

ISSN 0111-1760

**University of Otago  
Economics Discussion Papers  
No. 0213**

November 2002

---

## **General-to-Specific Modelling Using PcGets**

P. Dorian Owen

**Abstract:** This review examines the facilities provided by PcGets version 1.0, an OxMetrics module designed to implement automated general-to-specific model selection in the context of cross-section or dynamic time-series models that are linear in the parameters. A cross-section empirical example is used to illustrate the steps involved, the output produced and the options available for modellers.

**Keywords:** Econometric modelling; General-to-specific modelling; Model selection; Econometric software; PcGets; GiveWin

Contact details:

Dorian Owen  
Department of Economics  
School of Business  
University of Otago  
PO Box 56  
Dunedin, New Zealand

Email: [dowen@business.otago.ac.nz](mailto:dowen@business.otago.ac.nz)  
Tel: (64) 3 479 8655  
Fax: (64) 3 479 8174

## 1. Introduction

General-to-specific (Gets) model selection is a central feature of what is often referred to as the ‘LSE’ methodological approach to econometric modelling. It involves the formulation of a ‘general’ unrestricted model that is congruent with the data and the application of a ‘testing down’ process, eliminating variables with coefficients that are not statistically significant, leading to a simpler ‘specific’ congruent model that encompasses rival models (see, for example, Hendry, 1995, p.365).<sup>1</sup> This approach to carrying out empirical work has its roots in the work of Denis Sargan (e.g., Sargan, 1964) but is now most closely associated with David Hendry. Hendry, with a number of co-authors, has, over a period of more than 25 years, progressively developed and applied the methodology, in particular in a series of influential time-series studies of aggregate demand-for-money and consumption functions (e.g., Davidson, Hendry, Srba and Yeo, 1978; Hendry and Mizon, 1978).<sup>2</sup> The influence of the LSE/Hendry methodology has been boosted by its linkages to developments in the literature on cointegration (Pagan, 1995). Granger’s Representation Theorem (Engle and Granger, 1987) highlights the connection between cointegration (as a statistical characterization of a long-run relationship between the relevant variables) and the existence of an equilibrium- (or error-) correction mechanism (EqCM) representing the dynamics of adjustment to the long-run relationship, an early feature of Hendry’s models. While the LSE/Hendry approach has had a major influence on applied work (particularly in the UK, Europe and Australasia, but to a lesser extent in North America), it has not been without its critics (e.g., Keuzenkamp, 1995; Hansen, 1996; Faust and Whiteman, 1997) and the extent to which applied researchers draw on it has often been selective; e.g., while there is widespread use of EqCMs and increased emphasis on diagnostic testing and model evaluation, explicit applications of a Gets reduction strategy are less common. However, it is one of the few coherent and relatively comprehensive methodological bases for applied econometric modelling.

The Gets approach to model selection has been significantly enhanced by recent developments due to Hoover and Perez (1999), Hendry and Krolzig (1999, 2001, 2002) and Krolzig and Hendry (2001). In order to evaluate the properties of Gets modelling in a Monte Carlo study, Hoover and Perez (1999) take the first important steps in automating the Gets process. A common criticism of Gets modelling has been the lack of clear guidelines on the best way to proceed when simplifying the general model,

reflected in earlier published work by relatively imprecise statements as to how the specific model was obtained (Pagan, 1987).<sup>3</sup> Hoover and Perez's explicit specification of a simplification algorithm, a necessary condition for the application of Monte Carlo methods, deflects this criticism of earlier Gets modelling. They also introduce a number of other improvements. They examine multiple search paths, thus avoiding 'path dependency', which can seriously affect the properties of a simplification algorithm based on a single search path; e.g., a simple decision rule, such as successively removing the variable with the lowest absolute *t*-value, can easily result in being stuck in a search path that has deleted relevant variables. They also emphasize the importance of considering only model reductions that do not fail diagnostic tests in order to retain congruence, and the use of overlapping sub-sample testing to aid in the overall assessment of the 'reliability' of the significance of the coefficients.<sup>4</sup>

Hendry and Krolzig (1999, 2001) and Krolzig and Hendry (2001) develop the automated-algorithm approach into a publicly available program that can be used to implement Gets modelling in applied research. In doing so, they further refine the search algorithm to include: 'pre-search simplification', multiple reduction paths based on deletion of blocks of variables as well as single variables, implementation of encompassing tests to distinguish between competing candidate congruent models that emerge from different search paths, and the use of an information criterion to make a final selection if encompassing tests fail to pick a unique dominant final model. These features are discussed in more detail below.

PcGets 1.0 is the first release of Hendry and Krolzig's (2001) automated Gets modelling program. It is written in Ox, a powerful object-oriented matrix language (Doornik, 2001), but provides a similar, interactive, menu-driven interface to the other modules in the OxMetrics suite of programs (including PcGive, PcNaive, STAMP, and the GiveWin version of TSP). The discussion in the following sections relates to version 1.0 of PcGets, although some comparisons with the current beta version are also made.

## **2. Availability, Pricing and Documentation**

PcGets 1.0 was released in July 2001. PcGets and the other modules in the OxMetrics suite of programmes are distributed by Timberlake Consultants Ltd. Contact details for Timberlake in the UK and distributors in North America, Spain, Portugal, Italy and

Brazil can be found on their websites (<http://www.timberlake.co.uk> and <http://www.timberlake-consultancy.com>). Additional information and updates of OxMetrics software are available at the OxMetrics sites (<http://www.oxmetrics.net/>, <http://pcgive.com/pcgets/> and <http://www.nuff.ox.ac.uk/users/doornik/>).<sup>5</sup>

PcGets requires GiveWin, which acts as a common ‘front end’ for all the OxMetrics modules; GiveWin manages data and output files, provides facilities for loading, exporting, editing and transforming the data and has a comprehensive range of excellent graphics capabilities for plotting data and descriptive statistics. Further details of version 2 of GiveWin are discussed in Urga’s (2001) review of PcGive version 10.

A single-user (academic) license costs £250 for the PcGets and GiveWin programs, including both manuals (Hendry and Krolzig, 2001; Doornik and Hendry, 2001).<sup>6</sup> Multi-user and network licenses are also available (e.g., academic licenses cost £650 for five users or £1500 for an unlimited number). The PcGets manual is similar in style to that of recent versions of the PcGive manual; it doubles as a user manual, including six tutorials covering different aspects of the modelling process, and a textbook outlining the methodology and econometric theory underlying the Gets approach to model selection. Motivated by past criticisms of Gets modelling, the latter sections include a chapter reviewing and attempting to refute these criticisms, and, briefly, considering other competing methods for model selection; Hendry and Krolzig (2002) consider these issues in more detail. The help system available in PcGets is essentially an html-format version of the printed manual.

