

Appendix E. Subdirectories and GAMS Programs

build/		Working directory for dataset construction.
	<code>build.gms</code>	Command script to read, filter and aggregate a set of GTAP datasets. Default configuration: produces datasets <code>gtapingams.gdx</code> , <code>g20.gdx</code> , <code>g20_macro.gdx</code> and <code>g20_iea.gdx</code> for 2011. Calls <code>code/flex2gdx.gms</code> , <code>code/filter.gms</code> , <code>code/gtapaggr.gms</code> and <code>code/cdecalib.gms</code> .
	<code>filterchk.gms</code>	Command script to generate data for Table 6: Filtering results. Calls <code>code/flex2gdx.gms</code> , <code>code/filter.gms</code> . Results are written to a PivotData sheet in <code>filterchk.xlsx</code> .
	<code>test.gms</code>	Command script which calls <code>bmchk.gms</code> for dataset <code>g20_macro</code> , <code>mcpmge.gms</code> for dataset <code>g20</code> and <code>cdechk.gms</code> for dataset <code>g20</code> .
	<code>bmchk.gms</code>	Benchmark consistency is a necessary, but not a sufficient condition that the model is properly specified. This routine checks for the <code>g20_macro</code> dataset in both MGE and MCP formats. This routine produces a replication check for the GMR model and all the single region SOE models.
	<code>mcpmge.gms</code>	Is designed to perform a check of consistency for the MCP and MGE models at a point away from the benchmark equilibrium. It can be used to verify that a solution computed with <code>mge.gms</code> also solves <code>mcp.gms</code> , and that a solution computed with <code>mcp.gms</code> solves <code>mge.gms</code> . The GAMS save-point and loadpoint commands are used for this purpose.
	<code>cdechk.gms</code>	Verifies that the CDE demand system reproduces the exogenous own-price and income demand elasticities at the benchmark point.

code/		Code repository – not a working directory.
	<code>flex2gdx.gms</code>	Routine which translates the HAR files from a GTAP .zip archive into GAMS .gdx files. The routine produces an echoprint report of benchmark consistency of the database.
	<code>filter.gms</code>	Filter routine, based on environment parameter <code>-nd</code> , the number of decimal points in the filter.

gtap9data.gms	Utility routine for read a GTAPINGAMS version 9 dataset.
gtapaggr.gms	Dataset aggregation routine (call to cdecilib.gms following aggregation must be made by user).
aggr.gms chktarget.gms checkset.gms	Utility routines called by gtapaggr.gms.
domain.gms	Standard purpose libinclude routine for extracting the nonzero domain of a parameter, using the mysterious "option pd_l<p_l;" syntax.
cdecilib.gms	Utility routine to recalibrate CDE demand following dataset aggregation.
mcp.gms	The canonical static GTAPINGAMS model in algebraic format.
mge.gms	The canonical static GTAPINGAMS model in tabular MPSGE format.
gdpcalc.gms	Utility routine for reporting GDP on the basis of income, final demand or sectoral value-added, compatible with either the mcp.gms or mge.gms models. First call declares report parameters.
loadmdl.gms	Utility routines for switching demand systems (between Cobb-Douglas, LES and CDE) and model closures (GMR versus SOE). The macros included here are described below.

forensics/	Produces some calculations assessing the economic consequences of proportional unilateral changes in trade taxes.
ssagen.gms	This GAMS script generates a GAMS script for sensitivity analysis.
ssasolve.gms	is a GAMS programs written by ssagen.gms which processes tgrid.gms for several alternative model closures and demand system specifications.
ssamerge.gms	is a GAMS programs written by ssagen.gms which combines model results into a format suitable for generating an Excel pivot report.

tgrid.gms	Performs a sequence of counterfactual calculations for a single region, with proportional scaling tariff and export subsidy rates to average values between 0 and 10%. These counterfactual calculations are computed with Cobb-Douglas, LES and CDE demand. Macro results (welfare, model status, and global welfare) are returned in parameter results(dsys,tlvl,*). GDP calculations are stored in gdpresults(dsys,tlvl,gdpcat,r,gdpitem).
-----------	---

