

RunGTAP Help

Place your own product logo here and modify the layout of your print manual/PDF:
In Help & Manual, click "Tools" > "Print Manual Designer" and open this manual template to edit it.

Table of Contents

Part I Contents 7

1 What's new in RunGTAP ... 8
2 What was new in RunGTAP 1999 ... 11
3 What was new in RunGTAP 1998 ... 11
4 File Menu Item ... 12
5 Tools Menu Item ... 12
6 RunGTAP Don'ts ... 13

Part II Closure 14

1 Endogenous and Exogenous Variables ... 16
2 Condensation ... 16
3 New Closure for GTAP version 5 ... 16

Part III Shocks 19

1 Shocks to tax rates ... 21
2 Loading SHF files of shocks ... 22
3 Reading large numbers of shocks from a SHK or HAR file ... 22
4 Saving SHF files of shocks ... 24
5 Subtotals ... 24
6 More about the Theory of Subtotals ... 25
7 Specifying Subtotals ... 27

Part IV Loading and Saving Experiments 28

1 Loading and Saving Closures, Shocks, and Experiments ... 29
2 Exporting Experiments to WinGEM ... 30
3 Replicating WinGEM simulations in RunGTAP ... 30

Part V Solve 32

1 Solution Methods ... 33
2 Extrapolation Accuracy (XAC) File ... 34
3 Automatic Accuracy and Sub-intervals ... 35
4 Accuracy Summary ... 37
5 Running a Test Simulation ... 38
6 The SAVESIMS folder ... 38

Part VI Results 39

1 The Contents Window 40
2 Copying Results 41
3 Adjusting the look of the Results Page 41
4 Filtering for particular set elements 41
5 Cannot View a 3-D matrix 42
6 Number of Decimal Places 42
7 Copying to ClipBoard 42
8 Printing 42
9 Saving Output Files 42
10 CMF file 43
11 SL4 files 43
12 Examining Results using ViewSOL or ViewHAR 43
13 Interpreting Results and Data using AnalyseGE 44

Part VII Viewing the Database and Other Files 45

1 The Core Data File of Flows 47
2 The Parameters File 47
3 The Sets File 47
4 Subsets and Variables 47
5 CMFSTART file 48
6 XSETS and the CMFSTART file 49
7 Using parallel processing 50
8 The Log file 50
9 Derived Elasticities (Partial and General) 50
GE-Elast computations 51
General Equilibrium Elasticities 51
Use of the GEELAST header array file 52
Partial Equilibrium Elasticities 54
10 The RunGTAP Text File Editor 54
11 Choosing your text editor 55
12 ViewHAR 55
13 Header Array Files (HAR files) 55
14 TAB File 56
15 GTAPView 56
16 Tax Rates files 56
17 GTAPVol 57
18 DECOMP 57
19 The Shocks Program 57
20 Regenerating standard shock files 57

Part VIII Versions of the GTAP model 58

1 Files in the Version subdirectories ... 59
2 Derived files in version subdirectories ... 60
3 Using Version Archives ... 61
Creating Version Archives ... 62
Loading Version Archives ... 62
4 Creating new versions ... 62
The New Version Wizard ... 63
Name of new version ... 63
Creating a new version based on the current version ... 63
Creating a new version with a new aggregation ... 64
Files needed for a new aggregation ... 64
After you leave the New Version Wizard ... 65
Editing the Version Information ... 65
Altertax ... 66

Part IX Sensitivity Analysis 68

Part X Options dialog 70
1 Developer Mode ... 71
2 Show Levels Results. ... 72

Part XI Errors 73
1 Solution Not Valid ... 74
2 Job did not run ... 74
3 Error..See Log Window ... 74
4 No output file ... 74
5 Errors in Solution Process ... 75
6 Missing Files ... 75
7 Missing SLI file ... 75
8 Missing EXP files ... 76
9 Not a variable ! ... 76
10 Version directory does not exist ... 76
11 Bad Version Name ... 76
12 Not a HAR file ... 77
13 Miscellaneous Errors ... 77
14 Cannot Write to Version Directory ... 77
15 Old job still running ... 77
16 Formats of HAR and other Fortran files ... 78
17 RunGTAP and long file-names ... 79
18 Changes to the Database format for version 5 GTAP data ... 79
19 GEMPACK Licence May Be Required ... 80
20 RunGTAP Directory must have write access ... 81
21 The Work folder ... 81

22 INI File location 82

Part XII Contacts and Miscellaneous 83

1 The Center for Global Trade Analysis 84
2 The GTAP Model 84
3 The RunGTAP web page 85
4 The GEMPACK Modelling System 85
5 Hardware and Software Requirements 85
6 GEMPACK licence 86
7 Un-Installation 86
8 Hints for running large models 87
9 Valerie Powers, Tax Consultant 87
10 Purdue Pete 88
11 Indiana State Bird 89
12 Indiana State Flower(s) 89
13 The Rube Goldberg Machine Contest 91
14 A Prominent Hoosier 92
15 Debs on War 94
16 The Indiana State Song 96
17 Roger Bean 97
18 Hoosiers 99
19 The Triple XXX 102
20 Tippecanoe and Tyler too ! 103
21 The Beast of Busco 106
22 Corn and Soy 108
23 The Indiana Pi Bill 109
24 Euchre 112
25 The insectless state? 112
26 A vile murder 113
27 The forgotten pyramid 115
28 So it goes 117

Part XIII Additional Information for RunGTAP Developers 119

1 The RunGTAP concept 120
2 Files which must be present in the RUNGTAP directory 120
3 ONEGTAP 121
4 OPTIONS.INI 122
5 Using several different copies of RunGTAP 122
6 Map Files 122
7 Experiment and CMF files 123

8 Files in version sub-directories 124
9 File Descriptions in File Dialogs 125
10 How to share your RunGTAP experiments with others 125
11 How RunGTAP detects what versions are available 125
12 Standard Closure used in test simulations 126
13 Comments in experiment files 126
14 Creating a new version manually 126
15 The Help File 127
16 Check Versions 127
17 Can I alter the GTAP model and still use RunGTAP? 128
18 Un-Omitting Variables 128
19 Work directory for temporary files 128
20 Touch all versions 129
21 64-bit EXE files will not run under 32-bit Windows 129

Part XIV Changing the economics of RunGTAP -- Overview 130

1 Choosing and Disabling Modules 132
2 Notes for Module Developers 135
3 Creating Modules for others to use 138
4 Notes on altering GEELAST 139

Index 0

Part I

1 Contents

Contents

RunGTAP allows to you to conduct a variety of model simulations with the GTAP model. You can choose between several different versions of GTAP. Most versions allow you to choose from several closures. You can set values for the exogenous variables interactively, or read them from prepared files. You can choose among several methods to solve the model. You can view simulation results on-screen, print them, or export them to other programs. You can view, copy or print either the initial or the post-simulation model database. Finally, several tools are included to help explain or analyse results. Print out the tutorial document Ho2K.doc for a detailed guide to using RunGTAP.

RunGTAP uses a tabbed notebook or card index interface. The first 2 pages contain general information. The 3rd page contains information about the current version. The remaining 4 pages are arranged in the order in which they are typically used:

Closure · Shocks · Solve · Results

For information about the commands on the Menu bar see:

File · Copy · View · Version · Tools

There a few warnings that every RunGTAP user should heed: see RunGTAP Don'ts.

For any serious simulation, you need to learn how to compute and apply a large number of discrete shocks. See Reading large numbers of shocks from a SHK or HAR file

See also:

What's new · Problems? · Hardware and Software Requirements · More information about GTAP · The GTAP model · The GEMPACK Modelling System · GTAPView · GTAPVol · DECOMP · Endogenous and Exogenous Variables · Solution methods · Saving output files · Derived Elasticities · Printing · Un-Installation · Files in the Version subdirectories · Creating new versions · AlterTax · Replicating WinGEM simulations in RunGTAP · Exporting Experiments to WinGEM · GEMPACK Licence May Be Required · Hints for running large models · Using parallel processing ·

The file RUNGTAP.PDF in the RunGTAP directory may be printed to get a hard copy of all the topics in [a slightly outdated version of] this help file.

A service update (fixing recently noted problems) can be downloaded from:

http://www.copsmodels.com/rgtpatch.htm

1.1 What's new in RunGTAP

What's new in RunGTAP

This edition of RunGTAP contains a variety of improvements over the previous editions. If you have used RunGTAP before, you should be aware of these changes:

- [added May 2014] The SaveSims folder now contains an EXP file, generated at solve time from the contents of the Closure, Shocks and Solve pages.
- [added May 2014] 64-bit EXE files will not run under 32-bit Windows -- now RunGTAP attempts to warn you of this issue. See 64-bit EXE files will not run under 32-bit Windows.
- [added May 2012] Now works better if you press the "Stop" button while running a simulation.
- [added Sep 2010] Major changes and improvements have been made for made for Systematic Sensitivity Analysis (SSA)
- You can load SSA details from a file and save SSA details to a file.
- You can edit lines in the SSA Details memo.
- Each time you carry out SSA, the software prepares an SSA Summary Report CSV file. This CSV file gives considerable detail as to how the relevant shocks or parameters were varied. You can ask the software to do several SSA calculations concurrently. If you have a machine with say 3 free processors and you ask to do 3 SSA calculations concurrently, this should reduce the total time taken for your SSA calculations by about 67%.
- For more details, please see the SSA Help file which you can access via menu item Help on Systematic Sensitivity Analysis under the Tools Menu.
- [added Dec 2009] View menu now lets you edit the CMFSTART file and online help about CMFSTART has been expanded.
- [added Dec 2009] View menu now lets you view the solution file using ViewHAR. Unlike in ViewSOL, you can see values for variables with 4 or more dimensions. See ViewHAR online help topics *Whats New* or *Opening SL4 solution files* for more details.
- [added May 2007] Previously, RunGTAP imposed a time limit on simulations. This has been removed because, with 64-bit Windows it becomes feasible to solve very large versions of GTAP which might take hours.
- [added Dec 2006] New location for the temporary workfiles created by RunGTAP and the Windows programs that come with it. Hopefully this will end the TEMP problems experienced by a few people in the past. See Work folder.
- [added Dec 2006] New location for the INI file that stores your RunGTAP settings. Now, if you have several copies of RunGTAP, each has its own INI file -- previously all shared one INI file, sometimes leading to problems. See INI file location.
- [added Nov 2006]: Preparing for Windows Vista: new online help engine and adjustments to fonts.
- [added Sept 2006]: Form positioning adjusted to be more friendly to users of 2 monitors.
- [added June 2006]: **Developer...**Check Versions now warns if HAR or EXE files are of the older, LF90, type.

- [added June 2006] Note about ONEGTAP.
- [added June 2006] Note about Using several different copies of RunGTAP.
- [added June 2006] Note about OPTIONS.INI file.
- [added June 2006] Note about Using parallel processing.
- [added Jan 2006] Note about Reading large numbers of shocks from a SHK or HAR file.
- [added Sep 2003] Each time you successfully run an experiment, RunGTAP stores the SL4 and some other files in a subdirectory of the current version folder called SAVESIMS.
- [added Jan 2002] Modified the Accuracy Summary display.
- [added May 2001] The **Help..About** window now has a **Diagnostics** button which lists certain details of your system. If you have a problem with RunGTAP, you should certainly include this information with your error report: it might help track down the problem.
- [added May 2001] The View.Subsets and Variables dialog lets you copy data to a spreadsheet. And it shows coefficients.
- Ability to use different variants of the GTAP model which you or others may have developed. See Changing the economics of RunGTAP -- Overview.
- Ability to use TABLO programs compiled for GEMSIM -- assuming the user has installed GEMSIM. See Changing the economics of RunGTAP -- Overview.
- Version 5 of the GTAP model and database introduced a new data format. See New Data Format for GTAP Version 5. RunGTAP attempts to detect data in the old GTAP 4 format and to convert it automatically to the new format.
- Subtotals results can be calculated and shown.
- You can choose to use TABmate (or another editor) as your text editor.
- GEMPACK licencees can bring together model, results, and data with AnalyseGE.
- More flexibility in specifying the closure (swap and endogenous statements are now allowed)
- Additions to Options dialog.
- Tutorial document Ho2K.doc gives detailed introduction to using RunGTAP.
- Help item listing errors to avoid: see RunGTAP Don'ts.
- New edition of standard GTAP model with better treatment of margins means closures designed for previous editions must be updated. See New Closure for GTAP version 5.
- Buttons for loading and saving files of closures and shock statements have been removed to encourage users to instead use experiment (EXP) files to manage experiments. However these commands are still available from the Tools menu (if you are on the Closure or Shocks page).

- You can look at results with ViewSOL (but there are some limitations).

See also What was new in RunGTAP 1999

1.2 What was new in RunGTAP 1999

What was new in RunGTAP 1999

The 1999 edition of RunGTAP contained a variety of improvements over the previous editions.

- Help item listing errors to avoid: see RunGTAP Don'ts.
- Options dialog.
- You can view tax rates based on pre- or post-simulation databases.
- You can view Subsets and Variables
- New Version Wizard helps you create new versions based on fresh aggregations or on post-simulation databases.
- RunGTAP helps you do sensitivity analysis.
- RunGTAP calculates two types of derived elasticity for you to examine.
- Assistance in running AlterTax simulations so that the database reflects specified values for tax rates. See also the following item:
- Two important improvements to the Shocks page:
- (a) You can apply uniform shocks to subsets of a variable.
- (b) Tax rates can be shocked so they meet given ad valorem targets.
- You can view some results either as levels or changes.
- RunGTAP constantly searches for obsolete files -- if found they may be deleted without warning! See derived files and Run test simulation.

See also What was new in RunGTAP 1998

1.3 What was new in RunGTAP 1998

What was new in RunGTAP 1998

Following features were added February 1998:

- Details of experiments (closure, shocks, experiment description, parameter file and solution method) may be stored in EXP files.
- Version Archives for storing a snapshot of your work or exchanging experiments with other RunGTAP users.
- User can edit shocks and closure directly, or paste into them.

- Ranges of available closures and versions no longer hard-wired.
- Choose any solution method, rather than one of a small range.
- User can choose from a range of parameter files to use in simulations.
- User can edit the parameter file and save it with a new name for future use.
- User can save and reload closures and experiments as well as shocks.
- Option to save all solution outputs at once in a ZIP
- Option to save in a ZIP all the files needed to replicate a RunGTAP simulation using WinGEM.
- Simple procedures to replicate a WinGEM simulation using RunGTAP.
- Special File dialogs ensure that user opens or creates files with the right extension in the right subdirectory, and display a one line description of each file.
- More opportunities to print and copy.
- May view the condensation STI file.
- Task of version developer greatly eased.
- Cannot view Results page till solution has been made.

1.4 File Menu Item

File Menu Item

See:

Save Outputs

Export to WinGEM

Version Archive

Print

Print Setup

1.5 Tools Menu Item

Tools Menu Item

See:

Run test simulation

Options

AlterTax

Valerie

Sensitivity

1.6 RunGTAP Don'ts

RunGTAP Don'ts

TOO MANY COOKS SPOIL THE BROTH. RunGTAP maintains and modifies a multitude of files in the RunGTAP directory and subdirectories. Do not rename, modify or delete these files from outside the RunGTAP program, while RunGTAP is running. RunGTAP could get very confused.

THE SHOCKING TRUTH. When using the Shocks page to create new shocks, dont forget to remove any unwanted existing shocks that are already on the list. Use the **Clear All Shocks** button, or edit the shocks list directly to remove single lines.

DONT FORGET to save your experiments as EXP files after a successful simulation.

Part II

2 Closure

Closure

This page shows you the current closure (list of exogenous variables). The closure is stored in part of the experiment file and is loaded automatically when you load an EXP file.

You can edit the closure directly if you know the GEMPACK syntax for doing this.

Occasionally, you may use the menu commands **Tools...Load Closure File** and **Tools...Save Closure File** to save the current closure or to replace it with other closures that have been stored on disk.

See also Endogenous and Exogenous Variables and Load and Save File Dialog

If you choose to edit the closure directly, please bear the following in mind:

The simplest way of specifying a closure is to list the exogenous variables. A list of exogenous variables is built up with one or more statements of the form exogenous <list> ;, as in, for example,

```
exogenous p_cr p_a0 ;
```

There can be several such statements. It is a good idea to start each such exogenous ; statement on a new line.

You should put a statement

```
rest endogenous ;
```

at the end to indicate that the rest of the variables are endogenous. [Component numbers and components as arguments are allowed in exogenous <list> ; statements following the usual GEMPACK Command file syntax.]

RunGTAP **expects** (but does not force) the closure to be specified as above (ie, using only exogenous and rest endogenous keywords). However closures can be set up in GEMPACK Command files in various ways (for example, by using the endogenous and swap keywords, or by modifying the closure stored on an Environment file). For details, see the GEMPACK documentation. You **can** specify the closure in RunGTAP in any way which is allowed within GEMPACK.

How you specify the closure affects how the drop-down list of variables to shock on the Shocks page is formed. If your closure is simply a list of exogenous variables, only these variables will appear in the drop-down list. Otherwise, all variables of the model (even ones which are endogenous) will appear in that list, and it is your responsibility not to attempt to shock an endogenous variable.

2.1 Endogenous and Exogenous Variables

Endogenous and Exogenous Variables

Most economic models contain more variables than equations. The model is only able to determine values for as many variables as there are equations. Values for some variables must be set externally (by the user). Such variables are called exogenous. The rest -- whose values are determined by the model -- are called endogenous.

Although the number of endogenous variables is always equal to the number of equations, there may be several ways to pick the endogenous group from among the entire set of variables. A particular choice of exogenous and endogenous variables is called a closure.

2.2 Condensation

Condensation

GTAP has far too many variables and equations to solve in its original form. The model must be condensed to become soluble. Condensation consists of:

- **omitting** exogenous variables that will not be shocked.
- **substituting** out matrix variables using matrix equations: the variables must be endogenous and will not appear on the results.
- **backsolving** is like **substituting** except that the substituted variables appears in the results (but must be endogenous).

You can view the file containing the condensation instructions: it will be of interest mainly to GEMPACK experts. See also un-omitting variables.

2.3 New Closure for GTAP version 5

New Closure for GTAP version 5

New variables have been added to version 5 of the GTAP model, so closures designed for previous versions must be amended. Below we compare the **GTAP BOOK CLOSURE** for versions 4 and 5.

```
! GTAP 4 book GE closure: psave = pcgdswld common across regions
Exogenous pop
psave pcgdswld
profitslack incomeslack endwslack
cgdslack saveslack govslack tradslack
ao af afe ava atr
to txs tms tx tm
qo(ENDW_COMM,REG) ;
```

```
Rest Endogenous ;
! GTAP 5 book GE closure: psave = pcgdswld common across region
Exogenous pop
psave pcgdswld
profitslack incomeslack endwslack
cgdslack saveslack govslack tradslack
atm atf ats atd aosec aoreg avasec avareg
afcom afsec afreg afecom afesec afereg
to txs tms tx tm
qo(ENDW_COMM,REG);
Rest Endogenous ;
```

Thus, to convert from 4 to 5:

replace

```
ao af afe ava atr
```

with

```
atm atf ats atd aosec aoreg avasec avareg
afcom afsec afreg afecom afesec afereg
```

The New GE closure is like the BOOK closure except that in the NEW GE closure

we exogenize

```
psaveslack pfactwld
```

instead of

```
psave pcgdswld
```

```
exogenous ! standard
pop
psaveslack pfactwld
profitslack incomeslack endwslack
cgdslack tradslack
ams atm atf ats atd
aosec aoreg avasec avareg
afcom afsec afreg afecom afesec afereg
aoall afall afeall
au dppriv dpgov dpsave
to tp tm tms tx txs
qo(ENDW_COMM,REG) ;
Rest Endogenous ;
```

Here are the 2 closures for **GTAP model Version 6**:

```
exogenous ! standard
pop
psaveslack pfactwld
profitslack incomeslack endwslack
cgdslack tradslack
ams atm atf ats atd
aosec aoreg avasec avareg
```

```
afcom afsec afreg afecom afesec afereg
aoall afall afeall
au dppriv dpgov dpsave
to tp tm tms tx txs
qo(ENDW_COMM,REG) ;
Rest Endogenous ;

Exogenous ! GTAP book closure
pop
psave pcgdswld
profitslack incomeslack endwslack
cgdslack tradslack
ams atm atf ats atd
aosec aoreg avasec avareg
afcom afsec afreg afecom afesec afereg
aoall afall afeall
au uelas dpgov dpsave
to tp tm tms tx txs
qo(ENDW_COMM,REG) ;
Rest Endogenous ;
```

Part III

3 Shocks

Shocks

Use the controls on this page to build a list of shocks which is displayed in the panel at the bottom.

First use the combo box at top left to choose which variable to shock.

If you choose a matrix or vector variable a second row of combo boxes will appear, allowing you to choose whether to shock all of the elements, or only a subset.

Next, choose the value of the shocks. For most variables, a shock of -50 always mean a 50% decrease in the value of that variable. In that case the Type of Shock combo box at the right of the page shows only one option: **% change variable**. For some tax variables the options **% change rate** or **% change power** or **% target rate** may appear: these are explained in Shocks to tax rates.

For some variables pre-calculated shocks may be available (they will be stored on a file suffixed pre). The Type of Shock combo box will present this **file** option if it is available. If you choose to use pre-calculated shocks, the Shock Value edit box will disappear (it is irrelevant).

As you select the variable, its elements and the shock value, a shock specification line is built up (above the 4 buttons). The syntax of this line is that used by GEMPACK.

When you have constructed the shock you want, click the Add to Shock List button. Then go on to construct the next shock.

To remove some or all of the shocks you have added to the list, select and delete them.

The Shocks page only allows you to shock variables that have exogenous elements in the current closure. However, it does allow you to specify shocks to endogenous elements of variables that are only partly exogenous. Also, selecting a new closure does not automatically clear the shocks listed on the Shocks page, so that conceivably you could attempt to shock a variable which WAS exogenous in a previous closure, but was now endogenous. In that case, an error will be generated when you attempt to solve the model.

You can also edit the list of shocks directly if you know the GEMPACK syntax for doing this.

For example, you might add statements of the form:

```
subtotal varname = 1;
```

assuming varname was a shocked variable. see subtotals.

The menu command **Tools...Load Shocks** allows you to read in pre-existing files (suffixed SHF) which contain one or more lines of shock statements. See Loading Files of Shocks. The **Tools...Save Shocks File** command allows you to create such files yourself.

ATTENTION: When using the Shocks page to create new shocks, dont forget to remove any unwanted existing shocks that are already on the list. Use the **Clear All Shocks** button, or edit the shocks list directly to remove single lines.

See also:

Reading large numbers of shocks from a SHK or HAR file

Shocks to tax rates

Loading SHF files of Shocks

Saving SHF files of Shocks

3.1 Shocks to tax rates

Shocks to tax rates

Shocks to some tax variables are by default interpreted in a special way: a value of -50 means a 50% reduction in the ad valorem rate of the tax. In this case the Type of Shock combo box will indicate the option **% change rate**, and the shock statement will contain the keyword **rate%** to indicate the ad-valorem-rate interpretation.

If your shock is a percentage reduction in the ad valorem rate of a tax, the shock specification line will refer to a file. These variables are really powers of taxes (if tax rate is 20%, power is 1.2). The file shows what percent changes in powers are needed to move the ad valorem rates to zero. RunGTAP calculates what percent changes in powers are needed to reduce the ad valorem rates by the specified proportion. For example to eliminate an ad valorem rate of 20% (rate%=-100), the power of the tax must change from 1.2 to 1.0: a change of -16.67%. To halve the same rate (rate%=-50), the power of the tax must change from 1.2 to 1.1: a change of -8.33%. Consult Valerie if you have trouble with changes in powers and ad valorem rates.

If you are shocking an individual tax rate, RunGTAP works out these calculations in front of your eyes.

You may instead choose the option **% change power**. In this case a value of -50 means a halving of the power of the tax. That would change a power of 1.2 into 0.6: turning a 20% tax into a 40% subsidy.

Finally, you may instead choose the option **% target rate**. In this case a value of 10 means changing the tax rate so that it equals 10%. RunGTAP works out the corresponding shock to the power. The **% target rate** option is especially useful for altering database tax rates.