### **3. Installation**

OxMetrics modules, including PcGets and GiveWin, are supplied on a CD-ROM containing a menu-driven installation routine together with paper-copy licenses containing separate registration codes for each module; the code numbers have to be typed in during installation and before running the program.<sup>7</sup> The programs run on Windows 95/98/ME/NT/2000/XP.<sup>8</sup> GiveWin has to be installed before PcGets. After installation, opening GiveWin will reveal the set of available Ox modules, as shown in Figure 1. Trial installations on different computers revealed few problems; in one case some of the installed modules did not appear on the module list but this was relatively easily dealt with, either by repeating the installation procedure from the CD-ROM

menu or by using Add/Remove Module from the Modules pull-down menu.<sup>9</sup> Inside GiveWin, PcGets can be activated by clicking on PcGets in the left-hand workspace window or by clicking Start PcGets in the Modules menu.

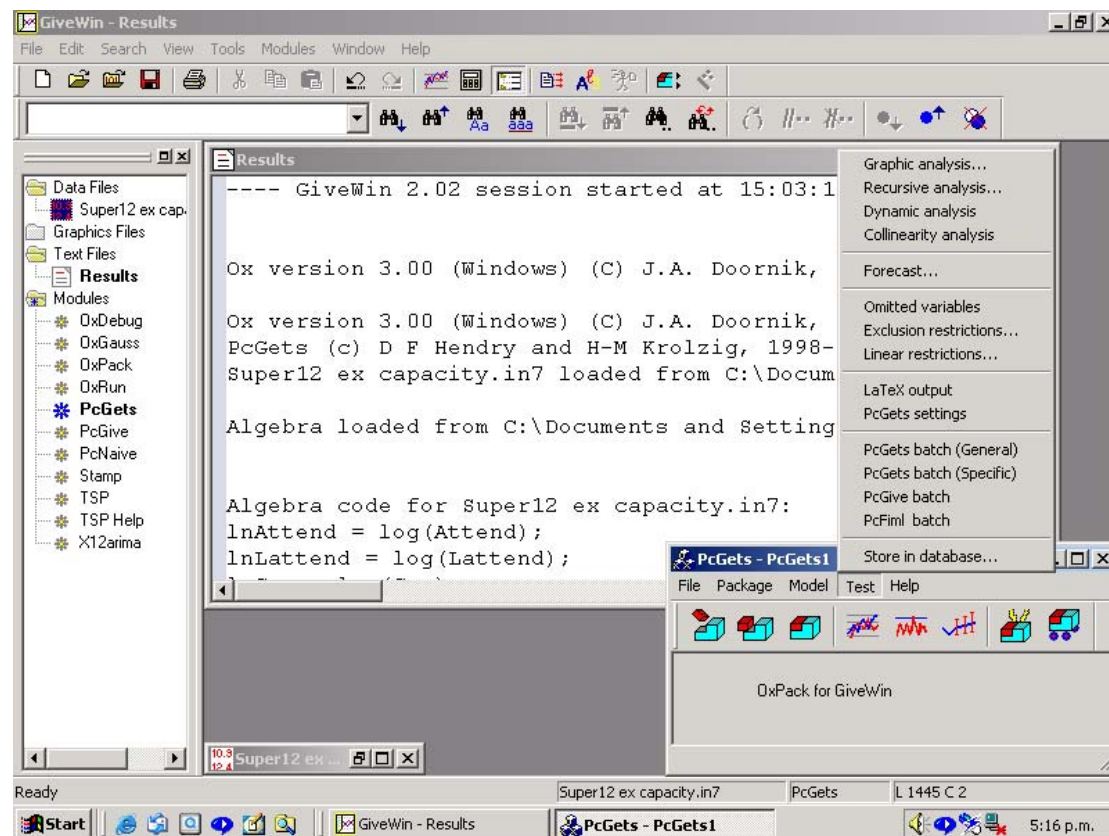


Figure 1. The GiveWin/PcGets interface.

#### 4. An Illustrative Example of Model Selection in PcGets

As an illustrative example, we consider some of the aspects of using PcGets in a study modelling attendance at Super 12 rugby union matches in New Zealand. The focus here is on the mechanics of running the ‘testimation’ procedures in PcGets and the output produced, rather than on the model formulation and interpretation of results, which are discussed in Owen and Weatherston (2002). A Gets approach was considered to be a useful choice for modelling a sports-demand function because the underlying theory is sufficiently loose to admit a wide range of candidate regressors. In such a situation, which is common to many other empirical applications in economics, the influence of particular variables of interest may be lost in the “morass of irrelevance” (Hendry and Krolzig, 1999, p.209). Reduction of the model, by elimination of statistically irrelevant

variables, to a parsimonious, congruent model allows for more efficient estimation and inference.<sup>10</sup>

Although most applications of the Gets approach to model selection involve time-series data, it can also be applied to cross-section data and a tutorial on cross-section model selection is included in Chapter 6 of the manual. Our application uses data from a pooled cross section (covering matches played over three seasons); for the purposes of this illustration, the data are treated as a standard cross section.

The first stage in using PcGets is to formulate the general unrestricted model (GUM), which, in the current version, is restricted to models that are linear in the parameters.<sup>11</sup> The GUM is specified using the **Formulate** command on the **Model** menu or by clicking the **Formulate** ‘building blocks’ icon (which, at the initial stage, would be the only active icon in the PcGets window) or by using a keyboard shortcut (**Alt+y**). This is representative of the alternative possible modes of operation throughout. This activates a ‘Data selection dialog’, which allows the user to select the variables from an active database, specify each variable’s status as regressand, regressor, endogenous, instrument and/or ‘fixed’ (discussed further below) and specify the number of lags of each of the variables to be included. Previous models formulated in the same session can be recalled (and models can be saved for later sessions). The **Tools** menu in GiveWin includes a **Calculator** command for transformations and allows algebra or models set up in previous sessions to be accessed via the **Algebra Editor** and **Batch Editor** commands. Once a model has been formulated the **Model Settings** and **Estimate** icons are activated and the GUM can be estimated by either Ordinary Least Squares Estimation (OLS) or Instrumental Variables Estimation (IVE) (if relevant instruments have been specified at the model formulation stage) using the **Estimate** icon or via the **Estimate** command on the **Model** menu.

The specification of the GUM from which reductions commence is, not surprisingly, crucial to the properties of any Gets approach (Hendry, 2000, p.482) because a poorly specified GUM stands little chance of leading to a ‘good’ final specific model. The aim is to start off with a congruent model, evaluated by applying a battery of mis-specification tests, and maintain congruence by applying the same tests, as diagnostic tests, through every stage of the simplification process. In general, specification testing of the GUM is based on five mis-specification tests, selected on the basis of their behaviour in Monte Carlo simulations: two  $F$ -tests for parameter constancy, denoted  $Chow(n)$ , for breakpoints at the sample mid-point ( $n = 0.5N$ , where

$N$  is the number of sample observations) and 90<sup>th</sup> percentile ( $n = 0.9N$ ); Doornik and Hansen's (1994)  $\chi^2$  test for normality of the error terms; the  $F$ -form of a Lagrange multiplier test for up to  $p$ -th order autocorrelation; the  $F$ -form of a test of autoregressive conditional heteroskedasticity (ARCH) for up to  $p$ -th order; and a  $\chi^2$  (or  $F$ -form) test for heteroskedasticity (denoted *hetero*) based on regressing the squared residuals on the original regressors and all their non-redundant squares (White, 1980). For a cross-section model-selection exercise, the autocorrelation and ARCH tests are not relevant; these can be 'turned off' by specifying an 'expert user's strategy', discussed below. The results for the mis-specification tests are appended to the usual estimates of coefficients,  $t$ -values, and goodness-of-fit statistics (not shown) as follows:<sup>12</sup>

	value	prob	alpha
Chow(36:1)	0.9978	0.5476	0.0100
Chow(64:1)	0.3396	0.9113	0.0100
normality test	0.7241	0.6963	0.0100
hetero test	11.0221	0.2005	0.0100

