Release 9.0-003 (August 2006)

Ken Pearson and Michael Jerie
August 2006

1. Introduction

This document describes Release 9.0-003 (August 2006) of GEMPACK.

This release fixes some bugs in Release 9.0 (April 2005) and contains two enhancements (see sections 2 and 3).

1.1 Installing Release 9.0-003 on a Windows PC

On a Windows PC, Release 9.0-003 is supplied on a GEMPACK CD.

You can carry out the installation following the same procedure you used to install Release 9.0. That is, the installation instructions are in GEMPACK document **GPD-6** for the Source-code Version, (GPD6.PDF on the CD), and **GPD-7** for the Executable-Image Version (GPD7.PDF on the CD).

1.2 Bug Fixes in Release 9.0-003

This Release fixes all known bugs in Release 9.0. This includes the following fixes:

* Fix to Shock file bug
* Fix to SLTOHT Spreadsheet Table bug
* Fix to TABLO-generated Program $POS Intertemporal bug

For more details see the GEMPACK web page http://www.monash.edu.au/policy/gp90bug0.htm

2. Enhancement – Sparse Header Arrays

Header Arrays containing real numbers which are of type RE (these contain set and element labelling) or type RL (these do not contain set and element labelling) can now be written in sparse form. This is normally done if at most 40% of the data values in the array are nonzero (that is, if at least 60% of the values are zero). When arrays are very sparse (that is, contain a high percentage of zero values), this can reduce the size of the Header Array file considerably. Arrays of integers and character strings are never written in sparse form, nor are 2-dinemsional arrays of type 2R. [See section 3.1.1 of GPD-4 for more about array type.]

> For example, the main data file for the USAGE model of USA is reduced from about 55 Mb to about 6 Mb by writing the relevant arrays in sparse form.

When a program writes a Header Array in sparse form, it just writes the nonzero values in the array and the corresponding positions (integers) to the file.

> For example, consider a 4x3 array in which the only nonzero entries are the (1,2) entry equal to 6.102 and the (4,3) entry equal to 7.366. Then this array would now normally be written in sparse form. On the file would go
>
> - the array containing the two nonzero values [6.102, 7.366], and
> - the array indicating the positions of these values in the whole array [5, 12]. [The positions are calculated by varying the first index fastest, and so on, as for component number as explained in section 5.3 of GPD-3.]

If you use any of the GEMPACK Fortran or Windows programs which can write sparse header arrays, you must also use appropriately recent versions of all relevant programs which can read such files. For example, the Release 9.0 (April 2005) versions of SEEHAR, CMPHAR, GEMSIM, ViewHAR, ViewSOL and AnalyseGE cannot read Header Array files which contain sparse headers. Accordingly

you must make sure that you are using recent versions of these programs which can read sparse headers.

For the present, the programs will only write sparse headers if you specifically request them to (see section 2.1 below). Then the program decides whether to write each header in sparse form. The programs will write arrays of types RE and RL in sparse form if at least 60% of the values in the array are zero.

The default for Release 9.002 (or later) is that the programs do not write sparse arrays. You have to take one of the actions described below in section 2.1 to write sparse arrays.

When we put out Release 10 (perhaps some time in 2007), we will probably change the default so that then the programs will write sparse headers whenever the array is sparse, unless you specifically request them not to.

2.1 Writing Sparse Headers

If you are preparing a Header Array file to send to a friend or colleague who does not have the recent GEMPACK software which can read sparse header arrays, you will want to make sure that you do not write any arrays in sparse form.

2.1.1 GEMPACK Fortran Programs

Versions of the GEMPACK programs beginning with Release 9.0-002 (November 2005) can all read header arrays written in sparse form. [But the Release 9.0 (April 2005) versions of these programs cannot read such header arrays.]

For any of the GEMPACK Fortran programs (for example, GEMSIM, TABLO-generated program, SLTOHT, MKHAR, RWHAR), you can ensure that they do write header arrays in sparse form by selecting program option at the first option screen (the screen where you can select such options as LOG, STI, SIF...)

WHS Write Headers in Sparse format

when you run the program.