See also:

Shocks

Reading large numbers of shocks from a SHK or HAR file

Loading SHF files of Shocks

Saving SHF files of Shocks

3.2 Loading SHF files of shocks

Loading SHF files of shocks

The menu command **Tools...Load Shocks File** allows you to read in pre-existing files (suffixed SHF) which contain one or more lines of shock statements. This could be useful for replicating more complex experiments. Shocks loaded from an SHF file replace all existing shocks; however you could first use the Copy command to save the old shocks in the Clipboard, then load the new shocks, then position the cursor at the bottom of the shock list and use Shift-Ins to paste the old shocks back in.

The syntax of the SHF files follows the same scheme as the shock statements you build up interactively. They may also include comments (prefixed by an ! character).

See also Load and Save File Dialog

See also:

Shocks

Reading large numbers of shocks from a SHK or HAR file

Shocks to tax rates

Saving SHF files of Shocks

3.3 Reading large numbers of shocks from a SHK or HAR file

Reading large numbers of shocks from a SHK or HAR file

The Shocks page makes it easy to apply individual shocks, but it is not intended that your list of shock statements be very long. You have two techniques to ease the burden.

Technique A: Define your own subsets...using the CMFSTART file.

For example, your shock memo might contain

```
Shock tms(FOOD_COMM,SSA_REG,NSSA_REG) = select from file TMS.SHK;
```

or

```
Shock tms(FOOD_COMM,SSA_REG,NSSA_REG) = target% 5 from file TMS.SHK;
```

Above, FOOD_COMM is a subset of REG, and SSA_REG and NSSA_REG are subsets of REG. The shocks statements above apply whole blocks of shocks simultaneously. The file TMS.SHK contains the shocks to powers of tariffs needed to eliminate the tariffs. So, the first shock statement abolishes tariffs on food going from SSA to the other, NSSA, regions. The second statement sets the same tariffs to 5%.

Technique B: Create your own shock files

To impose your own special structure of shock values, create your own shock file (suffix SHK in the version folder). For example, your shock memo might contain:

```
Shock tms(FOOD_COMM,SSA_REG,NSSA_REG) = select from file MyOwnTMS.SHK;
```

You would construct the file, MyOwnTMS.SHK, following the format of TMS.SHK (blank lines

and text following the ! character may be omitted).

The original file TMS.SHK was created by the program SHOCKS.TAB. If you have GEMPACK, you could adapt this to create and run (outside RunGTAP) your own version, MySHOCKS.TAB, which would create your file MyOwnTMS.SHK.

To learn more about GEMPACK shock statements, consult the GEMPACK manual at http://www.copsmodels.com/gpmanual.htm#gpd3.5.5. The following is a very brief summary:

Two independent choices when using shock files

Do you want to use HAR or SHK (GEMPACK text) files to store shock values?

- SHK (GEMPACK text) files can be created with Excel, a text editor, or a TABLO program. The disadvantage is that each SHK file can only hold one array of numbers. You can open SHK files (as GEMPACK text) from ViewHAR. Your shock statement will resemble:
  ```
  Shock tms(Food,REG,REG) = select from file TMS.SHK;
  ```
- To create HAR files, use a TABLO program or ViewHAR. The advantage is that a HAR file may hold several arrays of numbers on different headers, and is more convenient to view. Your shock statement will resemble:
  ```
  Shock tms(Food,REG,REG) = select from file MyShock.Har header H031;
  ```

Do you want GEMPACK to select just some of the numbers on file?

Suppose you want to shock every element of some variable. Then you can use shock statements like:

```
Shock tms = file TMS.SHK
Shock tms = file MyShock.Har header H031;
```

Suppose you have 3 regions and 3 goods, so that tms has 27 elements, GEMPACK will expect to find 27 numbers on the file.

However, a common situation is that the file holds shock values for all elements of the shocked variables, but you only want to use some of these. Suppose you have 3 regions and 3 goods. You might use shock statements like:

```
Shock tms(Food,REG,REG) = select from file MyShock.Har header H031;
```

Again GEMPACK will expect to find 27 numbers on the file, but only those 9 matching the Food element will be used.

If you wrote

```
Shock tms(Food,REG,REG) = file MyShock.Har header H031;
```

GEMPACK will expect to find just 9 numbers on the file (shocks to the Food element). But this variation is not used very frequently.

Note that the two shock statements:

```
Shock tms(TRAD_COMM,REG,REG) = select from file TMS.SHK;
Shock tms = file TMS.SHK;
```

have the same effect, since tms is dimensioned TRAD_COMM*REG*REG. In each case all 27 elements are shocked.

Mark Horridge recommends:

Normally, store shocks on a HAR file and use the* select from *syntax.

See also:

Shocks

Shocks to tax rates

Loading SHF files of Shocks

Saving SHF files of Shocks

3.4 Saving SHF files of shocks

Saving SHF files of shocks

The menu command **Tools...Save Shocks File** allows you to store the current shock statements on disk. Choose an eight-letter name that has not been used already.

See also Load and Save File Dialog

See also:

Shocks

Reading large numbers of shocks from a SHK or HAR file

Shocks to tax rates

Loading SHF files of Shocks

3.5 Subtotals

Subtotals

See also More about the theory of subtotals and how to Specify Subtotals.

Suppose you ran a simulation where Japan removed import tariffs on 5 different commodities, resulting in a welfare gain of 1 million for the USA.

You might wonder, how much of this gain is due to the tariff reduction on good 1 and how much to the other goods 2 to 4.

You might try running a series of experiments, all starting from the original data, in which you removed one tariff at a time. Probably you would find that the changes in USA welfare due to each tariff shock did not quite add up to one million. The discrepancy arises from the non-linearity of the GTAP model.

GEMPACK offers a more sophisticated approach to this problem, which has two advantages:

- the changes attributed to each shock exactly add up to the effect of the whole package.
- it is rather quick and easy to do.

In GEMPACK jargon, **subtotals** refers to the contribution made by one shock or group of shocks to the total changes caused by all the shocks. Suppose your complete shock package was specified by

```
Shock tms(TRAD_COMM,REG,JPN) = select from file tms.shk;
```

Then, if you added the following lines to the memo on the shocks page:

```
Subtotal tms(EXTRACT,REG,JPN) =Extract Effect;
Subtotal tms(FOOD,REG,JPN) =Food Effect;
Subtotal tms(AGRIC,REG,JPN) =Agric Effect;
Subtotal tms(LITMNFC,REG,JPN) =Litmnfc Effect;
Subtotal tms(HVYMNFC,REG,JPN) =Hvymnfc Effect;
```

RunGTAP would calculate the contributions of these individual tariff cuts to the total simulation results.

The **Define Subtotals** button on the Shocks page help you compose these statements.

After you specify shocks, you can ask for various subtotals to be calculated by clicking on the **Define Subtotal** button. You specify the shocks relevant to each subtotal via the Specify Subtotals form.

The subtotals results are shown in separate columns in the results pages. The subtotals results come after the simulation result and any levels results.

Note that subtotals results can only be calculated by programs from Release 7.0 (or later) of GEMPACK.

3.6 More about the Theory of Subtotals

More about the Theory of Subtotals

Suppose a model was written:

(1) Z= F(X,Y)

Here Z is an endogenous variable and there are two exogenous variables X and Y. In a real model, Z would be a vector and there would be many more exogenous variables. However, this simple model is enough to demonstrate the principle of subtotals.

For small changes around an initial solution we can write the linearized form:

(2) delZ= [Fx]delX + [Fy]delY

where Fx and Fy are the derivatives of F with respect to X and Y.

It seems natural to say that [Fx]delX is that part of delZ due to the change in X, and similarly that the change in Z due to Y is [Fy]delY.

GEMPACK solves non-linear equation systems by repeatedly solving systems such as (2). The changes (delZ) computed at each step are accumulated to give an exact change solution. Similarly, we can cumulate the terms [Fx]delX and so on to obtain contributions Cx and Cy from each exogenous variable. The contributions will add up to the total change in Z.

During the computation, a succession of tiny shocks to delX will cause X to move from its initial value X0 to its final value X1. Similarly Y will move from Y0 to Y1. We can think of the movement of these 2 exogenous variables as a path over a plane, from [X0,Y0] to [X1,Y1].

Clearly, there are many possible paths from [X0,Y0] to [X1,Y1]. Luckily, whichever path we take, as long as individual steps delX and delY are small enough, we will arrive at the correct total change in Z.

It turns out however, for non-linear models, that the values of the individual cumulated contributions Cx and Cy **do depend** [usually rather weakly] on the path followed by X and Y. Of course, the sum of these contributions does not so vary.

The path that GEMPACK chooses for exogenous variables is a **straight line** between initial and final values of exogenous variables. There are some theoretical reasons to suppose that **this path** gives the least biased estimates of Cx and Cy.

Obviously, the decomposition of total changes into those parts attributable to changes in individual exogenous variables must depend on which variables we think to be exogenous. Consider, for example, a simulation with GTAP where tariffs on two commodities, cloth and wheat, were shocked, giving rise to endogenous changes in the two corresponding import volumes. We could decompose the total effect on national welfare between the two tariff shocks.

Now imagine a companion simulation where this time the two tariffs are endogenous, and the corresponding import volumes (quotas) are exogenous and shocked so that they change by just the same amounts as in the previous simulation. We should find that all variables changed by the same amount in both simulations. In this sense, the changes to the tariffs and import quotas are equivalent.

If however we decomposed the total effect on national welfare in the second simulation between the two import volume shocks, we would normally obtain a **different** division of the same change in national welfare. The part of the welfare change due to the change in the cloth tariff in simulation 1 would not be the same as the part of the welfare change due to the change in the cloth quota in simulation 2. The reason for the difference is simply that the exogenous variables are different in the two simulations -- it does not require any non-linearities in the model or differences between the two simulations in the paths followed by variables.

The theory of subtotals is explained more fully in:

CoPS/IMPACT Working Paper Number IP-73

Title: Decomposing Simulation Results with Respect to Exogenous Shocks

Authors: W. Jill Harrison, J. Mark Horridge and K.R. Pearson

Download from: http://www.copsmodels.com/elecpapr/ip-73.htm

3.7 Specifying Subtotals

Specifying Subtotals

Each column of subtotals results shows the effect of just some of the shocks on the endogenous variables of the model. To specify a subtotal, you need to indicate which shocks you wish to see the effect of. You do this by indicating (on the **Specify Subtotal** form) the shocked variables in question.

For example

- suppose that in your simulation you are giving shocks to the variables tms (import tariffs), txs (export tariffs) and a0 (technological changes). You may wish to see the effect of the tariff shocks and also of the technological changes shocks. To do so, you would specify two subtotals. For the first one you would indicate the two variables tms and txs. For the second one you would indicate just the variable a0.
- suppose that in your simulation you are eliminating import tariffs on all of the 3 commodities food, manufactures and services by shocking all components of the variable tms (import tariffs). You may wish to see the effects of removing these tariffs on the individual commodities. If so you would specify 3 subtotals. For the first you would indicate the food component(s) of tms. For the second and third you would indicate the manufacturing and services components of tms respectively.

After you click on the Define Subtotal button on the Shocks page, you see the Specify Subtotal form (which is similar to the Shocks page). You select the variables (and their components, if relevant) which define the shocks making up those in the subtotals result you are interested. Click on the Add to subtotal button to add each variable to the list of variables for this subtotal Once you have selected all the variables whose shocks make up the current subtotal, click on the Ok button. You will then be asked to give the description for this subtotal.

See also More about the theory of subtotals.

Part IV

4 Loading and Saving Experiments

Loading and Saving Experiments

Two buttons on the Solve page allow you to record and recall the details of simulations. The details are stored on disk files (suffixed **EXP**) which contain the current closure, shocks, parameter file, solution method and experiment description. It is intended that you use these facilities to replicate experiments and to exchange them with other modellers.

An EXP file contains:

- full details of a closure
- full details of shocks
- the name of a parameter file
- a solution method
- a one-line description of the experiment

The **Save Experiment** button allows you to store current experiment details on disk. Choose an eight-letter name that has not been used already. Say you choose **FoodTax**. The experiment details will be stored on a file called **FoodTax.EXP** in the subdirectory of the current version. Edit the **Experiment Description** on the Solve page, to make it accurate, before saving the experiment.

The best times to save your experiment are just before or straight after solving the model.

Note that the **current** settings of closure, shocks, parameter file, solution method and experiment description are saved. These might not be the same settings as were used the last time the **Solve** button was clicked. Solve the model again just before saving an experiment if you are in doubt.

The **Load Experiment** button recalls experiment details from the file that you select. Note however that this **EXP** file contains only the instructions needed to run a simulation; it does not contain the solution or other output files generated by a simulation. You have to solve the model again in order to see the results of the experiment.

See also EXP Files, Load and Save File Dialog and Using Version Archives.

4.1 Loading and Saving Closures, Shocks, and Experiments

Loading and Saving Closures, Shocks, and Experiments

RunGTAP uses a specialised file dialog for loading and saving closures, shocks, experiments and versions.

For loading, the dialog displays all files of the appropriate type which apply to the current version. Since all these files are located in the same subdirectory and have the

same extension, both these details are hidden from the user and each file is identified simply by an 8-letter name. There may be a short message describing each file. For experiment files, this message is the experiment description; for versions it is just the first line of the Version Info; for parameter files it is the first line of the FIle History; otherwise the message is simply the first comment (line beginning with !) in the file.

For saving, the user must choose a new 8-letter file name. RunGTAP automatically adds the right extension and places the file in the correct directory. RunGTAP warns you before overwriting existing files: these are listed so you can avoid file names already in use. The Save dialog also gives you a chance to alter the file description before saving it.

4.2 Exporting Experiments to WinGEM

Exporting Experiments to WinGEM

The File..Export to GEMPACK menu command prepares a ZIP archive of all those files needed to replicate the most recent RunGTAP simulation within GEMPACK. These files are:

- GTAP.TAB and GTAP.STI (condensation) from the RunGTAP directory.
- BASEDATA.HAR (flows data) and the HAR parameters file (suffixed PRM).
- The CMF used for the last simulation.
- Any shock files referenced by the CMF.
- BATS.ZIP (Zip within Zip) various BAT, CMF, and INP files used by RunGTAP.

To re-run the simulation you would use the files GTAP.EXE, GTAP.AXS and GTAP.AXT from the RunGTAP directory (or use TABLO to process the files GTAP.TAB and GTAP.STI). You would need to edit the CMF to reflect your new file locations.

If you want to run the post-solution programs such as DECOMP and GTAPVol you will find the necessary executable files in the RunGTAP directory (plus the corresponding TAB files). MAP files are also needed. Look inside BATS.ZIP if you want to see how RunGTAP linked all these jobs together.

4.3 Replicating WinGEM simulations in RunGTAP

Replicating WinGEM simulations in RunGTAP

Within RunGTAP, all versions make use of the same economic model, as defined by the files GTAB.TAB and GTAP.STI in the RunGTAP directory. You must be able to replicate your WinGEM simulations using these files (without changing them at all).

The next question is, whether you can extend an existing version. This will usually be true if your WinGEM experiment uses the same level of aggregation as an existing version. If so, all you have to do is create a new EXP file for that version. You could do this by

starting up RunGTAP and at the same time opening your CMF file in NotePad or a similar editor. Paste **exogenous** and **rest endogenous** statements from the CMF into the RunGTAP Closure page. Similarly, paste **shock** statements from the CMF into the RunGTAP Shocks page. The first line of both Shocks and Closure sections should be an informative comment. Edit the experiment description on the Solve page and solve the model. If and when it works properly, save your experiment as an EXP file. You may also need to add text files of shocks (suffixed PRE) or parameters HAR files (suffixed PRM) to the version subdirectory. XSET and similar statements must be added to the file CMFSTART.

See also Files in the Version subdirectories

If you want to use a new aggregation you will have to create a new version.

Part V

5 Solve

Solve

This page is used for loading and saving experiments and for running simulations.

To run a simulation, (a) choose a different Solution method if necessary, (b) enter in a description of your experiment, and (c) click Solve to start the solution process.

While the model is being solved a small window with a spinning globe appears. Depending on the speed of your machine and the chosen solution method, the model could take between 10 seconds and 3 hours to solve.

It is probably better not to use the machine for other purposes while the model is being solved. However, you can **hide** the solution process if necessary, which will allow you to run other programs. RunGTAP will be minimized to the task bar, but will pop up again automatically when the simulation is completed.

You can also **stop** the solution process altogether (but see Old job still running).

If the solution is successful, you can save results permanently using the File..Save menu command.

See also Errors in Solution Process and Hints for running large models.

5.1 Solution Methods

Solution Methods

RunGTAP offers a variety of different solution methods:

- The SINGLE-STEP SOLUTION, or Johansen method, treats the model as a linear system, linearized around the initial solution. This approach is the simplest and quickest computationally. However, since GTAP is actually a non-linear system, Johansen results are not quite accurate, except for small shocks. The errors are super-proportional to the size of the shock: double the size of the shock, and the size of the errors will more than double.
- MULTI-STEP SOLUTION procedures are used to reduce linearization errors which arise from the default one-step or Johansen solution method. Briefly, the Euler multi-step procedure automatically divides the exogenous shock into a (user-specified) number of equal components. For example, a 10% increase in labour supply might be computed as two successive increases of 4.88% (1.0488 x 1.0488 = 1.1). Results for the first 4.88% instalment are calculated and the database is updated accordingly. Using the new database, results are calculated for the second 4.88% instalment. Since errors are super-proportional to the size of the shock, halving the shock leads to errors at each step which are less than half the size of the error produced by a single, full-size, step. Thus, the results from the two steps may be combined to produce a solution

that is more accurate than that obtained by a single step. The more steps, the more accuracy.

The Gragg and Midpoint methods are variations on the Euler method -- they can sometimes produce more accurate results for a given number of steps.

- SEQUENCE OF SOLUTIONS WITH EXTRAPOLATION -- Early experiments in solving models by the Euler method led to the following observations. The differences between a 8-step and a 16-step solution are often about half those between a 4-step and an 8-step solution. The differences between a 16-step and a 32-step solution are about half those between a 8-step and a 16-step solution, and so on. This rule enables us to predict what results would be generated by a solution with an infinite number of steps -- that is, the exact solution.

  For example, we might choose to run 3 Euler solutions with respectively 4, 8 and 12 steps. GEMPACK automatically uses these results to extrapolate to a solution that is more accurate than any of the 3 individual solutions. For Gragg and Midpoint methods the numbers of steps in a sequence of solutions must be either all odd or all even.

See also:

Automatic Accuracy and Sub-intervals

Accuracy summary

Extrapolation Accuracy (XAC) File Extrapolation Accuracy (XAC) File.

5.2 Extrapolation Accuracy (XAC) File

Extrapolation Accuracy (XAC) File

You can use the XAC file to estimate how accurate individual results are. The report has a complex format, which the GEMPACK documentation explains.

An XAC file will not be produced unless the line:

```
Extrapolation accuracy file = YES ;
```

appears in your CMFSTART file (see below).

Sometimes, apparently poor accuracy is nothing to worry about. Suppose you have three variables, related by the identity:

(A)	Saving	=	Earning -	Spending
Solution 1	2	=	1000001-	999999
Solution 2	-2	=	999999-	1000001

GEMPACK estimates accuracy by comparing two extrapolated solutions (solution 1, solution 2). In this case we might conclude that the results for Earning and Spending were accurate to about 2 parts in a million (5 figures). For Saving we dont even know the sign of the result. But, in the context of Equation (A) and given the relative magnitudes of the

variables, that is no cause for concern (*pace* Micawber). GEMPACK is not able to make use of economic insights or contextual information in reporting accuracy, and so **some** of the problems that it flags **may** be unimportant.

XAC file too large ?

WIth large models the XAC file can be rather large. You can replace the line:

```
Extrapolation accuracy file = YES ;
```

in your CMFSTART file in current version dir, with the line

```
Extrapolation accuracy file = NO ;
```

to have no XAC file.

Or, instead ADD to CMFSTART a line like:

```
XAC-retained %macro (REG)(TRAD_COMM)(TRAD_COMM,REG);
```

meaning that only variables of the dimensions listed will be kept in the XAC file.

See also

Solution methods.

Accuracy summary

Automatic Accuracy and Sub-intervals

5.3 Automatic Accuracy and Sub-intervals

Automatic Accuracy and Sub-intervals

See also solution methods.

GEMPACKs extrapolation solution methods produce an error bound for each variable. These bounds are conservative estimates of computation error -- they are not related to uncertainties in the setting of shocks or parameters. Error estimates are only available if you select a 3-solution method (eg, Euler 2-4-6 instead of Euler 2-4). They are listed in the (optional) XAC file.

What if you decided that your Gragg 2-4-6 solution was not accurate enough? You could get more accuracy by running a 4-8-12 step Gragg extrapolation.

However, it turns out that a more accurate solution could usually be produced as follows:

First, divide all the shocks into two equal instalments. Then use Gragg 2-4-6 to compute the effects of the first instalment. This produces an updated database, which can form the starting point for another Gragg 2-4-6 simulation to compute the effects of the second instalment. The total result is found by compounding the change results of both simulations. It would all take about as long as a 4-8-12 step Gragg extrapolation, but normally would be more accurate.

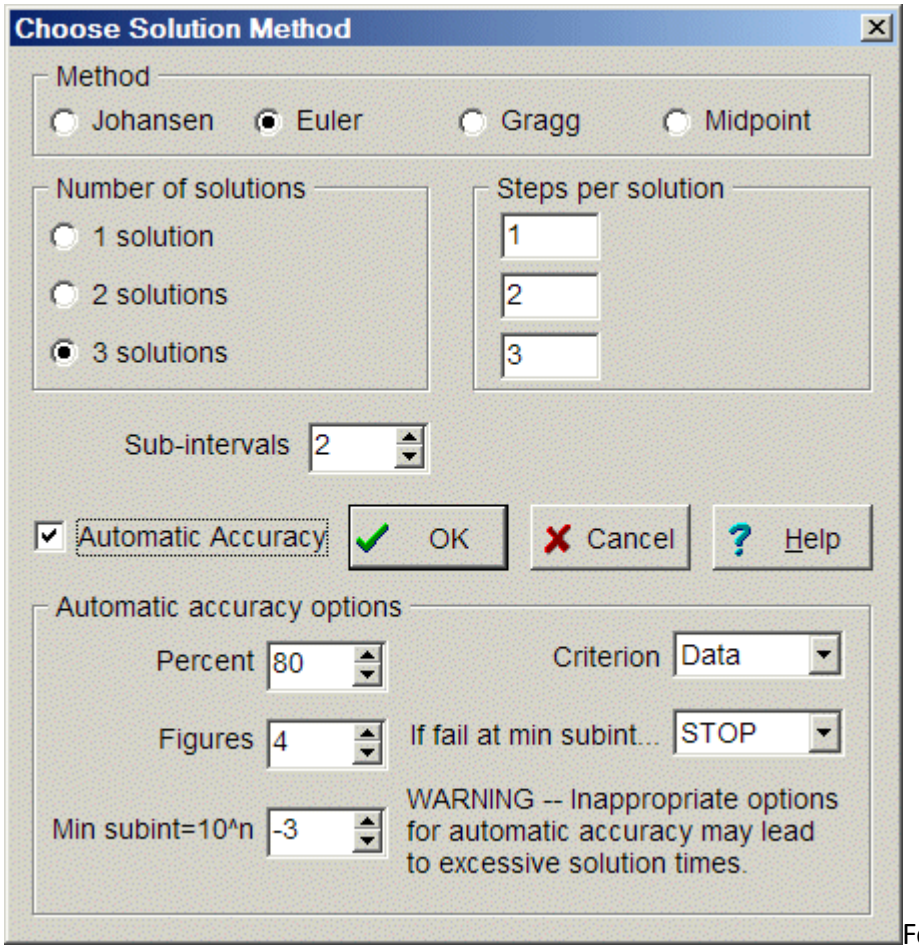
In GEMPACK language we describe this procedure by saying that the computation was broken into 2 ***sub-intervals***.

Note that in the example above we would obtain 2 sets of error estimates, one for each sub-interval. We might decide that the first sub-interval gave results that were accurate enough, but that the second sub-interval gave inaccurate results.

In that case we could split the second sub-interval into two halves, each with its own Gragg 2-4-6 computation. We would then have divided the entire computation into three sub-intervals, where the first sub-interval was twice as big as the other two. We could go on dividing up sub-intervals until we were satisfied with the accuracy (or until we got tired).

GEMPACK's ***automatic accuracy*** feature automates all these procedures. It decides, during the computation, how many sub-intervals are needed to achieve a user-specified level of accuracy.

If you switch on automatic accuracy from the **Choose Solution Method** dialog, additional options appear which allow you to specify how accurate a solution you want. See below:



For example you might ask for 80 **percent** of the updated data to be accurate to 4 **figures**.