For these tests, a 1% significance level is normally used throughout to help control the overall null-rejection probability, as suggested by the Monte Carlo evidence in Krolzig and Hendry (2001). However, if the user specifies a GUM that fails one of the mis-specification tests at the 1% level but still chooses to proceed, ill-advisedly, with this model, then a reduced significance level of that test, as determined by its prob-value in the GUM, is adopted in the algorithm. This is a useful device for Monte Carlo studies, but, in practical applications, non-congruence indicates the need to revise the GUM; in future versions of the program the authors intend to provide an explicit warning when the GUM is mis-specified.

If there are sufficient degrees of freedom, an  $F$ -form of the *hetero* test is calculated. Otherwise, the program issues a warning message that there are not enough observations for the hetero  $F$ -test and that the  $\chi^2$  test will be used instead (as in this example); this choice is then maintained throughout the simplification process, even as adequate degrees of freedom become available. Our experience of the *hetero* test in PcGets 1.0 suggests that it can give misleading values when the cross product data matrix,  $X'X$ , is numerically unbalanced. Future versions of PcGets will include an implementation of the *hetero* test, based on the code used in PcGive 10.1, which is numerically robust to unbalanced data sets.

The  $F$ -forms of the heteroskedasticity and ARCH tests in PcGets 1.0 use degrees-of-freedom corrections that 'overcorrect' for the number of parameters in the auxiliary

equations used to obtain these test statistics.<sup>13</sup> This reduces the numerical value of the test statistics, other things equal, leading to rejection of the null too rarely, especially for heavily parameterized models such as GUMs with large  $k$ . This is reflected in the quantile-quantile plots for the *hetero* and ARCH tests in Figure 5 of Hendry and Krolzig (2002). When applied to a heavily parameterized GUM, these tests show significant departures of the empirical distribution from their theoretical distribution (particularly for the *hetero* test, which will also be affected by overfitting). Using an alternative degrees-of-freedom correction (see fn. 13) improves the tests' properties significantly, and these will be used in future releases of PcGets.

After estimation, further evaluation statistics are available on the Test menu (shown activated in Figure 1), including graphical analysis of the residuals, recursive graphical analysis of parameter constancy, and a collinearity analysis.<sup>14</sup> Batch files can also be created for the general and specific models for re-running in later sessions.

Once a GUM has been specified, the next step is to simplify the model using 'pre-search' tests.<sup>15</sup> The aim of this pre-search testing stage is to remove any obvious 'dead wood' from the set of potential regressors, which can significantly reduce the number of search paths required at the next stage of the algorithm. Sequential  $F$ -tests are applied to sets of individually non-significant parameters ordered by their (absolute)  $t$ -values in the GUM. For time-series models lag-order pre-selection is the first stage, using  $F$ -tests on the longest-lag blocks until the null is rejected. For cross-section data, this step is skipped. Instead, the algorithm first applies 'top-down' testing, where the absolute  $t$ -values are ordered from the smallest up and  $F$ -tests are calculated cumulatively as the block size increases, until the null is rejected at the chosen significance level or a diagnostic test fails. Non-stringent significance levels are used (e.g., 75% or 50%) because variables dropped at this stage are permanently removed. This type of testing can be applied in two steps, the first time with a higher significance level (say 75%), which is then reduced for the second step (say to 50%). Representative output from version 1.0 for the top-down tests is shown below.

```
Stage-0 (Step 1): F presearch testing (top-down)
Remove 1 variable with t-prob > 0.9158 : F-prob =0.9158, Tests failed = 0;
Remove 2 variables with t-prob > 0.7691 : F-prob =0.9563, Tests failed = 0;
Remove 3 variables with t-prob > 0.7122 : F-prob =0.9727, Tests failed = 0;
Remove 4 variables with t-prob > 0.7009 : F-prob =0.9719, Tests failed = 0;
Remove 5 variables with t-prob > 0.6030 : F-prob =0.9348, Tests failed = 0;
Remove 6 variables with t-prob > 0.5630 : F-prob =0.9110, Tests failed = 0;
Remove 7 variables with t-prob > 0.5476 : F-prob =0.9065, Tests failed = 0;
Remove 8 variables with t-prob > 0.5019 : F-prob =0.9331, Tests failed = 0;
Remove 9 variables with t-prob > 0.4398 : F-prob =0.8888, Tests failed = 0;
```



```

Remove 10 variables with t-prob > 0.3679 : F-prob =0.8095, Tests failed = 0;
Remove 11 variables with t-prob > 0.3498 : F-prob =0.7529, Tests failed = 0;
Remove 12 variables with t-prob > 0.3341 : F-prob =0.7632, Tests failed = 0;
Remove 13 variables with t-prob > 0.2850 : F-prob =0.5039, Tests failed = 1;
Invalid reduction.

```

Note that failure of the  $F$ -test of the null that all the coefficients on the 13 deleted variables are jointly zero is rejected (using a 75% significance level in this case), rendering the last reduction invalid. Hence, in this example, 12 (of the 28 regressors in the GUM) are removed at this top-down (first step) pre-testing stage. If a second step of pre-testing is desired, the model with these 12 variables removed now becomes the new starting point and the top-down pre-search stage can be repeated using a more taxing significance level, say 50%. In this example, this removes an additional six variables before the  $F$ -test rejects further simplification at this significance level.

```

Stage-0 (Step 2): F presearch testing (top-down)
Remove 1 variable with t-prob > 0.6393 : F-prob =0.8058, Tests failed = 0;
Remove 2 variables with t-prob > 0.5828 : F-prob =0.8362, Tests failed = 0;
Remove 3 variables with t-prob > 0.4868 : F-prob =0.8664, Tests failed = 0;
Remove 4 variables with t-prob > 0.1420 : F-prob =0.8106, Tests failed = 0;
Remove 5 variables with t-prob > 0.1171 : F-prob =0.6984, Tests failed = 0;
Remove 6 variables with t-prob > 0.0711 : F-prob =0.6359, Tests failed = 0;
Remove 7 variables with t-prob > 0.0527 : F-prob =0.3266, Tests failed = 1;
Invalid reduction.

```

‘Bottom-up’ testing is then applied, the aim of which is to search for the smallest congruent reduction of the initial GUM. This starts with a model including a ‘core’ set of regressors that have significant  $t$ -values in the GUM; in this case eight variables have  $t$ -prob values less than 0.05.  $F$ -tests of jointly zero coefficients are then conducted on blocks of decreasing size (from the largest absolute  $t$ -values downwards), equivalent to sequentially adding the remaining regressors, one at a time, to the core set of regressors. This procedure stops when the  $F$ -test against the GUM is not rejected and no diagnostic test is rejected. In this application, the core set of eight regressors with significant coefficients satisfies both criteria.

```

Stage-0 (Step 3): F presearch testing (bottom-up)
Found 8 variables with t-prob < 0.0500.
Include 8 variables with t-prob < 0.0518 : F-prob =0.2355, Tests failed = 0;
Valid reduction found.

