Extension of ANSWERv6-MARKAL and GAMS-MARKAL code to permit Modeling of Lumpy Retirements

Overview and Acknowledgement

Intelligent Energy Systems, Sydney (IES) requested Noble-Soft Systems to implement an extension to ANSWERv6-MARKAL and the GAMS-MARKAL code to permit the modelling of Lumpy Retirements (for conversion and process technologies) for single-region MARKAL models, and provided funding support.

The modifications to the MARKAL mathematics to permit the modelling of Lumpy Retirements were suggested by Dr Ken Stocks of IES, and are detailed below.

The initial implementation of the new Lumpy Retirement facilities requested by IES was for single-region MARKAL models only, however more recently Noble-Soft Systems has made the necessary extensions to the GAMS-MARKAL code so that these facilities are now *also available for multi-region MARKAL models*. Note that these facilities are not available in conjunction with Stochastics, MACRO, MICRO, ENV_Damage or SAGE at this time.

To access the new Lumpy Retirement facilities, the user must have access to a GAMS-compatible optimizer (such as CPLEX, XPRESS) that has mixed integer programming (MIP) capabilities. The user is cautioned that the computer run time for solving a mixed integer program that involves a large number of integer variables can be many times that for solving a linear program of similar size. Hence use of the new facilities for modelling Lumpy Retirements should be confined to those technologies where retirement is intrinsically lumpy.

Rationale for the Lumpy Retirement Extension

The standard GAMS-MARKAL code does not permit the modelling of early retirement of technology capacity (retirement prior to the end of a technology's lifetime). The Lumpy Retirement extension permits more sophisticated modelling in that it allows for the early retirement of technology capacity (for conversion and process Technologies), and also allows the user to impose the requirement that the amount of capacity retired must occur in meaningful-sized amounts (hence "lumpy") or not at all. The modelling of this lumpy requirement is handled via mixed integer programming (MIP) techniques.

ANSWER Modelling of Lumpy Retirements, Comparison with Modelling of Lumpy Investments

ANSWER-MARKAL and the GAMS-MARKAL code have incorporated a facility for the modelling of Lumpy *Investments* since 2003. (See the Word document "Modeling Lumpy Investments in ANSWER-MARKAL.doc" in folder C:\AnswerV6\Doc".)

There are similarities between the modelling of Lumpy Retirements and Lumpy Investments, and there are differences:

- When modelling Lumpy Retirements, *the size of the 'lump' is specified by a new ANSWER time-series parameter RET_BLOCK*, where RET_BLOCK(TCH, YEAR) = minimum *positive* level of retirement of existing capacity for technology TCH in period YEAR. (We say *positive* because a zero level of retirement of existing capacity is allowable.)

  This is analogous to using ANSWER time-series parameter INV_BLOCK, where INV_BLOCK(TCH, YEAR) = minimum *positive* level of investment in new capacity for technology TCH in period YEAR when modelling Lumpy Investments.

- When modelling Lumpy Retirements, and where RET_BLOCK(TCH, YEAR) > 0, *retirement of existing capacity in period YEAR is restricted to being an integer multiple of RET_BLOCK, or zero*.

  This is analogous to option (a) in the "Modeling Lumpy Investments …" Word doc. But note that for option (a) in the Lumpy Investments case, ANSWER TID parameter INV_INT must also be specified, *whereas in the Lumpy Retirements case no new ANSWER TID parameter needs to be specified*.

- When modelling Lumpy Retirements, it is also allowable to specify two other options:

  (ii) RET_BLOCK(TCH, YEAR) = 0, to indicate that retirement is not allowed in that YEAR, or

  (iii) RET_BLOCK(TCH, YEAR) = blank, to indicate retirement is continuous in that YEAR.

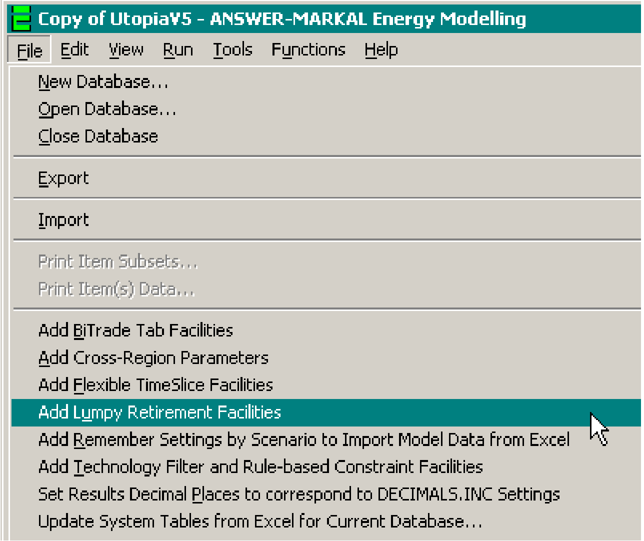
  There are no analogous options to the above in the Lumpy Investments case.

