to lie on level U)
1.               ;amount-out
                 ;the efficiency of the technology is amount-out/amount-in
*                ;id=*     => no more outputs for this technology
*                ;id=*     => no retirements for this technology
*                ;id=*     => no entry in a user defined relation of type one
                 :            related to operation
*                ;id=*     => no entry in a user defined relation of type one
                 :            related to construction or capacity
*                ;id=*     => no entry in a user defined relation of type two
                 :            related to operation
*                ;id=*     => no entry in a user defined relation of type two
                 :            related to construction or capacity
0                ;lag      (lag-time between inputs and outputs)
0                ;isw      (production pattern over load regions not changed)
0                ;isw      (investments are not distributed)
0                ;cmin     (no integer variable)
0                ;dummy
0                ;dummy
1.               ;pf       (plant factor)
0.               ;pl       (plant life)
-1.              ;capc     (investment costs, <0 => technology has no capacity
                 :          variables)
0.               ;fix      (fix operation and maintainance costs (linked to
                 :          the capacity variables)
0.               ;curc     (variable operation and maintainance costs (linked to
                 :          the production variables)
```

```
*                ;id=*    => no activity bounds
*                ;id=*    => no capacity bounds
*                ;id=*    => no dynamic constraints on operation
*                ;id=*    => no dynamic constraints on construction
0                ;ivar    (no technologies are linked to this one)
;The following 3 lines are comment for the report writer (REPO):
#Specific use of electricity.
#No costs are assumed due to variety of uses
#and lack of substitution possibilities.
;
c g-gasoline-use   u * 1980 0            ;dev id olev idur iref
g 1. * * g 1. * *                        ;idin am by-in inv idout am by-out ret
* * * *                                  ;rla rlc r2a r2c
0 0 0 0 0 0                              ;lag, chlr, distr, int - -
1. 0. -1. 0. 0.                          ;pf pl cap fix cur
* * * * 0                                ;actbnd capbnd mpact mpcap ivar
#Specific use of gasoline for transport.
;
;-------------------------------------------------------------------------------
;
;%%%%%%%%%%%%%%%%%%%
;% heating systems %
;%%%%%%%%%%%%%%%%%%%
;
;===============================================================================
;
v                ;its=v   =>conversion technology can have variable entries
l-loil~heating   ;ida-description (identifier of technology
u                ;idx      (output level)
*                ;idur     (exists in all energy density areas)
1975             ;iref     (first year)
0                ;lref     (no explicit last year)
1                ;id       (identifier of main input; this is searched on
                 ;          levels f,x,a,r--in this order)
1.               ;amount-in
*                ;id=*    => no more inputs for this technology
*                ;id=*    => no inventory
h                ;id       (identifier of main input--has to lie on level u)
1                ;isw      (switch to read time series = 1 => read single
                 ;          value per period)
.664 .69 .73 .75 ;amount-out
*                ;id=*    => no more outputs for this technology
*                ;id=*    => no last core
*                ;id=*    => no entry in a user defined relation of type one
                 ;           related to operation
*                ;id=*    => no entry in a user defined relation of type one
                 ;           related to construction or capacity
```

```
*                ;id=*    => no entry in a user defined relation of type two
                 ;           related to operation
*                ;id=*    => no entry in a user defined relation of type two
                 ;           related to construction or capacity
-1               ;isw=0   => read no lag-time
0                ;isw     (production pattern over load regions not changed)
0                ;isw     (investments are not distributed)
0                ;cmin    (no integer variable)
0                ;dummy
0                ;dummy
0 .23            ;pf      (plant factor)
0 15.            ;pl      (plant life in years)
0 124.           ;capc    (investment cost per unit of main output)
0 4.             ;fix     (fix operation and maintainance costs)
0 0.             ;curc    (variable operation and maintainance costs)
2600.            ;actin   (activity of the technology in the baseyear)
2                ;isw     (switch to read capacity in the baseyear,
                 ;         =2 => read total capacity and growth rate
11500. .9        ;capacity in baseyear and historic growth rate
*                ;id=*     => no activity bounds
up               ;id=up    => upper capacity bounds
1                ;isw=1    => read single values per period
300.             ;bnd      bound in the first period
1.e7 ,,          ;bnd      no bounds in the following periods
*                ;id=*     => no more capacity bounds
*                ;id=*     => no dynamic constraints on operation
up               ;id=up    => upper dynamic constraints on construction
0 40.            ;isw,g    (start-up parameter as time-series)
0 1.05           ;isw,gam  (growth rate as time-series)
*                ;id=*     => no more dynamic constraints on construction
0                ;ivar     (no technologies are linked to this one)
;The following line is the comment for the report writer (REPO)*
#Light oil heating system (75kW).
;
v-fx e-elec-heating        u * 1975 0      ;dev id olev idur iref
e 1. * * h 0 .96 * *                       ;idin am by-in inv idout am by-out ret
* * * *                                    ;r1a r1c r2a r2c
-1 0 0 0 0 0                               ;lag, chlr, distr, int - -
0 .23 0 15. 0 116. 0 3.4 0 0.              ;pf pl cap fix cur
58. 2 260. 1.1                             ;actin, capin
up 1 60. 1.e7 ,, * *                       ;actbnd capbnd
lo 0 .5 0 .9 * up 0 40. 0 1.1 * 0          ;mpact mpcap ivar
#Waterbased electric heating system (75kW).
;
```