WHS is a new basic program option which is available in all the Fortran programs (Release 9.0-002 or later). It does not show on the options screen when the program runs, but you can always select it. By default, option

–WHS Do not Write Headers in Sparse format

applies. You need to enter **WHS** to turn that off.

In Command files for GEMSIM or TABLO-generated programs (but not for SAGEM), you can include the either of the statements

```
WHS = yes|NO ;      ! NO is the default for the present.
                    ! This will change to YES for Release 10.
```

"WHS = no ;" ensures that the program will not write any header arrays in sparse form.

"WHS = yes ;" asks the program to write header arrays in sparse form when the array is sparse.

2.1.2 Windows Programs

The relevant programs are ViewHAR, ViewSOL and AnalyseGE. RunMONASH etc may also be affected since it sometimes reads Header Array files directly.

Versions of these programs supplied with Release 9.0-002 (November 2005) or later can all read header arrays written in sparse form. [But the Release 9.0 (April 2005) versions of these programs cannot read such header arrays.]

If you want to ensure that these programs do not write header arrays in sparse form, you need to go to the ViewHAR options menu via ***File..Options*** . Then make sure that the option

Use sparse disk storage

is NOT checked. This option in ViewHAR controls the writing of header arrays for the other relevant programs (such as AnalyseGE). In the version of ViewHAR supplied with Release 9.0-002 (or later) of GEMPACK, this option is greyed out so that it is never checked.

2.2 Download Programs Which can Read Sparse Headers

Versions of the GEMPACK Windows and Fortran programs produced before about October 2005 cannot read sparse headers on Header Array files.

If you produce Header Array files which contain sparse headers, or if you obtain such files from others, you will need to download from the GEMPACK web recent versions of the relevant programs (as supplied with Release 9.0-002 or later of GEMPACK) which can read such headers. This includes the Windows programs ViewHAR, ViewSOL, AnalyseGE and RunGEM and includes the Fortran programs SLTOHT, MKHAR, RWHAR, MODHAR and CMPHAR.

2.2.1 If You Use RunGTAP or RunMONASH etc

If you use RunGTAP or RunMONASH etc, you will need to copy versions of the relevant Fortran programs as indicated in sections 12.1 and 12.2 of GPD-5. You will also need to ensure that your RunGTAP or RunMONASH etc has access to recent versions of ViewHAR, ViewSOL etc.

3. Enhancements to Simulation Programs

This relates to the simulation programs GEMSIM, TABLO-generated programs and SAGEM.

3.1 LU Decomposition via MA48AG

As indicated in section 5.2.1 of GPD-5, compressions (that is, garbage collection) are not done by default. We have found that, for some models, this leads to unnecessarily large values of MMNZ etc. In some cases, the required values exceed the amount of memory available. Accordingly, if compressions are not turned on via the Command file, we have decided to let the programs turn on compressions in certain circumstances, as described below.

- When the programs increase MMNZ etc, the programs usually try to increase more than the minimum (usually about 10% more – the exact values depends on whether or not you have a statement of the form "MA48 increase_MMNZ = " in your Command file – see section 5.2.2 of GPD-5). If the program is not able to increase MMNZ by more than the minimum required for it to continue, and if this has happened a few times already, the program will turn on compressions within MA48AG.
- If the program is asked to increase MMNZ and if this has already been done a few times, the program will turn on compressions within MA48AG rather than increasing MMNZ etc.

In such cases, you will see a message of the form

```
%%INFORMATION. Looks like cannot increase MMNZ etc much more. So,
      instead of increasing MMNZ etc,',
   are turning on compression within MA48AG and
      continuing with the same MMNZ etc values.
```

[This is the message for the first case above. In the second case, the "Looks like…much more" is replaced by "Have increased MMNZ etc a few times before".]

In each case, compressions are only turned on when the routine MA48AG is the routine asking for the increase.

Once compressions are turned on, it is as if the statement

```
MA48 compress = yes ;
```

had been in the Command file.

4. New Command File Statements

4.1 For GEMSIM and TABLO-Generated Programs

WHS = YES|no ; ! See section 2.1.1

5. INDEX