User Manual for the HPZ Code Package

Yoram Halevy* Oriel Nofekh‡

Dotan Persitz† Lanny Zrill§

June 20, 2020

1 Matlab Preparation

The code package runs on Matlab (checked on versions 2015b and 2017a). The user needs to follow the following steps in order to run the code in the MATLAB environment:

- Extract the code package zipped file into a new folder named (for example) HPZ_Code.
- Set the MATLAB path to the HPZ_Code folder:
  - Click on the "set path" option.[1]
  - Choose the "Add Folder with Subfolders" option.
  - Choose the HPZ_Code library.

*Department of Economics, University of Toronto. Email: yoram.halevy@utoronto.ca.

†Coller School of Management, Tel Aviv University. Email: persitzd@post.tau.ac.il.

‡Eitan Berglas School of Economics, Tel Aviv University. Email: orielnofekh@mail.tau.ac.il.

§Department of Applied Economics, HEC Montreal, Email: lzrill@langara.bc.ca.

[1] In the 2010b version the button is under the File tab on the menu bar. In the 2012 version it is an individual button on the MATLAB toolbar. In the 2015b version this button is placed under the Home tab.

- After the files were added to the list, while still marked, click the "Move to Bottom" option.
- Save and Close.

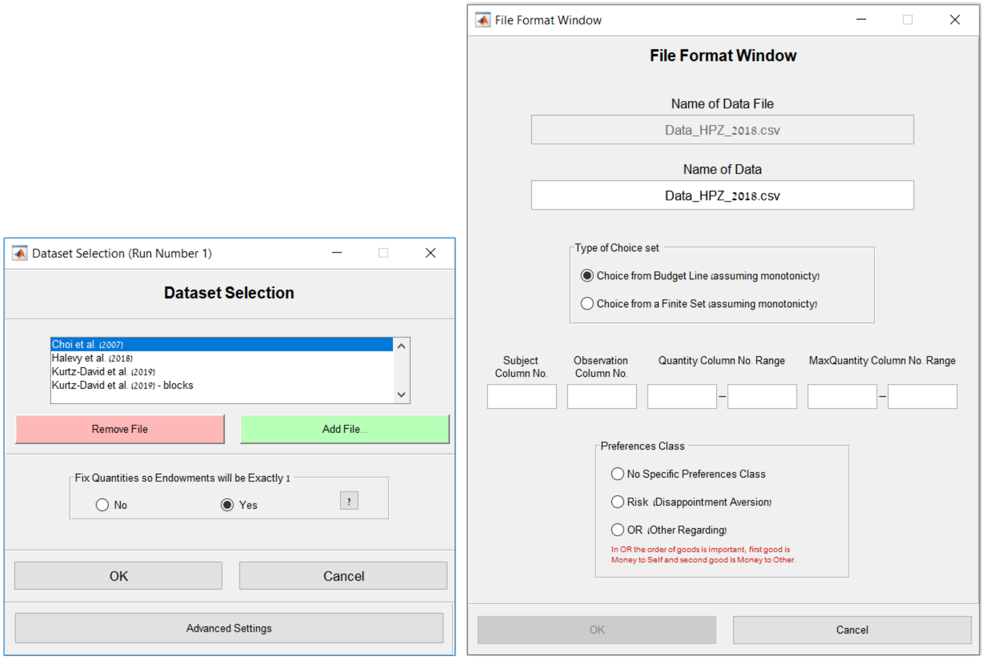
2 Start the Program

To run the code, the user should write HPZ_Interface on the MATLAB command window. The user interface (HPZ_Interface) includes a sequence of windows. Each window presents the user with questions regarding required operations. After the information is gathered, the "Running Plan" window appears. The user can then choose to insert the parameters for another run of the routine. In this way the user can insert several calculations to a single run of the routine. Once the "Run Now" button is clicked, all the pre-defined calculations are executed in the order of definition. When the routine is done, the results are saved into a file (the results of each run are stored separately) and a popup window is prompted including the path to the result file. The rest of this manual describes the sequence of windows.

