A generic decomposition approach for equations in GAMS models

Wolfgang Britz

Institute for Food and Resource Economics, University Bonn

Outline

- Why decomposition?
- Approaches to decomposition
- Generic approach based on GAMS plus Java (GGIG interface package)
  - Methodology
  - Required output from GAMS
  - Work with the „Equation and Variable Viewer“

Why decomposition?

- If a variable on the LHS depends on several variables on the RHS, it might not be easy to understand what drives the change of the LHS
- Examples:
  - Behavioral equations with own- and cross price effects
  - Profit changes depend on changes of multiple netputs quantities and prices
- Understanding these changes often crucial for policy analysis (and during debugging)

Approaches to decomposition

- Write case specific post-model code:
  – e.g. used in CAPRI for behavioral equations
  – e.g. used in GTAP to decompose welfare changes
  ++ flexible, can track changes across equations or summarize effects of blocks of variables (e.g. all cross price effects)
  -- coding and maintenance efforts
- Generic approach
  – Requires information from solver on Jacobian
  ++ Model independent
  -- Restricted to one equation

Generic approach in GAMS and Java

Change in LHS variable x approximately equal to changes in RHS variables y weighted with their Jacobian entries

CONVERT utility from GAMS delivers Jacobian

$$x = f(y)$$

$$\Delta x \approx \sum_i \left(\frac{\partial x}{\partial y_i} \right) \Delta y_i$$

CONVERT utility from GAMS delivers variable levels before and after the solve to calculate differences in x and y

Generic approach in GAMS and Java

In order to easier digest the information, contribution of variables y is expressed as % contribution to change in x, entries add up approximately to 100%

$$\Delta x \approx \sum_i \frac{\partial x}{\partial y_i} \Delta y_i$$

$$\frac{\Delta x(y_i)}{\Delta x} \% \approx \frac{\partial x}{\partial y_i} \Delta y_i \Big/ \Delta x \, 100$$

Approximate percentage share of change in x stemming for the change in each y

Application to your own model

- Solve your model with CONVERTD (a “pseudo solver”) before and after the actual model solve, variables should be at benchmark (no shock)
- Use the “Equation and Variable viewer” built in GGIG to decompose (does not require that your model runs under the interface)

Application to your own model

```
$iftheni.after "%1"=="after"

$onecho > convertd.opt

   gams %gamsDocDir%\convert_gtap8.gms
   dict %gamsDocDir%\convert_gtap8.txt
   jacobian %gamsDocDir%\convert_gtap8_after.gdx

$offecho

   gtap.optfile = 1;

$else.after

$onecho > convertd.op2

   jacobian %gamsDocDir%\convert_gtap8_before.gdx

$offecho

   gtap.optfile = 2;

$endif.after

      gtap.holdfixed = 0;
      option nlp=convertd;
      solve gtap using nlp minimizing dummy ;
      gtap.holdfixed = 1;

    );
$endif.CONVERT
```

Produces a GDX with the variable levels after the solve, the Jacobian, and a linearized version of your model as GAMS code

Produces a GDX with the variable levels before the solve

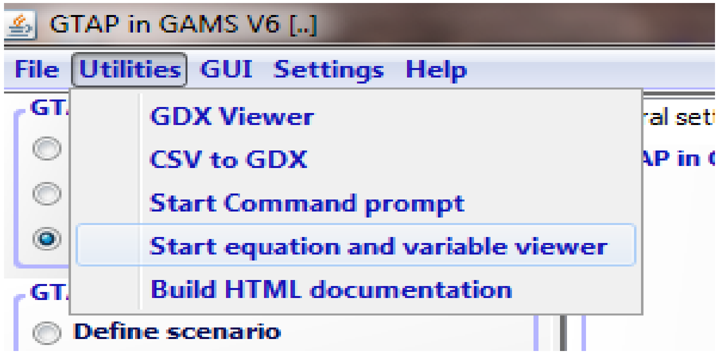
Call to CONVERT, Holdfixed = 0 allows to track effects of fixed variables

Equation and Variable Viewer

- Uses CONVERT output to show equations as generated by GAMS:
  - parameters are converted to numerical constants
  - individual instances are shown, reflecting e.g. $ operators to skip equations and variables
  - selection by variable / equation, filters for domains
  - if GDX with all symbols in model is present, parameters etc. can be easily inspected
  - link to model code possible