```
v-fx d-district-heat        u * 1975 0      ;dev id olev idur iref
d 1. * * h 1 .90 .92 .94 .95 * *            ;idin am by-in inv idout am by-out ret
* * dhr1 0 1. * *                           ;rla rlc r2a r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .23 0 15. 0 90. 0 3.2  0 0.               ;pf pl cap fix cur
890. 2 4050. 1.1                            ;actin, capin
up 1 -.5 ,,, lo 0 -.3 *                     ;activity bounds (0 => share of
                                            ;demand (i.e. <= 50%, >= 30%)
* * * 0                                     ;capbnd mpact mpcap ivar
#District heating system (75kW).
;
v h-elec~hpump             u * 1985 0       ;dev id olev idur iref
e 1. 1 0 .293 *                             ;idin am by-input light oil,
                                            ;amount relative to
                                            ;main output
* h 0 2.72  * *                             ;idin am by-in inv idout am by-out ret
* * * *                                     ;rla rlc r2a r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .23 0 15. 0 355. 0 9. 0 0.                ;pf pl cap fix cur
up 0 -.15  * * *                            ;actbnd capbnd mpact
up 1 20. ,, 0. 0 1.1 * 0                    ;mpcap: start-up as time series:
                                            ;technology has to be built
                                            ;before 2000, afterwards
                                            ;initial size zero: ivar
#Ground heatpump with oil back-up.
#The oil covers 30% of the production in
#the peak demand area.
;
;-------------------------------------------------------------------------------
;
;
;***********************
;* distribution systems *
;***********************
;
;===============================================================================
;
v e-elec~distribution   f * 1975 0          ;dev id olev idur iref
e 1. * * e 0 .9 * *                         ;idin am by-in inv idout am by-out ret
* * * *                                     ;rla rlc r2a r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .73 0 30. 0 350. 0 7. 0 0.                ;pf pl cap fix cur
340. 2 700. 1.08                            ;actin capin
* * * up 0 0. 0 1.05 * 0                    ;actbnd capbnd mpact mpcap ivar
#Distribution of electricity.
;
```

```
v d-DH~distribution     f * 1975 0        ;dev id olev idur iref
d 1. * * d 0 .91 * *                      ;idin am by-in inv idout am by-out ret
* * * *                                   ;rla rlc r2a r2c
-1 1 0 .152 0 .477 0 .299 0 .096          ;lag, chlr: fraction of production
                                          ;per load region (read as time series)
0 ,,,                                     ;distr, int - -
0 .75 0 30. 0 100. 0 2. 0 0.              ;pf pl cap fix cur
900. 2 3000. 1.06                         ;actin capin
* * * * 0                                 ;actbnd capbnd mpact mpcap ivar
#Distribution of district heat.
;
v l-loil~distribution   f * 1980 0        ;dev id olev idur iref
l 1. * * l 0 l.    * *                    ;idin am by-in inv idout am by-out ret
* * * *                                   ;rla rlc r2a r2c
-1 0 0 0 0 0                              ;lag, chlr, distr, int - -
0 1. 0 30. 0 -1. 0 0. 0 14.               ;pf pl cap fix cur
* * * * 0                                 ;actbnd capbnd mpact mpcap ivar
#Distribution of light refinery products.
;
v g-mfuel~distrib.      f * 1980 0        ;dev id olev idur iref
g 1. * * g 0 1.    * *                    ;idin am by-in inv idout am by-out ret
* * * *                                   ;rla rlc r2a r2c
-1 0 0 0 0 0                              ;lag, chlr, distr, int - -
0 1. 0 30. 0 -1. 0 0. 0 14.               ;pf pl cap fix cur
* * * * 0                                 ;actbnd capbnd mpact mpcap ivar
#Distribution of gasoline.
;
;------------------------------------------------------------------------------
;
;
;**********************
;* central conversion *
;**********************
;
;
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;% district heat production %
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;
;==============================================================================
;
```

```
v f-foil~heatpl          x * 1975 0         ;dev id olev idur iref
f 1. * * d 1 .85 .87 .90 .92 * *            ;idin am by-in inv idout am by-out ret
dhcp 0 1. *                                 ;rla: coefficient 1 in relation to
                                            ;main output
dhcp:t 0 -.83 *                             ;rlc: coefficient -.83 per total
                                            ;installed capacity
SO2 1 17.8 17.4 16.8 16.4                   ;
NOx 1  5.6  5.4  6.3  5.1 * *               ;r2a r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .95 0 25. 0 116. 0 5.8 0 7.2              ;pf pl cap fix cur
900. 2 3500. 1.005                          ;actin capin
* * * * 0                                   ;actbnd capbnd mpact mpcap ivar
#Fueloil heatplant (25MW).
;
;-------------------------------------------------------------------------------
;
;
;****************
;* cogeneration *
;****************
;
;===============================================================================
;
v c-coal~co-gen          x * 1960 0         ;dev id olev idur iref
c 1. * * d 1 .56 .55 ,   ,                  ;
         e 1 .19 .20 .23 .27 * *            ;idin am by-in inv idout am by-out ret
dhcp   0 1.   elcp   1  .34  .36  .42  .49     ;rla
pkel   0 .83                            *      ;rla
dhcp:t 0 -.83 elcp:t 1 -.26 -.28 -.32 -.38 *   ;rlc
SO2 1 27. 27.5 ,, NOx 1 15.75 16.04 ,,      ;r2a
cgup   0 1.   cglo   0 1.   *               ;r2a
cgup:t 0 -.68 cglo:t 0 -.46 prel:t 0 -.05 *;r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .88 0 25. 0 540. 0 16. 0 5.               ;pf pl cap fix cur
40. 1 88. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ;
* fx 1 0. 1.e7 ,, * * * 0                   ;actbnd capbnd mpact mpcap ivar
#Coal co-generation plant.
#(50MW(e)/100MW(th))
;
;-------------------------------------------------------------------------------
;
;
```