In assessing accuracy, it makes a difference whether we look at updated data, or at results. By default, GEMPACK uses the data **criterion**, since percentage changes might refer to an infinitesimal base, and so be irrelevant.

Normally GEMPACK starts with two sub-intervals and increases the number until sufficient accuracy is attained. If your experience tells you that 6 **sub-intervals** are needed, you can speed up solution time by making 6 the starting point.

Some mis-specified models never converge! This could cause GEMPACK to go on forever dividing up sub-intervals into smaller parts, without ever getting accurate results. The **Min**imum **subint**erval option prevents this happening. By default it is set to -6, meaning that no subinterval smaller than one millionth (=10E-6) will be tried.

The **If fail at min int** option controls whether computations stop or continue on, if the minimum sub-interval is reached before results are sufficiently accurate.

See also:

Solution methods.

Accuracy summary

Extrapolation Accuracy (XAC) File

5.4 Accuracy Summary

Accuracy Summary

If you choose a solution method with extrapolation, RunGTAP displays an accuracy summary showing how many variables are accurate in the levels to 1, 2, 3, 4, 5, or 6 figures. A modified Chernoff statistic displays overall accuracy. Accuracy of the updated data is summarised in the same way.

GEMPACK estimates a rather conservative lower bound on accuracy. In most cases results will be considerably more accurate than is indicated by the summary.

There is one summary for accuracy of variables, another for accuracy of data on file.

If you used automatic accuracy to break your simulation into several sub-intervals there will be an accuracy summary for each subinterval. Use the combobox to display which you want to see. The combo box also indicates which is the least accurate sub-interval. With GEMPACK 8 or later there is also an overall summary for variables (but not data) which allows for the accumulation of errors in all the sub-intervals.

Acccuracy is measured by an ordinal score 0...10. For each score value there is a corresponding face. Click-and-hold on the little key at the bottom of the window to see the whole range of faces.

The View menu item allows you to look again at the accuracy summary later, if it still exists (it is deleted if you change versions or load another experiment.

See also:

Solution methods.

Automatic Accuracy and Sub-intervals

Extrapolation Accuracy (XAC) File

5.5 Running a Test Simulation

Running a Test Simulation

Use the menu command **Tools..Run Test Simulation** to run a simple simulation with the current version. You might do this to check that RunGTAP is working properly, or to force RunGTAP to regenerate missing files.

See also missing SLI file.

5.6 The SAVESIMS folder

The SAVESIMS folder

Each time you successfully run an experiment, RunGTAP stores some results from that experiment in a subdirectory of the current version folder called SAVESIMS. The files have the same name as your experiment: they are:

- the CMF file used to run the main simulation,
- the UPD (update) file from the main simulation,
- the LOG file from the main simulation,
- the SL4 and SLC files,
- an EXP file, generated at solve time from the contents of the Closure, Shocks and Solve pages. [new May 2014]

These files might be useful for debugging purposes, or in case you want to examine several solutions at once using ViewSOL. You can delete the contents of the SAVESIMS folder to save disk space, if necessary.

Part VI

6 Results

Results

This page allows you to inspect the variable values computed during the last successful solution process. Initially, a Contents window is displayed, listing the model variables. There are two ways to examine a particular item:

(a) simply click twice on the item,

(b) click or arrow keys to select the item; then press the space bar.

The Contents window will be replaced by the **Numbers** window, showing numerical values. Exogenous variables are shown in red. You can click on variable names that appear as row or column labels to see a longer description in the status bar at the bottom of the screen.

To return from the Numbers window to the Contents window, click twice on any numerical result.

Using the Copy menu command, either the Contents window or the Numbers window may be copied to the clipboard. Each window can also be printed (using File..Print). You can also adjust the size of the font used to display results.

See also Filtering for particular set elements.

Advanced topic: Usually RunGTAP is set up to show only the more interesting variables on the results page (only the variable listed in the GTAP.MAP file). You can see **all** the variables by using ViewSOL or by by switching OFF the option Use map file in the **Tools...**Options dialog.

6.1 The Contents Window

The Contents Window

The Contents window on the Results page lists all the variables that may be viewed. If filtering is in effect, only variables containing the filter element are shown. Scalar (macro) variables appear first as one item, followed by the other variables in alphabetical order.

The little button marked V at top left allows you to arrange the contents list in 2 ways:

- vectors listed individually, or
- vectors ranging over the same set grouped together.

The second option allows you to display a number of vectors together in the Numbers window, which can be very handy.

6.2 Copying Results

Copying Results

The Copy menu command causes the current window (either Contents or Numbers) to be copied to the Windows Clipboard. From there it can be pasted to various programs. For example, you can paste directly into an Excel spreadsheet. You can also paste into Word. In the latter case, a series of items separated by tabs appears in your document; select them all and use Words Convert Text to Table command.

When copying real matrices to a spreadsheet, remember that the data copied is only as accurate as that displayed on screen. You may want to increase the number of decimal places before copying.

6.3 Adjusting the look of the Results Page

Adjusting the look of the Results Page

The six small buttons at the bottom right of the results page can be used to either increase or decrease:

- the font size used,
- the row spacing,
- the column widths.

Adjust them to suit your own taste.

6.4 Filtering for particular set elements

Filtering for particular set elements

A combo box at top left of the Numbers window allows you to select just one set element for display. Supposed your solution file contained vectors and matrices, dimensioned over sets some of which contained the element Agriculture. You could ask to see just the Agriculture results.

The effect of filtering in this way is that:

- macro variables are not shown,
- each vector that contains Agriculture turns into a scalar,
- each matrix turns into 1 or 2 vectors (2 if Agriculture is both a row and a column label), and
- each 3-D array turns into 1, 2 or 3 matrices.

The items in the combo box are all the elements of every set. Choose the first item, Everything, to switch filtering off.

6.5 Cannot View a 3-D matrix

Cannot View a 3-D matrix

The Results page does not allow you to see a whole 3-D matrix at once. However, you can view a 2-dimensional slice of one, by filtering.

6.6 Number of Decimal Places

Number of Decimal Places

You can select from 0 to 6 decimal places, from a combo box at the top of the Numbers window of the Results page. Numbers that are exactly zero appear as a single digit. Exogenous numbers appear in red.

6.7 Copying to ClipBoard

Copying to ClipBoard

The Copy menu command causes the current window to be copied to the Windows Clipboard. Many other programs can paste in data from there.

See also Copying Results.

6.8 Printing

Printing

RunGTAP allows you to print out most of what you can view. For results, only those currently being viewed are printed, so to print ALL the results you would have to select them in turn and print them.

Use the File..Print Setup menu command to change printers or to change from portrait to landscape paper orientation. Printed output will be in 9 point Courier.

6.9 Saving Output Files

Saving Output Files

The File..Save Outputs submenu allows you to save various files produced by the most recent successful simulation. The files are:

- Solution file (SL4)
- XAC file

- Updated Core data.
- Updated GTAPView output.
- Updated Tax Rates output.
- DECOMP output.
- GTAPVol output.

All or some of the above packed into one ZIP file. You can choose which are to be included.

The SL4 and CMF files would only be useful in conjunction with the complete GEMPACK system.

6.10 CMF file

CMF file

The CMF file is a text file used by GEMPACK to specify the details of a simulations (closure, shocks, and names of input and output files). The shock statements that you construct on the Shocks page, and the list of exogenous variables on the Closure page form the main part of a CMF file that RunGTAP creates each time you solve the model.

The CMF file might be useful as a record of your simulation.

6.11 SL4 files

SL4 files

SL4 files are used by GEMPACK to store solutions. You can use the ViewSOL or GEMPIE programs (if you have them) to view the contents of SL4 files. In some cases, these programs may be more convenient for viewing or printing results than the simpler facilities built into RunGTAP.

6.12 Examining Results using ViewSOL or ViewHAR

Examining Results using ViewSOL or ViewHAR

Experienced GEMPACK users may prefer to use the ViewSOL program to examine results. This option is available from the **View** menu only after simulation is complete. The ViewSOL interface is similar to, but more complicated than, the Results Page in RunGTAP.

One advantage of ViewSOL is that it allows you to compare results from several simulations side by side. You can do this by opening each solution using **View.. Results using ViewSOL** as soon as RunGTAP completes each simulation. Results from previous experiments may be found in the SAVESIMS folder.

You can also view the solution file using ViewHAR. Its only advantage is that, unlike in ViewSOL, you can see values for variables with 4 or more dimensions. See ViewHAR online help topics *Whats New* or *Opening SL4 solution files* for more details.

6.13 Interpreting Results and Data using AnalyseGE

Interpreting Results and Data using AnalyseGE

AnalyseGE is a program which lets you simultaneously view equations, data, results, and expressions computed from data and results. It is available from the **View** menu only after simulation is complete and if you have a GEMPACK licence. It has its own online help.

AnalyseGE requires a solution file (SL4) and another file with the same name but with the SLC extension. The SLC files are produced automatically by TABLO-generated programs made with GEMPACK Version 7.0 and later.

If you save an SL4 file with **File..Save Outputs** the matching SLC is also saved.

Part VII

7 Viewing the Database and Other Files

Viewing the Database and Other Files

The View menu item allows you to examine a number of files used or produced by RunGTAP:

The **Base data**, showing the data base prior to the experiment. This is divided into:

- The Core data file of flows.
- GTAPView output.
- Tax Rates output.
- Both general and partial equilibrium elasticities.

Updated data, showing values after the last experiment. This is available only after a simulation has been completed. It is divided into:

- The Core data file of flows.
- GTAPView output.
- Tax Rates output.
- DECOMP output.
- GTAPVol output.

As well there is:

- The Sets file and also Subsets and Variables information.
- The Parameters file.
- The TAB files.
- The main Log file (available only after a simulation has been completed).
- Accuracy Summary (available only after some simulations have been completed).
- The CMFSTART file.
- An arbitrary text file.
- Results using ViewSOL or ViewHAR (available only after simulation is complete).

Most of these files (except TAB, Log and Text) are in the special Header Array format used by GEMPACK. If you choose to look at them, RunGTAP will launch the GEMPACK program ViewHAR in its own window.

The text files may be viewed in the GEMPACK editor TABmate, or in a simpler text file viewer which is part of RunGTAP. It allows you to modify the parameters file and the

arbitrary text file; otherwise it is read only (unless you are in Developer mode). See also using TABmate as your text editor.

7.1 The Core Data File of Flows

The Core Data File of Flows

This is the main data file used by the GTAP model. It consists mainly of flows (transaction values) valued at various prices. These flows describe the initial (pre-simulation) economy.

After a simulation a second, **updated**, core data file is produced, depicting the post-simulation economy.

Both initial and core data files lack various subtotals that you might want to see. GTAPView calculates these for you.

7.2 The Parameters File

The Parameters File

Contains the various elasticities and other parameters used by GTAP. When you start RunGTAP, or change to a new version, a default parameters file will be used, probably called DEFAULT.PRM. The Change Parameter File button on the Solve page allows you to load another parameters file instead (if there is another).

The View..Parameters menu item allows you to examine and edit the current parameters file. If you change it you can save it with a new name. Be careful to adhere to the original format, and to alter the first line of the file history to a 1-line summary of your changes.

Parameters files must have the suffix (extension) PRM.

7.3 The Sets File

The Sets File

Contains the various sets used by GTAP, mainly groups and subgroups of commodities and primary factors. See also Subsets and Variables.

7.4 Subsets and Variables

Subsets and Variables

The **View..Subsets and Variables** dialog has three pages -- **Sets**, **Variables**, and **Coefficients**.

The **Sets** page shows on the left a list of all the sets in the model (including any XSETS). Click on the left-hand list to see details of the selected set, in particular:

- set elements (members),
- other sets which are subsets of the selected set; and
- supersets -- sets of which the selected set is a subset.

The **Variables** page simply shows a list of all the variables, with dimensions and descriptions.

The **Coefficients** page simply shows a list of all the coefficients, with dimensions and descriptions.

Each page has a **Copy** button: useful for pasting data into a spreadsheet.

7.5 CMFSTART file

CMFSTART file

In each version folder there is a file called CMFSTART, which contains some additional instructions which are sent to GEMPACK prior to solving the model.

Use **View...CMFSTART file** to view or edit this file. When a new version is created, the CMFSTART file is created as a copy of a file called CMFSTART.DEF in the main RunGTAP folder.

CMFSTART.DEF contains the following lines:

```
CPU = yes; ! log show simulation times
NDS = yes; ! no displays
! servants=1; ! use 2 processors at once, if possible
Extrapolation accuracy file = NO ; ! No XAC file
```

If you want to produce an Extrapolation accuracy (XAC) file (but beware, the XAC file can be very large) you would change the line above to:

```
Extrapolation accuracy file = YES; ! Make XAC file
```

Characters following an exclamation mark are treated as a comment (ie, ignored). So the line:

```
! servants=1; ! use 2 processors at once, if possible
```

is inactive. You could remove the first exclamation mark to activate it then GEMPACK would try to speed up the solution by parallelizing the computation: see Using parallel processing.

A main use of the CMFSTART file is to create new sets which you can use in shock statements: see XSETS and the CMFSTART file.

Conceivably, you could add other types of statements to the CMFSTART file -- assuming you have a deep and long-standing knowledge of the GEMPACK system. The contents of that file become the first part of the CMF file sent to GTAP.EXE each time a simulation is run.

One rash fellow added the following 2 lines to CMFSTART:

```
file gtapDATA2 = VersionDir\basedat2.har; ! in version dir
Updated file gtapDATA2 = gdat2.upd; ! in work dir
```

His intention was to use an extra input data file for a specially modified GTAP model. Even though this contradicts the clear instructions given in Notes for Module Developers, the above lines seemed to work!

Indeed RunGTAP seems to encourage such behaviour: if you include the string VersionDir in a line of CMFSTART, RunGTAP replaces that string with the name of the version folder. Similarly, HomeDir is replaced with the name of the RunGTAP folder.

If you keep changing the CMFSTART file between simulations in strange ways, your EXP experiment files might no longer be capable of reproducing previous simulations -- when you run a simulation, the **current** CMFSTART is used. There are 2 ways you might be able to recover previous versions of CMFSTART. First, when you save an EXP file, the CMFSTART file is saved within it as a comment. Second, after a simulation the CMF file (which contains CMFSTART) is stored in the SAVESIMS subdirectory of the current version folder.

7.6 XSETS and the CMFSTART file

XSETS and the CMFSTART file

A number of sets and subsets are declared in the file GTAP.TAB, which defines the GTAP model. The actual membership of these sets, which depends upon aggregation level, is given by the version-specific file SETS.HAR.

It can sometimes be very handy, especially in devising shock statements, to define your own additional sets. For example, you might like to define a set of agricultural commodities or food-importing regions.

You can persuade GEMPACK to let you do this by adding pairs of XSET and XSUBSET statements to the CMFSTART file which is unique to each version [use **View...CMFSTART file**]. For example, the CMFSTART file for the version ASA7x5 contains the lines:

```
XSET AG_COMM # Agric comms # (agric, food) ;
XSUBSET AG_COMM is subset of TRAD_COMM ;
XSET NONAG_COMM = TRAD_COMM - AG_COMM ;
```

Your new sets will not be visible to RunGTAP until after you next solve the model. Then, you should be able to **View..**Subsets and Variables to see the new sets listed.

See also CMFSTART file

7.7 Using parallel processing

Using parallel processing

Most PCs have more than one processor. In this case it MAY be advantageous and possible to use parallel processing to speed up your GEMPACK computation. The conditions are:

- Your PC must have more than one processor (eg, a dual-core CPU) or 2 separate CPUs.
- You must have enough RAM to run 2 simulations at once (if not, both simulations will run very slowly).
- You must be using a multiple simulation solution method, eg 3-5-7 Gragg, or 2-3-4 Euler. A single simulation (eg, 3-step Euler) cannot be parallelized.
- Your TABLO-generated EXE file (eg, GTAP.EXE) must be produced using GEMPACK version 10 or later.
- Your RunGTAP.EXE was produced June 2006 or later (use **Help...About** to check this).

First, there is no point in using 2 CPUs unless your simulation takes a minute or more to run.

To activate parallel processing add the line (ending in semicolon):

```
servants=1;
```

to your CMFSTART file. If you have a quad-core cpu, you could try servants=2.

7.8 The Log file

The Log file

This file records the process of solving the model. If there is an error during the solution process, the last few hundred lines of the log file are displayed automatically, but even if there is no error you can use the View..Log File menu item to see the whole log file. Search for %% to find warnings.

If you choose a solution method that uses extrapolation an accuracy report will appear at the end of the log file.

7.9 Derived Elasticities (Partial and General)

Derived Elasticities (Partial and General)

The behavioural elasticities (such as demand elasticities) used by GTAP are derived from the data and parameter files stored in each version. Depending on what is imagined to be held constant, either partial equilibrium elasticities or general equilibrium elasticities

may be defined. Each may be useful in explaining results, so RunGTAP calculates and stores on file both sets of values for you to look at.

Both sets of elasticities are a function of the base flows data and of the currently selected parameters file. By default, RunGTAP recalculates each whenever the file is missing or seems older than the files from it is derived. For each parameter file (suffixed PRM) there should be a matching file of partial elasticities (suffixed PEL) and a matching file of general equilibrium elasticities (suffixed GEL). Both are HAR files which you can examine with ViewHAR.

7.9.1 GE-Elast computations

GE-Elast computations

RunGTAP calculates or re-calculates general equilibrium elasticities (using the base flows data and the currently selected parameters file) whenever it seems necessary. The process can be quite time consuming and requires a GEMPACK license (SAGEM is used) so it is switched off by default.

Use the **Version..Modules** menu item to switch on automatic calculation of general equilibrium elasticities.

See also Derived Elasticities

For theory of the general equilibrium elasticities see: Derivation and Interpretation

For how to use the GE-Elast output file see: Use of the GEELAST header array file.

7.9.2 General Equilibrium Elasticities

General Equilibrium Elasticities

See also: GE-Elast computations

Derivation and Interpretation

One of the big problems in general equilibrium analysis stems from the fact that firms do not face a single demand schedule, from which we can read off the demand elasticity for a given product. Rather, the typical firm/industry sells to many different agents in the economy. Some of these are domestic firms, some are domestic households and some consist of overseas buyers of exported goods. Thus the demand elasticity faced by a sector in the GTAP model is a composite of many different agents responses to a price change. This is still manageable analytically, in the sense that the market demand elasticity can be computed as a share-weighted average of the individual agents responses. For example, if 10% of steel output is sold to the domestic automobile industry and that industrys conditional, own-price elasticity of demand for steel is 0.10, then the contribution of the auto industry to steels total, partial equilibrium demand elasticity would be 0.10 * 0.10 = 0.01.

However, in general equilibrium, automobile firms response to an increase in the price of steel will be more complicated. In particular, this will tend to force the price of automobiles up. As a result, the total number of autos sold is expected to decline. This output effect must be added to the substitution effect captured by the conditional price elasticity of demand to get the total effect. However, the story does not stop there. Because the increase in the price of steel raises the price of other products, some of which may themselves be substitutes for steel, one must also take into account cross-price effects. Finally, income in the economy may be affected, in which case income effects will also come into play. The only way to capture all of these components of the general equilibrium demand elasticity is via model simulation.

Simulation of the general equilibrium elasticities of demand proceeds as follows. The output tax on steel is perturbed by 1 percent. This in turn causes the market price of steel to rise by something less than 1 percent, say 0.6% (the remainder is borne by producers in the form of a decline in the agents supply price). We then observe the reduction in market demand, which might be 0.9%. In this case, the general equilibrium elasticity of demand for steel in the model would be -0.9%/0.6% = -1.5%. But how much of this is attributable to different agents? To answer this question, we need to observe the change in demand by all of the different agents in the model. Auto producers might show a reduction in steel demand of 0.08%. We then divide this by the market price change to find the auto sectors GE, own-price elasticity of demand for steel = -0.08%/0.6% = -.13. If we then multiply this elasticity by the auto sectors share of total steel demand we obtain their contribution to the total GE demand elasticity for steel = 0.1 * -0.13 = -0.013. For a small perturbation, the share-weighted agents responses will add up to the total GE demand elasticity.

Now read: Use of the GEELAST header array file

7.9.3 Use of the GEELAST header array file

Use of the GEELAST header array file

Chapter 13 of the GTAP book is an example of how the GE demand elasticity information might be used to facilitate analysis. In this chapter, George Frisvold examines the global incidence of technological progress in agriculture and food production. One of his findings is that an increase in total factor productivity in Australian agricultural crops benefits Australian farmers while the reverse is true in North America (Table 13.3). Upon further reflection, he attributes this to the fact that the general equilibrium elasticity of demand for Australian crops is elastic, while this same parameter is inelastic in the case of North America. Thus an increase in output in North America is more than offset by the resulting decline in price and overall revenue falls. This translates into a decline in land rents, which is the measure of farmer welfare used by Frisvold. Hence the importance of the GE demand elasticity in economic analysis.

But why is the demand for crops elastic in Australia, but not in North America? The GEELAST program is designed to permit users to quickly decompose the GE demand

elasticity into its component parts, thereby facilitating explanation of the results in terms of fundamentals. The general method is one of beginning with a summary array: **O**, which simply reports the GE own-price elasticity of demand by commodity and region. From this array we can quickly verify that the Frisvolds assertion is correct: NAMs elasticity is -0.75 versus -1.22 for Australia/New Zealand. (Note that these figures are different from those in Table 13.2 of the GTAP book. This is because the figures in Table 13.2 were based on GTAP version 1 data and parameters, whereas all of the results are based on version 2 data and parameters.)

The subsequent arrays with the prefix **O** decompose the aggregate own-price elasticity of demand. **O1**, shows how much is derived from domestic agents versus exports. Here, we see that domestic agents contribute less to the ANZ elasticity than to the NAM elasticity (-.13 vs. -.22), and a further breakdown by firms and households in **O11** show this to be true across the board. (**O111** breaks the domestic firms demands down by sector.) On the other hand, from **O1** we can see that export demands are sufficient to make ANZ crop demand elastic, since the qxm entry is equal to -1.1, vs. -.54 for NAM. **O2** breaks the export demand down by destination region. Selecting CROPS and ALL REG destinations, then requesting a row share from VIEWHAR we see that East Asia accounts for 40% of ANZs export demand elasticity. This is followed by ROW and SE Asia.

Having rather exhaustively broken down the sectoral demand elasticity, we now turn to the task of explaining why the export demand elasticitys contribution is larger in the case of ANZ. Is this because exports are more important for ANZ than for NAM? Or is it because the export demand elasticity faced by ANZ exporters is larger? To answer this question we need to decompose each agents contribution into a share and a behavioural response. The next set of headers, with prefix **S**, do precisely this. Thus selecting **S1**: ALLCOMP/qxm/CROPS/AllREG, we can compare the export share of total output in NAM (.21) to that in ANZ (0.30). We can also compare the average behavioural response (own-price elasticity) on the part of export markets, to the two regions price changes. Here we see again that the ANZ figure is larger: -2.57 for NAM vs. -3.67 for ANZ. The prod entry simply reports the product of the share and the elasticity, which is what was reported previously in header **O1**.

Thus we can conclude that the large contribution of exports to the GE elasticity of demand for ANZ crops comes from two factors. First, export demands are generally more elastic than domestic demands, and ANZ has a larger share of exports in total output. Secondly, since ANZ is, on average, smaller in the export markets, the own-price elasticity of demand facing its goods in export markets is larger. (The reader will need to review the Armington equations in the model to fully grasp the second point. Since the elasticity of substitution among imports from different sources is the same for all regions, the import shares are the sole cause of differing import demand elasticities facing different exporters.)

The other headers with the **S** prefix permit one to decompose the domestic and export demand elasticities into their share and elasticity components. In both cases, the first thing to do is to specialise the last two drop down boxes (which initially are Sum TRAD_COMM/Sum REG) to the commodity/region in question eg., crops/ANZ).

The last set of headers in this file permit one to obtain cross-price elasticities as well. **X1** reports the cross-price elasticities ETA(i,r,j,s) = q(i,r,j,s)/p(j,s), ie. the quantity response of the (i,r) pair to a change in the price of good j in s. For example, selecting the drop-down box combinations: crops/AustrNZ/crops/All REG, we can observe the change in demand for AustrNZ crops when crops prices in other regions rise. For example, when NAM crops prices rise by one percent, the model predicts an increase in demand for ANZ crops of 0.27%. If crops prices rise worldwide by 1%, then the model predicts a decline in ANZ crop demand of -0.217%, which is simply the column sum in this array.