```

```

Stage-0 (Step 4): Test of the additional restrictions imposed by the bottom-up reduction.
F test 5.5470 [0.0062] Rejected.

```

The next step (Step 4 in the output above) is to perform an encompassing test of the models selected by the top-down and bottom-up pre-search stages. This amounts to jointly testing exclusion restrictions for the additional variables identified for removal

by the bottom-up search relative to the model implied by the top-down search. In this case, zero restrictions on the coefficients of two additional variables are jointly tested and soundly rejected, so the variables identified as candidates for removal in the bottom-up pre-search are retained.

A new GUM is formed by removing the variables deleted by the pre-search testing procedures. The multiple-path search procedure commences sequential estimation and testing from this model. Each variable with a coefficient that is not statistically significant, and has not previously commenced a reduction path, is removed and starts a separate search path. If the diagnostic tests are passed, the variable with the next lowest absolute  $t$ -value is removed. Failure of any of the diagnostics (or an  $F$ -test of the model reduction relative to the GUM) leads to rejection of that reduction and a reinstatement of the variable. In such cases, the algorithm returns to the last accepted model and the variable with the next lowest absolute  $t$ -value is removed. Simplification continues until a diagnostic test fails or all remaining parameters are statistically significant. In the latter case the simplified model becomes the terminal model for that search path. As well as starting off search paths by deleting a single variable, search paths also commence by deleting blocks of variables, with the composition of the block determined by the prob-values of the individual coefficients'  $t$ -statistics (e.g., different blocks of variables with  $t$ -prob values greater than 0.001, 0.01 and 0.05). As noted earlier, a major advantage of employing multiple-path searches is that the final selected model is not 'path dependent'. In the current example, six different paths were examined, yielding the following output:<sup>16</sup>

```

Stage-1: Multiple-path encompassing search
Path 1: Check variables with t-prob > 0.0010.
Remove Time, Dcake, Htries, Trad, AB, Oppmax, Tests failed = 1   Path is
not continued.
Path 2: Check variables with t-prob > 0.0100.
Remove Time, Htries, AB, Tests failed = 0   Terminal specification found.
Path 3: Check variables with t-prob > 0.0500.
Remove Time, Htries, AB, Path converged to a previously found specification.
Path 4 of 6 started.
Remove AB           : Tests failed = 0
Remove Time         : Tests failed = 0   Terminal specification found.
Path 5 of 6 started.
Remove Time         : Tests failed = 0
Remove AB           : Path converged to a previously found specification.
Path 6 of 6 started.
Remove Htries       : Tests failed = 0
Remove AB           : Tests failed = 0   Terminal specification found.

```

Path 1, commencing with a block of variables deleted, leads to one of the diagnostic test rejecting its null, so this path is not continued. While PcGets 1.0 reports the

number of diagnostic tests failed, even the detailed output option does not provide any information about which tests have failed. As Hendry consistently emphasizes, rejection of the null does not imply acceptance of the alternative; however, some researchers might find it useful to know if failure of a particular test is responsible for truncating or modifying several search paths, as this may signal problems with the GUM that were not quite reflected in the initial mis-specification tests. As part of the output of the detailed reporting option, the current beta version includes model estimates and mis-specification test results for each different terminal model and for each of the final models in non-continuing paths.

Paths 2 and 3, with different criteria for defining the variables to include in the initially deleted block of variables, lead to the same terminal specification, in which all the remaining variables have statistically significant coefficients. Similarly paths 4 and 5 lead to the same terminal specification.

In this example, there are three different competing ‘final’ models. PcGets provides estimates of all the final models and their union. Encompassing  $F$ -tests are then used to choose between these terminal models by testing each model against the union.

```

Encompassing tests
Model 1:  F test      3.4973 [0.0365] Removed.
Model 2:  F test      2.5293 [0.1169]
Model 3:  F test      2.8738 [0.0951]

```

Here, Model 1 is rejected against the union and is dropped, but the  $F$ -tests cannot select between the other two models. In such cases, the union of Models 2 and 3 becomes the new GUM and the whole multiple-path reduction process is repeated. In this example, this yields the same two non-encompassing final models. In such a situation, PcGets chooses a unique final model based on an information criterion. These are chosen from the Akaike, Schwarz, Hannan-Quinn (HQ) or ‘HK’ criteria. The last of these is not defined in the manual or help system but is a modification of the HQ criterion that exploits sub-sample significance to penalize models for lack of ‘reliability’, a notion discussed in the next paragraph.<sup>17</sup>

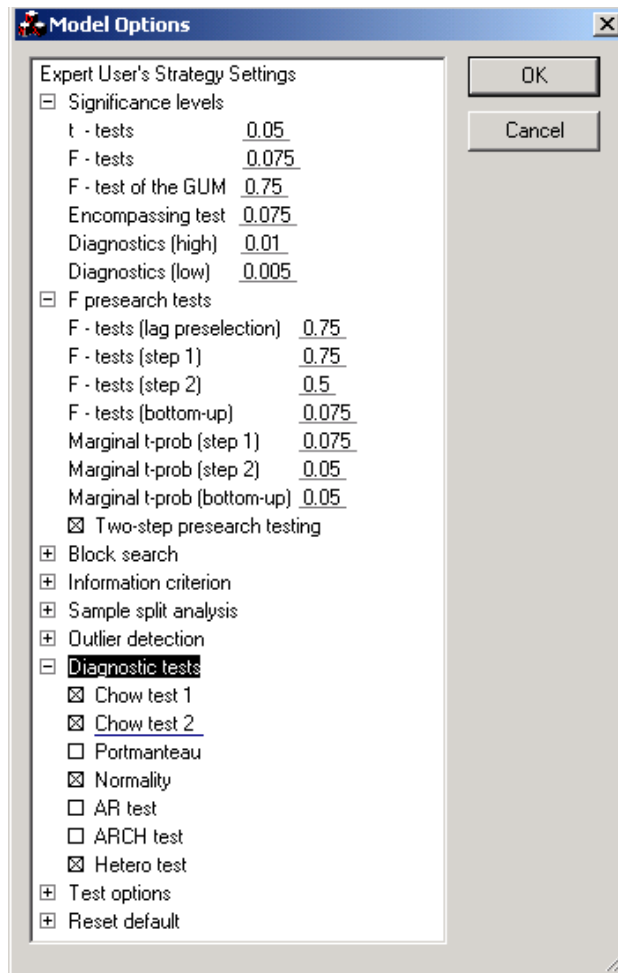
In our example, the basic ‘liberal’ strategy settings lead to eight regressors being retained from the original set of 28, while the ‘conservative’ strategy retains five regressors; these strategies are discussed further below.<sup>18</sup> Results for the specific model selected by this ‘testimation’ process are then reported in full. At this stage, PcGets, following Hoover and Perez’s (1999) suggestion, also evaluates the significance of the parameters in the final selected model in two overlapping sub-samples (the first and last

75% of the sample) and reports the relevant  $t$ -prob values. This provides a device for identifying ‘adventitiously significant variables’, which are significant by chance in the full sample but are likely to show up as not statistically significant in one or both of the sub-samples.<sup>19</sup> ‘Reliability statistics’ are assigned and reported. 100% reliability (a score of 1.000 in the reported output) corresponds to statistical significance in the full sample and both sub-samples, whereas 0% reliability reflects lack of statistical significance in the full sample and both sub-samples.<sup>20</sup> Intermediate cases are given scores determined by arbitrary but sensible penalties; for example, the penalty for a failed  $t$ -test is 0.4 in the full sample and 0.3 in either sub-sample. Reliability for coefficients on dummy variables is not to be taken too literally as such variables can be zero for all observations in one of the sub-samples and this is treated as non-significant.