```
;***************
;* powerplants *
;***************
;
;=================================================================================
;
v f-foil~ppl              x * 1968 0        ;dev id olev iref
f 1. * * e 0 .372 * *                       ;idin am by-in mat idout am by-out
elcp 0 1. pkel 0 1. * elcp:t 0 -.77 *       ;rla rlc
SO2 0 16.935 NOx 0 12.715 *                 ;r2a
prel:t 0 -.1              *                 ;r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 .9  0 25. 0 434. 0 5.2 0 12.6             ;pf pl cap fix cur
0. 2 150. 1.1                               ;historic capacities
* * * up 0 .2 0 1.05 *  0                   ;mpact mpcap
#Heavy destillate oil base load steam power plant.
;
v j-diesel~ppl             x * 1975 0       ;dev id olev iref
1 1. * * e 0 .28 * *                        ;idin am by-in mat idout am by-out
elcp 0 1. pkel 0 -32.33 * elcp:t 0 -.77 *   ;rla rlc
SO2 0 15.75 NOx 0 16.893 * prel:t 0 1. *    ;r2a r2c
-1 0 0 0 0 0                                ;lag, chlr, distr, int - -
0 1.  0 25. 0 260. 0  5.2 0 28.             ;pf pl cap fix cur
0. 2 250. .9                                ;historic capacities
* * * * 0                                   ;mpact mpcap
#Light destillate oil peak power plant.
;
v h-hydropower             x * 1975 0       ;dev id olev iref
* h-q 0 1. * e 0 1. * h-q 0 1. *            ;idin am by-in mat idout am by-out
elcp 0 1. pkel 0 1. * elcp:t 0 -.77 *       ;rla rlc
* prel:t 0 -.1 *                            ;r2a r2c
-1 1 0 .032 0 .129 0 .129 0 .16             ;lag, chlr
0 0 0 0                                     ;distr, int - -
0 .63 0 60. 0 425. 0 18. 0 0.               ;pf pl cap fix cur
240. 2 500. 1.1                             ;actin capin
* up 1 15. 1.e7 ,, *                        ;actbnd capbnd
lo 0 0. 0 1. up 0 0. 0 1.05 * * 0           ;mpact mpcap
#Run of river hydropower plant,
#average plant factor .45
;
c hydro-link        q * 1980 0              ;dev id olev idur iref
h-r 1. * * h 1. * *                         ;idin am by-in inv idout am by-out ret
* * * *                                     ;rla rlc r2a r2c
0 0 0 0 0 0                                 ;lag, chlr, distr, int - -
1. 0. -1. 0. 0.                             ;pf pl cap fix cur
* * * * 0                                   ;actbnd capbnd mpact mpcap ivar
;
;---------------------------------------------------------------------------------
```

```
;
;
;****************************
;* other central conversion *
;****************************
;
;================================================================================
;
c n-refinery    a * 1975 0                 ;dev id olev idur iref
o 1. * * g .15 1 .3 f-r .5 * *             ;idin am by-in inv idout am by-out
* * SO2 5. NOx 1.6 * *                     ;rla rlc r2a r2c
0 0 0 0 0 0                                ;lag, chlr, distr, int - -
.8 25. 24. 0. .75                          ;pf pl cap fix cur
804. 2 1200. .98                           ;actin capin
* * * * 0                                  ;actbnd capbnd mpact mpcap ivar
;
c c-catcracker    a * 1980 0               ;dev id olev idur iref
f-r 1. * * g .095 1 .57 f .285 * *         ;idin am by-in inv idout am by-out
* * SO2 8. NOx 2.5 * *                     ;rla rlc r2a r2c
0 0 0 0 0 0                                ;lag, chlr, distr, int - -
.8 25. 220. 0. 6.5                         ;pf pl cap fix cur
* * * * 1                                  ;actbnd capbnd mpact mpcap ivar
;
c 2-catcracker    a * 1980 0  3.           ;dev id olev idur iref
f-r 1. * g .285 1 .38 f .285 *             ;idin am by-in idout am by-out
* SO2 3.65 NOx .83 *                       ;rla r2a
0 0 0 0                                    ;lag, chlr, distr, int - -
6.5                                        ;pf pl cap fix cur
* *                                        ;actbnd capbnd mpact mpcap ivar
;
c 1-link          a * 1980 0               ;dev id olev idur iref
f-r 1. * * f 1. * *                        ;idin am by-in inv idout am by-out
* * * *                                    ;rla rlc r2a r2c
0 0 0 0 0 0                                ;lag, chlr, distr, int - -
1. 1. -1. 0. 0.                            ;pf pl cap fix cur
* * * * 0                                  ;actbnd capbnd mpact mpcap ivar
;
;--------------------------------------------------------------------------------
;
;
```

```
;**************
;* extraction *
;**************
;
;==============================================================================
;
v c-coal~extraction       a * 1975 0       ;dev id olev iref
c 1. * * c 0 1. * *                         ;idin am by-in mat idout am by-out
* * * *                                     ;rla rlc r2a r2c
-1 0 5 1 * * 0 0 0                          ;lag, chlr, distr, int - -
0 1. 0 20. 3 200. 1. 1.02 1.0073            ;pf pl cap
           3 10.  1. 1.02 1.0073            ;fix
           3 5.   1. 1.02 1.0073            ;cur
60. 2 60. .8                                ;actin capin
* * * * 0                                   ;mpact mpcap
;
@03                                         ;return to unit 3
```

STIM.r

```
;*****************
;* INPUT BLOCK 6 *
;*****************
;                 GLOBAL COST MULTIPLIERS
;===============================================================================
;
1 0 0 0 0 0 0 0 0 0 ; dummy switches
;
;
-1                    ; aresp  [no multiplier for all domestic resource costs]
3 180. 1. 1.02 1.0073 ; aimpp  [multiplier for all import prices]
-1                    ; aexpp  [no multiplier for all export costs]
;
;-------------------------------------------------------------------------------
;
;                 RESOURCE/IMPORT/EXPORT INPUT
;===============================================================================
;
;                 CRUDE OIL IMPORTS
;-------------------------------------------------------------------------------
o                ;id      identifier of energy form: crude oil
r                ;lev     identifier of level: resource level
i                ;ideq    identifier for kind of equation: import (r/i/e)
-1               ;spres   no cost multiplier for import of this fuel
-2               ;aresl   no limit on total availability per year
a                ;id      identifier of country imports come from
-1.              ;tres    no limit on total availability from this country
                 ;        over the whole horizon
-2               ;aresgl  no limit on annual availability from this country
1                ;idela   identifier of 1st elasticity class
0 1.             ;aprel   price of crude oil from country a, elasticity class 1
                 ;        (relative to global cost multiplier)
*                ;areseu  no annual upper limit on this elasticity class
*                ;idela   no more elasticity classes
*                ;dyncon  no dynamic constraint per country
0.               ;bsyrv   imports in baseyear from country a
-2               ;depl    no depletion constraint
*                ;id      no more countries
*                ;dyncon  no dynamic constraint for total import
#Imported crude oil.
*                ;id      no other kinds of equations => end of this fuel
;
```