Finally, **X11** permits us to decompose these cross-price elasticities by domestic and export changes. Selecting: crops/AustrNZ/crops/All REG/All QOCOMP reveals that the -0.217 cross price elasticity of demand for AustrNZ crops with respect to the NAM crops price comes almost entirely from the export market (0.25 of the total 0.27 elasticity).

7.9.4 Partial Equilibrium Elasticities

Partial Equilibrium Elasticities

The PEELAST TABLO program calculates conventional demand elasticities from the values contained in the parameters file and from the initial flows data. The results are contained in files suffixed PEL in each version directory.

7.10 The RunGTAP Text File Editor

The RunGTAP Text File Editor

RunGTAP allows you to view (and sometimes change) a variety of files. By default, text files are viewed using the RunGTAP editor. While you are in the RunGTAP editor, the remainder of RunGTAP is disabled: you must finish editing (Back to RunGTAP) before resuming other work.

You can choose to use TABmate as your text editor.

The editor is in read-only mode, unless:

(a) you chose the View..Other text file option, or

(b) you are in Developer mode.

If you do modify and save a file, remember that RunGTAP is fussy about file extensions (suffixes). For example, RunGTAP only recognises a TABLO input file if its name ends in .TAB. You should normally keep the same extension, even if you save a file with a new name. File names may be limited to 8 letters, plus a 3-letter extension.

7.11 Choosing your text editor

Choosing your text editor

The **Tools...Options** command allows you to choose your text editor, from:

- the default or built-in editor (fairly primitive)
- the TABmate editor
- another favourite editor.

You can specify the filenames (location) of the last two. We recommend that you choose the TABmate editor.

7.12 ViewHAR

ViewHAR

RunGTAP uses the GEMPACK program ViewHAR for viewing various header array files used or produced by the GTAP model.

The ViewHAR interface is similar to that used by the RunGTAP results page. ViewHAR has its own online Help system, which you should consult. You may see there that ViewHAR has the capacity to modify files; however this option is disabled when it is launched from RunGTAP,unless you are in Developer mode and have a GEMPACK Licence.

Unlike the other parts of RunGTAP, ViewHAR is a separate program that you can leave open while you return to other tasks. You might have several copies of ViewHAR open at the same time. It is your responsibility to keep track of these and close them when not required.

7.13 Header Array Files (HAR files)

Header Array Files (HAR files)

Data for economic models (for example, input-output tables) are often held within GEMPACK on files called Header Array or HAR files. Header Array files contain one or more arrays each containing data values. An individual array of data on a Header Array file is accessed by supplying the unique 4-character identifier (or Header) for that array of values.

The data values held by an individual array can be either all real numbers, all integer numbers or all character strings. Depending on the type of data that is to be stored, the number of dimensions allowed varies.

Headers for arrays on any one file must be unique since the header acts as a label or primary key to identify the associated array.

You can learn more about HAR files from the ViewHAR online help.

7.14 TAB File

TAB File

TAB file is short for TABLO input file. The main TAB file is used by GEMPACK to specify the equations and variables of the GTAP model. It is written in the TABLO language, native tongue of GE modellers.

You can also view the TAB files which specify the DECOMP, GTAPView, GTAPVol and SHOCKS programs.

RunGTAP allows you to consult TAB files, but does not encourage you to modify them. You must close the file viewer before going back to RunGTAP.

7.15 GTAPView

GTAPView

The GTAPView program uses the core data file to compute various subtotals and ratios which it may be useful to know, especially for understanding and explaining results.

GTAPView can read either the original core data file or the updated core data file. The updated GTAPView output summarises the database after the last simulation.

GTAPView also calculates Tax Rates files.

7.16 Tax Rates files

Tax Rates files

The GTAPView program uses the core data file to compute various tax rates which you should know, especially for understanding and explaining results.

You can look at these files using ViewHAR. The tax rates are **percentages off the base**, for example:

If the purchaser pays $13, of which tax is $3, the file shows a rate of 30%.

GTAPView computes rates for both the original core data file or the updated core data file. The latter relates to the database after the last simulation.

7.17 GTAPVol

GTAPVol

The main results of a GTAP simulation are stored in two files:

- The solution file (SL4). You can see the contents of this on the Results page. It consists mainly of percentage changes in variables.
- The Updated Core data. This consists mainly of flows (transactions value).

The GTAPVol program combines and supplements these results by computing various **volume** changes. The volume changes are measured using initial market prices (IMP). These might be useful for understanding and explaining results.

7.18 DECOMP

DECOMP

The DECOMP program uses simulation results to break down the welfare changes in each region into a large number of components, thus exposing the sources of welfare change.

7.19 The Shocks Program

The Shocks Program

The SHOCKS program calculates from the GTAP database what percentage changes in the powers of various taxes would be required to abolish those taxes. The values of the changes are stored on files that can be used to calculate shocks for GTAP simulations.

See also Regenerating standard shock files

7.20 Regenerating standard shock files

Regenerating standard shock files

Each version sub-directory should contain files showing what percentage changes in the powers of various taxes would be required to abolish those taxes. The files are called TO.SHK, TMS.SHK, and so on. RunGTAP re-generates these files automatically during the Solve process if they are missing or obsolete (perhaps because the underlying flows database has changed).

Part VIII

8 Versions of the GTAP model

Versions of the GTAP model

RunGTAP allows you to choose between several different versions of GTAP. Switch between versions with the **Version..Change** menu command.

Most of the versions use the same underlying GTAP model (as specified in the TAB file) and so share the same theoretical specification. The main difference between these versions is the number of goods and countries. Other differences might include the parameter settings, or any specially prepared files of shocks, closures, or experiments.

A few versions use their own special variants of the GTAP model, and so have their own special TAB and associated files. The **Version..Modules** menu command displays which TAB files are used for the current version.

The Version page should contain information about the current version.

Each version corresponds to a subdirectory of the RunGTAP directory. The subdirectory has the same name as the version and contains a collection of files that make up the version.

The **Version** menu item also allows you to Delete or create New versions.

8.1 Files in the Version subdirectories

Files in the Version subdirectories

Each version corresponds to a subdirectory of the RunGTAP directory. The subdirectory has the same name as the version and contains a collection of files that make up the version. RunGTAP offers you various ways to create files in the version subdirectory, but provides no way to delete such files (unless you use **Version..Delete** -- which removes the whole version). You would have to close RunGTAP and then use Explorer, File Manager, or the DOS command line to delete individual files. The purpose of this section is to describe these files in general terms and in particular to list the files that you must **not** delete.

Files which you should not delete

VERSION.TXT contains the text on the Version Info page. If this file is missing RunGTAP will not see the version subdirectory.

BASEDATA.HAR contains the base flows data

SETS.HAR contains the sets

DEFAULT.PRM is the default parameters file. Like the preceding two files it is in HAR format.

An EXP file: there must be one file suffixed **EXP**; if there are several, RunGTAP will look for one it used before or one called DEFAULT.EXP -- if it doesnt find that it picks by alphabetical order.

VERSION.SLI file is needed for the assisted method of creating shocks.

BASEVIEW.HAR and BASERATE.HAR are the base data GTAPView outputs

CMFSTART contains information that RunGTAP transmits to GTAP.EXE

Be careful about deleting files with the following extensions (suffixes):

SHK and **PRE** files contain shock values

PRM files are HAR files of parameters.

These three file types may be referenced by EXP files. In addition RunGTAP expects that there will be an **SHK** file for most tax variables. RunGTAP attempts to regenerate all the SHK files if any are missing or obsolete.

You can delete the following file types:

PEL and **GEL** files contain the partial and general equilibrium elasticities derived from the corresponding PRM file. RunGTAP attempts to regenerate them if they are missing or obsolete.

SHF files are specifications of complete sets of shocks.

EXP files contain data needed to recreate experiments (NB, leave at least one file suffixed **EXP**).

CLS files are used to store closures.

See also Derived files in version subdirectories

8.2 Derived files in version subdirectories

Derived files in version subdirectories

Some of the files stored in version subdirectories are derivative -- they can be regenerated by RunGTAP from other, more basic files.

For example, the file BASERATE.HAR, which shows initial tax rates, is generated by the GTAPView program from the core flows data file BASEDATA.HAR. Hence, if either of the files GTAPView.EXE or BASEDATA.HAR is changed, BASERATE.HAR should be regenerated.

The file VERSION.SLI in each version subdirectory is the most important derivative file. It contains information about the names and dimensions of sets and variables, and depends on the files GTAP.EXE and SETS.HAR. If either of these change, RunGTAP deletes VERSION.SLI and issues a warning.

RunGTAP contains three mechanisms to assist in keeping derivative files up-to-date.

- When RunGTAP starts up, or whenever you load a new version, RunGTAP scans the current version and deletes obsolete files.
- Whenever the model is solved, any missing or obsolete derivative files are regenerated.
- In Developer mode, an independent checking routine reports missing or obsolete files for the current or all versions.

Normally, it is easy to run a simple Johansen simulation to regenerate any missing or obsolete derived files. See Run test simulation. However, this will not be possible for the user who is not able to change or create files in the version subdirectories -- see Check Versions.

List of Derived Files in Version Subdirectories

BASEVIEW.HAR (data summary) produced by GTAPView.EXE from BASEDATA.HAR.

BASERATE.HAR (tax summary) produced by GTAPView.EXE from BASEDATA.HAR.

VERSION.SLI produced by GTAP.EXE from SETS.HAR and CMFSTART (for XSETS).

Each parameter file (suffixed PRM) should give rise to the following two files:

???.PEL (partial equilibrium elasticities) produced by PEElast.EXE from ???.PRM and BASEDATA.HAR.

???.GEL (general equilibrium elasticities) produced by SAGEM and Pricelas.EXE from ???.PRM and BASEDATA.HAR.

Files of standard tax shocks, TO.SHK, TMS.SHK, TGI.SHK, etc, produced by SHOCKS.EXE from BASEDATA.HAR

8.3 Using Version Archives

Using Version Archives

Each version corresponds to a subdirectory of the RunGTAP directory. The subdirectory has the same name as the version and contains a collection of files that make up the version. RunGTAP allows you to store the whole collection as a compressed (ZIP) archive. You can also load such an archive to create a new RunGTAP version.

Version archives are most useful if you have made a habit of saving your experiments immediately after successful simulations. In that case the version archive will enable you to repeat these experiments.

You could use version archives in several ways:

(a) Create an archive as a snapshot of your work.

(b) Create an archive on floppy disk that you post to another RunGTAP user. They can load the archive as a new version and then replicate all your simulations (as long as as their version of RunGTAP is not too old).

(c) Create a new version based on the current version. First save the current version (say, **MYMODEL**) as an archive. Then load that archive with a new version name (say, **MYMODEL2**). The current RunGTAP version will then be **MYMODEL2**.

(d) Create a backup version based on the current version - which you propose to alter. First save the current version (say, **MYMODEL**) as an archive. Then load that archive with a new version name (say, **OLDMODEL**). The current RunGTAP version will then be **OLDMODEL**. Use the Version..Change command to go back to **MYMODEL**.

See also Creating Version Archives and Loading Version Archives

8.3.1 Creating Version Archives

Creating Version Archives

The File..Version Archive..Create ZIP menu command causes all the files associated with the current version to be stored in a ZIP archive. You must choose the name and location of the ZIP file.

See also Using Version Archives

8.3.2 Loading Version Archives

Loading Version Archives

The File..Version Archive..Load ZIP menu command causes all the files stored in a ZIP archive to be installed as a new version. It is expected that the ZIP file was created by the RunGTAP Version Archive..Create ZIP command

RunGTAP will first ask you to choose the name and location of the ZIP file. Then you must supply a name for the new version: RunGTAP will suggest the same name as the zip file. RunGTAP will then create the new version and load it as the current version.

If you choose a version name that is already in use, RunGTAP will ask if you want to overwrite the existing version. You will be presented with 3 options:

Yes: existing version will be replaced with the new version.

No: You will be asked to suggest a new name for the version you are loading.

Cancel: Abandon the whole operation.

See also Using Version Archives

8.4 Creating new versions

Creating new versions

You must first choose between the following two options:

Creating a new version based on the current version

Creating a new version with a new aggregation

If the number or contents of your regions and sectors are different, you must choose the second option.

8.4.1 The New Version Wizard

The New Version Wizard

The New Version Wizard will lead you through the steps involved in creating a new version. Although a particular version may have multiple closure, shock, and parameters files, there can only be one core data file of flows. If you want to have a different flows database, you need to create a new version.

If this is your first time, explore the relevant Help topics thoroughly (hint: use the >> and << buttons above).

Now read : Creating New Versions

PS: If you think the Wizard looks a little crazy, you should know that he was photographed on the last day of an exhausting GTAP training course !

8.4.2 Name of new version

Name of new version

The name of a new version must consist of up to 8 letters or numbers without spaces. The first character must be a letter, not a number.

RunGTAP warns you if there is already a version with the name you chose. That would imply that you are going to overwrite existing version files. RunGTAP will allow you to do this, but **at your own risk**.

Make sure to choose a name that you like -- the only way to change it later is to close RunGTAP altogether and use the Explorer to rename the corresponding directory.

8.4.3 Creating a new version based on the current version

Creating a new version based on the current version

You can base a new version on the current version if both versions have the same regions and sectors. There are 3 possible cases:

New core data file of flows is made as byproduct (update) of last simulation

The idea here is that the base data for your new version is the post-simulation data from the last experiment. You should successfully run the experiment just before you create

the new version. Then ask the Wizard to **use updated database from last simulation**. RunGTAP will copy all the *other* files from the existing version subdirectory.

An important special case of this is the Altertax procedure. In that case your simulation creates an updated database consistent with your desired tax rates. Then you must create a new version that contains this updated flows database.

New core data file of flows is same as existing version

In this case a new version is not essential. One reason to create it would be as a backup (but then File..Version Archive is more appropriate). Another reason might be that you wanted to have a different selection of experiments, closures, parameters or shock files.

If this is what you want, follow the Wizards suggestion of **simply copying everything**.

Neither of the above

In this case you want to use a core data file of flows which is created in some non-standard way (eg, by directly modifying the existing core data file of flows, BASEDATA.HAR). The best thing to do is to first create a new version which is identical to the current version (choose **simply copy everything)**. After you have finished with the Wizard, modify or replace the file BASEDATA.HAR in the new version subdirectory.

8.4.4 Creating a new version with a new aggregation

Creating a new version with a new aggregation

To create a new version based on a fresh aggregation you must specify a flows (core data) file, a sets file, and a parameters file. Alternatively you may specify a zip archive (created by the GTAPAgg program) which contains all three files. See files needed for a new aggregation.

RunGTAP will create some files for you, including:

- A default experiment file.
- A default version description, for you to edit later.

More files will be created during the test simulation that takes place after you leave the New Version Wizard.

8.4.5 Files needed for a new aggregation

Files needed for a new aggregation

To create a new version based on a fresh aggregation you must specify three files, all in HAR (Header Array) format:

- A flows data file.
- A sets file
- A parameters file

Normally these three files would be supplied to you by GTAP or you would have created them yourself using programs and data from GTAP.

Alternatively you may specify a ZIP archive which contains all three files named according to the standard RunGTAP file naming system (see below). The program GTAPAgg creates a ZIP archive of this type.

RunGTAP will copy the files you choose to a directory named after your new version. The copies will be renamed, if necessary, to follow the standard RunGTAP file naming system:

- The flows data file will be named BASEDATA.HAR.
- The sets file will be named SETS.HAR.
- The parameters file will be named DEFAULT.PRM.

The New Version Wizard does some basic checking to see if the files you chose are of the right type. Other problems might come to light during the test simulation that takes place after you leave the New Version Wizard.

8.4.6 After you leave the New Version Wizard

After you leave the New Version Wizard

If you quit the New Version Wizard, or just close it by clicking on the top corner, nothing more will be done.

However if you click FINISH to leave the New Version Wizard, RunGTAP will automatically load the new version and start running a test simulation. This simulation has two purposes:

- It makes sure that the data files for the new version are useable.
- It creates various ancillary files, such as the standard files of tax shocks and the GTAPView output.

The test simulation uses a standard closure and a numeraire shock.

Assuming the simulation works, YOUR work starts. You must select or create new experiment files and edit the version information.

8.4.7 Editing the Version Information

Editing the Version Information

The version information which is shown on the second RunGTAP page is stored in a file called VERSION.TXT. You can edit this file using any text editor, including that built into RunGTAP. The changes only appear on the Version page when RunGTAP next loads that version.

An easier way to edit the version information, if you are adventurous, is to use **Tools..Options** to switch into Developer mode. It allows you to edit the Version

Infomation directly, so you can see what it will look like. Dont forget to use **Developer..Save version.txt**, when you are done.

8.4.8 Altertax

Altertax

This topic is more advanced -- the instructions assume that

you already have some RunGTAP experience.

Suppose you wish RunGTAP to simulate the effect of abolishing your countrys crippling 80% tax on imported wine-corks. Using the **View..Base Data.. Tax Rates** command you discover that the GTAP database shows the cork tax to be only 30%. Why ? Perhaps the constructors of the GTAP database overlooked this detail whilst completing their enormous task. Or, maybe the tax rate *was* 30% for the data base reference year, but has risen to 80% since then.

What can you do about this problem? An obvious possibility is to edit the database directly to reflect todays 80% rate. This turns out to be difficult to do. There are so many flows to edit, and you have to ensure that important adding-up conditions are not violated.

Antipodean seer Gerard Malcolm advocates another approach -- using the model itself to create a new database. A simulation is run where tax rates are shocked to their desired value and the updated post-simulation database is used for subsequent policy experiments.

Malcolm suggests using a special closure and a special parameter file to ensure that the rate-changing simulation left other cost and sales shares as little changed as possible. His closure fixes regional trade balances, whilst his parameter settings amount to Cobb-Douglas everywhere -- this keeps budget shares fixed.

The **Tools..AlterTax** command loads these special closure and parameter settings. Then, it is up to you to set new target tax rates. The Shocks page has been designed to make this easy -- see Shocks to tax rates and Valerie. Next solve the model, and use the **View..Updated Data.. Tax Rates** to check that your targets have been reached. If all is well, create a new version based on the post-simulation database. **Version..New** launches the New Version Wizard, which helps you do this.

Finally, dont forget to change back to your original closure and parameter settings for policy experiments.

There is a Technical Working Paper by Malcolm on the GTAP web site which explains the theory of AlterTax in more detail.

Advanced Notes

The Altertax closure holds DTBALR exogenous for all regions except one, and CGDSLACK exogenous for that one region (which can be any one). Hence the closure depends on the

regional aggregation. RunGTAP has a way of manufacturing such a closure when a new version is created:

Whenever RunGTAP starts, or if the **Version..Change** or **Version..New** commands are used, RunGTAP checks that ALTERTAX.CLS exists in the new version directory. If not, RunGTAP adds to GTAP.CLP to create a new ALTERTAX.CLS in the version directory.

Whenever a simulation is run, if ALTERTAX.PRM does not exist, RunGTAP uses the ALTPAR module (if active) to create a new ALTERTAX.PRM in the version directory.

Warning: Previously, the convention was to exogenize DTBAL and SAVESLACK [instead of DTBALR and CGDSLACK]. Since RunGTAP only creates ALTERTAX.CLS if necessary, you may find that the previous Altertax closure is still being used. Edit the Altertax closure (and save it as ALTERTAX.CLS) to rectify this.

Part IX

9 Sensitivity Analysis

Sensitivity Analysis

Sensitivity analysis addresses such questions as:

How reliable are the results from a general equilibrium simulation? Is a policy conclusion very sensitive to the particular values assumed for parameters? Does our economic forecast depend critically on particular shocks to exogenous variables?

More detailed help is available from within the **Tools..Sensitivity** dialogs.

Part X

10 Options dialog

Options dialog

The **Tools..Options** Dialog allows you to customise certain features of RunGTAP:

- Show Levels Results
- **Music at startup** controls whether the GTAP tune plays when the program starts. The option is disabled if your PC has no sound card.
- If **Use Map file** is checked, only those variables listed in the GTAP map file appear on the results page; otherwise, all variables appear.
- Developer Mode
- Choose your text editor

You can also control which copy of various GEMPACK programs will be used by RunGTAP. Not all of these programs are included with the standard RunGTAP distribution.

RunGTAP uses a temporary or work directory for calculations -- this will be a subdirectory WORK beneath the RunGTAP folder.

10.1 Developer Mode

Developer Mode

Use **Tools..Options** to switch into Developer Mode, if you dare.

Developer Mode is designed to assist in preparing new versions and debugging RunGTAP problems. It has the following effects:

- You can go on even if vital files are missing.
- ViewHAR allows file modification.
- The RunGTAP text file editor allows you to modify files.
- You can edit the Version Info page within RunGTAP, and save it to disk.
- You can view the contents of the temporary, working subdirectory used by RunGTAP
- You can run checks on the files in version directories to identify missing or obsolete files.

These capabilities can be very handy, but you should be aware that they are associated with a risk. Developer mode is like driving without a seat belt -- many of the normal RunGTAP safety features are disabled. You could quite easily cause RunGTAP to crash.

Therefore, use developer mode when you need to, but switch it off at other times.

You can switch developer mode off and on using the Tools..Options menu item. It is always OFF when RunGTAP starts up.

Please dont report bugs that only occur in Developer Mode. In Developer Mode, *all bugs are features !*

10.2 Show Levels Results.

Show Levels Results.

GEMPACK can generate solutions in 4 formats for some variables:

Initial Level, Post-Solution Level, Ordinary Change, and Percentage Change.

If you switch the feature off, only the default change (percent or ordinary) will be available.

You can alter this setting using the Tools..Options menu item.

Part XI

11 Errors

Errors

See:

Errors in Solution Process

Miscellaneous Errors

or use the contents pane at left to go through the errors in sequence.

11.1 Solution Not Valid

Solution Not Valid

You may view the most recently calculated solution, but be aware that variable values shown on the results page have not been updated to reflect your changes to closure, shocks, parameter file, or solution method.

You must explicitly invoke the Solve procedure if you want to calculate new values for endogenous and exogenous variables.

11.2 Job did not run

Job did not run

This error message means that the named program produced no log file -- suggesting that it did not run at all. See miscellaneous errors.

11.3 Error..See Log Window

Error..See Log Window

If you get this message at the GTAP stage, there is something wrong with your shocks or closure. Examine the end of the log file for a clue.

If you get this message at some other stage (SLTOHT, DECOMP, GTAPView, or GTAPVol) there is some other problem. The log file may give a clue. See also miscellaneous errors.

11.4 No output file

No output file

This error message means that the named program ran normally but did not produce the expected output file. See miscellaneous errors.

11.5 Errors in Solution Process

Errors in Solution Process

There are 4 main reasons why the model may fail to solve:

- The experiment may not be well formulated. For example, it is possible to specify a shock to ALL elements of a variable with some endogenous elements. In this case, a window will appear containing an error report, generated by GEMPACK. Towards the end of this report, there should be a message indicating the cause of the error.
- RunGTAP not properly installed or the original files corrupted. See Missing Files.
- Your machine may have insufficient free memory or free disk space (see Hardware and Software Requirements).
- An error in the RunGTAP program; see Miscellaneous Errors.

11.6 Missing Files

Missing Files

This message probably means that RunGTAP was not properly installed or some original files have been deleted or moved (by your children or colleagues, perhaps?). Try installing the latest version of the program, which you can download from the RunGTAP web page.

11.7 Missing SLI file

Missing SLI file

The file VERSION.SLI contains details of variable names, sizes and elements. These are needed for the assisted method of creating shocks. If there is no VERSION.SLI file, RunGTAP creates it the first time a simulation is run. Until then, shocks must be entered manually, or loaded from file.

The usual reason for the missing SLI file message is that RunGTAP has deleted the SLI file because it is older than one of the main model EXEs (normally GTAP.EXE or SLTOHTA.EXE). This might occur if you installed a RunGTAP patch or update.

DONT PANIC !!

Simply use the menu command **Tools..Run Test Simulation** to run a simulation that will regenerate the missing file. You might have to do this (once only) the first time you use a different version.

11.8 Missing EXP files

Missing EXP files

RunGTAP expects to find at least one experiment file (a file with extension EXP) in each version subdirectory.

11.9 Not a variable !

Not a variable !

This message appears if the closure box contains an expression that RunGTAP cannot recognise as a model variable. Manually edit the closure to fix the problem, or load another closure.