PcGets includes two pre-specified calibrations of the algorithm. Its ‘liberal strategy’ aims to minimize the probability of not selecting relevant variables; by attempting to retain relevant variables, it runs the risk of also retaining irrelevant variables. By contrast, the program’s ‘conservative strategy’ aims to minimize the probability of not eliminating nuisance variables, though by increasing the probability of deleting irrelevant variables it risks also deleting relevant variables (Hendry and Krolzig, 2001, pp.197-8). The following are some of the settings for the liberal [conservative] strategy in version 1.0 of PcGets: significance levels for pre-search ‘top-down’  $F$ -tests are 75% [50%] at step 1 and 50% [25%] at step 2, and for ‘bottom-up’  $F$ -tests are 7.5% [2.5%]; in the multiple path searches, significance levels are 5% [1%] for  $t$ -tests, 7.5% [2.5%] for  $F$ -tests and 1% [1%] for diagnostics.

Alternatively, users can calibrate their own ‘expert user’s strategy’ by choosing their own settings; this can be activated using the **Model Settings** command on the **Model** menu or by clicking the **Model Settings** icon or with the shortcut **Alt+s**. As the default strategies include tests for autocorrelation and ARCH errors, which are not relevant for our cross-section model, we need to use the expert settings option to remove these before testimation.<sup>21</sup> The chosen settings can be adjusted via **Options** on the **Model** menu. Because the expert settings are ‘remembered’ between runs, it is first necessary to set one of the pre-calibrated strategies as a base, by expanding **Reset Default**, selecting, for example, the liberal strategy and clicking **OK**. Re-activating **Options** and unselecting the **AR** test and **ARCH** test in the **Diagnostic tests** section will remove these tests and, unless other changes are made at this stage, will leave all the other settings as in the liberal strategy; see Figure 2 and note that the calibrations for

the significance levels and the pre-search tests (revealed by expanding these headings) match those of the liberal strategy which was used as a base.<sup>22</sup> Selecting Estimate on the Model menu or clicking the Estimate icon leads to the Estimate Model dialog box where OLS, ‘testimation’ (the Gets algorithm using current settings), instrumental variables (IV) estimation or IV testimation can be selected.



**Figure 2.** Defining an expert user’s strategy

If different calibrations of the expert user’s strategy are used in the same run it can be useful to include a record of the current settings in the output using PcGets settings on the Test menu (as in Figure 1) after a model has been testimated.<sup>23</sup> For our cross-section example, the amended version of the liberal strategy gives the following settings, which gives a good idea of the large number of choices required to calibrate the complete algorithm.

```
PcGets Settings
Testimation algorithm
  Outlier correction                TRUE
  F presearch testing (lag order preselection)  FALSE
```

F presearch testing (top - down)	TRUE
F presearch testing (bottom - up)	TRUE
Sample split analysis	TRUE
Sample-size adjusted significance levels	TRUE
Significance levels	
t - tests	0.0500
F - tests	0.0750
F - test of GUM	0.7500
Encompassing test	0.0750
Diagnostics (high)	0.0100
Diagnostics (low)	0.0050
F presearch tests	
F - tests (lag order preselection)	0.7500
F - tests (step 1)	0.7500
F - tests (step 2)	0.5000
F - tests (bottom-up)	0.0750
Marginal t-prob (step 1)	0.0750
Marginal t-prob (step 2)	0.0500
Marginal t-prob (bottom-up)	0.0500
Two-step presearch testing	TRUE
Block search	
Check groups with t-pvals > 0.90	TRUE
Check groups with t-pvals > 0.70	TRUE
Check groups with t-pvals > 0.50	TRUE
Check groups with t-pvals > 0.25	TRUE
Check groups with t-pvals > 0.10	TRUE
Check groups with t-pvals > 0.05	TRUE
Check groups with t-pvals > 0.01	TRUE
Check groups with t-pvals > 0.001	TRUE
Information criterion	
Selection criterion for final model	HQ
Sample split analysis	
Significance level	0.0750
Size of the subsample (fraction)	0.7500
Penalty for failed t-test in full sample	0.4000
Penalty for failed t-test in subsample 1	0.3000
Penalty for failed t-test in subsample 2	0.3000
Outlier correction	
Size of marginal outlier (in std.dev.)	2.5600
Diagnostic tests	
Chow test 1	TRUE
Chow test 2	TRUE
Portmanteau	FALSE
Normality	TRUE
AR test	FALSE
ARCH test	FALSE
Hetero test	TRUE
Test options	
Chow test breakpoint 1	0.50
Chow test breakpoint 2	0.90
Portmanteau max lag	12
AR test min lag	1
AR test max lag	4
ARCH test min lag	1
ARCH test max lag	4

A very useful feature of the algorithm is the ability to force the search procedure to include specified variables of particular interest (defined as ‘fixed’ variables in the original model formulation). This allows for the examination of the sensitivity of the specific model to the inclusion of these variables and a comparison with the unconstrained algorithm’s chosen specific model. A variation that could perhaps be included in future versions would be the option to group variables to be included or

excluded as a block; this could be useful for handling, for example, sets of dummy variables or different proxies for the same variable that could be difficult to separate due to high degrees of collinearity.

While the above concentrates on a cross-section example, the same features apply to time-series applications. In addition, for dynamic models, PcGets implements a lag-order pre-search testing stage and provides long-run solutions and roots of the autoregressive polynomial for the final estimated models. An interesting aspect of the application of PcGets to time-series data is that, in its current form, testing for cointegration and the adoption of an EqCM is not part of the automated algorithm, which is initially surprising given the long association of EqCMs with the LSE/Hendry approach. If cointegration is present, there are obvious advantages in adopting an EqCM formulation. The EqCM and dynamic adjustment terms are all  $I(0)$  and it provides a representation that is closer to orthogonality than a levels model; the latter improves the algorithm's ability to disentangle the influence of relevant variables and helps avoid ending up with several mutually encompassing final models. While Hendry and Krolzig regard an  $I(0)$  representation as the ideal, automating testing for cointegration involves additional complications. They emphasize that care needs to be taken to avoid mixing variables with different orders of integration in the GUM; unit-root pre-testing could therefore be useful at the GUM formulation stage to avoid mixing orders of integration. In version 1.0, specifying the GUM in levels is recommended, as a second-best solution, when the cointegration properties of the system are unknown, or cointegration analysis is not feasible. Hendry and Krolzig (2001, pp.112-3) note, following the results of Wooldridge (1999) and Sims, Stock and Watson (1990), that the mis-specification tests and most selection tests will still be valid even with  $I(1)$  levels variables.<sup>24</sup>

Hendry and Krolzig (2001, p.112) emphasize that “it would be useful to implement cointegration tests and appropriate transformations prior to reduction” and work is underway to incorporate these features in future releases of the program. The unit root tests in PcGive and transformations from a levels GUM to an EqCM have been implemented, but the more difficult task of including an automatic procedure running a Johansen-type cointegration analysis is a longer-term goal.

