Extension of ANSWERv5 and GAMS-MARKAL code to permit Modeling of Lumpy Investments

Overview

To permit more sophisticated modeling in respect of lumpy investments (such as the construction of gas pipelines), version 5 of ANSWER and the GAMS-MARKAL code have been extended to handle the all-or-nothing building of new capacity via mixed integer programming (MIP) techniques. Three different facilities for modeling lumpy investments are provided. Each of these facilities requires the user to identify technologies TCH where investment in new capacity is 'lumpy'. The size of the 'lump' is specified by a new ANSWER time-series parameter INV_BLOCK, where INV_BLOCK(TCH, YEAR) = minimum *positive* level of investment in new capacity in technology TCH in period YEAR. (We say *positive* because a zero level of investment in new capacity is allowable.) The three facilities that are provided allow the user to require that:

(a) Investment in new capacity in each period is either an integer multiple of INV_BLOCK, or zero; *or*

(b) Investment in new capacity in each period is either INV_BLOCK, or zero; *or*

(c) Investment in new capacity in each period is either INV_BLOCK, or zero and investment is restricted to occur in at most one period.

The user specifies which of the three facilities (a) or (b) or (c) is to be adopted for a particular technology TCH via new ANSWER TID parameters INV_INT or INV_BIN or INV_SOS respectively. How this is done is detailed below.

The new facilities that permit the modeling of lumpy investments are available under version 5 of ANSWER for both single-region and multi-region MARKAL models. However they are not available in conjunction with Stochastics, MACRO, MICRO, ENV_Damage or SAGE at this time.

The user is cautioned that the computer run time for solving a mixed integer program that involves a large number of integer/binary variables can be many times that for solving a linear program of similar size. Hence use of the new facilities for modeling lumpy investments should be confined to those technologies (such as gas pipelines) where investment is intrinsically lumpy.

Acknowledgement
The approach taken is similar to that previously implemented in an OMNI version of MARKAL by Dr Ken Stocks of Intelligent Energy Systems, Sydney.

To make the lumpy investment extension available in ANSWER-MARKAL

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

2. Use ANSWER's 'File, Update System Tables from Excel' facility to apply the supplied XLS file *UpdateAnsSysTables_V52.XLS* to all existing version 5 ANSWER databases. Also apply this update XLS to your version 5 template database TemplateV5.MDB in C:\ANSWERv5, or alternatively replace your existing TemplateV5.MDB in C:\ANSWERv5 by the supplied TemplateV5.MDB. (The effect of applying this XLS file to a version 5 ANSWER database is to make that database aware of the new data parameters that are needed to specify that investment in new capacity is to be modeled as 'lumpy'; and also aware of the new data and results parameters that are needed for the recent GAMS-MARKAL enhancements: goal programming model variant, global trade, DMD vintaging and derived technology emissions.)