```
;                    COAL RESOURCES
;-------------------------------------------------------------------------------
;
c r r               ; id lev ideq
-1 -2               ; spres aresl
a 10000.            ; id tres
-2 1 0 0. * *       ; aresgl nela aprel areseu end ela
up 0. 1.1 * 70.     ; dyncon bsyrv
1 .5 .41 .65 ,      ; depletion
b 20000.            ; id tres
-2 1 0 20. * *      ; aresgl nela aprel areseu end ela
* 0. -2             ; no dyncon, bsyrv depletion
*                   ; id  no more grades
up 0 1.12 *         ; dyncon all grades
                    ;
                    ; comment for REPORT:
#Costs of coal extraction is included in the according
#technology. Cost of second grade represents additional cost.
*                   ; id  no other kinds of equations!
;                    GASOLINE IMPORTS
;-------------------------------------------------------------------------------
g a i               ; id lev ideq
-1 -2               ; spres aresl
a -1.               ; id tres
-2 1 0 1.46         ; aresgl nela aprel
up 0 100. * *       ; areseu end ela
* 0. -2             ; no dyncon, bsyrv, depletion
*                   ; id  no more import countries
up 10. 1.1          ; upper dyncon
lo 10. .9 *         ; lower dyncon
#Price is set to 1.46 times the price of crude oil.
* ; id  no other kinds of equations!
;
;                    LIGHT LIQUIDS IMPORTS
;-------------------------------------------------------------------------------
l a i               ; id lev ideq
-1 0 650.           ; spres aresl
a -1.               ; id tres
-2 1 0 1.183 * *    ; aresgl nela aprel areseu end ela
lo 0. .9 *          ; lower dyncon
500. -2             ; bsyrv no depletion
* *                 ; id  no more import countries!, no dyncon
#Price is set to 1.183 times the price of crude oil.
*                   ; id  no other kinds of equations!
;
```

```
;                  HYDROPOWER POTENTIAL
;-----------------------------------------------------------------------------
h r r              ; id lev ideq
-1 -2              ; spres aresl
a -1.              ; id tres
-2 1 0 1000. * *   ; aresgl nela aprel areseu end ela
* 0. -2            ; no dyncon, resex
* *                ; id  no more import countries!, no dyncon
#Capacity for building hydropower as resource.
#unit: MW
*                  ; id  no other kinds of equations!
;
* ; id  no more fuel identifiers!
;
;-----------------------------------------------------------------------------
;
@@ ; end of information
```

APPENDIX 2

Example Input and Output Files for *CHIN*

The following example shows the file resulting from two input files. The main input file was read from UNIT 3, the additional one from UNIT 2. The output file was written to UNIT 8. Unit 4 does not contain any data. Additionally the example shows how to process two scenarios. The sort input example in chapter 4.2.1 shows th input formats for the first 6 lines of the file on UNIT 3.

Standard Input File (on UNIT 5)

```
01    ; select variables belonging to scenario 1
      ; blank line (i.e., change no default UNITS)
```

Main Input File (on UNIT 3)

```
#Begin of Information
#--------------------
;
!+01#The following entry belongs to scenario 01:
!+01 1 3.5             ; integer (switch), real value (for scenario 1)
!-01 0                 ; integer value (for all scenarios except 1)
5                      ; integer value
!+01 1. 3.10476e3 0.43 ; real values belonging to scenario 1
!+02 2. 6.20952e3 0.86 ; real values belonging to scenario 2
a-string1              ; identifier with explanatory string
25. ,,                 ; repetition of variables
@02                    ; continue to read from file on UNIT 2
67 b-string2 100 , 23.4; mixing various string types
;
#End of Information
#------------------
@@                     ; end of input
```

Additional Input File (on UNIT 2)

```
#begin of information read from file on UNIT 2
!+01#      data for scenario 1 chosen
!+01  1. 2. 3.
!+02#      data for scenario 2 chosen
!+02 11. 22. 33.
#end of information read from file on UNIT 2
@03 ; return to file on UNIT 3
```

Output File (on UNIT 8)

```
#           Begin of Information
#           --------------------
#           The following entry belongs to scenario 01:
     1
3.5
     5
1.
3.10476e3
0.43
     a-string1
25.
25.
25.
#           begin of information read from file on UNIT 2
#                data for scenario 1 chosen
1.
2.
3.
#           end of information read from file on UNIT 2
    67
     b-string2
   100
   100
23.4
#           End of Information
#           ------------------

 numerical selector chosen was 01
```

APPENDIX 3

Report of the input data for *STIM*
(Small Test and Implementation Model)

The report writer (*REPO*) as described in chapter 4.4 produces a control output of the data in the input files. The following listing shows the report for *STIM*, which is defined by the input files listed in Appendix 1. Before the report the corresponding table of content is listed. The technology data are preceeded by some additional text, which was included in the report by means of switch 4 as described in table 2, chapter 4.4.

TABLE OF CONTENTS

```
REPORT OF INPUTS TO MESSAGE II RUN FOR: TEST.inp
****************************************************


Produced at  7. 4, on 1984- 7-28


     All costs are given in $'80 per kW or per kWyr,
     capacities in MW and energy quantities in MWyr
     if not stated differently.


GENERAL INPUTS
==============


Labels of time periods:
-----------------------

  1980  1985  1990  2000

  Length of historic periods:    5 years


Definition of load regions
--------------------------

 The year is ordered into 4 load regions according to the
 average load per month. The correct peak installations of
 powerplants and heating plants is assured by user defined
 relations (elcp and dhcp).