Equation and Variable Viewer

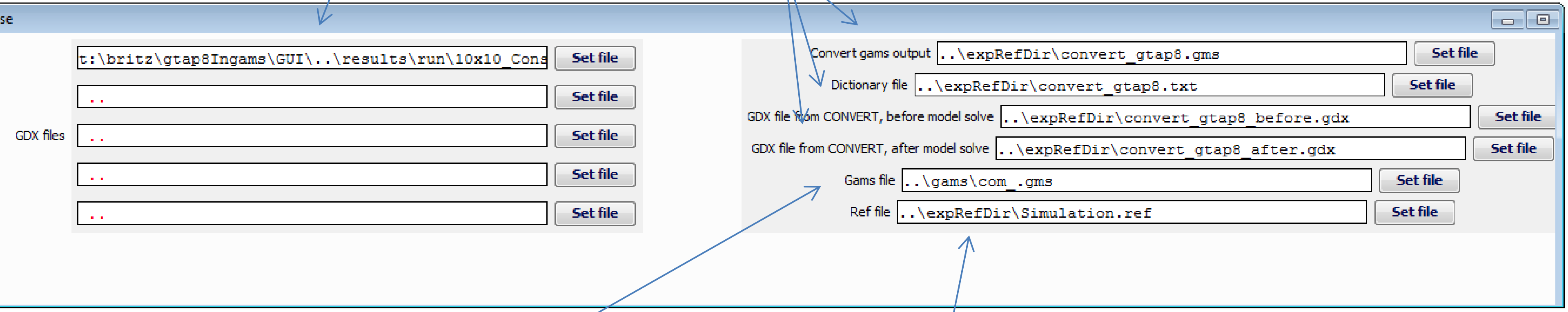
- Open Equation and Variable Viewer from model GUI based on GGIG



Equation and Variable Viewer

GDX from execute_unload
after model solve (all symbols)