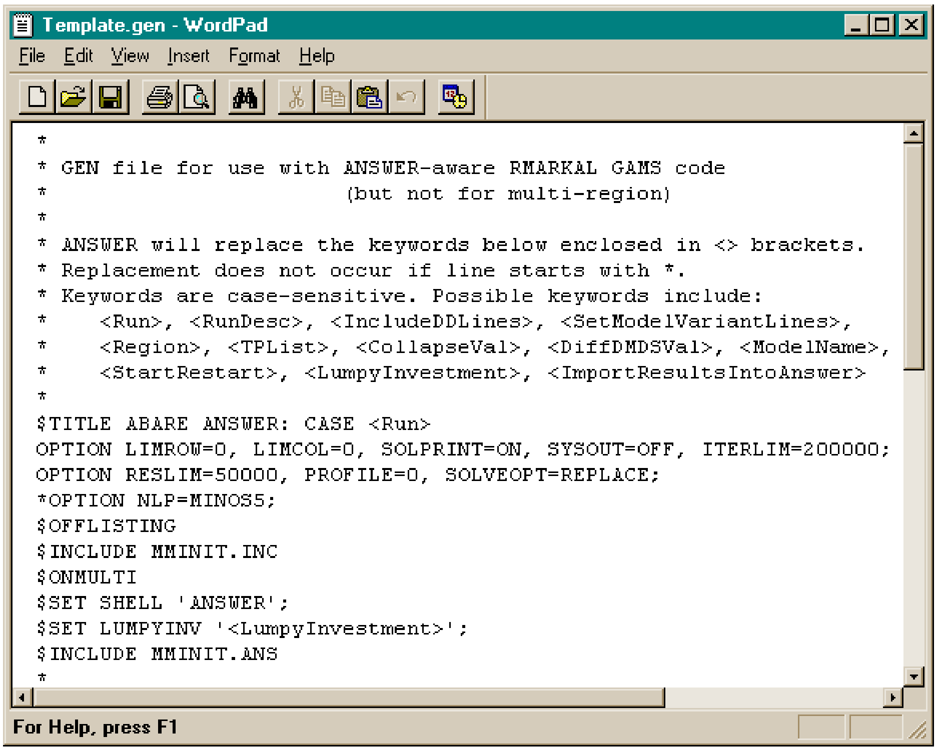
3. Unzip the supplied zip file *Gams_srcPRD_V52.zip* into C:\ANSWERv5\Gams_srcPRD folder, over-writing. (If you are cautious, and wish to test these enhancements while retaining your existing Gams_srcPRD files, then rename your existing Gams_srcPRD folder and create a new Gams_srcPRD folder prior to unzipping Gams_srcPRD_V52.zip.)

4. 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 investment facility in conjunction with ANSWER version 5.1.3 or higher requires the line:

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

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

```
$SET SHELL 'ANSWER';
```



```
*
* GEN file for use with ANSWER-aware RMARKAL GAMS code
*                       (but not for multi-region)
*
* ANSWER will replace the keywords below enclosed in <> brackets.
* Replacement does not occur if line starts with *.
* Keywords are case-sensitive. Possible keywords include:
*    <Run>, <RunDesc>, <IncludeDDLines>, <SetModelVariantLines>,
*    <Region>, <TPList>, <CollapseVal>, <DiffDMDSVal>, <ModelName>,
*    <StartRestart>, <LumpyInvestment>, <ImportResultsIntoAnswer>
*
$TITLE ABARE ANSWER: CASE <Run>
OPTION LIMROW=0, LIMCOL=0, SOLPRINT=ON, SYSOUT=OFF, ITERLIM=200000;
OPTION RESLIM=50000, PROFILE=0, SOLVEOPT=REPLACE;
*OPTION NLP=MINOS5;
$OFFLISTING
$INCLUDE MMINIT.INC
$ONMULTI
$SET SHELL 'ANSWER';
$SET LUMPYINV '<LumpyInvestment>';
$INCLUDE MMINIT.ANS
*
```

When running ANSWER, the keyword **`<LumpyInvestment>`** will be replaced by **`YES`** or **`NO`** according to whether the "Lumpy Investment" checkbox on the Run Model form is checked or unchecked. (See para 9 below.) The standard Template.GEN file that is in Gams_srcPRD_V51d.zip already contains the **`$SET LUMPYINV '<LumpyInvestment>';`** line.

5. If you are running a multi-region model, and have been using your own, non-standard, RegionTemplate.GEN/RPT/SLV files then note that use of the lumpy investment facility in conjunction with ANSWER version 5.1.3 or higher requires the line:

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

    to be inserted into each of the files RegionTemplate.GEN/RPT/SLV in your GAMS Source folder immediately below the line:

    ```
    $SET SHELL 'ANSWER';
    ```

    When running ANSWER, the keyword **`<LumpyInvestment>`** will be replaced by **`YES`** or **`NO`** according to whether the “Lumpy Investment” checkbox on the Run Model form is checked or unchecked. (See para 9 below.) Each of the standard RegionTemplate.GEN/RPT/SLV files in Gams_srcPRD_V51d.zip already contains the **`$SET LUMPYINV '<LumpyInvestment>';`** line.