Identifiers            nr of     nr per
                     sub peaks  time unit
1-load                   4          1

Length (fraction per year) :

  0.084  0.333  0.333  0.250

Discount rate
-------------

constant   4.0 %
```

DEMANDS

Annual demand figures:

years	e-specific~electr	g-gasoline	h-thermal~uses
1980	344.00	892.00	3200.00
1985	360.00	940.00	3150.00
1990	375.00	990.00	3100.00
2000	390.00	1060.00	3100.00

Comments:

e-specific~electr
Non-thermal requirements of electricity.

g-gasoline
Specific uses of gasoline for transportation.

h-thermal~uses
Useful energy demand for thermal uses of energy.

Distribution of demands to load regions:

for demand	1. lr	2. lr	3. lr	4. lr
e-specific~electr	0.107	0.393	0.300	0.200
g-gasoline	0.084	0.333	0.333	0.250
h-thermal~uses	0.148	0.466	0.292	0.094

Power level relative to average load:

for demand	1. lr	2. lr	3. lr	4. lr
e-specific~electr	1.271	1.181	0.902	0.798
g-gasoline	1.000	1.000	1.000	1.000
h-thermal~uses	1.767	1.399	0.877	0.375

LISTS OF ENERGY FORM NAMES

Energy forms:

level: r	level: a	level: x	level: f	level: u
o crude~oil	l light~oil	e electricity	e electricity	e specific~electr
c hard~coal	g gasoline	d district~heat	d district~heat	g gasoline
h hydrocap	f fueloil		l light~oil	h thermal~uses
f fueloil	c hard~coal		g gasoline	
	x so2-dummy			

Comments:

Energy Forms on Level f

e-electricity
Final electricity.

d-district~heat
Final district heat.

l-light~oil
Light oil products for thermal uses.

g-gasoline
Gasoline for utilization as motor fuel.

```
Energy Forms on Level x

e-electricity
   Secondary electricity.

d-district~heat
   Secondary district heat.

Energy Forms on Level a

l-light~oil
   Light oil products for thermal uses.

g-gasoline
   Gasoline for utilization as motor fuel.

f-fueloil
   Heavy refinery products that can either be used
   in powerplants or heating plants.

Energy Forms on Level r

f-fueloil
   Heavy refinery products that can be cracked again
   or transferred to the level of use.

Stock piles: (level q)
----------------------

id name                stock in base year
 h hydrocapacity                0.

Comments:
---------

h-hydrocapacity
   Interlink for the build-up of new hydropowerplants (Unit: MW).
```

DEFINITION OF ADDITIONAL RELATIONS

Used for accounting only :

SO2-emissions
Counter on the emissions of SO2 from central conversion plants.

NOx-emissions
Counter on the emissions of NOx from central conversion plants.

Used as Constraints :

years	elcp:1 (le)	elcp:2 (le)	elcp:3 (le)	elcp:4 (le)
1980	0.	0.	0.	0.
1985	0.	0.	0.	0.
1990	0.	0.	0.	0.
2000	0.	0.	0.	0.

Comments:

elcp (lr 1)
Constraint to force an overcapacity of 30%
for electricity generation, load region 1.

elcp (lr 2)
Same constraint to for load region 2.

elcp (lr 3)
Same constraint to for load region 3.

elcp (lr 4)
Same constraint to for load region 4.

years	dhcp:1 (le)	dhcp:2 (le)	dhcp:3 (le)	dhcp:4 (le)
1980	0.	0.	0.	0.
1985	0.	0.	0.	0.
1990	0.	0.	0.	0.
2000	0.	0.	0.	0.

Comments:

dhcp (lr 1)
Constraint to force an overcapacity of 20%
for district heat production, load region 1.

dhcp (lr 2)
Same constraint to for load region 2.

dhcp (lr 3)
Same constraint to for load region 3.

dhcp (lr 4)
Same constraint to for load region 4.

years	pkel (le)	dhrl (le)	prel (ge)	ogup (le)
1980	0.	none	0.	0.
1985	0.	0.	0.	0.
1990	0.	0.	0.	0.
2000	0.	0.	0.	0.

Comments:

pkel
Constraint to force a minimum electricity production
from peak powerplants of 3%.

dhrl
The share of district heat shall not decline
in relation to the demand.

prel-peak~power
10% of installed capacity has to be peak powerplants.

cgup-max~cogen
Maximum operation time of cogeneration per year: 6000h

years	cglo (ge)
1980	0.
1985	0.
1990	0.
2000	0.

Comments:

cglo-min~cogen
Minimum operation time of cogeneration per year: 4000h

```
RESOURCES - IMPORTS - EXPORTS
------------------------------


 1. l-light~oil          on level: 2
-----------------------------------

Imports
-------
              total                 1. grade
------------------------------------------------
annual   lolim    uplim    lolim    uplim    cost
 1980     0.      650.      0.               213.
 1985     0.      650.      0.               213.
 1990     0.      650.      0.               235.
 2000     0.      650.      0.               253.
------------------------------------------------
total     0.    19500.      0.

Dynamic constraints on imports
------------------------------------------

for              type   hist.extr.  gam    g
---------------------------------------------
      1. grade   lo        500.00   0.90   0.

Price is set to 1.183 times the price of crude oil.


 2. g-gasoline           on level: 2
-----------------------------------

Imports
-------
              total                 1. grade
------------------------------------------------
annual   lolim    uplim    lolim    uplim    cost
 1980     0.      100.      0.      100.     263.
 1985     0.      100.      0.      100.     263.
 1990     0.      100.      0.      100.     290.
 2000     0.      100.      0.      100.     312.
------------------------------------------------
total     0.     3000.      0.     3000.
```

```
Dynamic constraints on imports
-------------------------------------------

for              type   hist.extr.  gam     g
---------------------------------------------
total resource    up        0.      1.10  10.00
total resource    lo        0.      0.90  10.00

Price is set to 1.46 times the price of crude oil.


 3. o-crude~oil          on level: 1
------------------------------------

Imports
-------
            total                1. grade
------------------------------------------------
annual   lolim     uplim     lolim    uplim    cost
 1980     0.                  0.               180.
 1985     0.                  0.               180.
 1990     0.                  0.               199.
 2000     0.                  0.               214.
------------------------------------------------
total     0.                  0.

Imported crude oil.
```

```
4. c-hard~coal          on level: 1
-----------------------------------

Domestic resources
------------------
            total                1. grade                 2. grade
-------------------------------------------------------------------------
annual   lolim     uplim     lolim    uplim     cost   lolim   uplim    cost
 1980    0.                  0.                 0.     0.               20.
 1985    0.                  0.                 0.     0.               20.
 1990    0.                  0.                 0.     0.               20.
 2000    0.                  0.                 0.     0.               20.
-------------------------------------------------------------------------
total        0.   30000.         0.  10000.                0.  20000.