Note: The message:

```
atr is not a variable
```

typically means that the closure you have loaded was designed for a previous version of the GTAP model. See New Closure for GTAP version 5.

11.10 Version directory does not exist

Version directory does not exist

A warning occurs if RunGTAP discovers that there is no subdirectory corresponding to the current version. That might occur if you deleted or renamed the version subdirectory (using Explorer, for example) from outside RunGTAP *while RunGTAP was running*.

You should close RunGTAP immediately and start again.

You should not alter the files in the RunGTAP directory or its subdirectories while RunGTAP is running except by using the RunGTAP program itself.

If you want to rename a version directory, close RunGTAP first, and remember that version names are limited to 8-letters only.

See RunGTAP Don'ts

11.11 Bad Version Name

Bad Version Name

Version Names should consist of no more than 8 letters or digits. Spaces and most special characters are not allowed.

You should close RunGTAP and use Explorer to rename the version (a subdirectory of the RunGTAP directory) to a name that conforms to RunGTAP rules.

11.12 Not a HAR file

Not a HAR file

This message appears if you were supposed to choose a file of the GEMPACK Header Array type, but instead you selected a file in a different format.

11.13 Miscellaneous Errors

Miscellaneous Errors

These could indicate a shortage of memory, poor installation, or a programming error.

Probably you should download the most recent patch (update pack) from:

http://www.copsmodels.com/rgtpatch.htm

See also: Hardware and Software Requirements

The **Help..About** window has a **Diagnostics** button which lists certain details of your system. If you have a problem with RunGTAP, you should certainly **include this information** with your error report: it might help track down the problem.

11.14 Cannot Write to Version Directory

Cannot Write to Version Directory

Before allowing you to save closures, shocks, or experiments, RunGTAP checks to see if you have write permission to the subdirectory where that version is stored. You might not have permission, if for example, RunGTAP was stored on a network drive, or on a CD-ROM.

The remedy would be to install RunGTAP on your local hard drive.

11.15 Old job still running

Old job still running

If you cancel a simulation while it is running you might get the message Old job still running. Although RunGTAP has stopped waiting for the old simulation to finish, you may not start another simulation until the previous job has either:

(a) finished by itself, or

(b) been terminated by you.

If you do try to start another simulation before (a) or (b) has occurred you will again get the message Old job still running.

To perform (b) look on the task bar (often at the bottom of the screen) and find the command-prompt icon labelled "RunGTAP task in progress". Click on that icon to enlarge the window, then terminate the job by clicking on the X mark at the top right of the DOS window. Ignore any protests by Windows.

If you cant find the "RunGTAP task in progress" icon, it may have finished by itself, while you were reading these instructions.

11.16 Formats of HAR and other Fortran files

Formats of HAR and other Fortran files

The Header Array files used by GEMPACK are stored on disk in a special format compatible with the Lahey Fortran compilers. Some years ago, Lahey decided to change this disk-file format. So now there are two special formats:

- the old format, which we call **Lahey**, is compatible with the obsolete F77L3-EM32 and LF90 compilers, which are not supported by modern GEMPACK.
- the new format, which we call **Fujitsu**, is compatible with the modern compilers supported by GEMPACK.

ViewHAR opens either style. A tiny panel at bottom right indicates the type of the currently open file. It shows **blue** for **Lahey** files, and **green** for **Fujitsu** files.

Fortran-based EXES supplied with or created with GEMPACK will produce one or the other type of file according to which compiler was used to create those EXEs.

Fortran-based EXES supplied with or created with GEMPACK 6.0 or below **will not read** the **Fujitsu** type of file and will only create the **Lahey** type of file.

Fortran-based EXES supplied with or created with GEMPACK 7.0 or later will read **both** types of files but will only write **one** type (according to which compiler was used to create those EXEs).

When a user installs the source code version of GEMPACK he or she decides which compiler is to be used and whether supplied EXE files are of the LF90 or LF95 type. So on a given PC, the GEMPACK system is also either of **Lahey** or **Fujitsu** flavour. The executable image version of GEMPACK (which needs no compiler) is of **Fujitsu** flavour.

When a GEMPACK EXE compiled with LF90 has to read a **Fujitsu** file, it first creates a temporary copy in **Lahey** format and then reads that. Similarly, when a GEMPACK EXE compiled with LF95 has to read a **Lahey** file, it first creates a temporary copy in **Fujitsu** format and then reads that. The process is automatic and transparent to the user -- although there will be a performance cost.

To avoid this performance hit, or to enable GEMPACK 6 users to read data stored in **Fujitsu** files, you may need to translate data files between **Lahey** and **Fujitsu** formats. GEMPACK offers several conversion programs for this purpose. The official program is DOS-based, and called CONLF.EXE. ConvHAR is a Windows based program, written by

Mark Horridge, which converts groups of files simultaneously. Also, ViewHAR can read or save in either format (so could be used to convert).

Note: When a GEMPACK EXE internally converts a HAR file by making a temporary copy, the log file will refer to the temporary copy as the input file -- which might be confusing.

11.17 RunGTAP and long file-names

RunGTAP and long file-names

Although recent releases of GEMPACK support long filenames, RunGTAP enforces 8-letter version and experiment names to maintain compatibility with earlier releases of GEMPACK.

11.18 Changes to the Database format for version 5 GTAP data

Changes to the Database format for version 5 GTAP data

The formats of version 5 GTAP data files differ in several ways from previous versions:

- The Parameters file is stored as a HAR file, not a text file. SLUG and RORDELTA are stored as integers, not real numbers.
- GTAP treats international trade margins in a more complete way. On the BaseData file:
- Item VST now of size MARG_COMM*REG was TRAD_COMM*REG.
- New item VTWR size MARG_COMM*TRAD_COMM*REG*REG
- A new vector of reals at header DPSM in BaseData file, size REG, containing 1s (initially); allows more sophisticated cardinalization of national utility.
- Sets file has new header MARG, listing the margin commodities.
- Sets, Parameters, and BaseData files all have a new header DVER, containing the single real number 5.0. This identifies to RunGTAP and other programs that the data matches the version 5 format. RunGTAP refers to this DVER number as the vintage. If the DVER header is missing, RunGTAP assumes that the file is in the previous, version 4 GTAP, format.

The change to file formats going from GTAP 4 to GTAP 5 will be an issue if:

- You attempt to create a new version using data made by the OLD version of GTAPAGG, which creates data for GTAP 4.
- You attempt to load a version archive which was created by an OLD version of RunGTAP, which uses data in GTAP 4 format.

In each case, RunGTAP will detect that the data is in GTAP 4 format (the clue is the absence of the DVER header) and will automatically convert it to the new format the next

time a simulation is run. You get a warning when this happens; the old files are saved with HR4 and PR4 extensions.

If some **but not all** of the Sets , Parameters, and BaseData files have the new header DVER, RunGTAP will stop with an error, issuing the message:

Sets and BaseData files have different vintages

See also New Closure for GTAP version 5.

11.19 GEMPACK Licence May Be Required

GEMPACK Licence May Be Required

See also GEMPACK Licence and http://www.copsmodels.com/gpgtap.htm

The GEMPACK program used by RunGTAP to solve the GTAP model is able to solve the model with highly aggregated data sets (for example, a 3-commodity, 3-region aggregation of the GTAP data) or, provided you have enough memory on your computer, with highly disaggregated data sets (for example, a 20-commodity, 15-region aggregation of the GTAP data).

If you are working with an aggregation with up to 9 tradeable commodities and up to 9 regions, you will not require a GEMPACK licence to solve the model.

If you are working with an aggregation of the GTAP data with more than 9 tradeable commodities and more than 9 regions, it is highly likely that you will require a **Large-simulations GEMPACK licence**, or a more powerful GEMPACK licence (for example, a Source-code or Executable-image licence).

If you are working with an aggregation with more than 10 tradeable commodities but less than 10 regions (for example, 14 tradeable commodities and 7 regions), or if you are working with an aggregation with more than 9 regions but less than 9 tradeable commodities (for example, 7 tradeable commodities and 15 regions), you may need such a GEMPACK licence. [As a rough rule of thumb, multiply the number of tradeable commodities by the number of regions. If this exceeds 100, it is highly likely that you will need such a licence. If this is less than 90, you probably wont.]

If your simulation is large enough to require at least a Large-simulations GEMPACK licence and you do not have such a licence, RunGTAP will report that your simulation ended with an error. The LOG file displayed on your screen will indicate clearly that you need a GEMPACK licence.

To obtain a Large-simulations GEMPACK licence (the cost is approximately $US300), you must contact GEMPACK (not the GTAP Project at Purdue University).

Information about GEMPACK licences, including Large-simulations licences, and their prices is available on the GEMPACK Web site at address http://www.copsmodels.com/gpprice.htm.

11.20 RunGTAP Directory must have write access

RunGTAP Directory must have write access

When you start RunGTAP, it checks to see that it has been installed in a folder which

- has write access (ie, you have permission to write files there).
- has a short and simple name, containing no spaces or Asian characters

You might get an error message if the above conditions are not satisfied. If that happens, you should rename the RunGTAP folder to something short and simple, or even re-install RunGTAP in a folder with a short and simple name -- such as C:\RunGTAP6.

See also:

The Work folder

INI File location

11.21 The Work folder

The Work folder

RunGTAP uses a work folder to store temporary files using during calculations. The folder is called WORK and is a subdirectory of the RunGTAP directory (the folder where RunGTAP.EXE is). For example, if you installed RunGTAP in C:\RunGTAP6, the work folder would be C:\RunGTAP6\Work.

RunGTAP deletes all files in the work folder on startup, and whenever you press the Solve button. However, you can exit RunGTAP and examine files (eg, log files) in the work folder. This can be useful in debugging problems.

RunGTAP uses other GEMPACK programs such as GTAP.EXE, SLTOHTA.EXE, ViewHAR.EXE and others. To ensure that these programs use the Work folder for temporary files, RunGTAP sets environment variables TMP, TEMP and GPTEMP to point to the Work folder. This setting applies only to the current RunGTAP session and to programs (such as ViewHAR) launched by that session. It does not apply to other programs running at the same time.

Note the change: Previous versions of RunGTAP used the Windows TEMP folder for calculations. This caused a number of problems, particularly if the name of that folder was long, or contained spaces or Asian characters. Hence the change.

Technical note: You can go **Help...About..Diagnostics** to get a report which includes (about halfway through) a listing of the RunGTAP environment block. This shows the current TMP and TEMP settings (pointing to the Work folder) and the previous settings (under ORIGINAL_TMP and ORIGINAL_TEMP) which still apply to other programs.

See also:

RunGTAP Directory must have write access

INI File location

11.22 INI File location

INI File location

RunGTAP stores user settings in the file RUNGTAP.INI in the RunGTAP folder (where RunGTAP.EXE is). Thus, if you have several copies of RunGTAP, each has its own INI file.

Implications:

- You must have write access to RunGTAP Directory
- You should not put RunGTAP on a network drive to be used by several people.

Note: In previous versions of RunGTAP, several copies of RunGTAP would all share one INI file, sometimes leading to problems.

See also:

RunGTAP Directory must have write access

The Work folder

Part XII

12 Contacts and Miscellaneous

Contacts and Miscellaneous

See

The Center for Global Trade Analysis

The GTAP Model

The RunGTAP web page

The GEMPACK Modelling System

Hardware and Software Requirements

Un-Installation

12.1 The Center for Global Trade Analysis

The Center for Global Trade Analysis

The Center for Global Trade Analysis undertakes international computable general equilibrium (CGE) modelling, and provides services to other CGE modellers.

The Center's main activity is contributing to and coordinating the Global Trade Analysis Project. GTAP provides data, software, and training in multi-country CGE analysis. It also maintains the celebrated GTAP model.

The Center is part of the Department of Agricultural Economics, Purdue University, West Lafayette, Indiana, USA.

http://www.gtap.org/

12.2 The GTAP Model

The GTAP Model

The standard GTAP model is a multi region, computable general equilibrium model, with perfect competition and constant returns to scale. Bilateral trade is handled via the Armington assumption. Innovative aspects of this model include: the treatment of private household preferences using the non-homothetic CDE functional form, explicit treatment of international trade and transport margins, and a global banking sector which intermediates between global savings and consumption. The standard model also gives users a wide range of closure options, including a selection of partial equilibrium closures that facilitate comparison of results to studies based on partial equilibrium assumptions. The model is implemented using the GEMPACK software suite.

12.3 The RunGTAP web page

The RunGTAP web page

This is located at:

http://www.gtap.agecon.purdue.edu/products/rungtap/default.asp

You can download the standard version of RunGTAP there. There may also be a list of problems that users have encountered, with suggested solutions.

The web page http://www.copsmodels.com/rgtpatch.htm

lets you download a service patch, which fixes various known problems.

While you are running RunGTAP, you can use the Help..About menu command to find the creation date of your copy of the program. The Web page should show the date of the most recent version. If the two are the same, there is no point in downloading the files.

12.4 The GEMPACK Modelling System

The GEMPACK Modelling System

GEMPACK is a suite of general-purpose economic modelling software especially suitable for formulating and solving general equilibrium models. The user specifies the economic behaviour in algebraic form and supplies data defining some initial solution of the model: GEMPACK will find new solutions of the model corresponding to different values for the exogenous variables. It also contains a range of utility programs for viewing and manipulating economic data and the results of simulations. On Windows PCs, GEMPACK may be manipulated via a visual interface, **WinGEM**.

To find out more about GEMPACK see the web page at:
http://www.copsmodels.com/gempack.htm

Manuals can be downloaded from: http://www.copsmodels.com/gpdoc.htm

12.5 Hardware and Software Requirements

Hardware and Software Requirements

Operating system

RunGTAP requires a X86-compatible PC running Microsoft Windows XP or later.

Memory

You will need at least 256Mb of RAM *in addition to that required by your operating system*. In practice this means that you will need at least 2GB of RAM. If you run versions with many commodities and regions, you may need yet more RAM.

Licensing

In essence RunGTAP is a free good which you use or share without payment. However, if you run versions with many commodities and regions, you may need a GEMPACK licence. See GEMPACK Licence May Be Required and GEMPACK Licence. You may also be charged if you wish to make use of the full (60+ regions, 57 sector) version of the GTAP database.

Hard disk

The installation requires around 100 Mb of free disk space. Another 20 megabytes may be needed during the solution process to store temporary files.

Read-Only Access

You **might** be able to run RunGTAP from a read-only network drive or CD-ROM. In each case the operation of RunGTAP will be restricted, since many procedures require that RunGTAP write to version subdirectories.

Installing on a network drive

You **might** be able to run RunGTAP from a network drive, but problems would arise if two people tried to use RunGTAP at the same time, since RunGTAP writes to version subdirectories (which would be shared network folders).

12.6 GEMPACK licence

GEMPACK licence

There are some RunGTAP activities for which you need a GEMPACK licence.These are:

(a) Solving very large models.

(b) Doing sensitivity analysis.

RunGTAP uses the file TLICA.EXE and TLICA.PIF to check licences -- these files must be located in the same subdirectory as RUNGTAP.EXE.

The GEMPACK licence is a file called LICEN.GEM which must be located in the directory C:\GP (or in the subdirectory pointed to by the DOS environment variable GPDIR).

You can view your licence status by selecting the **Help..GEMPACK licence** menu item.

See also http://www.copsmodels.com/gpgtap.htm

and GEMPACK Licence may be required.

12.7 Un-Installation

Un-Installation

At installation, a number of non-version-specific files are created in one home directory (the directory containing RUNGTAP.EXE). This will have several version-specific subdirectories.

No other changes to the system are made, and so you can expunge all trace of RunGTAP by deleting the home directory, and any desktop icons pointing to RunGTAP.

12.8 Hints for running large models

Hints for running large models

RunGTAP imposes no limit on the size of your model. If your PC has enough memory you can have as many regions or commodities as you like. However, the following considerations apply:

For more than 10 regions or commodities you may need an additional special GEMPACK licence.

If you double the number of regions and commodities, solution time and memory requirements will increase by much more than double.

Use the Tools..Options menu command to switch off the calculation of GE elasticities.

While the model is being solved a small window with a spinning globe appears. If you want to use the PC for other tasks during the simulation, you must click the **Hide** button underneath the spinning globe. RunGTAP will be minimized to the task bar, but will pop up again automatically when the simulation is completed. Meanwhile you should be able to perform other small tasks, such as reading your email. However, other programs may seem very slow, as the bulk of the PCs resources are devoted to the GEMPACK computation.

Windows will attempt to use virtual memory if your PC lacks suffficient physical memory for the jobs you are running. This can cause GEMPACK jobs to run VERY slowly. A symptom of the problem is constant hard disk activity, as data is paged from hard disk to RAM and back again. Fix the problem by installing more RAM (eg, a gigabyte) in your machine.

12.9 Valerie Powers, Tax Consultant

Valerie Powers, Tax Consultant

The GTAP project is pleased to announce the appointment of Valerie Powers as a new Associate Emeritus. Previously Senior Executive Partner (Conversion) of taxation lawyers Gruff Coddington Grabbet (Lagos and Cayman Islands), Ms. Powers is better known in academic circles for her research on the measurement of tax changes.

Due to temporary legal problems, it is not yet possible to consult Valerie in person. However, RunGTAP incorporates an expert system that brings to all her extraordinary facility at converting between powers of taxes and ad valorem rates. Use **Tools..Valerie** to try it.

Enter numbers in any two cells. Then click on Valerie -- she uses the blue cells to work out the rest.

At the bottom of the panel, Valerie shows the steps she followed to work out the answers. These formulae refer to variables, iR, fP, dR and so on. Each variable name has two letters, which follow a system.

First Letter: i (**i**nitial) f (**f**inal) d (or**d**inary change) p (**p**ercent change)

Second Letter: R (ad valorem % **R**ate) P (**P**ower)

Hence, iR = initial rate, dP = ordinary change in power, pR = % change in rate, and so on. One letter in each row and column label is underlined, to help you remember the system.

For other queries, please email Valerie at valpowers@gmail.com

and she will respond when her crowded itinerary allows her an opportunity.

12.10 Purdue Pete

Purdue Pete



Purdue Pete is the unofficial mascot of the Purdue football team -- the BoilerMakers.

Legend has it that in the early years of Purdues football, there was a shortage of players, so the coaches recruited men from the nearby Monon Railroad Factory. Pete is one of these tough workers who worked (with hammers) on the big steam engines (Boilers).

The Purdue Pete image was created in 1940 as an advertising logo for the University Book Store. From 1956 mascots dressed as Purdue Pete have appeared at football games.

The official team mascot is a train -- the BoilerMaker Express:



12.11 Indiana State Bird

Indiana State Bird



The Cardinal (Richmondena cardinalis) was adopted as the state bird by the 1933 Indiana General Assembly. The male is bright red; the female is brown with dull red crest, wings and tail. The birds remain in Indiana year round and nest in thickets of brambles or low saplings. The eggs are bluish-white with brown markings.

The Cardinal is the state bird of 7 states: Illinois, Indiana, Kentucky, North Carolina, Ohio, Virgina, and West Virginia.

12.12 Indiana State Flower(s)

Indiana State Flower(s)

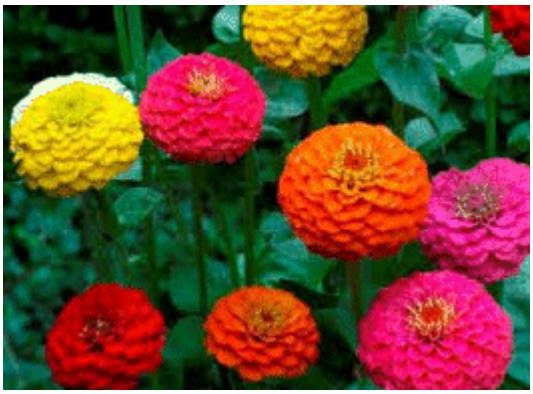
Indiana adopted the carnation as its state flower in 1913:



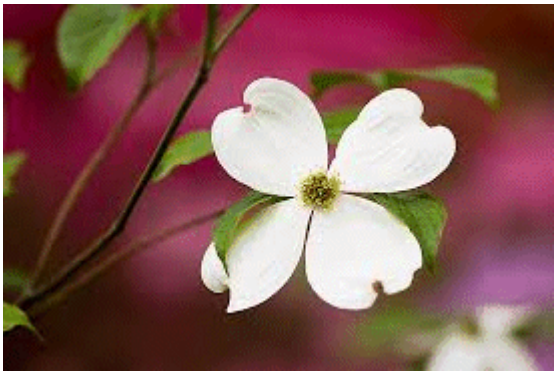
but the General Assembly of 1923 chose instead the blossom of the tulip tree:



In 1931 the Assembly changed its mind again, choosing the zinnia (persuaded by a powerful zinnia grower):



Then in 1957, a Senate committee proposed that this in turn be replaced by the dogwood flower:



However, one state representative (who was also a commercial peony grower) persuaded the House to adopt instead the peony:



-- which has remained the official state flower ever since.

12.13 The Rube Goldberg Machine Contest

The Rube Goldberg Machine Contest

began as a rivalry between two Purdue engineering fraternities and was popular at Purdue in the 1940s and 1950s. College students compete to design a machine that uses the most complex process to complete a simple task - put a stamp on an envelope, screw in a light bulb, make a cup of coffee - in 20 or more steps.



The cartoon above, showing a machine that helps a numbskull become an engineer, was drawn for Purdue students by cartoonist Rube Goldberg in the early 50s. According to

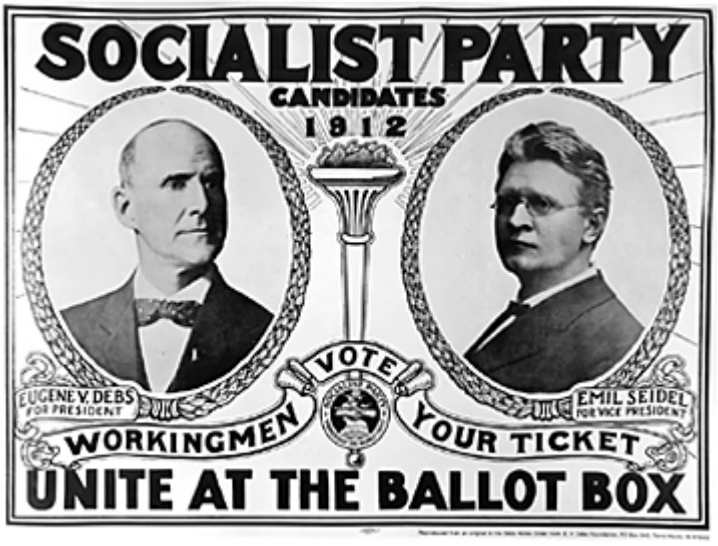
Goldberg: Seal balances ball and applauds self, causing string to start one-man band. Co-ed, hearing music, does Charleston, kicking over oil lamp and burning string, dropping basketball on switch, which starts atomic machine, lifting numbskull so he can crib answers and become a successful engineer.

12.14 A Prominent Hoosier

A Prominent Hoosier



Probably the most famous Hoosier was Eugene V. Debs. He ran for President five times [1900, 1904, 1908, 1912, 1920] as candidate for the Socialist Party, in some of the most dynamic campaigning ever seen in the United States. His greatest showing was the campaign of 1908 which featured the RED SPECIAL train which went all over the country.



Eugene Victor Debs was born in Terre Haute, Indiana in 1855. He found work as a railroad fireman in 1870 and eventually became active in the trade union movement. Debs worked as editor of the Locomotive Firemens Magazine, before being elected national secretary of Brotherhood of Locomotive Fireman in 1880. Debs, a member of the Democratic Party, was elected to the Indiana Legislature in 1884.