To specify that investment in new capacity for a particular technology is to be modeled as ‘lumpy’

6. For each technology TCH where modeling of lumpy investment is required, create an instance of the INV_BLOCK parameter to specify the size of the ‘lump’ for investment in new capacity in each period. For example, if the size of the ‘lump’ for investment in new capacity is 100 PJ in each period, then create an instance of the INV_BLOCK parameter with values of 100 in each period. (Note that it is allowable for an instance of the INV_BLOCK parameter to have different values in different periods.)

7. For each technology TCH where the INV_BLOCK parameter has been specified at step 6, decide which of the three lumpy investment facilities (a) or (b) or (c) is to be applied, and follow the steps below at (a) or (b) or (c) respectively.

    (a) If wanting to specify that for technology TCH investment in new capacity in each period is an integer multiple of INV_BLOCK or zero, for this TCH create an instance of the **INV_INT** parameter with TID value = 1.

    (b) If wanting to specify that for technology TCH investment in new capacity in each period is INV_BLOCK or zero, for this TCH create an instance of the **INV_BIN** parameter with TID value = 1.

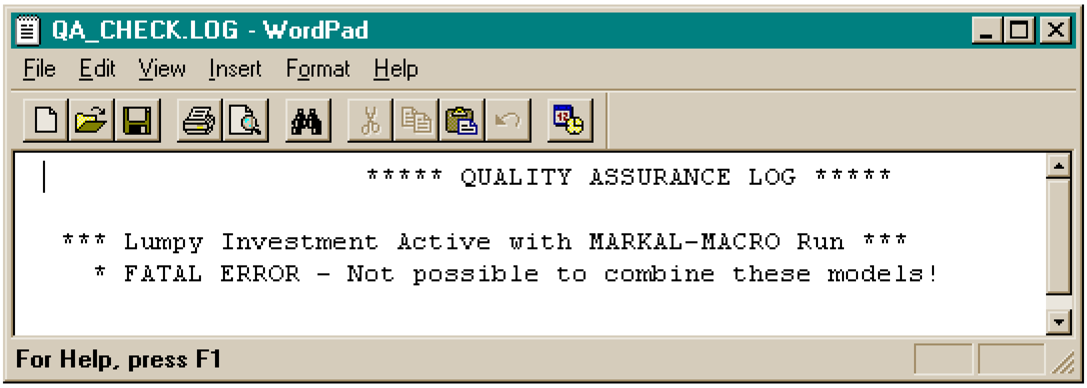
    (c) If wanting to specify that for technology TCH investment in new capacity in each period is INV_BLOCK or zero, *and that investment is restricted to occur in at most one period*, for this TCH create an instance of the **INV_SOS** parameter with TID value = 1.

Note that ANSWER does not prevent the user from associating instances of INV_INT, INV_BIN and INV_SOS with the same technology. Paras 2 and 3 of the "Additional Considerations" section indicate how this ambiguous situation is resolved.

"Run Model" considerations

8. Before invoking "Run Model", ensure that your GAMS Work folder contains an options file (for example OSL.OPT if OSL 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.

9. When you invoke "Run Model", ensure that the "Lumpy Investment" checkbox (just below the Model Variants combobox) is checked. Note that it is not possible to carry out a model run that invokes the lumpy investment 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, for example:



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

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

New Data Parameters added to ANSWER re lumpy investments

TS Data Parameter INV_BLOCK

Description: Min. investment level in new capacity, if investment occurs.

Domain: May be defined for any non-resource technology, and for each time period.