To make the Lumpy Retirement extension available in an ANSWERv6-MARKAL Database

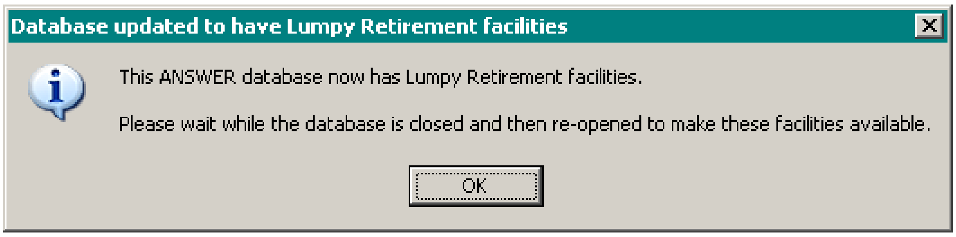
1. You must have access to a GAMS-compatible optimizer (such as CPLEX, XPRESS) that has mixed integer programming (MIP) capabilities.

2. You must be running AnswerV6.exe version 6.3.01 (or higher), and you must be running GAMS-MARKAL version 5.9a (or higher).

3. Run AnswerV6.exe and open the single-region or multi-region ANSWER-MARKAL database that you wish to make Lumpy Retirement aware. Then, invoke the new "File, Add Lumpy Retirement Facilities" menu item, as per the screen snapshot below. Do this immediately – if you move off the Home screen to the Data/Results screen then return to the Home Screen, you will find that this menu item has been disabled, and you will need to close the database and then re-open it.


Copy of UtopiaV5 - ANSWER-MARKAL Energy Modelling
File
Edit
View
Run
Tools
Functions
Help
New Database...
Open Database...
Close Database
Export
Import
Print Item Subsets...
Print Item(s) Data...
Add BiTrade Tab Facilities
Add Cross-Region Parameters
Add Flexible TimeSlice Facilities
Add Lumpy Retirement Facilities
Add Remember Settings by Scenario to Import Model Data from Excel
Add Technology Filter and Rule-based Constraint Facilities
Set Results Decimal Places to correspond to DECIMALS.INC Settings
Update System Tables from Excel for Current Database...

4. After a few seconds the following message will appear to confirm that the database now has Lumpy Retirement facilities:


Database updated to have Lumpy Retirement facilities

This ANSWER database now has Lumpy Retirement facilities.

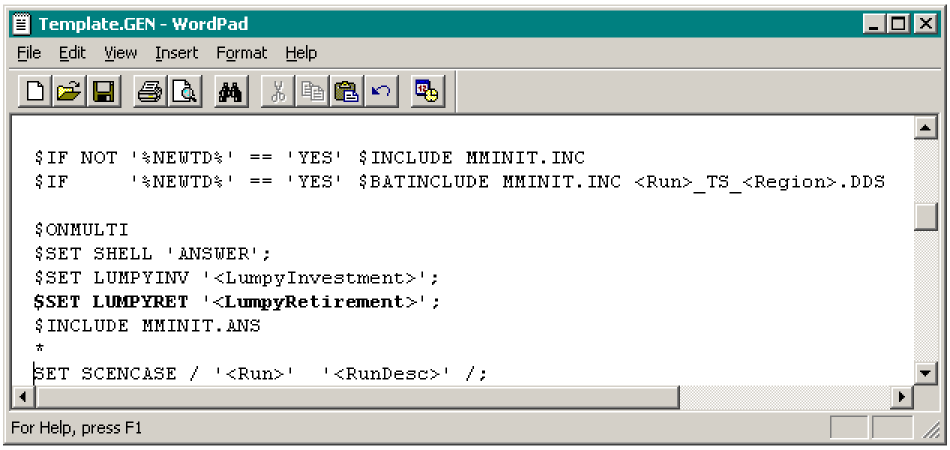
Please wait while the database is closed and then re-opened to make these facilities available.