Dynamic constraints on resource extraction
------------------------------------------

for              type   hist.extr.  gam    g
---------------------------------------------
total resource    up        70.00   1.12   0.
      1. grade    up        70.00   1.10   0.
```

Costs of coal extraction is included in the according technology. Cost of second grade represents additional cost.

```
 5. h-hydrocap          on level: 1
-----------------------------------

Domestic resources
------------------
            total                1. grade
------------------------------------------------
annual   lolim    uplim     lolim   uplim    cost
 1980     0.                  0.             1000.
 1985     0.                  0.             1000.
 1990     0.                  0.             1000.
 2000     0.                  0.             1000.
------------------------------------------------
total     0.                  0.

Capacity for building hydropower as resource.
unit: MW
```

DESCRIPTION OF TECHNOLOGIES

This section summarizes the economic and technological data as well as the scenario related parameters for the technologies included in the energy supply system modeled. The following types of tables are used:

ECONOMIC AND TECHNOLOGICAL PARAMETERS

This table shows the investment and operation costs of the system as well as the most important technological parameters. The cost figures are expressed as cost per unit of total output or per unit of main output (see switch 14 in table 2, chapter 4.4).

Entries to the Table:

capcost	Specific Investment Cost per kW Installed. This item includes costs of: -- components and material, -- direct and indirect labour, -- architect/engineers fee, -- owners cost (site preparation, -- infrastructure, licensing, preliminary studies, etc.), and -- interest paid during construction
var O+M	Annual Variable Operation and Maintainance Cost per kWyr of energy produced. This figure comprises: -- labour cost, -- cost of repair, -- replacement of parts, -- non-energy raw material consumption, and -- waste disposal.
fix O+M	Annual Fixed Operation and Maintainance Cost per kW installed capacity. Includes the same items as stated above.

pllife	Technical Lifetime of Technologies.
plfctr	Capacity Factor. For end use technologies and technologies handling fuels without load regions this factor represents the time the system is actually working. For other technologies it represents the maximum availability factor.
avg cst	Average Cost of Producing one Unit of Output. This figure represents only the costs related to the technology as such. In order to determine the total cost per unit of energy output the fuel costs have to be added to the value stated here. The given value is calculated using the maximum capacity factor and a depreciation rate equal to the discount rate. As the capacity factor can be lowered by the model in the case an energy form with load regions (e.g. electricity, district heat) is produced, these costs have to be interpreted as the minimum cost per unit of output.

ENERGY FLOWS AND EFFICIENCIES

This list shows the amounts of energy in- and outputs relative to one unit of main input (1. energy form in table) followed by a number indicating the energy form level on which this energy form is defined. These energy form names are listed in the table 'LIST FO ENERGY FORM NAMES' above.

Entries to the Table:

input	Energy Inputs.
output	Energy Outputs.
initrequ	Initial Core Requirements for Nuclear Power Plants.
finalret	Last Core of Nuclear Power Plants.

If the load pattern of a technology was fixed exogenously an additional table shows the fraction of the energy in- or output per load region and the according power levels relative to the annual load average.

USER DEFINED RELATIONS

This list is only included if the technology is linked to user defined relations. These can be used for various purposes, like e.g. accounting only, limiting the sum of outputs for a number of technologies, or pricing of non energy inputs.

The figures in this table show the values deducted from or added to the according accounting row per unit of new capacity or per unit of main output (1. energy output shown in the previous list).

A full list of the accounting row names, together with an explanation on their meaning, is included in the previous section.

LIMITATIONS

The parameters shown in the following tables comprise the constraints and bounds imposed on the capacity expansion and/or the energy outputs.

Dynamic Constraints on Capacity Build Up:

These bounds constrain the annual new capacity build-up of the annual production $Z(t)$ as a function of the new built capacity or produduction during the previous year $Z(t-1)$

$$Z(t) <= gam * Z(t-1) + g ,$$

gam	Annual Growth Factor [% per year],
g	Initial Capacity or Production for new Technologies [MW].

To calculate the maximum annual biuld-up during a period (T) of n years this formula results in

$$Z(T) <= gam * Z(T-1)^{n} + g * (1 + gam + .. + gam^{n-1}).$$

Annual Absolute Bounds:

These values represent absolute limits imposed on the annual build-up of new capacities or on the amount of main output per year. Each of them is given as a time series with an entry for each time period defined.

VARIABLE PARAMETERS

This table includes all time dependent economic and technical technology descriptors as well as entries to user defined relations.

1 ueee elec-appliance

Short system description:

Specific use of electricity.
No costs are assumed due to variety of uses
and lack of substitution possibilities.

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
0.	0.	0.	0.	1.00	0.

energyforms	lev	input	output
electricity	f e	1.00	
specific~electr	u e		1.00

2 uggg gasoline-use

Short system description:

Specific use of gasoline for transport.

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
0.	0.	0.	0.	1.00	0.

energyforms	lev	input	output
gasoline	f g	1.00	
gasoline	u g		1.00

```
3   ullh   loil heating
------------------------------

    Short system description:
    -------------------------

    Light oil heating system (75kW).

     capcost    var O+M    fix O+M pllife  plfctr  avg cst
      US$/kW    US$/kWa  US$/kW/a  years    fr    US$/kWa
    -------------------------------------------------------
      124.00       0.         4.00   15.    0.23     65.88

    energyforms      lev     input   output
    ---------------------------------------
    light~oil        f l     1.00
    thermal~uses     u h              0.66

    Annual new capacities in historic periods
    (from 1961 until 1975 in  5 year steps)
    -----------------------------------------
     1186.1,  700.4,  413.6,

    Dynamic bounds on build up of:
    ------------------------------------
                             gam       g
    capacity (up)           1.05     40.00

    Annual bounds on new capacities:
    ---------------------------------

    upper bounds
             300.00  none    none    none


    Variable parameters
    -------------------


    energy forms
         h (output)      0.664      0.690      0.730      0.750
```

```
  4   ueeh   elec heating
-------------------------------

        Short system description:
        --------------------------

        Waterbased electric heating system (75kW).