While PcGets is designed primarily for single-equation models it can also be applied to vector autoregressive (VAR) models. Krolzig (2000) shows that equation-by-equation Gets modelling is appropriate if the variance-covariance matrix of the VAR

system is diagonal. He also outlines extensions to the PcGets algorithm that could deal with unrestricted variance-covariance matrices in a stationary VAR set-up and cointegration testing in non-stationary situations. An application of the equation-by-equation approach is given in the tutorial in Ch. 8 of the manual and automatic VAR reduction, with each equation taken in turn without further user intervention, will be included in future releases. However, as noted above, the systems approach and the cointegration-testing aspects are yet to be implemented.

## **5. General Comments**

The movement to an ‘expert systems’ approach to model selection, based around the automation of an explicit rules-based decision process, is a natural development given currently available computing power and the demand for modelling facilities by non-expert users (Oxley, 1995). In this vein, the development of an algorithm that makes explicit the set of decision rules involved in model reduction should provide a major boost to the more widespread use of the Gets aspect of the LSE/Hendry methodology. However, this will require potential users to get past the perceived evils of something that is both ‘automatic’ and has first-impression connotations of data mining. Both of these pre-conceptions are misleading. The automatic aspects embodied in the pre-specified search strategies can be amended within the limits of the current set-up of the algorithm. More importantly, the now-explicit set of decision rules can serve as a basis for further critical evaluation and progress to achieve stable and/or improved performance across different types of data generating process (DGP) as more theoretical developments and Monte Carlo evidence on the Gets algorithm become available. Also, Hendry and Krolzig (2001, pp.124-5) emphasize that, while Gets modelling clearly relies heavily on ‘data basing’, it is not, given its emphasis on encompassing rival models, ‘data mining’ in the derogatory sense of trying several different models to find one that fits the data arbitrarily well. In addition, the ease of applying the reduction process should encourage users to put more care and attention into specifying the initial GUM, although it is possible that some could use it for ‘higher level’ data mining (replacing OLS estimation with testimation), which is not sanctioned by the methodology.

Critics of Gets modelling are commonly concerned with the problems of controlling the overall size of tests in a sequential testing process and of interpreting the end results



from a classical viewpoint. However, because of the interactions between its various components, not all of which can currently be modelled theoretically, the overall properties of the Gets algorithm can really only be evaluated from a holistic perspective.<sup>25</sup> Monte Carlo evidence to date (Krolzig and Hendry, 2001; Hendry and Krolzig, 1999, 2001, 2002) suggests that the different elements of the overall algorithm combine to give impressive properties, with the size of the model selection process close to the nominal size of the tests used in the search and power approaching that obtained if commencing from the DGP. In this context, ‘power’ and ‘size’ relate to the probabilities of retaining in the final model variables that are, respectively, included and not included in the DGP. The use of multiple search paths and encompassing tests, the removal of clearly redundant variables through pre-search simplification, and the use of post-selection sub-sample tests combine to reduce the overall size, despite the large number of tests conducted (Krolzig and Hendry, 2001; Hendry and Krolzig, 1999). The overall size properties imply that the *t*-values in the selected final models can effectively be interpreted as ‘one-shot’ values. The algorithm’s impressive performance is reflected in its low overall ‘search’ costs.<sup>26</sup>

Since the release of version 1.0, Hendry and Krolzig (2002) have adjusted the calibrations of PcGets and achieved more stability in its properties, such as the ratio of actual to nominal size, across different DGPs in their Monte Carlo studies. The main substantive changes are at the pre-search testing stage for dynamic time-series models. In the current beta version, as well as checking for zero coefficients on all variables at a given lag (‘lag-order pre-selection’), the program also tests for zero coefficients on all the lags for each variable in turn (‘variables pre-selection’). For these block pre-search *F*-tests, critical *F*-prob values of 0.5 [0.25] are used in the pre-specified liberal [conservative] strategy.

What is now desirable is more extensive evidence on the performance of PcGets across the range of different types of DGP likely to be encountered in applied work (including situations where, for example, there are different signal-to-noise ratios, where the ‘true’ model is not in the GUM (Granger and Timmermann, 2000), where regressors are endogenous, and where the data are from cross-sections and panels).<sup>27</sup> Also, we need evidence on its head-to-head performance against rival, practical, model selection methods, including reliance on information criteria, non-sequential approaches (e.g., different versions of Leamer’s (1983) approach, some of which, such as Pesaran and Timmermann (2000), involve specific-to-general aspects) and

alternative automated systems (e.g., Phillips (1996) provides the theoretical basis, in a Bayesian framework, for an automated approach to model selection). Such investigations could also lead to more insights into whether different settings for the algorithm's calibration are better suited to different types of data and different modelling situations.

Hendry and Krolzig (2001, p.6) claim that “in practical empirical modelling, [PcGets] regularly delivers performance at least as good as models developed by some of the world's expert econometricians – but in a fraction of the time they take!” If this is indeed the case for a broader set of models than have so far been examined, then the concern that while “the Hendry methodology generates neat and elegant constructions in the hands of the master ... the chance is that it will spawn monsters” when applied by lesser mortals (Davidson, 1991, p.407) will be turned on its head.

## **6. Conclusion**

Overall, PcGets is an exciting addition to the proverbial ‘econometrician’s toolkit’. While all tools have the potential to be misused, PcGets is a huge step forward in the application of Gets modelling and is likely to encourage much more widespread use of this key aspect of the LSE/Hendry methodology in applied work. As befits the type of progressive research strategy consistently advocated by David Hendry, it also holds considerable promise for further practical and theoretical developments in the crucial area of model selection, which, in turn, are likely to further enhance the scope and performance of future versions of the program.

## **Acknowledgements**

I am grateful to David Hendry and Hans-Martin Krolzig for very helpful responses to a range of queries and for providing a copy of the current beta version of PcGets, as well as some Monte Carlo results on alternative forms of some of the tests. Parts of this review are based on a seminar given at the Reserve Bank of New Zealand (August, 2002); thanks are due to participants, especially Chris Plantier, and Clayton Weatherston for helpful comments and queries. Financial support from the Marsden Fund (Royal Society of New Zealand) is also gratefully acknowledged.