Units: Technology Capacity

Comment: The INV_BLOCK parameter must be specified for each technology TCH where modeling of lumpy investment is required. INV_BLOCK(TCH,YEAR) indicates, for technology TCH, the size of the ‘lump’ for investment in new capacity in period YEAR.

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

TID Data Parameter INV_INT

Description: Indicator that new capacity investment level is INT * BLOCK.

Domain: May be defined for any non-resource technology.

Units: None

Comment: The INV_INT parameter must be specified (with TID value = 1) for each technology TCH where modeling of lumpy investment is required, and where investment in new capacity in each period is restricted to be an integer multiple of INV_BLOCK, or zero.

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

TID Data Parameter INV_BIN

Description: Indicator that new capacity investment level is BLOCK, or 0.

Domain: May be defined for any non-resource technology.

Units: None

Comment: The INV_BIN parameter must be specified (with TID value = 1) for each technology TCH where modeling of lumpy investment is required, and where investment in new capacity in each period is restricted to be INV_BLOCK, or zero.

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

TID Data Parameter INV_SOS

Description: Indicator new cap. invest. in <=1 period, level BLOCK or 0.

Domain: May be defined for any non-resource technology.

Units: None

Comment: The INV_SOS parameter must be specified (with TID value = 1) for each technology TCH where modeling of lumpy investment is required, and where investment in new capacity in each period is restricted to be INV_BLOCK or zero, *and where investment is restricted to occur in at most one period.*

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

How the New Model Equations and Variables related to lumpy investment appear in the GAMS .LST file

For a single-region MARKAL model:

The new model equations and variables related to lumpy investments appear in the GAMS listing file <casename>.LST.

As already noted, at step 7 above the user decides which of the three lumpy investment facilities (a) or (b) or (c) is to be applied to technology TCH. The new model equations and variables that are generated are as follows:

(a) Where INV_BLOCK and INV_INT=1 are specified for technology TCH, new model equations **EQ_INVBLK1**(YEAR,TCH) and new integer model variables **INVINT**(YEAR,TCH) are created, where:

```
EQ_INVBLK1(YEAR,TCH)..
        -INV(YEAR,TCH) + INV_BLK(TCH,YEAR) * INVINT(YEAR,TCH) =E= 0
```

and
INVINT(YEAR,TCH) has an upper bound of TCH_IBND(TCH,'UP',YEAR)/INV_BLK(TCH,YEAR) if parameter instance TCH_IBND(TCH,'UP',*) exists, else an upper bound of 10. (Note that TCH_IBND appears in ANSWER as data parameter IBOND(BD).)

The effect of EQ_INVBLK1 together with the requirement that INVINT be an integer variable, is that in each period, investment in new capacity, if positive, must occur in integer multiples of INV_BLK.

(b) Where INV_BLOCK and INV_BIN=1 are specified for technology TCH, new model equations **EQ_INVBLK2**(YEAR,TCH) and new binary model variables **INVBIN**(YEAR,TCH) are created, where:

```
EQ_INVBLK2(YEAR,TCH)..
        -INV(YEAR,TCH) + INV_BLK(TCH,YEAR) * INVBIN(YEAR,TCH) =E= 0
```

The effect of EQ_INVBLK2 together with the requirement that INVBIN be an binary variable, is that in each period, investment in new capacity, if positive, must be equal to INV_BLK.

(c) Where INV_BLOCK and INV_SOS=1 are specified for technology TCH, new model equations **EQ_INVBLK2**(YEAR,TCH) and **EQ_INVSOS**(TCH) and new binary model variables **INVBIN**(YEAR,TCH) are created, where:

```
EQ_INVBLK2(YEAR,TCH)..
        -INV(YEAR,TCH) + INV_BLK(TCH,YEAR) * INVBIN(YEAR,TCH) =E= 0
EQ_INVSOS(TCH)..
        SUM(YEAR, INVBIN(YEAR,TCH)) =L= 1
```

with the understanding that the summation in EQ_INVSOS(TCH) is taken over all time periods YEAR >= START(TCH).