Macros for model configuration

Two user-callable macros provided in loadmdl.gms:

\$macro loadsys(ff,r) Loads demand system ff for region r where ff is one of LES, CD or CDE and r is a region identifier, either a quoted set element or a subset of the regions r. This function alters:

cd(r) Cobb Douglas demand system flag for model regions
les(r) LES demand system flag for model regions
cde(r) CDE demand system flag
vdfm(),vifm() Levels of subsistence and discretionary demand
vom() Output level which activate aggregate consumer demand, subsistence and discretionary demand.

\$macro loadrm(rr) Load regions rr(r) as endogenous elements of the current model. If rr is a set which includes all regions in r, then a global multiregional model is produced. If rr is singleton or a subset of regions in r, then a small open economy closure is provided. This routine alters:

rm(r) Set of regions in the model,
rx(r) Set of regions in rest of world,
rnum(r) Set defining the numeraire region – the region in the model with the largest consumption.
vem(i,r) Exports to ROW regions by regions in the model,
rtxs_row(i,r) Average subsidy rates on exports to ROW regions.
rowpfx Current account balance for the rest of world regions.

The user-callable macros provided in `loadmdl.gms` are invoked as follows in the `tgrid.gms` program:

```
*      Load regions in the model, either a global multiregional
*      model containing all regions or a small open economy model
*      with a single region.

$if not set mdl          $set mdl gmr
$if not set rcalc        $set rcalc chn

$if %mdl%==gmr loadrm(r);
$if %mdl%==soe loadrm("%rcalc%");
```

In the same program, a loop over functional forms is used to calculate scenario results for three different demand systems, as shown below.

GDP reporting

`gdpcalc.gms` declares the following identifiers:²⁴

```
set      gdpcat  GDP Categories  /
          expend  Expenditure (C + G + I - (X-M)),
          income  Income (Factor income + taxes),
          valueadded  Sectoral factor earnings plus tax payments,
          total    Total GDP/,

          gdpitem /"X-M",set.g,set.f,
                  revto,revtfd,revtfd,revtfd,revtxs,revtms,
                  expend,income,valueadded,chksum/;

alias (gdpitem, gdpi);

parameter  gdp(gdpcat,*,gdpitem)  Real GDP accounting,
          vadd(g, gdpitem, r)      GDP on a value-added basis;
```

GDP reporting in `tgrid.gms` consists of an include statement at the beginning of the program, just after having read `mge.gms`, in a context permitting declarations:

²⁴ These names may not be used in the calling program.

```
*      Read the model and calculate GDP at the benchmark point:

$include %code%mge

$include %code%gdpcalc
```

The GDP routine is included following scenarios solutions inside the loop over alternative demand systems:

```
loop(dsys,

*      Load the demand system:

    loadsys(dsys,r);

*      Initialize tax instruments at benchmark values:
    rtms(i,s,r) = rtms0(i,s,r);
    rtxs(i,s,r) = rtxs0(i,s,r);

    loop(tlvl,

*      Assign tax rates for the counterfactual simulation:

        rtms(i,s,rcalc)$max(rtms0(i,s,rcalc),0) =
            rtms0(i,s,rcalc) * tlvl.val/averate;
        rtxs(i,rcalc,s)$max(-rtxs0(i,rcalc,s),0) =
            rtxs0(i,rcalc,s) * tlvl.val/averate;

*      Compute the equilibrium values:

        $include gtap9.gen
        solve gtap9 using mcp;
        abort$(gtap9.objval>1e-3) "Simulation fails: gtap9.";

*      Store the GDP results for this simulation:

        $include %code%gdpcalc

        gdpresults(dsys,tlvl,gdpcat,r,gdpitem) = gdp(gdpcat,r,gdpitem);

    ...

))
```