## References

- Davidson, J. (1991) Review of Modelling Economic Series: Readings in Econometric Methodology [C.W.J. Granger (ed.), Oxford University Press, Oxford]. *Economica*, 58, 405-408.
- Davidson, J.E.H., Hendry, D.F., Srba, F. and Yeo, S. (1978) Econometric modelling of the aggregate time-series relationship between consumers' expenditure and income in the United Kingdom. *Economic Journal*, 88, 661-692.
- Doornik, J.A. (2001) *Ox 3.0 - An Object-Oriented Matrix Programming Language*. London: Timberlake Consultants Ltd.
- Doornik, J.A. and Hansen, H. (1994) An omnibus test for univariate and multivariate normality. Working Paper, Nuffield College, Oxford.
- Doornik, J.A. and Hendry, D.F. (2001) *GiveWin: An Interface to Empirical Modelling*. London: Timberlake Consultants Ltd.
- Engle, R.F. and Granger, C.W.J. (1987) Co-integration and error correction: representation, estimation and testing. *Econometrica*, 55, 251-276.
- Faust, J. and Whiteman, C.H. (1997) General-to-specific procedures for fitting a data-admissible, theory-inspired, congruent, parsimonious, encompassing, weakly-exogenous, identified, structural model to the DGP: a translation and critique. *Carnegie-Rochester Conference Series on Public Policy*, 47, 121-161.
- Favero, C.A. (2001) *Applied Macroeconometrics*. Oxford: Oxford University Press.
- Granger, C. and Timmermann, A. (2000) Data mining with local model specification uncertainty: a discussion of Hoover and Perez. *Econometrics Journal*, 2, 220-225.
- Hansen, B.E. (1996) Methodology: alchemy or science? *Economic Journal*, 106, 1398-1413.
- Hendry, D.F. (1995) *Dynamic Econometrics*. Oxford: Oxford University Press.
- Hendry, D.F. (2000) Epilogue: the success of general-to-specific model selection, in D.F. Hendry, *Econometrics: Alchemy or Science?* (New Edition). Oxford: Oxford University Press.
- Hendry, D.F. and Krolzig, H.-M. (1999) Improving on 'Data mining reconsidered' by K.D. Hoover and S.J. Perez. *Econometrics Journal*, 2, 202-219.
- Hendry, D.F. and Krolzig, H.-M. (2001) *Automatic Econometric Model Selection Using PcGets*. London: Timberlake Consultants Ltd.

- Hendry, D.F. and Krolzig, H.-M. (2002) New developments in automatic general-to-specific modelling, paper presented at the ESAM02 conference, Brisbane, July 2002, forthcoming in B.P. Stigum (ed.), *Econometrics and the Philosophy of Economics*. Princeton, NJ: Princeton University Press.
- Hendry, D.F. and Mizon, G.E. (1978) Serial correlation as a convenient simplification, not a nuisance: a comment on a study of the demand for money by the Bank of England. *Economic Journal*, 88, 549-563.
- Hendry, D.F. and Mizon, G.E. (1990) Procrustean econometrics: or stretching and squeezing data, in C.W.J. Granger (ed.), *Modelling Economic Series: Readings in Econometric Methodology*. Oxford: Oxford University Press.
- Hoover, K.D. and Perez, S.J. (1999) Data mining reconsidered: encompassing and the general-to-specific approach to specification search. *Econometrics Journal*, 2, 167-191.
- Hoover, K.D. and Perez, S.J. (2001) Evaluating specification search in cross-country growth regressions: extreme bounds vs. general-to-specific methodologies. Department of Economics, University of California, Davis, mimeo (<http://www.econ.ucdavis.edu/faculty/kdhoover/pdf/GrowthERN.pdf>).
- Keuzenkamp, H.A. (1995) The econometrics of the Holy Grail – a review of *Econometrics: Alchemy or Science? Essays in Econometric Methodology*. *Journal of Economic Surveys*, 9, 233-248.
- Krolzig, H.-M. (2000) General-to-specific reductions of vector autoregressive processes. Working Paper, Nuffield College, Oxford.
- Krolzig, H.-M. and Hendry, D.F. (2001) Computer automation of general-to-specific model selection procedures. *Journal of Economic Dynamics & Control*, 25, 831-866.
- Leamer, E.E. (1983) Let's take the con out of econometrics. *American Economic Review*, 73, 31-43.
- Levine, R. and Renelt, D. (1992) A sensitivity analysis of cross-country growth regressions. *American Economic Review*, 82, 942-963.
- Owen, P.D. and Weatherston, C.R. (2002) Uncertainty of outcome and Super 12 rugby union attendance: application of a general-to-specific modelling strategy. University of Otago, Economics Discussion Papers 0211.
- Oxley, L.T. (1995) An expert systems approach to econometric modelling. *Mathematics and Computers in Simulation*, 39, 379-383.

- Pagan, A. (1987) Three econometric methodologies: a critical appraisal. *Journal of Economic Surveys*, 1, 3-24.
- Pagan, A. (1995) Three econometric methodologies: an update, in L. Oxley, D.A.R. George, C.J. Roberts and S. Sayer (eds), *Surveys in Econometrics*. Oxford: Basil Blackwell.
- Pesaran, M.H. and Timmermann, A. (2000) A recursive modeling approach to predicting UK stock returns. *Economic Journal*, 110, 159-191.
- Phillips, P.C.B. (1996) Econometric model determination. *Econometrica*, 64, 763-812.
- Sala-i-Martin, X.X. (1997) I have just run two million regressions. *American Economic Review*, 87, 178-183.
- Sargan, J.D. (1964) Wages and prices in the United Kingdom: a study in econometric methodology, in P.E. Hart, G. Mills and J.K. Whitaker (eds), *Econometric Analysis for National Economic Planning*. London: Butterworth.
- Sims, C.A., Stock, J.H. and Watson, M.W. (1990) Inference in linear time series models with some unit roots. *Econometrica*, 58, 113-144.
- Urga, G. (2001) Theory and practice of econometric modelling using PcGive10. *Journal of Economic Surveys*, 15, 571-588.
- White, H. (1980) A heteroskedastic-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48, 817-838.
- Wooldridge, J.M. (1999) Asymptotic properties of some specification tests in linear models with integrated processes, in R.F. Engle and H. White (eds), *Cointegration, Causality, and Forecasting: A Festschrift in Honour of Clive W. J. Granger*. Oxford: Oxford University Press.

---

## Notes

<sup>1</sup> Congruency and encompassing are key requirements in this approach. ‘Congruency’ relates to the matching of the model with the evidence in the data with respect to criteria such as homoskedastic, normal, innovation errors; parameter constancy; and weak exogeneity of the conditioning variables for the parameters of interest. Essentially, congruent models lack evidence of mis-specification. Encompassing is concerned with avoiding loss of information in the reduction process; if model A is a reduction compared to model B, then A encompasses B if there is no loss of information in A compared to B.

<sup>2</sup> For a recent textbook treatment see Favero (2001, Ch. 5).

<sup>3</sup> While Hendry and Mizon (1990) argue that the “model is the message”, and that exactly how the model is arrived at is not that important, less-experienced Gets modellers will readily identify with Pagan’s (1987, p.7) view that this “is cold comfort to those who are implementing the methodology or who are learning it for the first time”.