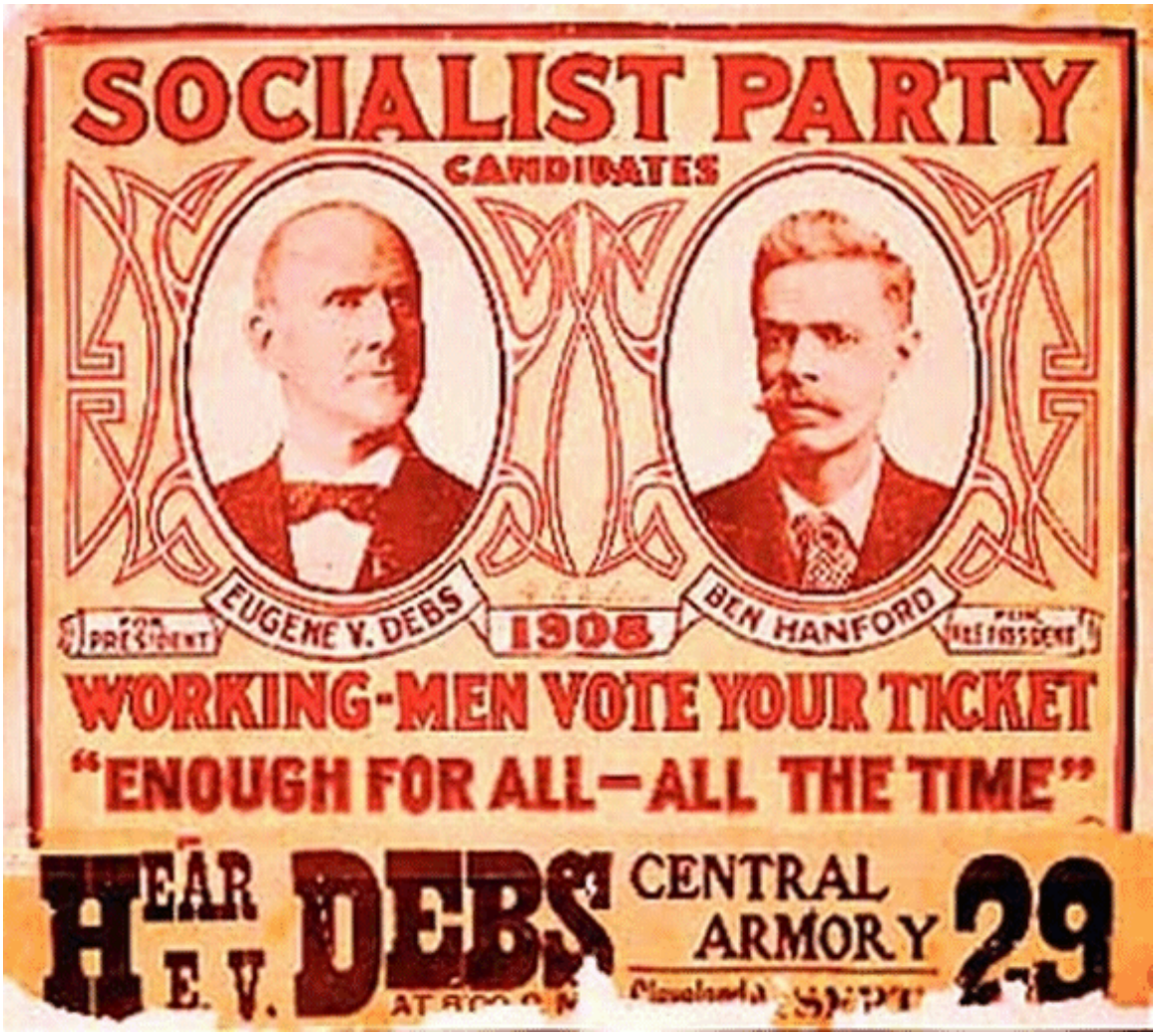
In 1893 Debs was elected the first president of the American Railway Union (ARU). During the Pullman Strike in 1894, Debs was arrested and charged with contempt of

court. Despite being defended by Clarence Darrow, he was found guilty and sentenced to six months in prison.

Debs now became a socialist and believed that capitalism should be replaced by a new cooperative system. Although he advocated radical reform, Debs was opposed to the revolutionary violence supported by the Communist Party.

In 1897 Debs joined with Victor Berger and Ella Reeve Bloor to form the Social Democratic Party (SDP). Debs was the partys presidential candidate in 1900 but received only 97,000 votes. The following year some members of the SDP left the party and established the Socialist Party of America.

In 1904 Debs was the new party's presidential candidate and got 400,000 votes. He was also the party's candidate in 1908, and 1912, when with his running-mate, Emil Seidel, he won 897,011 votes.



Debs believed that the First World War had been caused by the imperialist competitive system. Between 1914 and 1917 Debs made several speeches explaining why he believed the United States should not join the war. After the USA declared war on the Central Powers in 1917, several Socialist Party members were arrested for violating the

Espionage Act. After making a speech in Canton, Ohio, on 16th June, 1918, criticizing the Espionage Act, Debs was arrested and sentenced to ten years in Atlanta Penitentiary. He was still in prison when as the presidential candidate of the Socialist Party, he received 919,799 votes in 1920.



His program included proposals for improved labour conditions, housing and welfare legislation and an increase in the number of people who could vote in elections. President Warren G. Harding pardoned Debs in December, 1921. Debs died in 1926: his house in Terre Haute is preserved as a museum.

12.15 Debs on War

Debs on War

Some excerpts from a Debs speech:

Wars throughout history have been waged for conquest and plunder. In the Middle Ages when the feudal lords who inhabited the castles whose towers may still be seen along the Rhine concluded to enlarge their domains, to increase their power, their prestige and their wealth they declared war upon one another. But they themselves did not go to war any more than the modern feudal lords, the barons of Wall Street go to war.

The feudal barons of the Middle Ages, the economic predecessors of the capitalists of our day, declared all wars. And their miserable serfs fought all the battles. The poor, ignorant serfs had been taught to revere their masters; to believe that when their masters declared war upon one another, it was their patriotic duty to fall upon one another and to cut one anothers throats for the profit and glory of the lords and barons who held them in contempt. And that is war in a nutshell. The master class has always declared the wars; the subject class has always fought the battles. The master class has had all to gain and nothing to lose, while the subject class has had nothing to gain and all to lose--especially their lives.

They have always taught and trained you to believe it to be your patriotic duty to go to war and to have yourselves slaughtered at their command. But in all the history of the

world you, the people, have never had a voice in declaring war, and strange as it certainly appears, no war by any nation in any age has ever been declared by the people.

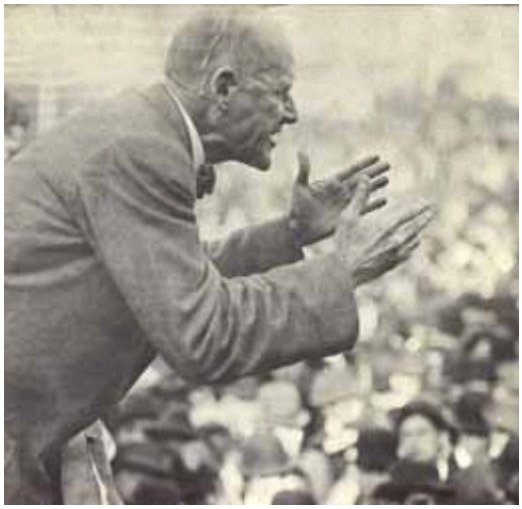
And here let me emphasize the fact--and it cannot be repeated too often--that the working class who fight all the battles, the working class who make the supreme sacrifices, the working class who freely shed their blood and furnish the corpses, have never yet had a voice in either declaring war or making peace. It is the ruling class that invariably does both. They alone declare war and they alone make peace. Yours not to reason why; Yours but to do and die. That is their motto and we object on the part of the awakening workers of this nation. If war is right let it be declared by the people. You who have your lives to lose, you certainly above all others have the right to decide the momentous issue of war or peace.

You need at this time especially to know that you are fit for something better than slavery and cannon fodder. You need to know that you were not created to work and produce and impoverish yourself to enrich an idle exploiter. You need to know that you have a mind to improve, a soul to develop, and a manhood to sustain.

They are continually talking about your patriotic duty. It is not their but your patriotic duty that they are concerned about. There is a decided difference. Their patriotic duty never takes them to the firing line or chucks them into the trenches. And now among other things they are urging you to cultivate war gardens, while at the same time a government war report just issued shows that practically 52 percent of the arable, tillable soil is held out of use by the landlords, speculators and profiteers. They themselves do not cultivate the soil. Nor do they allow others to cultivate it. They keep it idle to enrich themselves, to pocket the millions of dollars of unearned increment....

And now for all of us to do our duty! The clarion call is ringing in our ears and we cannot falter without being convicted of treason to ourselves and to our great cause.

Do not worry over the charge of treason to your masters, but be concerned about the treason that involves yourselves. Be true to yourself and you cannot be a traitor to any good cause on earth.



Debs outside Canton jail Ohio delivering the speech for which he was himself jailed.

12.16 The Indiana State Song

The Indiana State Song

On the Banks of the Wabash, Far Away

Round my Indiana homesteads wave the cornfields,
In the distance loom the woodlands clear and cool.
Oftentimes my thoughts revert to scenes of childhood,
Where I first received my lessons, nature's school.
But one thing there is missing in the picture,
Without her face it seems so incomplete.
I long to see my mother in the doorway,
As she stood there years ago, her boy to greet.
[CHORUS]
Oh, the moonlight's fair tonight along the Wabash,
From the fields there comes the breath of new-mown hay.
Through the sycamores the candle lights are gleaming,
On the banks of the Wabash, far away.



Many years have passed since I strolled by the river,
Arm in arm, with sweetheart Mary by my side,
It was there I tried to tell her that I loved her,
It was there I begged of her to be my bride.
Long years have passed since I strolled thro there churchyard.
Shes sleeping there, my angel, Mary dear,
I loved her, but she thought I didn't mean it,
Still I'd give my future were she only here.

On the Banks of the Wabash, Far Away, written by Terre Haute native Paul Dresser and dedicated to 14-year-old Mary E. South of Terre Haute, whom Dresser had never met, is the state song of Indiana. First published in July 1897, the song was adopted as the official

state song on March 14, 1913, by the Indiana General Assembly. The decision was affirmed in 1998. However, a perky imitation of Wabash, plagiarized in 1917, and popularized by a 1942 film, is more widely sung today (Indiana).

Paul Dresser was the brother of noted Hoosier writer Theodore Dreiser. It is said that Paul was so scandalized by his brothers frank writings that he changed his name from Dreiser to Dresser.

Hear the song: http://www.copsmodels.com/ftp/tgm/wabash.mp3

12.17 Roger Bean

Roger Bean



Chic Jackson July 10, 1920

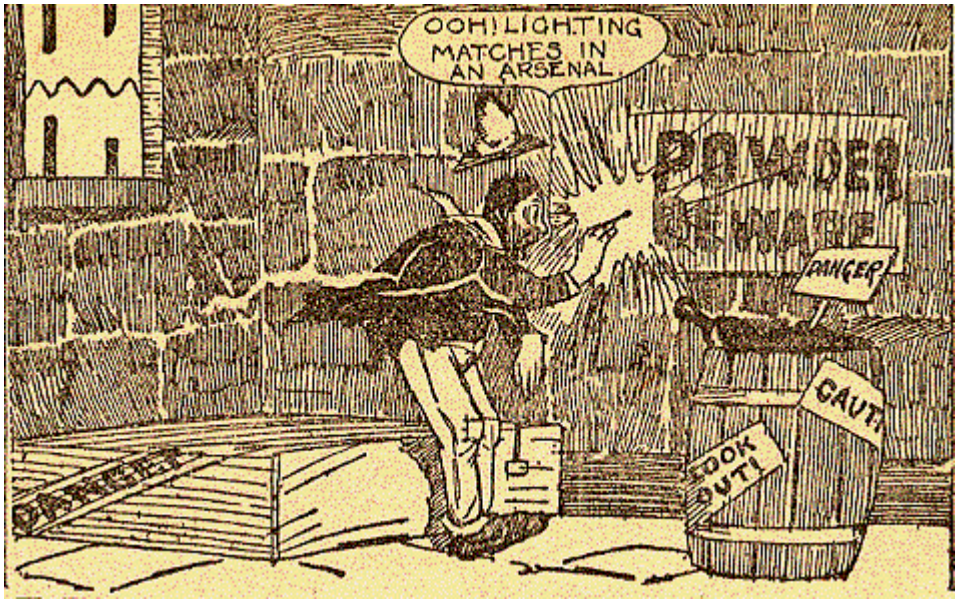
On April 22, 1913, Mr. and Mrs. Roger Bean stepped off a train at Union Station, and a new, locally produced comic strip made its debut. Created by cartoonist Chic Jackson, Roger Bean depicted the trials and tribulations of a typical Hoosier family. It ran for 21 years and resulted in eight books of collected strips. From a review of one such collection:

This years [1916] Roger Bean Book by Chic Jackson, cartoonist of the Indianapolis Star, is attracting widespread attention. There are 62 splendid illustrations, each one telling a story that will hold the readers attention and interest.

To those who are familiar with Jacksons daily strip, which deals with everyday life as it really is, this book will appeal in particular, as it is like having the best parts of an interesting continued story compiled and ready to read.

Those who are not acquainted with Roger Bean will find it profitable to meet the square good natured regular guy he is. His plump little wife, exquisitely feminine, the baby boy they adopted one Christmas eve, and Golduh Stubbins, the inexperienced maid of all work make up the rest of Rogers household. The title of the book is Roger Bean On The Firing Line.

Chic Jackson hails from the state of Indiana, which has produced so many stellar lights in the literary field. His progress has been rapid, and today he is a household idol in Indiana.





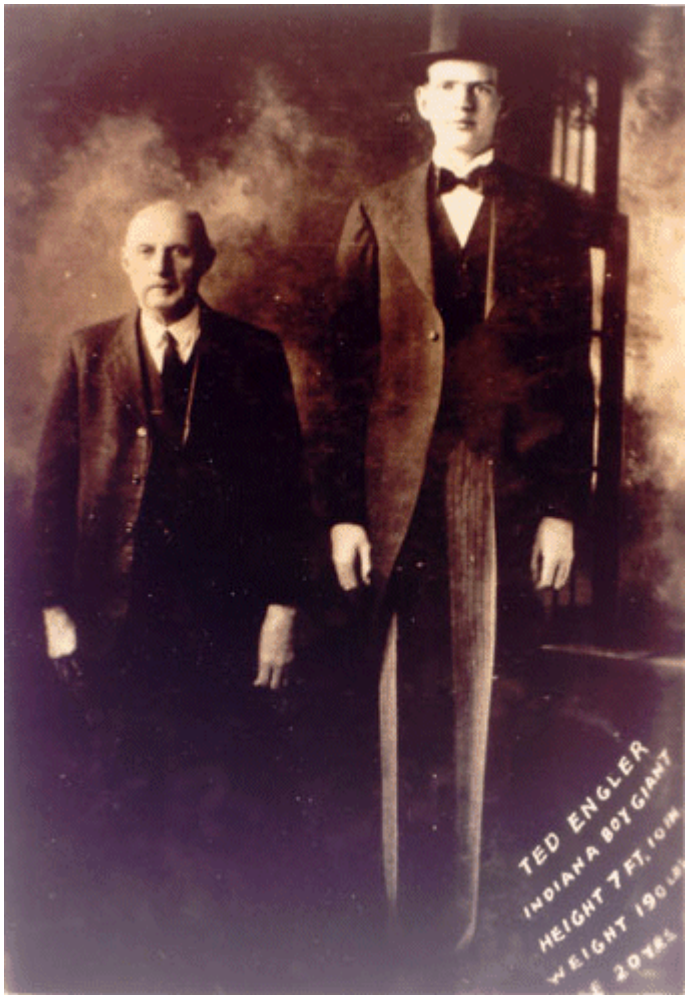
12.18 Hoosiers

Hoosiers

A Hoosier is a person from Indiana. No-one is quite sure how the word originated. On January 12, 2017, the Federal Government officially changed the nickname of people from the state of Indiana from "Indianans" to "Hoosiers", making Indiana the first state to not have a version of their state name in their nickname ("Illinoisans", "Texans", etc.).



2002 commemorative banknote



A 7'10" giant from Indiana

12.19 The Triple XXX

The Triple XXX



Opened in 1929 by Bert Wright, the 24-hour restaurant on the Levee near Purdue University is the first and oldest Drive-in Restaurant in Indiana. It continues to serve hamburgers, root beer, and shakes to West Lafayette residents. The name derives from a brand of root beer.



12.20 Tippecanoe and Tyler too !

Tippecanoe and Tyler too !

Tippecanoe and Tyler too was the 1840 US presidential campaign slogan of William Henry Harrison -- the first president to campaign actively for office.

While Tyler was Harrison's running-mate, Tippecanoe was a nickname referring to a battle fought by Harrison while he was governor of the newly formed Indiana Territory.

As governor, Harrison arranged to buy large areas of Indian land for use by white settlers. The land cessions were opposed by the Shawnee chief Tecumseh, who organized armed resistance.



Harrison and Tecumseh met in battle in 1811 in Tippecanoe near present-day West Lafayette. Although the battle was indecisive (Harrison lost more men than the Indians) it was claimed as a government triumph.

Harrison's military record figured strongly in his successful campaign against incumbent Van Buren. His theme song (another first) was written by Alexander Coffman Ross, a jeweler of Zanesville, Ohio, with music based on the tune Little Pigs. The first verse was:

What's the cause of this commotion, motion, motion,

Our country through?

It is the ball a-rolling on

For Tippecanoe and Tyler too.

For Tippecanoe and Tyler too.

And with them well beat little Van, Van, Van,

Van is a used up man.

And with them well beat little Van.

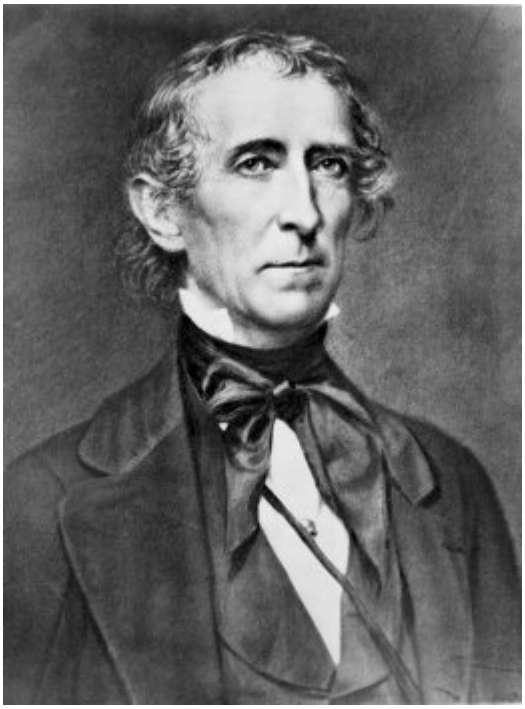


Harrison, aged 68, was criticized as too old for the presidency. After an opposing newspaper claimed

Give him a barrel of hard cider, and a pension of two thousand a year and he will sit the remainder of his days in his log cabin.

Harrison campaigned as the log cabin and hard cider candidate -- a man of the common people.

When elected, Harrison delivered the longest inaugural address in American history -- without a coat in bad weather. A month later he was dead of pneumonia - to be succeeded by Tyler who served the rest of the 4-year term.

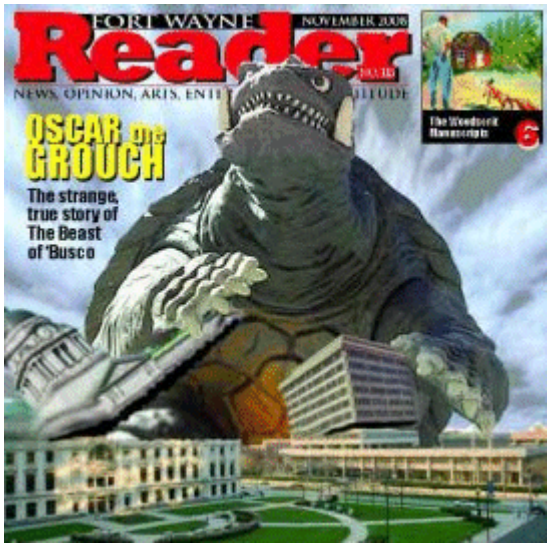


John Tyler

12.21 The Beast of Busco

The Beast of Busco

During the late 1950s the town of Churubusco, Indiana (pop. 1800, 18 miles south of Fort Wayne) was terrorized by a giant turtle.



A local newspaper pictures Oscar destroying Fort Wayne

The monster was first seen in 1898 in a lake owned by Oscar Fulk, a local farmer. Fulk carved his name, Oscar, on the shell, and this also became the name of the turtle. Another sighting occurred in 1914.

Oscar was spotted again in 1948 by two fisherman. Later the lake's new owner, Gale Harris, and the local minister also saw the beast. According to reports, its head was as big as a child's, and its shell was around 6 feet long and five feet wide. Locals complained that chickens and other small livestock, even young cows, had been disappearing.

Harris organized attempts to trap or kill the creature which always escaped. A film (since lost) showed Oscar breaking free of a trap. So Harris invested in huge pumps to drain the lake; to pay for his search he charged admission to tourists. His quest became a media sensation and visitor numbers swelled. In October 1949, a number of these actually glimpsed the enormous turtle.

Unfortunately, Harris became ill, the lake refilled and Oscar was never seen again. Since 1950, Churubusco has organized an annual Turtle Festival in his honour.



A much smaller turtle

12.22 Corn and Soy

Corn and Soy

Corn (maize) is Indianas most popular crop, with around 7 million acres planted in 2010. Traditionally, corn must be rotated every 2 or 3 years with soybean to avoid exhausting the soil. However, high subsidies to corn ethanol have recently encouraged successive corn-on-corn cropping at the cost of higher fertilizer inputs.

A favourite Hoosier snack is the **corn dog** -- a hot dog coated in cornbread batter, deep fried in hot oil, and served on a wooden stick.



Enjoying a corn dog at the Indiana State Fair

12.23 The Indiana Pi Bill

The Indiana Pi Bill

An eccentric medical doctor and amateur mathematician, Dr. Edwin Goodwin of Solitude, Indiana, discovered (he thought) how to square the circle, and encouraged his local representative to enact the discovery into State law. A bill was submitted to the Indiana General Assembly in January 1897, which would allow Indiana residents to freely use his discoveries -- others would pay a copyright fee. The Bill was passed unanimously by the lower House in February 1897 and passed on to the Senate for review.

The bill was full of self-contradictory nonsense, including a decree that "the ratio of the diameter and circumference [of a circle] is as five-fourths to four." ie, that P = 3.2.

Luckily the head of the Purdue University Mathematics Department, Professor Clarence Abiathar Waldo, happened to overhear the Assembly debate. Of course he was appalled.



CLARENCE ABIATHAR WALDO,
A. M., Ph. D.,
Head Professor of Mathematics.

Professor Waldo quickly moved to inform Senators about the dangers of a bill which exposed Indiana to ridicule. In the next session, Senators laughed at the bill and referred it indefinitely to a sub-committee (on public drunkenness!).

The official history of the Indiana General Assembly records:

"Thanks mainly to this alert professor, who convinced the Senate not to tamper with unsolvable mysteries . . . above man's abilities to comprehend, the Indiana General Assembly failed to do in 1897 what no one before or since has done, i.e. square the circle."

Above account was abbreviated from:

http://www.agecon.purdue.edu/crd/localgov/second level pages/indiana_pi_story.htm

Text of Indiana Bill No. 246, 1897

A Bill for an act introducing a new mathematical truth and offered as a contribution to education to be used only by the State of Indiana free of cost by paying any royalties whatever on the same, provided it is accepted and adopted by the official action of the Legislature of 1897.

Section 1

Be it enacted by the General Assembly of the State of Indiana: It has been found that a circular area is to the square on a line equal to the quadrant of the circumference, as the area of an equilateral rectangle is to the square on one side. The diameter employed as the linear unit according to the present rule in computing the circle's area is entirely wrong, as it represents the circle's area one and one-fifth times the area of a square whose perimeter is equal to the circumference of the circle. This is because one fifth of the diameter fails to be represented four times in the circle's circumference. For example: if we multiply the perimeter of a square by one-fourth of any line one-fifth greater than one side, we can in like manner make the square's area to appear one-fifth greater than the fact, as is done by taking the diameter for the linear unit instead of the quadrant of the circle's circumference.

Section 2

It is impossible to compute the area of a circle on the diameter as the linear unit without trespassing upon the area outside of the circle to the extent of including one-fifth more area than is contained within the circle's circumference, because the square on the diameter produces the side of a square which equals nine when the arc of ninety degrees equals eight. By taking the quadrant of the circle's circumference for the linear unit, we fulfill the requirements of both quadrature and rectification of the circle's circumference. Furthermore, it has revealed the ratio of the chord and arc of ninety degrees, which is as seven to eight, and also the ratio of the diagonal and one side of a square which is as ten to seven, disclosing the fourth important fact, that the ratio of the diameter and circumference is as five-fourths to four; and because of these facts and the further fact that the rule in present use fails to work both ways mathematically, it should be discarded as wholly wanting and misleading in its practical applications.