5. *In the single-region case* the Template.GEN file that is in the GAMS-MARKAL version 5.9a (or higher) code is already set up for use with the Lumpy Retirement facility. However if you are running a single-region MARKAL model, and have been using your own, non-standard, Template.GEN file then note that use of the Lumpy Retirement facility requires the line:

```
$SET LUMPYRET '<LumpyRetirement>';
```

to be inserted into the file Template.GEN in your GAMS Source folder immediately below the line:

```
$SET LUMPYINV '<LumpyInvestment>';
```



```
$IF NOT '%NEWTD%' == 'YES' $INCLUDE MMINIT.INC
$IF     '%NEWTD%' == 'YES' $BATINCLUDE MMINIT.INC <Run>_TS_<Region>.DDS

$ONMULTI
$SET SHELL 'ANSWER';
$SET LUMPYINV '<LumpyInvestment>';
$SET LUMPYRET '<LumpyRetirement>';
$INCLUDE MMINIT.ANS
*
SET SCENCASE / '<Run>'  '<RunDesc>' /;
```

When running ANSWER, the keyword `<LumpyRetirement>` will be replaced by **`YES`** or **`NO`** according to whether the “Lumpy Retirement” checkbox on the Run Model form is checked or unchecked. (See section 10 below.)

Note also that when using the Lumpy Retirement facility, the Template.GEN file <u>must not</u> contain the line:

```
        $SET CPTEL 'L'
```

6. *In the multi-region case* similar considerations to those at 5 above apply: the RegionTemplate.GEN/SLV/RPT files in the GAMS-MARKAL version 5.9a (or higher) code are already set up for use with the Lumpy Retirement facility

7. Invoke “Run, Mapping of Nulls to GAMS DD” and check that the Upper Bound setting is +INF. **If wishing to use option (iii) below (RET_BLOCK(TCH,YEAR) = blank, indicating that retirement is continuous) it is <u>essential</u> that the Upper Bound setting is +INF** (otherwise an Upper Bound setting of space is OK)**.**

8. Before invoking “Run Model”, ensure that your GAMS Work folder contains an options file (for example CPLEX.OPT if CPLEX is your MIP optimizer) with appropriate settings for the OPTCR parameter (and perhaps OPTCA parameter) that controls a mixed integer programming run. Be aware that the OPTCR parameter is a relative termination tolerance, meaning that the MIP solver will stop and report the first integer solution found that is within 100*OPTCR of the best possible solution. That is, if OPTCR is set at 0.01, the MIP solver will stop when it obtains an integer solution that is within 1% of the best possible solution. (If OPTCR is set at 0.0, the MIP solver will continue until it has found the optimal integer solution, and verified that this integer solution is indeed optimal. This may involve a prohibitive amount of time for a MIP problem that has a large number of integer/binary variables.) For additional details, see the GAMS Users Guide, and the documentation that is specific to the MIP optimizer that you are running.

To specify that Retirement of Existing Capacity for a Particular Technology is to be modelled, possibly as ‘Lumpy’

9. For each Conversion or Process Technology TCH where modelling Retirement, possibly Lumpy, is required, use the AddRow on either the Technology tab or the Parameter tab to create an instance of the RET_BLOCK parameter where:

   (i) RET_BLOCK(TCH,YEAR) > 0 indicates the size of the ‘lump’ for retirement of existing capacity for technology TCH in period YEAR. (For example, if the size of the ‘lump’ for retirement of existing capacity is 100 PJ in each period, then create an instance of the RET_BLOCK parameter with values of 100 in each period.)

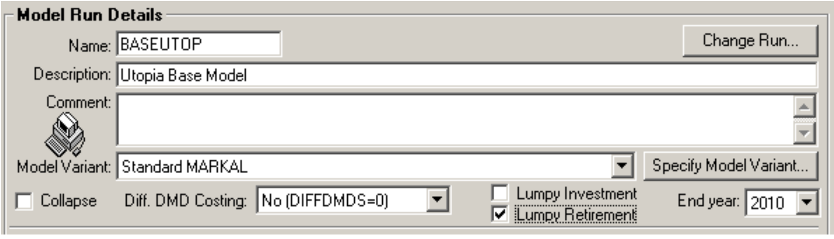
   (ii) RET_BLOCK(TCH,YEAR) = 0 indicates that retirement is not allowed for technology TCH in period YEAR.