         capcost    var O+M   fix O+M pllife  plfctr  avg cst
          US$/kW    US$/kWa  US$/kW/a  years    fr    US$/kWa
        ----------------------------------------------------------
          116.00      0.         3.40   15.    0.23     60.14

        energyforms       lev    input   output
        ----------------------------------------
        electricity       f e    1.00
        thermal~uses      u h             0.96

        Annual new capacities in historic periods
        (from 1961 until 1975 in  5 year steps)
        -------------------------------------------
           10.0,    16.1,    25.9,

        Dynamic bounds on build up of:
        -------------------------------------
                              gam        g
        activity (low)        0.90      0.50
        capacity (up)         1.10     40.00

        Annual bounds on activity:
        ---------------------------

        upper bounds
                  60.00  none     none     none
```

```
 5   uddh   district-heat
-------------------------------

         Short system description:
         -------------------------

         District heating system (75kW).

          capcost    var O+M   fix O+M pllife  plfctr  avg cst
           US$/kW    US$/kWa  US$/kW/a  years    fr    US$/kWa
         -----------------------------------------------------
            90.00       0.          3.20   15.    0.23     49.11

         energyforms      lev     input   output
         ---------------------------------------
         district~heat    f d     1.00
         thermal~uses     u h               0.90

         relations          id  lr      coeff. relative to
                                      capacity       main output
         -----------------------------------------------------
                          dhr1                              1.00

         Annual new capacities in historic periods
         (from 1961 until 1975 in  5 year steps)
         ----------------------------------------
           155.6,  250.7,  403.7,

         Annual bounds on activity:
         --------------------------

         upper bounds
                  1600.00 1540.76 1484.04 1468.42

         lower bounds
                   960.00  924.46  890.43  881.05
```

```
Variable parameters
-------------------


energy forms
     h (output)      0.900      0.920      0.940      0.950



 6   uehh    elec hpump
----------------------------

     Short system description:
     -------------------------

     Ground heatpump with oil back-up.
     The oil covers 30% of the production in
     the peak demand area.
     First year of operation: 1985

      capcost    var O+M    fix O+M pllife  plfctr  avg cst
       US$/kW    US$/kWa   US$/kW/a  years    fr    US$/kWa
     -----------------------------------------------------
       355.00      0.          9.00   15.     0.23    177.95

     energyforms       lev     input   output
     ----------------------------------------
     electricity       f e     1.00
     thermal~uses      u h              2.72
     light~oil         f l     0.29

     Dynamic bounds on build up of:
     -----------------------------------

                             gam        g
     capacity (up)           1.10     20.00

     Annual bounds on activity:
     --------------------------

     upper bounds
              480.00  472.50  465.00  465.00
```

```
Variable parameters
-------------------


lower dynamic bounds
         g          20.000    20.000    20.000     0.



  7   feee   elec distribution
----------------------------------

      Short system description:
      -------------------------

      Distribution of electricity.

        capcost    var O+M   fix O+M pllife  plfctr  avg cst
         US$/kW    US$/kWa  US$/kW/a  years    fr    US$/kWa
      ------------------------------------------------------
         350.00       0.        7.00   30.     0.73    37.32

      energyforms       lev    input   output
      ---------------------------------------
      electricity       x e     1.00
      electricity       f e             0.90

      Annual new capacities in historic periods
      (from 1946 until 1975 in  5 year steps)
      -----------------------------------------
           7.3,    10.7,    15.7,    23.0,    33.8,    49.7,

      Dynamic bounds on build up of:
      ------------------------------
                             gam       g
      capacity (up)          1.05      0.
```

```
  8   fddd   DH distribution
--------------------------------

      Short system description:
      -------------------------

      Distribution of district heat.

       oapcost    var O+M    fix O+M pllife  plfctr  avg cst
        US$/kW   US$/kWa  US$/kW/a  years     fr    US$/kWa
      ----------------------------------------------------
        100.00      0.          2.00   30.    0.75     10.38

      energyforms      lev    input   output
      -------------------------------------
      district~heat    x d    1.00
      district~heat    f d             0.91

      output pattern fixed
      energy:     0.15  0.47  0.29  0.09
      power:      1.77  1.40  0.88  0.38

      Annual new capacities in historio periods
      (from 1946 until 1975 in  5 year steps)
      -----------------------------------------
         42.8,    57.3,    76.6,   102.5,   137.2,   183.6,


  9   f111   loil distribution
----------------------------------

      Short system description:
      -------------------------

      Distribution of light refinery products.
```

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
0.	14.00	0.	0.	1.00	14.00

energyforms	lev	input	output
light~oil	a l	1.00	
light~oil	f l		1.00

10 fggg mfuel distrib.

Short system description:

Distribution of gasoline.

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
0.	14.00	0.	0.	1.00	14.00

energyforms	lev	input	output
gasoline	a g	1.00	
gasoline	f g		1.00

11 xffd foil heatpl

Short system description:

Fueloil heatplant (25MW).

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
116.00	7.20	5.80	25.	0.95	21.12

energyforms	lev	input	output
fueloil	a f	1.00	
district~heat	x d		0.85

relations	id	lr	coeff. relative to capacity	coeff. relative to main output
	dhcp	t	-0.83	1.00
emissions	S02			17.80
emissions	NOx			5.60

Annual new capacities in historic periods (from 1951 until 1975 in 5 year steps)				
133.1,	136.5,	139.9,	143.4,	147.1,

Variable parameters

energy forms d (output)	0.850	0.870	0.900	0.920

12 xccd coal co-gen

Short system description:

Coal co-generation plant.
(50MW(e)/100MW(th))

```
 capcost    var O+M   fix O+M pllife  plfctr  avg cst
 US$/kW     US$/kWa  US$/kW/a  years    fr    US$/kWa
-----------------------------------------------------
  403.20       3.73      11.95   25.    0.88     46.64

energyforms       lev    input   output
---------------------------------------
hard~coal         a c    1.00
district~heat     x d             0.56
electricity       x e             0.19

relations          id  lr      coeff. relative to
                             capacity      main output
-----------------------------------------------------
                  dhcp  t     -0.83              1.00
                  elcp  t     -0.26              0.34
                  pkel                           0.83
emissions         SO2                           27.00
emissions         NOx                           15.75
max~cogen         cgup  t     -0.68              1.00
min~cogen         cglo  t     -0.46              1.00
peak~power        prel  t     -0.05