Section 3

In further proof of the value of the author's proposed contribution to education and offered as a gift to the State of Indiana, is the fact of his solutions of the trisection of the angle, duplication of the cube and quadrature of the circle having been already accepted as contributions to science by the American Mathematical Monthly, the leading exponent of mathematical thought in this country. And be it remembered that these noted problems had been long since given up by scientific bodies as insolvable mysteries and above man's ability to comprehend.

*Note: information above refers to the mathematical symbol **pi**. Since January 23, 2009, the Indiana state **pie** is officially sugar cream pie*



12.24 Euchre

Euchre

Euchre is a trick-taking card-game (like bridge) played with a reduced deck. It is the game responsible for introducing the joker card into modern packs, and is related to Loo, a seventeenth-century game of bad repute. During the 19th century, it was widely regarded as the national card game of the USA, but it is now less popular. However, it retains a strong following in Indiana and other Midwest states.

See the Wikipedia article.

A similar game, called 500, is popular in Australia.



During July 2016, the Indiana Gaming Commission, as part of a campaign against illegal gambling, warned several Senior Citizens Centers that playing Euchre for small prizes (such as canned peaches and toilet paper) was illegal. After widespread protests, the Governor, Mike Pence, instructed the Gaming Commission to "ensure common sense prevails" in its actions.

12.25 The insectless state?

The insectless state?

45 of the 50 US states have a state insect -- but not Indiana [as of 2016]!

It is proposed that Say's firefly (pictured below) be the official state insect.

Say's firefly is one of about 175 species of fireflies in the United States.



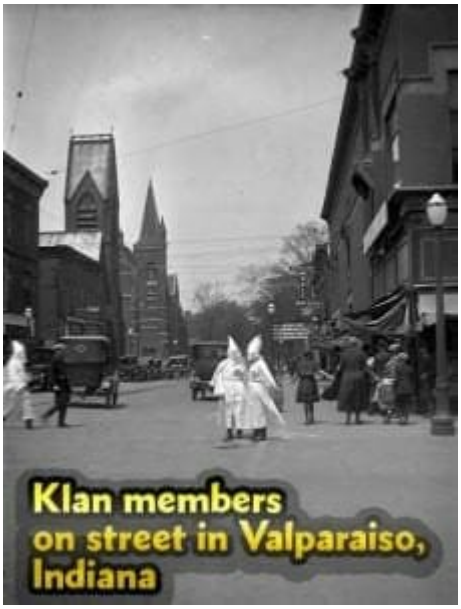
12.26 A vile murder

A vile murder

The Ku Klux Klan was an American white supremacist hate group, formed to terrorize black people after the defeat of the Confederacy in the American Civil War. Confederate General Nathan Bedford Forrest was elected the first leader or Grand Wizard. However, in 1870, a federal grand jury determined that the Klan was a "terrorist organization". Klan members were prosecuted, and by 1872 the movement had been suppressed.

After the film The Birth of a Nation was released, mythologizing and glorifying the former Klan, a second Ku Klux Klan was founded in 1915 in Georgia. It adopted regalia featured in the film; membership was kept secret by wearing masks in public.

The new movement spread to midwestern states. A young man named D.C. Stephenson soon became the Klan's best recruiter in Indiana, and state leader (Grand Dragon) in 1922. Stephenson preached a new anti-immigrant, anti-Catholic, Prohibitionist, anti-union and anti-Semitic agenda. Membership grew by leaps and bounds, including many clergymen and local politicians. Around 250,000 white Indiana men (about 30% of the native-born Caucasians) joined the Klan in the early 1920s -- the highest proportion of any state. Stephenson received a fee for each new recruit, becoming extremely rich.



Soon Stephenson was the most powerful political figure in Indiana. Even the Governor, Republican Edward Jackson was his protégé. Stephenson broke with the existing

national KKK, and formed a rival KKK that supported the Republican Party. Yet rumors of his dark side emerged. Numerous reports surfaced of attempted rapes and sexual assaults--usually after he had been drinking.



Stephenson felt himself above the law. In January 1925 he formed a brief association with Madge Oberholtzer, a young female literacy teacher. She dropped him 2 months later. On March 15 Stephenson and his bodyguards forced Oberholtzer onto Stephenson's private train to Chicago. In the train she was repeatedly raped and bitten. She was then imprisoned in the Indiana Hotel, of Hammond, Indiana.



Planning suicide, she managed to buy a box of poisonous mercuric chloride tablets, but was only strong to swallow three. She vomited blood throughout the remainder of the

day. Stephenson panicked and ordered his thugs to drive her back home to Indianapolis. She died on April 14, 1925 from infections from the bites, plus kidney failure from mercury poisoning.

Publicly a Prohibitionist and a defender of "Protestant womanhood," Stephenson was tried in 1925 for the rape and murder of Madge Oberholtzer.The Klan's image as upholders of law and morality was gravely weakened as it was proven that Stephenson and many of his associates were in private womanizers and alcoholics. Stephenson was convicted of the abduction, forced intoxication, and rape of Oberholtzer. Because her suicide attempt eventually caused death, Stephenson was also charged with murder. He was sentenced to life in prison on November 16, 1925.

To Stephenson's surprise, Governor Jackson refused to grant clemency or a sentence reduction. In revenge, Stephenson released lists of public officials who were or had been on the Klan payroll. The Indianapolis Times interviewed Stephenson and investigated the Klan's political ties, winning a Pulitzer Prize for its reporting. This publicity and the state's crackdown on Klan activity caused a huge decline of the organization by the end of the 1920s -- many local chapters quit *en masse*. The Indiana Klan was virtually wiped out.

The state indited Governor Jackson; George V. "Cap" Coffin, chairman of the Marion County Republican Party; and attorney Robert I. Marsh, charging them with conspiring to bribe former Governor Warren McCray. The mayor of Indianapolis, John Duvall, was convicted and sentenced to jail for 30 days. Republican commissioners of Marion County resigned after being charged with accepting bribes from the Klan and Stephenson.

Stephenson was paroled in December 1956, on condition that he leave Indiana and never return.

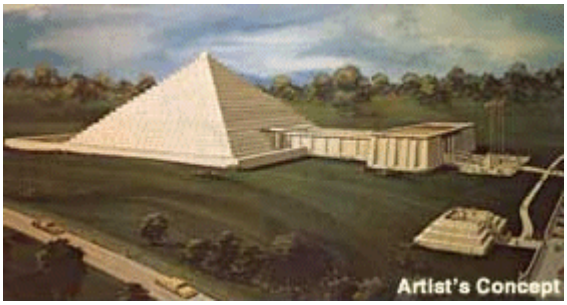
See also:
en.wikipedia.org/wiki/Madge_Oberholtzer
en.wikipedia.org/wiki/D._C._Stephenson
famous-trials.com/stephenson/74-home

12.27 The forgotten pyramid

The forgotten pyramid

Bedford, Indiana (pop 14000), is known as the limestone capital of the world. Its quarries supplied stone for many of Washington's major buildings and for the Empire State Building.

In the late 1970s townspeople planned to celebrate these achievements (and attract tourists) by building a 1/5th scale replica of the Great Pyramid of Giza, and received government funding of $700,000 to complete the project .

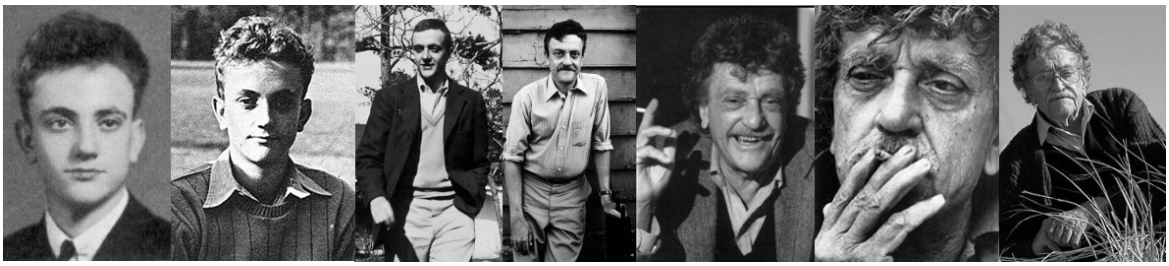


Unfortunately the plan was noticed in 1981 by Democratic Senator William Proxmire, who gave the project a "Golden Fleece" award for its perceived waste. Funds were withdrawn, and the project abandoned. The remnants can still be viewed about nine miles north of Bedford.



12.28 So it goes

So it goes



Kurt Vonnegut Jr. was born in Indianapolis on Armistice Day, 1922, to a wealthy family of architects and brewers. After studying to be an engineer, he joined the Army during World War II, and was captured by the Germans. In Dresden as a POW, he survived the night of Allied fire-bombing which killed around 150,000 German civilians -- more than died at Hiroshima and Nagasaki. Afterwards, he helped other surviving POWs to gather and cremate corpses. This provided the backdrop for his most famous novel, Slaughterhouse-Five (1969). Other famous works included Player Piano (1952), Cats Cradle (1963) and Breakfast of Champions (1973).

Vonneguts personal life was surrounded by tragedies. His mother committed suicide; and his sister and her husband died during the same week in 1958 -- leaving 3 children, whom Vonnegut adopted. His son struggled with madness but emerged. Vonnegut himself, after a low period in the 1970s attempted suicide in 1984.

In later years Vonnegut returned to writing, and lent his support to various causes. He served as Honorary President of the American Humanist Association,and appeared in advertisements for the American Civil Liberties Union. In 2005 he publicly criticized the US invasion of Iraq.

In 2000, a fire destroyed the top story of his home. Vonnegut was hospitalized and his personal archives destroyed. Early in 2007, a fall at home caused irreversible brain injuries. He died aged 84 in April 2007.

Some Vonnegut sayings

A purpose of human life, no matter who is controlling it, is to love whoever is around to be loved.

I really wonder what gives us the right to wreck this poor planet of ours.

I want to stay as close to the edge as I can without going over. Out on the edge, you see all kinds of things you cant see from the center.

It happens too fast for you ever to think about it. If you could just persuade people of this, but they insist on amassing information.

If people think nature is their friend, then they sure don't need an enemy.

Laughter or crying is what a human being does when there's nothing else he can do.

Writers can treat their mental illnesses every day.

Maturity is a bitter disappointment for which no remedy exists...unless laughter can be said to remedy anything.

Human beings are about 1,000 times dumber and meaner than they think they are.

The universe is a big place, perhaps the biggest.

Unanticipated invitations to travel are dancing lessons from God.

Please notice when you're happy.

Another flaw in the human character is that everybody wants to build and nobody wants to do maintenance.

Those who believe in telekinetics, raise my hand.

Be careful what you pretend to be because you are what you pretend to be.

A good idea will keep you awake during the morning, but a great idea will keep you awake during the night.

Educating a woman is like pouring honey into a fine Swiss watch: everything stops.

Don't be reckless with other peoples' hearts, and don't put up with people who are reckless with yours.

Part XIII

13 Additional Information for RunGTAP Developers

Additional Information for RunGTAP Developers

The following advanced topics are aimed at:

(a) those who wish to create or revise versions for RunGTAP;

(b) those who wish to revise the GTAP model which underlies all versions; and

(c) RunGTAP maintainers at Purdue.

An intimate knowledge of both GEMPACK and the GTAP model is assumed. Use the >> button above to move through the topics.

See also Can I alter the GTAP model and still use RunGTAP?

13.1 The RunGTAP concept

The RunGTAP concept

RunGTAP is a visual interface to the GTAP model, aimed at the beginner or intermediate GTAP user. It supports various **versions** of GTAP which are distinguished chiefly by level of aggregation. Most versions use the same executable files, as defined by the files GTAP.TAB and GTAP.STI and other TAB files in the RunGTAP directory. Other versions have their own executable files.

Integral to RunGTAP is a rigid system of file names. Some filenames are fixed: eg, for each version, BASEDATA.HAR contains initial flows data. In other cases particular file extensions or suffixes are compulsory: eg, parameters files must be suffixed PRM. The rigid system makes it possible for the RunGTAP user to handle the plethora of files produced by GEMPACK with relatively little pain.

RunGTAP assumes that your PC is fast relative to the size of the model you wish to solve. Files containing simulation results have only a temporary existence, unless explicitly saved. Normally, only experiment inputs are stored: if you want to see the outputs later, you run the experiment again. Intermediate files, such as LU, EN4, and EQ4 files, are not used. Post-solution programs such as DECOMP and GTAPVol, which are optional outside RunGTAP, are all run for every simulation.

13.2 Files which must be present in the RUNGTAP directory

Files which must be present in the RUNGTAP directory

Examine the file VITAL.FIL to see a complete list of the files which should be present in the RUNGTAP directory. RunGTAP reads VITAL.FIL and checks for the presence of the listed files on startup, and stops if they are not found.

Most of the files should only be changed by Mark Horridge:

If you have the source code version of GEMPACK Version 7 (or later), you can recompile the TAB files (use LTG) if the underlying model changes. Two batch files MKONEEXE and MKALLEXE help you do this. For example, type:

```
MKONEEXE Decomp
```

to recompile DECOMP.

Type:

```
MKALLEXE
```

to recompile all the TAB files.

Some files of special interest

File GTAP.STI contains the condensation.

See MAP files.

The following files may be also be changed:

RUNGTAP.BMP	the picture (paint editor needed)
RUNGTAP.TIT	text on opening page (always 4 lines)
RUNGTAP.TXT	text on 2nd page. 1 line in this file is a paragraph on screen. Edit in Microsoft Word or WordPad but remember to save as a text file.
OPTIONS.INI	self documenting INI file
INTRO.WAV	opening tune

13.3 ONEGTAP

ONEGTAP

[This note is aimed mainly at those preparing an edition of ONEGTAP for distribution.]

ONEGTAP is a version of RunGTAP used for training purposes which uses a simplified, one-region version of the GTAP model. The main program (RunGTAP.EXE) is the same as RunGTAP, but additional TAB files and special versions are supplied. The right-hand pane of **Version...Modules** dialog is used to ensure that these versions (Base, foodsub, labortax, land) use the special TAB files (Decompo, Gtapo, Gtpvewo, Gtpvolo) and that the SHOCKS and GEELAS modules do not run (nb., all these settings should be version-specific).

In addition, the following files are different for OneGTAP:

RUNGTAP.BMP (the picture), RUNGTAP.TIT (text on opening page), RUNGTAP.TXT (text on 2nd page), and OPTIONS.INI (self documenting INI file). (see Files which must be present in the RUNGTAP directory).

In the OPTIONS.INI file you could set PlayMusic=0 (activate lines by removing leading semicolon).

See also: Using several different copies of RunGTAP and OPTIONS.INI.

13.4 OPTIONS.INI

OPTIONS.INI

The OPTIONS.INI file, located in the same folder as RunGTAP.EXE, is used to:

- set some default options which might be overwritten later by users choosing **Tools...Options**
- set some obscure options which can only be altered in this way.

To learn more, view the file with a text editor.

OPTIONS.INI is mainly of interest to those preparing a new copy of RunGTAP or ONEGTAP for distribution.

See also: Using several different copies of RunGTAP and ONEGTAP

13.5 Using several different copies of RunGTAP

Using several different copies of RunGTAP

You might have several different copies of RunGTAP on your hard disk. If so, bear in mind:

- You are not allowed to run two copies of RunGTAP at once.
- Each copy of RunGTAP will store user options in its own RunGTAP.INI file, located in the same folder as RunGTAP.EXE, and use its own work directory for temporary files created during calculations.

13.6 Map Files

Map Files

Before the solution file can be viewed within RunGTAP or used by modules such as DECOMP or GTAPVOL, it must be converted to a Header Array (HAR) File using SLTOHTA. A text file, suffixed **MAP**, may be used to control which variable results are copied to the HAR file, and the headers at which such variables are located.

The MAP files, such as GTAP.MAP, should consist of lines as follows:

cols 1-4	header
col 5	blank
cols 6 plus	varname NOT followed by any dimensions

The header names in GTAP.MAP are irrelevant: the file serves only to select which variables will be visible within RunGTAP. [Exogenous variables which are not mentioned in GTAP.MAP may not be available for shocking.] The **Tools..Options** dialog allows you not to use GTAP.MAP -- in this case all variables appear on the RunGTAP results page.

The header names in other map files are important. DECOMP.MAP includes mappings for GTAPVOL.

13.7 Experiment and CMF files

Experiment and CMF files

While shock files (*.SHF) and closure files (*.CLS) are useful, experiment files (*.EXP) are the preferred way of managing simulations. They contain a subset of the commands found in CMF files.

An EXP file contains:

- full details of a closure
- full details of shocks
- the name of a parameter file
- a solution method
- a one-line description of the experiment

When RUNGTAP loads an EXP file it extracts all the EXOGENOUS statements and puts them in the box on the Closure page. It extracts all the SHOCK statements and puts them in the box on the Shocks page. It extracts METHOD and STEPS statements and updates the currently selected solution method accordingly. It extracts the VERBAL DESCRIPTION, if any, and loads it into an edit box on the Solve page. It uses the line file gtapparm = .. to determine which parameter file is to be used. Other statements in the EXP file are ignored {although future versions of RUNGTAP might use some of these; also see notes below on comments}

When RUNGTAP does a simulation it constructs and runs a temporary CMF file as follows. First it takes the entire contents of the CMFSTART file usually present in the version subdirectory. Then it adds various hard-wired file names and the name of the current parameter file. Then it adds METHOD and STEPS statements based on the currently selected solution method. Using the contents of the Description edit box on the Solve page, it adds a VERBAL DESCRIPTION statement. Then it appends the contents of the box on the Closure page, followed by the contents of the box on the Shocks page.

13.8 Files in version sub-directories

Files in version sub-directories

These fall into three groups: those which ***must*** be present, those which ***ought*** to be present, and those which ***may*** be present:

Files which must be present in each version sub-directory

VERSION.TXT contains the text on the Version Info page. If this file is missing RunGTAP will not see the version subdirectory. Each line in VERSION.TXT is a paragraph on screen. Max 255 chars per para. Edit in Microsoft Word or WordPad but remember to save as a text file. If the first character of the file VERSION.TXT is a blank, RunGTAP will display the file using a monospace font (Courier). First line of the file VERSION.TXT is used as a one-line descriptor in the Change Version file dialog.

Base data files have fixed names:

BASEDATA.HAR (flows)

SETS.HAR (sets)

DEFAULT.PRM (HAR parameters file)

An EXP file: there must be one file suffixed EXP. EXP files must contain a closure and some shocks. They may optionally contain lines:

Method = and Steps = - if not Johansen is assumed.

Verbal description = - if not ? is assumed.

file gtapparm = - if not default.prm is assumed.

The file named in the file gtapparm line should NOT have a full path; eg, correct is:

```
file gtapparm = exp2.prm;
```

RunGTAP may remember which EXP file a user was using with a particualr version last time. Otherwise, if there is a file named DEFAULT.EXP, RunGTAP will load that first. Otherwise, the alphabetically first EXP will be run.

It is good practice to ensure that whenever a version is newly opened, the EXP file which is loaded by default does contain a valid experiment.

Files which may optionally be present in version sub-directories

CMFSTART contains lines you want RunGTAP to add to the CMF file.

CLS (closure) files are simply that part of a CMF which refers to the closure (from Exogenous down to and including Rest Endogenous). You may not read the closure from an LU or environment file.

SHK files are text files of shock values. If RunGTAP finds a file named as follows: varname + .SHK it assumes that varname is the power of a tax and that the file shows what % changes in the power would be needed to remove the tax.

RunGTAP automatically use the SHOCKS program to make a series of 100% liberalization shocks, named as follows: to.shk tf.shk tpi.shk tpd.shk tgi.shk tgd.shk tfi.shk tfd.shk txs.shk tms.shk. Any or all of these files are optional.

PRE files are text files of shock values. If RunGTAP finds a file named as follows:

varname + digit 0-9 + .PRE

it assumes that it is a text file containing shock values appropriate to that variable.

SHF files are specifications of complete sets of shocks as they would appear in a CMF file. You can write values directly or refer to files of shock values. File names are presumed to refer to the version subdirectory. Comments are allowed. The first comment line should be a short description of the files contents - this will appear in the Load Shock file dialog.

If VITAL.FIL is found in a version directory it NOT used -- this is a change from previous editions of RunGTAP.

13.9 File Descriptions in File Dialogs

File Descriptions in File Dialogs

Some of the file dialogs used by RunGTAP show a one-line description of each file. For EXP files, this description is the same as the Verbal Description. For CLS and SHF files the description is the first comment in the file. More precisely it is the first line whose leftmost non-blank character is ! and which contains other non-blank characters. For Versions, the description is the first non-blank line of the VERSION.TXT file. For PRM files, it is the first line of the file history.

13.10 How to share your RunGTAP experiments with others

How to share your RunGTAP experiments with others

Use the Version Archive..Create ZIP command to save the current version on a floppy. Your colleague uses the Version Archive..Load ZIP command to install the floppy contents as a new version on her machine.

13.11 How RunGTAP detects what versions are available

How RunGTAP detects what versions are available

When RunGTAP starts up, or whenever the Change Version command is issued, RunGTAP automatically detects what versions are available as follows: any subdirectory of the RunGTAP home directory that contains a file called VERSION.TXT is presumed to contain a version. The name of a version is the same as the directory it resides in. When

RunGTAP starts up, it loads the version used most recently, if possible. Otherwise, available versions are sorted alphabetically and the first one is loaded.

13.12 Standard Closure used in test simulations

Standard Closure used in test simulations

The file GTAP.EXP in the RunGTAP directory contains the closure and other details used for the test simulation which RunGTAP performs when you create a new version. This could be edited if desired.

13.13 Comments in experiment files

Comments in experiment files

When RUNGTAP loads an EXP file it puts all the EXOGENOUS statements in the Closure box and all the SHOCK statements in the Shocks box. Of course all the informative comments that go with these statements travel to the right place. So there has to be a convention that decides whether a comment belongs in the Shocks Box or the Closure box. The fine print is: A comment line is a blank line or a line whose first non-blank character is !. All the other lines in an EXP or CMF are deemed to be statement lines. ***Comments are attached to the first statement line below them.***

13.14 Creating a new version manually

Creating a new version manually

Create a new subdirectory beneath the RunGTAP directory. The name of this subdirectory (8 letters max) will be the name of the new version. The new subdirectory must contain the following files:

VERSION.TXT text on the Version Info page

BASEDATA.HAR flows data

SETS.HAR sets

DEFAULT.PRM HAR parameters file

DEFAULT.EXP default experiment file

If the new version uses the the same aggregation as some existing version, you can copy all these files from the existing version subdirectory. It is probably a good idea to copy the file CMFSTART also.

The DEFAULT.EXP file must contain at least the following statements, which each appear just as they would in a CMF file:

```
EXOGENOUS ...     ;
```

```
REST ENDOGENOUS;
SHOCK ;
```

You should now be able to start up RunGTAP, change to the new version, and solve the model. The first time you do this, RunGTAP will automatically create the files VERSION.SLI and BASEVIEW.HAR.

Now you use can use RunGTAP to create new EXP, CLS, SHF, and PRM files. Modify closures and shocks by typing directly into the boxes for shocks and closures on the appropriate pages. Edit verbal descriptions and choose new solution methods and parameter files.

It is best to save EXP, CLS or SHF files just after successfully computing a solution, so you know that they work properly.

XSET and XSUBSET statements, if needed, must appear in the file CMFSTART.

When RunGTAP starts up, or when a new version is loaded, it loads the EXP which was last used for that version, if possible. Otherwise it loads DEFAULT.EXP if it exists. Otherwise it loads the EXP file which comes alphabetically first

13.15 The Help File

The Help File

Only Mark Horridge can change the help file. Send revised snippets to MH for him to incorporate.

13.16 Check Versions

Check Versions

The **Developer..Check Versions** command causes RunGTAP to scan through all versions checking for missing or out-of-date files. Most of the problems which might be discovered for a particular version will be fixed automatically when a simulation is run using that version, **as long as the user is able to create or alter version files**. The latter condition would **not** be met if, for example, all the RunGTAP files were being executed off a CD-ROM, or were stored in read-only network directories. If you plan to distribute RunGTAP in those ways, it is especially important to check all versions carefully before use.

Addition June 2006: **Check Versions** now reports if HAR or EXE file are of the older, LF90, type. It is preferable that these be converted to the more modern LF95 (Fujitsu) format.

- For EXE files, re-run TABLO and LTG (using LF95 or Intel or GFortran compiler).
- For HAR files, use ConvHAR or similar tool to convert to newer Fujitsu format.