   (iii) RET_BLOCK(TCH,YEAR) = blank indicates that retirement is continuous (not lumpy) for technology TCH in period YEAR.

   Note that it is allowable for an instance of the RET_BLOCK parameter for a particular TCH to have a mix of positive, zero and blank values in different periods if desired.

If when you use the AddRow you do not see the RET_BLOCK parameter, this means that the database you are working with is not Lumpy Retirement aware. Rectify this by carrying out steps 2 and 3 above.

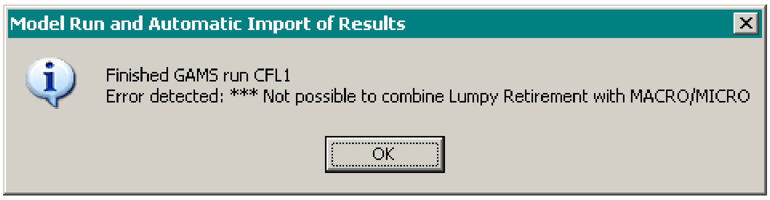
10. When you invoke “Run Model”, ensure that the new “Lumpy Retirement” checkbox (immediately below the “Lumpy Investment” checkbox) is checked:



Model Run Details
Name: BASEUTOP
Description: Utopia Base Model
Comment:
Model Variant: Standard MARKAL
□ Collapse
Diff. DMD Costing: No (DIFFDMDS=0)
□ Lumpy Investment
☑ Lumpy Retirement
End year: 2010

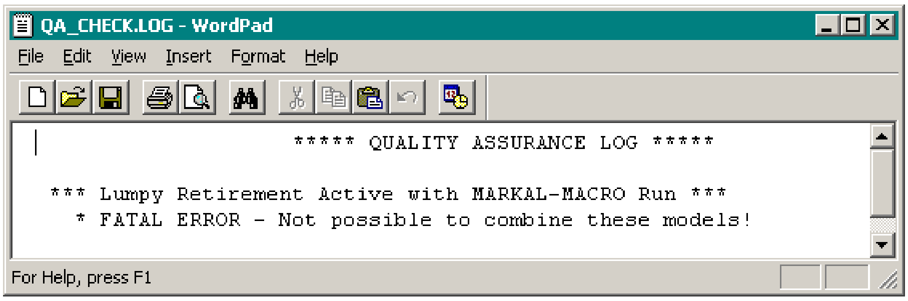
If when you invoke “Run Model” you do not see the “Lumpy Retirement” checkbox, this means that the database you are working with is not Lumpy Retirement aware. Rectify this by carrying out steps 2 and 3 above.

Note that it is not possible to carry out a model run that invokes the Lumpy Retirement facilities and where MARKAL-MACRO or MARKAL-MICRO is selected as the Primary Variant. (Such a run would require an optimizer capable of handling both mixed integer programming, and non-linear programming.) If such a run is attempted, a fatal error is generated, and an appropriate error message is written to the screen and to the QA_CHECK.LOG file:



Model Run and Automatic Import of Results

Finished GAMS run CFL1
Error detected: *** Not possible to combine Lumpy Retirement with MACRO/MICRO



QA_CHECK.LOG - WordPad

```
                    ***** QUALITY ASSURANCE LOG *****

 *** Lumpy Retirement Active with MARKAL-MACRO Run ***
   * FATAL ERROR - Not possible to combine these models!
```

For Help, press F1

New Data Parameter added to ANSWER re Lumpy Retirements

TS Data Parameter RET_BLOCK

Description: Min. retirement level of existing capacity, if retirement occurs.
Domain: May be defined for any Conversion or Process Technology, and for each time period.
Units: Technology Capacity
Comment: The RET_BLOCK parameter must be specified for each technology TCH where modelling of Retirement, possibly Lumpy, is required.

(i) Specify RET_BLOCK(TCH,YEAR) > 0 to indicate the size of the 'lump' for retirement of existing capacity for technology TCH in period YEAR.
(ii) Specify RET_BLOCK(TCH,YEAR) = 0 to indicate that retirement is not allowed for technology TCH in period YEAR.
(iii) Specify RET_BLOCK(TCH,YEAR) = blank to indicate that retirement is continuous (not lumpy) for technology TCH in period YEAR.