The effect of EQ_INVBLK2 together with the requirement that INVBIN be an binary variable, is as for (b) above. The additional effect of the the EQ_INVSOS equation is that investment is restricted to occur in at most one period.

For a multi-region MARKAL model:

The new model equations and variables related to lumpy investments appear in the GAMS listing file <casename+TRADE>.LST.

As already noted, at step 7 above the user decides which of the three lumpy investment facilities (a) or (b) or (c) is to be applied to technology TCH. The new model equations and variables that are generated are as follows:

(a) Where INV_BLOCK and INV_INT=1 are specified for technology TCH in region REG, new model equations **MR_INVBLK1**(REG,YEAR,TCH) and new integer model variables **R_INVINT**(REG,YEAR,TCH) are created, where:

```
MR_INVBLK1(REG,YEAR,TCH)..
        -R_INV(REG,YEAR,TCH) + INV_BLK_R(REG,TCH,YEAR) * R_INVINT(REG,YEAR,TCH) =E= 0
and
R_INVINT(REG,YEAR,TCH) has an upper bound of
TCH_IBND_R(REG,TCH,'UP',YEAR)/INV_BLK_R(REG,TCH,YEAR) if parameter instance
TCH_IBND_R(REG,TCH,'UP',*) exists, else an upper bound of 10.
```

(b) Where INV_BLOCK and INV_BIN=1 are specified for technology TCH in region REG, new model equations **MR_INVBLK2**(REG,YEAR,TCH) and new binary model variables **R_INVBIN**(REG,YEAR,TCH) are created, where:

```
MR_INVBLK2(REG,YEAR,TCH)..
        -R_INV(REG,YEAR,TCH) + INV_BLK_R(REG,TCH,YEAR) * R_INVBIN(REG,YEAR,TCH) =E= 0
```

(c) Where INV_BLOCK and INV_SOS=1 are specified for technology TCH in region REG, new model equations **MR_INVBLK2**(REG,YEAR,TCH) and **MR_INVSOS**(REG,TCH) and new binary model variables **R_INVBIN**(REG,YEAR,TCH) are created, where:

```
MR_INVBLK2(REG,YEAR,TCH)..
        -R_INV(REG,YEAR,TCH) + INV_BLK_R(REG,TCH,YEAR) * R_INVBIN(REG,YEAR,TCH) =E= 0
MR_INVSOS(REG,TCH)..
        SUM(YEAR, R_INVBIN(REG,YEAR,TCH)) =L= 1
```

with the understanding that the summation in MR_INVSOS(REG,TCH) is taken over all time periods YEAR >= START(REG,TCH).

Additional Considerations

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

2. If for the same technology TCH both INV_INT=1 and INV_BIN=1 parameter instances are

specified, the INV_INT=1 instance is ignored and it is assumed that the user wishes to invoke facility (b).

3. If for the same technology TCH both INV_INT=1 (or INV_BIN=1) and INV_SOS=1 parameter instances are specified, the INV_INT=1 (or INV_BIN=1) instance is ignored and it is assumed that the user wishes to invoke facility (c).

4. Use of the name INV_SOS (for the TID variable that is specified when the user wishes to invoke facility (c)) might lead some experienced mathematical programming practitioners to expect that special ordered sets (SOS) would be used. That is, to expect that perhaps for lumpy investment facility (c) the GAMS formulation may have used variables INVSOS(TCH, YEAR) defined as SOS1 variables, rather than the approach that was taken of using variables INVBIN(YEAR,TCH) defined as binary variables. However, section 17.2.2 of the GAMS User Guide advises that:

   "Not all MIP solvers allow SOS1 variables. Furthermore, among the solvers that allow their use, the precise definition can vary from solver to solver. Any model that contains these variables may not be transferable among solvers."

   For these and other reasons, the use of SOS1 variables in the GAMS formulation was avoided.