LF90 has not been updated since Windows XP was introduced -- LF90-generated EXEs may lead to various obscure problems.

See also:

Lahey and Fujitsu file formats

Touch All Versions.

13.17 Can I alter the GTAP model and still use RunGTAP?

Can I alter the GTAP model and still use RunGTAP?

Yes, you can, if have GEMPACK and know how to use it. You can create either GEMSIM or Fortran models.

See Changing the economics of RunGTAP -- Overview.

See also un-omitting variables.

To create new EXE files (which run faster, cope with larger datasets, and can be distributed to others), you need the full source code version of GEMPACK with Fortran compiler.

Obviously it is assumed that you are a complete expert in GEMPACK.

13.18 Un-Omitting Variables

Un-Omitting Variables

You can safely **not** omit variables which **are** omitted in the supplied GTAP.STI file. Of course, this makes your model bigger. You need GEMPACK, Version 7 or later.

Edit GTAP.STI and place an exclamation mark ! in column 1 of the line where the omitted variable is mentioned. From DOS, type TABLO<GTAP.STI, then LTG GTAP.

To see the new variable on the results page, youll need to include it in the file GTAP.MAP. This contains lists of variables and headers. Make up any new unique header for the un-omitted variable.

Youll need to add the new variable to the closure files GTAP.EXP, GTAP.CLP and GTAP.CLG.

You should now be able to use RunGTAP as normal. RunGTAP will delete some files which depend on GTAP.EXE (such as VERSION.SLI) but these will be recreated the next time you run a simulation.

13.19 Work directory for temporary files

Work directory for temporary files

When RunGTAP is first run, a subdirectory WORK (of the RunGTAP folder) is created for temporary files. All files in the directory are deleted at the start of each solution process. Developer Mode makes it easy for you to look at the contents of this directory.

13.20 Touch all versions

Touch all versions

The **Touch all versions** command is a specialized option designed for GTAP developers at Purdue. Be **very cautious** about using it.

Touch is computer jargon for changing the date of a file.

The **Touch all versions** command goes through all the version subdirectories, setting the date stamp of each file to the current moment.

There are 2 reasons you might want to do this.

First, it is a way to frustrate RunGTAPs habit of deleting obsolete files. Normally, if you recompile GTAP.EXE, RunGTAP will want to recreate very many files (such as SLI files). **Touch all versions** command will fool RunGTAP into thinking that all version files are up to date.

Second, if you **Touch all versions** just prior to building a release, it will be easy to see later which files have been modified by a user since RunGTAP was installed.

A possible procedure:

First run Check All Versions to identify out-of-date files. Fix any problems.

Then recompile GTAP.EXE, if necessary.

Then run Touch All Versions to mark all files as up-to-date.

13.21 64-bit EXE files will not run under 32-bit Windows

64-bit EXE files will not run under 32-bit Windows

Each Windows operating system is either 32-bit or 64-bit.

Also, each Windows EXE file is either 32-bit or 64-bit.

You can run 32-bit EXEs under 64-bit Windows, but you cannot run 64-bit EXEs under 32-bit Windows.

You should be aware of this if you create Version Archives for others to use: if the Archive contains a 64-bit EXE, that Archive is not useable by those who have only a 32-bit Windows operating system.

RunGTAP tries to warn you:

- if you create a Version Archive that contains a 64-bit EXE.
- if you are running 32-bit Windows and the current version contains a 64-bit EXE.

Part XIV

14 Changing the economics of RunGTAP -- Overview

Changing the economics of RunGTAP -- Overview

This help topic is long and complex. Try maximizing the Help window: you will still need to scroll to

read the whole topic. We suggest that you PRINT the topic from within Help (via right-click).

Then read the printed sheet twice. Later, refer to the printed sheet as you work with RunGTAP.

See also: Choosing and Disabling Modules

The specification of the GTAP model and of the tools for interpreting results (such as GTAPView and the welfare decomposition) are contained within a host of ancillary files supplied with RunGTAP. Those files are divided into 8 groups (or modules) each of which contains one TAB file. The files comprising the standard RunGTAP modules are listed in the table below. The files within a single module have the same name, but with different extensions.

Module	TAB file	Other files*
Main model	GTAP.TAB	GTAP.MAP, GTAP.EXP, GTAP.MIN, GTAP.SIN, GTAP.CLG, GTAP.CLP
Shocks	SHOCKS.TAB	
GTAPView	GTPVEW.TAB	
GTAPVol	GTPVOL.TAB	
Welfare Decomp	DECOMP.TAB	DECOMP.MAP
Partial Elasticities	PEELAS.TAB	
GE Elasticities	GEELAS.TAB	GEELAS.MAP
Altertax	ALTPAR.TAB	

* As well as the files listed, each TAB file requires a STI file, and **either** GSS and GST files **or** AXT, AXS, and EXE files. For example, RunGTAP comes with GTAP.STI, GTAP.AXS, GTAP.AXT, and GTAP.EXE for the standard GTAP model. If **both** GSS/GST **and** AXT/AXS/EXE files are found, the GSS will be used.

You can specify that RunGTAP will use a different module, if you can supply or have been given the appropriate files. For example, you might create your own improved DECOMP module. You would have to create your own complete set of files perhaps called DECOMPX.TAB, DECOMPX.STI, DECOMPX.AXS, DECOMPX.AXT, DECOMPX.EXE and DECOMPX.MAP. You need to have a recent version of GEMPACK to create these files (recent means version 7.0 or later).

These files could be stored in the main RunGTAP directory. In that case, you could specify that DECOMPX was used either for all versions (aggregations) of the model, or just for specific versions. Or, if DECOMPX is to be used for one version only, you should store the DECOMPX files in that version directory.

RunGTAP also lets you disable any module (except for the main model). This might be especially useful while developing new modules. Possibly, your version of GTAP, GTAPQ, might prevent the original GTPVOL module from running properly. You could switch off GTPVOL for a while, until you had time to create your own GTPVOLZ module which would cooperate with GTAPQ. Again, you can disable a module either globally (for all versions) or locally (for specific versions).

To see an example of model customization, change to the TP14 version which is supplied with RunGTAP. Then choose the menu item **Version...Modules**. You will see that for this version the GTAP module is replaced by the GTAPTP14 module (which adds monopolistic competition).

Notice that in the examples above, the names of the customized TABLO programs (DECOMPX, GTAPTP14) are all formed by adding letters to the names of the original TABLO programs (DECOMP, GTAP). RunGTAP insists you follow this rule.

See also:

Choosing and Disabling Modules

Notes for Module Developers

14.1 Choosing and Disabling Modules

Choosing and Disabling Modules

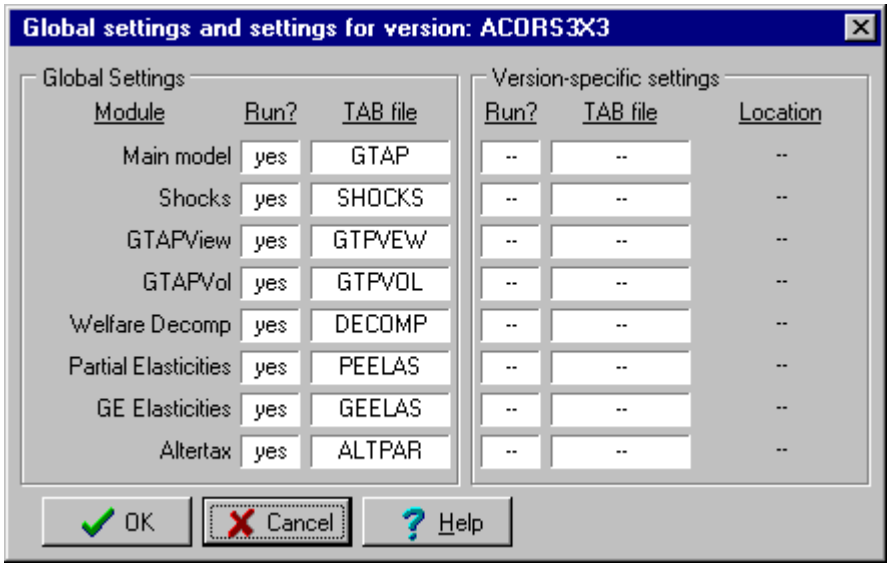
This help topic is long and complex. Try maximizing the Help window: you will still need to scroll to

read the whole topic. We suggest that you PRINT the topic from within Help (via right-click).

Then read the printed sheet twice. Later, refer to the printed sheet as you work with RunGTAP.

8 standard modules (or groups of files) supply the economic content of RunGTAP. To learn more about modules, see Changing the economics of RunGTAP -- Overview.

The **Version...Modules** dialog is used to disable selected modules, or to specify alternative replacement modules.



The **left-hand panel** of the dialog controls the default global settings which apply to all versions (aggregations), unless over-ridden for particular versions. In the left panel, the smaller boxes in the **Run?** column can be used to disable a module (ie, prevent it from running). By clicking on the **Run?** boxes you can toggle between **yes** (*do* run the module) and **no** (do *not* run the module). However, you may not disable the first, Main model, module.

Still in the left panel, the boxes in the **TAB file** column specify which TAB file in the main RunGTAP directory is used for a particular module (row). If you click on one of these boxes a TAB file selection dialog is launched which is described below.

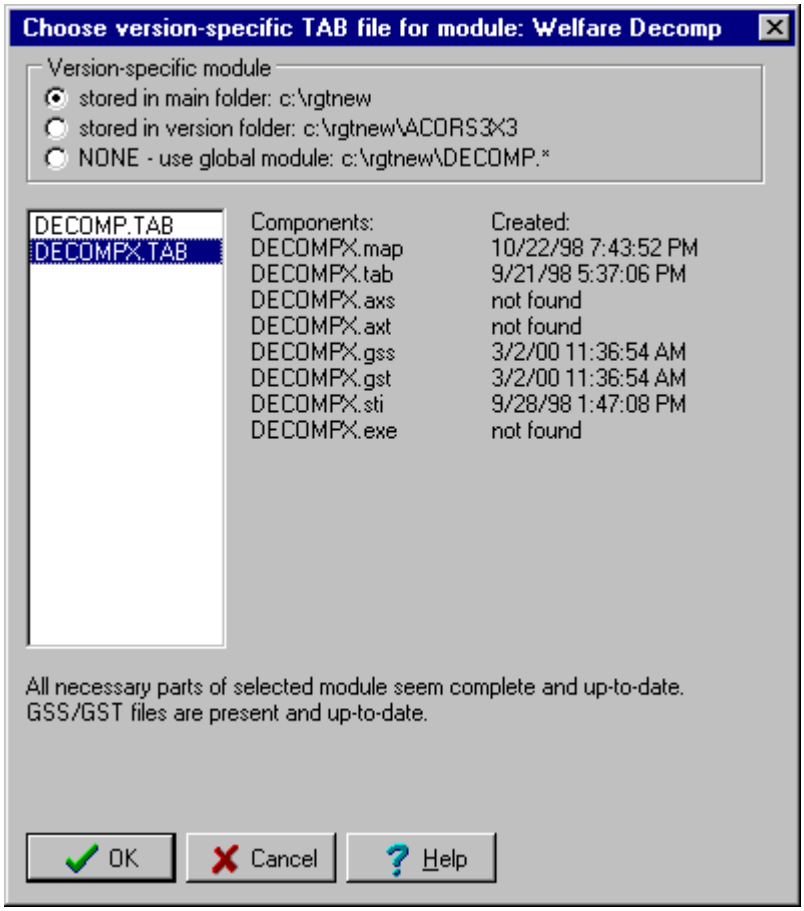
The **right-hand panel** of the dialog allows you to over-ride the default global module settings for the current version only (current version name appears in window title bar). In this panel, the symbol -- has a special meaning: it means follow the corresponding global setting in the left panel. You can click on the boxes in the **Run?** column to toggle between three settings: **yes** (*do* run the module); **no** (do *not* run the module); and -- (follow global setting). Again, you may not disable the first, Main model, module.

Still in the right panel, the **TAB file** boxes specify which TAB file is to be used for this module and version. Here, the -- symbol means that this version uses the default global TAB file in the main RunGTAP directory. If you click on one of these boxes a TAB file selection dialog is launched which is described below.

If a version-specific TAB file has been selected, the last, **Location**, column indicates whether that TAB file is located in the version directory or in the main RunGTAP directory. If no version-specific TAB file has been selected, the **Location** column shows --.

Normally most of the boxes in the right-hand panel are filled by dashes (--), indicating that this version is following the default global settings.

If you click on the version-specific **TAB file** column, the following dialog appears:



The dialog controls which files are to be used for a specific module (here Welfare Decomp) for the current version. The first thing for you to do is to choose an option at top left.You can tell RunGTAP to search for files in either the main RunGTAP directory or in the subdirectory of the current version, or you can choose for this version to follow the global settings (in which case there is nothing more to do). Above, the first option has been chosen, and RunGTAP has displayed at left a list of TAB files in the main RunGTAP directory which might be used for the DECOMP module. These are DECOMP.TAB (the original file) and DECOMPX.TAB (a customized version). The other TAB files in the main directory are not listed because RunGTAP insists that replacements for DECOMP.TAB are named by adding letters to the original name.

In this case DECOMPX.TAB has been chosen, and the right hand side of the dialog lists the other components that are needed. There are no AXS, AXT or EXE files, but this is OK since GSS and GST are present. Otherwise, missing files cause a warning to appear. RunGTAP also checks that the AXS/AXT/EXE or GSS/GST files have been created since the TAB and STI files were last modified. If all necessary files are present and up-to-date you can press OK to exit the dialog.

If **both** GSS/GST **and** AXT/AXS/EXE files are found, the GSS will be used.

The same dialog is used to choose which default global modules are used. In this case the 3 options at top left disappear -- you must specify a TAB file located in the main RunGTAP directory.

RunGTAP stores both global and version-specific settings in INI files. Each version has its own VERSION.INI file stored in the version folder. This means that if you send someone else an archive of your own version which is configured to use your own modules, the other person can load your archive into a new version folder and the new version will automatically be configured to use the right modules. This works especially well if you choose to store your custom module in the version folder -- then the version archive will automatically contain all the right module files.

Warning: you cannot revert to default module settings simply by deleting (using Windows Explorer) specialized module files from a version directory. That will make RunGTAP quite upset! The correct approach is to use **Version..Modules** to set module usage for that version back to global default settings. Then run a test simulation. ***Then*** you can delete the version-specific files which are no longer needed.

See also:

Notes for Module Developers

14.2 Notes for Module Developers

Notes for Module Developers

This help topic is long and complex. Try maximizing the Help window: you will still need to scroll to

read the whole topic. We suggest that you PRINT the topic from within Help (via right-click).

Then read the printed sheet twice. Later, refer to the printed sheet as you work with RunGTAP.

See also Creating Modules for others to use

Can I create my own version of GTAP.TAB?

NO! In the context of RunGTAP, version means database. You must learn to talk the talk. Instead think about creating a **custom module** which will replace the standard GTAP module and will be be named GTAPxxxx. Your new module will be used with one or more versions (aggregations). The most important **component** (file) in your new module will be called GTAPxxxx.TAB.

Getting started with your own modules

First create a new version, called perhaps NEWVER1, which is based on some existing version. You will work in the NEWVER1 folder so that your mistakes do no lasting damage. Now, in DOS go to the RunGTAP folder (probably C:\RunGTAP) and issue the following command:

```
copy GTAP.* NEWVER1\GTAPCOOL.*
```

The effect is to create copies of all files in the standard GTAP module. The copies are located in the NEWVER1 subdirectory and are named GTAPCOOL.

Now start up RunGTAP, change to the NEWVER1 version and launch the **Version..Modules** dialog. Click on the Version-specific, Main model, TAB file box (at top right): the **Choose TAB file** dialog will appear. Choose the **stored in version folder** option; GTAPCOOL should appear in the list below. Click on GTAPCOOL; the module components will be listed at right. They should be all present and up-to-date. Click OK to return to the **Version..Modules** dialog where you will now see that the GTAPCOOL files will be used instead of GTAP for the current version. Click OK again and run an experiment -- the progress indicator should inform you that your GTAPCOOL module is being used.

Thus far, of course, all GTAPCOOL files are identical to their GTAP counterparts. Probably your next step would be to exit RunGTAP, modify GTAPCOOL.TAB in some way, edit GTAPCOOL.STI (changing GTAP to GTAPCOOL in 3 places) and use TABLO to create AXS/AXT/EXE or GSS/GST files. Then go back into RunGTAP and test the modified GTAPCOOL model. Remember that if you add a new variable, it must be listed in GTAPCOOL.MAP to appear on the results page. You might also have to modify the closures in GTAPCOOL.EXP, GTAPCOOL.CLP and GTAPCOOL.CLG.

When RunGTAP notices that you have changed a TAB file, it may delete older files that depend on that TAB file. You might be warned that these files are missing. Usually the files will be regenerated next time you run a simulation and so the warnings will go away.

Module Names

You must choose a name for your module by adding a few letters to the original name. The new name cannot be more than 8 letters long. Since most of the original module names are 6 letters, this means that often your module is distinguished by an additional 1 or 2 letters. For the main module (GTAP) you can add up to 4 letters. The 8-letter restriction is required so that RUNGTAP works with GEMPACK 6.0. The requirement that the first letters of your module name match those of the orginal module is enforced so that RunGTAP can tell the purpose of a TAB file from its name. Thus RunGTAP does not allow the user to choose, say, DECOMPX.TAB as a replacement for SHOCKS.TAB.

Should I put my new module in the main RunGTAP folder or in a version folder?

Bear the following in mind.

The advantages of locating your new module in a version folder are:

(a) easier to tell what is going on; mistakes only affect one version.

(b) The **Version..Archive** command stores all version files within a single ZIP -- including your new module. So you can back up your work. You could also send the zip archive by email to another RunGTAP user; she can load the archive to create a new version which will automatically be set to use your new module. Even if she had herself recently created her own module with the same name as yours, there would be no conflict.

The disadvantage of locating your new module in a version folder is that it can be used only for that version. You might end up with multiple copies of your new module, duplicated in several version folders. You could avoid that by locating it in the main RunGTAP folder.

Suppose you created a new improved DECOMPX module which worked with the standard GTAP module and data files. It would be logical to place it in the main RunGTAP folder and you could specify that it be used by default in place of the supplied DECOMP module.

Suppose again that you created a new GTAPOIL module to replace the main GTAP model. Suppose you have 5 alternative databases for GTAPOIL -- so you need 5 different version folders. And suppose that GTAPOIL requires some extra coefficients on the parameter files so that it will not run with the standard versions. In this case it would be natural to place GTAPOIL in the main RunGTAP directory so that all your 5 versions could use it. However, you should not assign GTAPOIL to be the global replacement for GTAP, since it requires special parameter files. Instead, you must change to each of the five versions in turn, and use the **Version..Modules** command to specify that GTAPOIL is used for the current version.

Is there a limit to the changes I can make?

You have to ensure that your changes to the main model (GTAP module) do not conflict with the requirements of the other modules (Decomp, GtapVol, etc), or else you must disable them until you can create new modules to replace them. In general, it will be easier to add things than to take them away. If you remove or redimension data items used by the other programs, problems will arise. Similarly, omitting, substituting, or backsolving variables which were not so treated before could cause problems.

Another factor that constrains your changes is that RunGTAP writes the CMF files. Therefore, your modules must use the same set of file-names for input and output as the original modules. You cannot add new input or output files (but see XSETS and the CMFSTART file). Instead you should put new input data on one of the existing input files: SETS.HAR, BASEDATA.HAR, or XXX.PRM. To create new output headers, add write statements to the GTAPView TAB file.

Variables with more than 3 dimensions do not appear in the RunGTAP user interface.

If you find you are diverging too far from the RunGTAP framework, consider using the standard GEMPACK program RunGEM. Although similar to RunGTAP, it lacks many of its GTAP-specific features. It will work with nearly any GEMPACK model.

I only want to alter one little file. Must I duplicate the entire module?

Sorry, yes. Of course, you have the option of directly editing (without renaming) a standard file (perhaps GTAP.MAP) which is used by default for all versions. But be careful !

Short Description of less obvious module components

GTAP.MAP:	controls which results the user sees.
GTAP.EXP:	The default experiment.

GTAP.MIN:	Model Information file made by TABLO.
GTAP.SIN:	Sensitivity Information file: if in doubt, copy existing file. First line of SIN file must be #Parameters; any more is optional (see SSA.CHM for syntax rules).
GTAP.CLG:	the closure used for SAGEM GE Elasticities.
GTAP.CLP:	(most of) the closure used for ALTERTAX runs.
DECOMP.MAP	pipes results to DECOMP and GTAPVOL programs.
GEELAS.MAP	pipes results from SAGEM to GEELAS program.

Is RUNGTAP compatible with modules created by GEMPACK version 6.0?

Mostly, yes, although RunGTAP does not officially support GEMPACK 6.0. But you should be aware that some new features of later GEMPACK versions will not be supported by modules created with GEMPACK version 6.0. Long directory names are an example. If you installed RunGTAP in C:\My Projects\RunGTAP you might find that modules created with GEMPACK 6.0 would not run. The AnalyseGE tool also requires that the main GTAP module was made with GEMPACK 7.0 or later. Also, executables made with GEMPACK 7.0 or later will flexibly handle both Formats of HAR files.

See also Creating Modules for others to use

14.3 Creating Modules for others to use

Creating Modules for others to use

You need a GEMPACK licence to create new modules. There are two main sorts of licence:

- Cheaper, **Executable-Image** licence allows you to make GEMSIM modules. You need GEMSIM to run these, and you need a GEMPACK licence for GEMSIM. So this technology only lets you share modules with other GEMPACK licencees [who must have the ***same*** version of GEMSIM]. Also, larger models cannot be solved with GEMSIM.
- More expensive **Source-Code** licence allows you to make EXE modules. If you do this right, anyone can run these EXEs. They also run faster than GEMSIM modules.

This note concerns the second route, and tells the right way to compile EXE modules that others may use. You will have one of these Fortran compilers:

- LF90
- Lahey LF95
- Intel Fortran
- GFortran

LF90 is NOT recommended at this time. LF90 uses files (such as HAR files) of a different format to newer HAR files. This may raise difficulties if you attempt to share LF90 EXEs and HAR files with users of modern RunGTAP installations (see Formats of HAR files).

Always test distributed EXEs

To do this, create the EXES, then rename your GEMPACK directory by adding an x to the name. Perhaps it would become C:\GPX.

Now your system will be unable to find your GEMPACK licence LICEN.GEM. Test all EXEs which are to be distributed under these conditions.

For RunGTAP, testing means: solving the model and testing the systematic sensitivity options. Each GEMPACK EXE must be exercised. For GTAPAgg, time series must be tested if available. After the testing, rename your GEMPACK directory back to original by removing the x from their name.

14.4 Notes on altering GEELAST

Notes on altering GEELAST

Prof Hertel inquired:

I want to use GEELAST to compare two GTAP-type models. One is relatively standard and one is rather different. After a bit of fiddling, I cant seem to figure out where to dig in to modify the appropriate CMF file do this for the non-standard model (sagem.cmf?). Can you direct me to the right doco?

Reply:

Search in the RunGTAP help for keywords PRICELAS, GEELAST and GEELAS. In topic Notes for Module Developers, the short Description of less obvious module components lists

GTAP.CLG: the closure used for SAGEM GE Elasticities.

GEELAS.MAP: pipes results from SAGEM to GEELAS program.

You should study also GEELAS.TAB. In version modules, you can activate or disactivate the GEELAST -- it is often switched off.

Note that

(a) if your non-standard model has very different theory, GEELAS.TAB will need to be changed.

(b) Substituted variables, needed by GEELAS, are recovered by repeating in GEELAS equations which are in main model -- the two TABs must be consistent.

(c) GEELAS.TAB re-uses some coefficients to conserve memory -- leading to less transparent code.

Delete the GEL file to force RunGTAP to recompute the GEL. Switch on developer mode and jump into the work folder...You will see the PRODUCED files:

File SAGEM.cmf:

```
Verbal Description = GEelast run;
solution file = GEelast;
```

File GESLTOHT.inp:

```
geelast ! sl4 file
```

File PricElas.cmf: [input to GEELAS]

```
File GSLTOHAR = c:\rungtap5\work\geelast.sol;
```

showing sequence of jobs and files. RunGTAP.BAT had:

```
SAGEM.exe -cmf SAGEM.cmf >SAGEM.log
SLTOHTA.exe -sti GESLTOHT.inp >GESLTOHT.log
GEELAS.exe -cmf PricElas.cmf >PricElas.log
```