In GAMS: The ANSWER parameter RET_BLOCK is mapped to the GAMS parameter RET_BLK in the *.DD/*.DDS files created by ANSWER.

How the New Model Equations and Variables related to Lumpy Retirement appear in the GAMS .LST file

For a single-region MARKAL model:

Where RET_BLOCK(TCH,YEAR) is specified for technology TCH in ANSWER, RET_BLOCK(TCH,YEAR) will be mapped to TABLE RET_BLK(TCH,YEAR) in the GAMS code. Then provided that the "Lumpy Retirement" checkbox on the Run Model form is checked:

1. New continuous variables **RETIRE**(YEAR,TCH) will be created. New integer model variables **RETIRINT**(YEAR,TCH) may also be created, this depends on the value specified for RET_BLK(TCH,YEAR) and is discussed below.

2. The existing capacity interperiod transfer equations EQ_CPT2 (for Conversion Technologies) and EQ_CPT3 (for Process Technologies), which currently appears in the GAMS as follows (simplified):

```
EQ_CPT2/3(YEAR,TCH)..
    CAP(YEAR,TCH) – SUM(appropriate YEAR1,  INV(YEAR1,TCH))  =E= TCH_RES(TCH,YEAR)
```

will be modified as follows (simplified):

```
EQ_CPT1/2/3(YEAR,TCH)..
    CAP(YEAR,TCH) – SUM(appropriate YEAR1,  INV(YEAR1,TCH))
                  + SUM(YEAR1<=YEAR, RETIRE(YEAR1,TCH))  =E= TCH_RES(TCH,YEAR)
```

3(i) Where RET_BLK(TCH,YEAR) > 0, integer model variables **RETIRINT**(YEAR,TCH) and new model equations **EQ_RETBLK**(YEAR,TCH) will be created, where:

```
EQ_RETBLK(YEAR,TCH)..
        -RETIRE(YEAR,TCH) + RET_BLK(TCH,YEAR) * RETIRINT(YEAR,TCH) =E= 0
```

and
RETIRINT(YEAR,TCH) has an upper bound of TCH_RES(TCH,YEAR)/RET_BLK(TCH,YEAR) if parameter instance RESID(TCH,*) exists, else an upper bound of 10. (Note that TCH_RES appears in ANSWER as data parameter RESID.)

The effect of EQ_RETBLK together with the requirement that RETIRINT be an integer variable, is that in each period, retirement of existing capacity, if positive, must occur in integer multiples of RET_BLK.

3(ii) Where RET_BLK(TCH,YEAR) = 0, integer model variables **RETIRINT**(YEAR,TCH) and

new model equations **EQ_RETBLK**(YEAR,TCH) will not be created.
Variable **RETIRE**(YEAR,TCH) will be assigned an upper bound of zero, forcing **RETIRE**(YEAR,TCH) = 0.

So by specifying RET_BLK(TCH,YEAR) = 0 in some YEAR, the user is indicating that they don't wish to allow Retirement to occur in that YEAR.

3(iii) Where RET_BLK(TCH,YEAR) = blank, integer model variables **RETIRINT**(YEAR,TCH) and new model equations **EQ_RETBLK**(YEAR,TCH) will not be created.
No upper bound will be assigned to variable **RETIRE**(YEAR,TCH) – it will be allowed to take any positive value (of course consistent with capacity remaining non-negative).

So by specifying RET_BLK(TCH,YEAR) = blank in some YEAR, the user is indicating that they wish to allow **RETIRE**(YEAR,TCH) to be a continuous variable in that YEAR.

For a multi-region MARKAL model:

Similar considerations apply to those described above for a single-region MARKAL model, but since this is the multi-region case, the new continuous variables that are created will be named R_RETIRE(REG,YEAR,TCH) and if new integer model variables are created they will be named R_RETIRINT(REG,YEAR,TCH). Likewise instead of the new model equations being named EQ_RETBLK(YEAR,TCH), they will be named MR_RETBLK(REG,YEAR,TCH) and with other standard adjustments that apply in the multi-region case as compared with the single-region case.

Additional Considerations

1. If the "Lumpy Retirement" checkbox on the Run Model form is unchecked then the Lumpy Retirement facilities will be disabled, and any Lumpy Retirement data that may be present in the database will be ignored when "Run Model" is invoked.