Annual new capacities in historic periods
(from 1952 until 1975 in  5 year steps)
-----------------------------------------
   17.6,     0. ,     0. ,     0. ,


Annual bounds on new capacities:
--------------------------------

fixed bounds
           0.     none     none     none


Variable parameters
-------------------


energy forms
     d (output)     0.560      0.550      0.550      0.550
     e (output)     0.190      0.200      0.230      0.270
```

13 xffe foil ppl

Short system description:

Heavy destillate oil base load steam power plant.

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
434.00	12.60	5.20	25.	0.90	49.25

energyforms	lev	input	output
fueloil	a f	1.00	
electricity	x e		0.37

relations	id	lr	coeff. relative to capacity	coeff. relative to main output
	elcp	t	-0.77	1.00
	pkel			1.00
emissions	SO2			16.93
emissions	NOx			12.72
peak~power	prel	t	-0.10	

Annual new capacities in historic periods (from 1951 until 1975 in 5 year steps)				
1.9,	3.0,	4.8,	7.8,	12.5,

Dynamic bounds on build up of:	gam	g
capacity (up)	1.05	0.20

```
14   xlje   diesel ppl
---------------------------

         Short system description:
         -------------------------

         Light destillate oil peak power plant.

          capcost    var O+M    fix O+M pllife   plfctr   avg cst
           US$/kW    US$/kWa   US$/kW/a  years     fr     US$/kWa
         -------------------------------------------------------
           260.00      28.00       5.20   25.      1.00      49.84

         energyforms      lev    input   output
         --------------------------------------
         light~oil        a l    1.00
         electricity      x e             0.28

         relations          id   lr      coeff. relative to
                                      capacity        main output
         -------------------------------------------------------
                            elcp  t    -0.77               1.00
                            pkel                         -32.33
         emissions          SO2                           15.75
         emissions          NOx                           16.89
         peak~power         prel  t     1.00

         Annual new capacities in historic periods
         (from 1951 until 1975 in  5 year steps)
         -----------------------------------------
            22.1,   13.0,    7.7,     4.5,    2.7,
```

15 x.he hydropower

Short system description:

Run of river hydropower plant,
average plant factor .45

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
425.00	0.	18.00	60.	0.63	58.39

energyforms	lev	input	output	initrequ	finalret
electricity	x e		1.00		
hydrocapacity	q h			1.00	1.00

output pattern fixed

energy:	0.07	0.29	0.29	0.36
power:	0.85	0.86	0.86	1.42

relations	id	lr	coeff. relative to capacity	main output
	elcp	t	-0.77	1.00
	pkel			1.00
peak~power	prel	t	-0.10	

Annual new capacities in historic periods
(from 1916 until 1975 in 5 year steps)

0.2,	0.3,	0.5,	0.8,	1.4,	2.2,	3.5,
5.7,	9.1,	14.7,	23.6,	38.0,		

Dynamic bounds on build up of:

	gam	g
activity (low)	1.00	0.
activity (up)	1.05	0.

```
Annual bounds on new capacities:
--------------------------------

upper bounds
          15.00  none    none    none



16   qhoh   link
---------------------


 capcost    var O+M    fix O+M pllife  plfctr  avg cst
  US$/kW    US$/kWa   US$/kW/a  years    fr    US$/kWa
------------------------------------------------------
     0.         0.         0.      0.    1.00      0.

energyforms       lev    input   output
---------------------------------------
hydrocap          r h    1.00
hydrocapacity     q h              1.00



17   aong   refinery
-------------------------


 capcost    var O+M    fix O+M pllife  plfctr  avg cst
  US$/kW    US$/kWa   US$/kW/a  years    fr    US$/kWa
------------------------------------------------------
    3.79       0.12       0.     25.    0.80      0.42

energyforms       lev    input   output
---------------------------------------
crude~oil         r o    1.00
gasoline          a g              0.15
light~oil         a l              0.30
fueloil           r f              0.50
```

relations	id	lr	coeff. relative to capacity	coeff. relative to main output
emissions	S02			5.00
emissions	NOx			1.60

Annual new capacities in historic periods (from 1951 until 1975 in 5 year steps)				
58.2,	52.6,	47.5,	42.9,	38.8,

18 afcg catcracker

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
22.00	0.65	0.	25.	0.80	2.41

energyforms	lev	input	output
fueloil	r f	1.00	
gasoline	a g		0.09
light~oil	a l		0.57
fueloil	a f		0.28

relations	id	lr	coeff. relative to capacity	coeff. relative to main output
emissions	S02			8.00
emissions	NOx			2.50

Alternative operation

af2g catcracker

var O+M US$/kWa	avg cst US$/kWa
1.95	3.71

energyforms	lev	input	output
fueloil	r f	1.00	
gasoline	a g		0.28
light~oil	a l		0.38
fueloil	a f		0.28

relations	id	lr	coeff. relative to capacity	coeff. relative to main output
emissions	SO2			3.65
emissions	NOx			0.83

20 aflf link

capcost US$/kW	var O+M US$/kWa	fix O+M US$/kW/a	pllife years	plfctr fr	avg cst US$/kWa
0.	0.	0.	0.	1.00	0.

energyforms	lev	input	output
fueloil	r f	1.00	
fueloil	a f		1.00

```
21   accc    coal extraction
---------------------------------


   capcost    var O+M    fix O+M pllife   plfctr   avg cst
    US$/kW    US$/kWa   US$/kW/a  years     fr     US$/kWa
  -------------------------------------------------------
    200.00       5.00       10.00   20.     1.00      29.72

  energyforms       lev     input   output
  ---------------------------------------
  hard~coal         r c     1.00
  hard~coal         a c              1.00

  Annual new capacities in historic periods
  (from 1956 until 1975 in  5 year steps)
  -----------------------------------------
      8.2,     2.7,     0.9,     0.3,


  Variable parameters
  --------------------


  capital costs       200.000    200.000    220.816    237.476
  variable costs        5.000      5.000      5.520      5.937
  fix costs            10.000     10.000     11.041     11.874
```