<sup>4</sup> Hoover and Perez (1999) perform multiple searches over overlapping sub-samples and require variables to have statistically significant coefficients in both sub-samples as well as the full sample. By contrast, PcGets utilizes sub-sample estimation and testing at the post-selection stage.

<sup>5</sup> A restricted demo version of PcGive Professional 10.1, including the demo versions of STAMP 6.2, GiveWin 2.1 and PcGets 1.0, is available for downloading from the last of these sites (under ‘program upgrades’).

<sup>6</sup> The data sets used in Hendry (1995) and Davidson, Hendry, Srba and Yeo (1978) are also included. The former is used extensively in the tutorials in the PcGets manual.

<sup>7</sup> For GiveWin, a licensing code is required for one of the modules for which it acts as a front end, e.g., PcGets, PcGive, TSP or STAMP.

<sup>8</sup> Our experience is based on running PcGets on Windows 2000.

<sup>9</sup> In this review, commands on menus, toolbar buttons and dialog items, as well as instructions for typing, are in Sans Serif font and output is in Courier New font. The font for the output from PcGets can be changed in GiveWin using the View menu (Preferences/Change Font...) and the actual appearance of PcGets depends on the settings of the operating system.

<sup>10</sup> Even in cases where economic theory implies ‘tight’ specifications, alternative theories often abstract in quite different dimensions and identify different elements as ‘crucial’ (e.g., theories of the demand for money) so an encompassing general model is a natural starting point. Moreover, Hendry (2000, p.474) emphasizes that “since theoretical models are incomplete, and possibly incorrect, an econometrics that was forced to use such theories as the only permissible starting-point for data analysis could contribute no useful knowledge, except perhaps rejecting the theories”.

<sup>11</sup> Models that are linear in the parameters are very widely used in empirical work, although Granger and Timmermann (2000) argue that this set of models is too restrictive. As noted by Hansen (1996), a Gets approach is harder to implement in situations where a broader ‘general’ model could include non-linearities, complex interactions, structural changes and non-Gaussian errors. While Hansen argues that this is a limitation of the methodology, it does make sense to ‘walk before attempting to run’; with the insights gained from applying Gets in the linear-in-the parameters case, Hendry and Krolzig are working on extending the algorithm to generic non-linear (single-equation) problems.

<sup>12</sup> The output from the current beta version of PcGets also includes details of the type of test ( $F$  or  $\chi^2$ ) and degrees of freedom for the mis-specification tests.

<sup>13</sup> For example, for the *hetero* test in version 1.0, the  $F$ -form of the test statistic has  $q$  and  $T-k-q-1$  degrees of freedom, where  $T$  is the number of sample observations,  $k$  the number of estimated parameters in the original model and  $q+1$  the number of parameters (including a constant term) in the auxiliary regression of  $e^2$  (the squared residuals from the original regression) on the original regressors ( $X$ ) and all their (non-redundant) squared values ( $X^2$ ); see, for example, Hendry and Krolzig (2001, Table 13.1). An alternative degrees-of-freedom

---

adjustment is to use  $T-q-1$  in the denominator (analogous to the case of the  $F$ -form of the usual LM test for autocorrelation).

<sup>14</sup> The collinearity analysis option reports eigenvalues of the second-moment matrix of the data on the explanatory variables and produces a correlation matrix. Note that the latter does not mean-adjust the data, so it differs from the standard correlation matrix (as, for example, reported in PcGive).

<sup>15</sup> The program also checks for outliers, with the option available to add dummy variables to ‘neutralize’ their impact.

<sup>16</sup> The user can choose whether the output consists of just the final selected model, a condensed summary of the multiple path searches or a detailed summary. The last of these can produce large amounts of output when there are many paths being searched, but provides a much clearer guide to what the algorithm is actually doing.

<sup>17</sup> The Schwarz, HQ and HK criteria are consistent, i.e., they ensure that if a DGP is nested in a model it will be selected with probability one as the sample size tends to infinity.

<sup>18</sup> Krolzig and Hendry (2001, p.864) characterize the process based around searching multiple paths as follows: “Intuitively, the iterative loops around sequences of path searches could be viewed as ‘sieves’ of ever-decreasing meshes filtering out the relevant from the irrelevant variables: as an analogy, first large rocks are removed, then stones, pebbles, so finally only the gold dust remains.”

<sup>19</sup> This is based on the respective properties of non-central  $t$ -values, which diverge as the sample size increases, and central  $t$ -values, which vary around zero. In the latter case, relevant when the null of a zero coefficient value is true, there is a low probability of significant  $t$ -values in both sub-samples (Krolzig and Hendry, 2001, p.834).

<sup>20</sup> Note that the selected specific models can include variables with a zero reliability rating, if omission of a variable leads to a significant diagnostic test statistic and non-congruence.

<sup>21</sup> We also need to unselect the F presearch testing (lag order preselection) option in the Model Settings dialog box, as this is also not relevant in a cross-section application. Although the default values will still be shown in the Model Options dialog, as in Figure 2, these tests will not be applied.

<sup>22</sup> The pre-specified liberal or conservative strategies include sample-size adjustment of significance levels, which are lost in the expert settings option.

<sup>23</sup> While the output from PcGets 1.0 does not label the current strategy, the output file can easily be edited to show which of the pre-specified strategies is being used for each final specific model. The current beta version automatically includes a label.

<sup>24</sup> Sims, Stock and Watson’s (1990) results imply that the selection test statistics can often have standard distributions in the presence of integrated regressors, although they do not guarantee that this is always the case; in particular this does not apply if the hypotheses being tested cannot be expressed as restrictions on parameters on mean-zero-stationary linear combinations of the variables. Although it is possible that adopting a levels GUM that includes variables that are potentially  $I(1)$  could, in some instances, involve estimating a spurious regression, residual autocorrelation should act as a warning indicator in such cases.

<sup>25</sup> Note, however, that Hendry (2000), Hendry and Krolzig (1999, 2001, 2002) and Krolzig and Hendry (2001) include discussions of components of the relevant theory.

<sup>26</sup> Hendry and Krolzig (2001) distinguish between the costs of inference and the costs of search. The former arise from errors in inference, even when tests are conducted on the DGP. If  $p^{\text{DGP}}$  and  $p^{\text{GUM}}$  represent the probabilities of retaining a particular variable (at a specific significance level) when testing the DGP and the GUM respectively, then  $1 - p^{\text{DGP}}$  represents the costs of inference. Search costs =  $|p^{\text{DGP}} - p^{\text{GUM}}|$  and reflect the costs of including irrelevant variables and additional costs of failing to retain relevant variables (Hendry, 2000).

<sup>27</sup> Hoover and Perez (2001), in a Monte Carlo study designed to reflect the ‘realistic’ setting of cross-country growth regressions, show that a cross-section version of a Gets algorithm has near to the nominal size and high power, and clearly outperforms Levine and Renelt’s (1992) and Sala-i-Martin’s (1997) versions of Leamer’s (1983) extreme-bounds approach to model

---

selection. The former turns out to be too stringent, rejecting relevant variables too frequently, while the latter is too permissive, including irrelevant variables too frequently.