Files generated by CONVERT
(variable levels, Jacobian, linearized equations

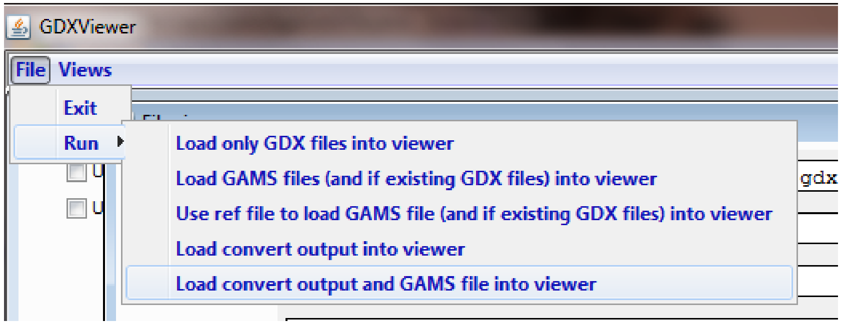


Main GAMS file
of your model

Ref file from running for model
(allows e.g. to find included files

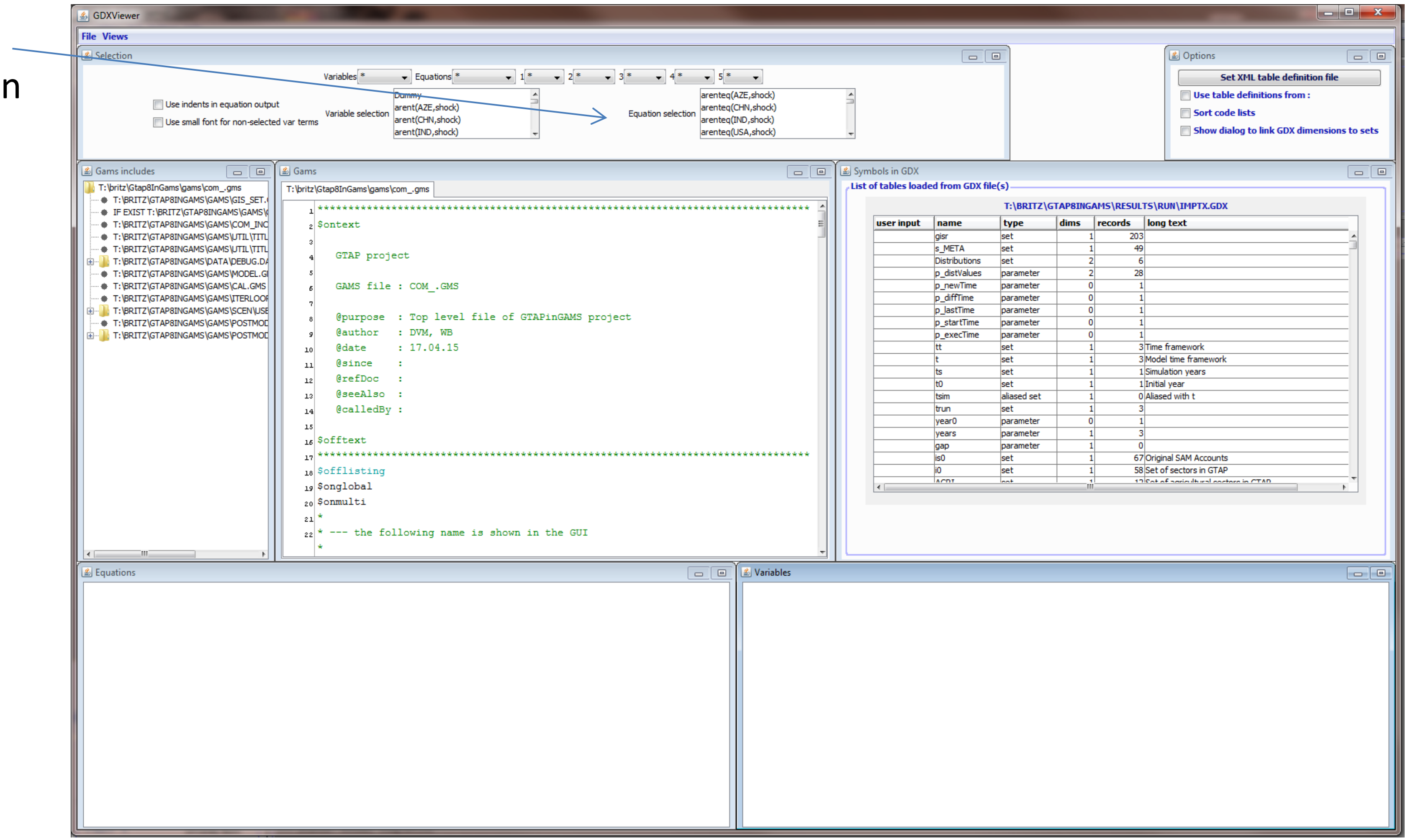
Equation and Variable Viewer

Load the files into the viewer … that might take up to minute, depending on how large the model is



Equation and Variable Viewer

Select
an equation



Equation and Variable Viewer

The viewer will an equation as seen below, parameters become numerical constants

Equation instance

Variable with Jacobian==1 (possibly LHS)

```
arenteq(EU_25,shock) "Aggregate rate of return " ..

  -1.077*pft(EU_25,Capital,shock){-0.003328,0.003328,-0.3328%}*(1.0
  -kappaf(EU_25,Capital,shock){0.0})
  +0.1349*arent(EU_25,shock){-0.02467,-0.003328,-100.0%}=E=0.0;
```

Variable in equation

Change in levls

Change times Jacobian

Contribution to change of LHS

Equation and Variable Viewer

A more complex example with non-linearities

```
xdeq(EastAsia,TransComm-c,hhsld,shock) "Agents demand for domestic goods " ..

  -7.57E-4*(
    pa(EastAsia,TransComm-c,hhsld,shock){0.00965,-0.0162,1290.0%}/(
      ps(EastAsia,TransComm-c,shock){0.0101,0.0169,-1350.0%}*(
        1.0
        +dintx(EastAsia,TransComm-c,hhsld,shock){0.0}
        +itxshft(EastAsia,hhsld,shock){0.0})
      )
    )
  **1.9*xa(EastAsia,TransComm-c,hhsld,shock){3.01,-0.00202,161.0%}
  +7.55E-4*xd(EastAsia,TransComm-c,hhsld,shock){1.66,0.00125,0.141%}=E=0.0;
```

Note: The number of significant digits shown can be changed.

Summary

- Generic approach applicable to any GAMS model to decompose change on LHS of equation to changes of variables on the RHS
- Minimal coding in efforts in GAMS needed
- Integrated in viewer which offers additional information
- Main dis-advantage is that the approach cannot track changes across a set of equations