3 Dataset Selection Window

In the Dataset Selection window (see Figure 3.1a) the user is required to choose the dataset for analysis. All datasets that involve choice from linear budget sets with any number of goods (≥ 2) are supported for non-parametric analysis. Parameter estimation is currently supported for two goods in the contexts of risk and other regarding preferences. Datasets that involve choice from a finite set with two goods are supported for non-parametric analysis and for NLLS-based Parameter estimation.

The user can add a dataset to the list by clicking the "Add File" option. A browser window pops up where the user is asked to choose the new dataset. Then, the File Format window (see Figure 3.1b) appears where the user can edit the dataset name, the preferences class (no preferences class, risk preferences or other regarding preferences) and match the columns in the file to the



(a) Dataset Selection Window (b) File Format Window

Figure 3.1: Dataset Selection

observation level data required by the software – column with subject number, column with observation number (observation numbers for each subject must be $1, 2, 3, \ldots$ to make sure the data file is not corrupted), range of columns with the chosen quantities and a corresponding range of columns with the maximal quantities (the reciprocal of the prices). The software allows the user to add a dataset with more than 2 goods as one with no specific preferences class, but then in the Action Selection Window (see section 4) only the "Consistency analysis" option will be available. The software also allows the user to add a dataset of choice from a finite set (rather than a linear budget set), in which case the file format must be as follows: the first column is the subject number, the second is the observation number, the third and fourth columns contain the chosen quantities, and every two consecutive columns from then onward contains one optional choice that was available to the user.

In the Dataset Selection Window the user can also remove a dataset from

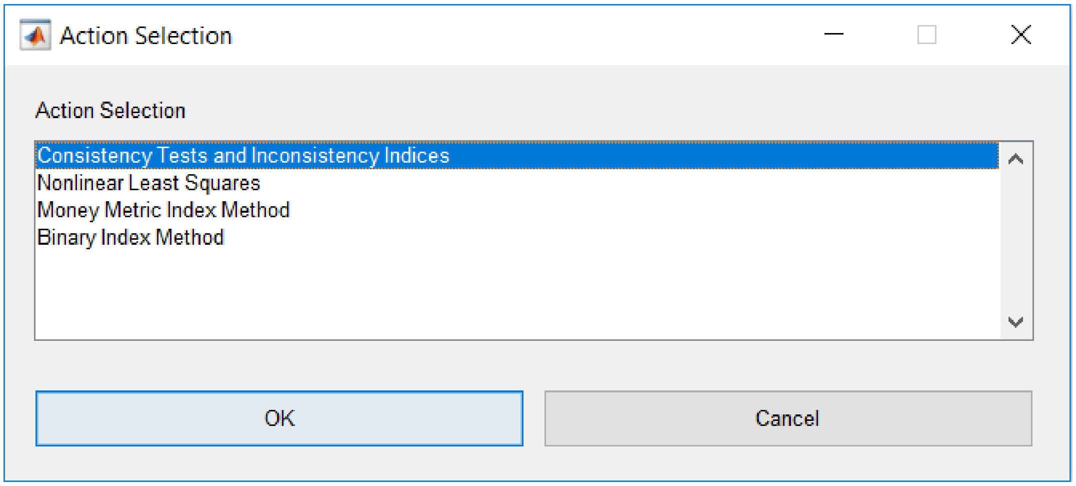


Figure 4.1: Action Selection Window

the list by choosing the dataset and clicking the “Remove File” button.

Finally, the user can choose whether to fix the endowments to exactly 1, i.e. exactly on the budget line, or leave the data as is. This option might be useful when quantities are rounded in the original data file. In this case, the endowment may equal 1 only approximately and some calculations may be slightly sensitive to these discrepancies.[2] If the user asks to fix the endowment, the quantities are changed accordingly while keeping their ratio constant.[3]

In the bottom of the Dataset Selection Window there is a button labeled "Advanced Settings". It leads to a window that allows to control various elements of the software, such as waitbar settings, residuals printing settings and more. For more details, see Appendix.

4 Action Selection Window

In the Action Selection window (see Figure 4.1) the user is required to choose one of the following four actions:

[2]Discrete indices are more prone to be affected by these adaptations than continuous indices.

[3]If the required adjustment is larger than 5% the software suspends the process and an error pop up window is prompted.

- Consistency analysis.
- Nonlinear Least Squares recovery.
- Money Metric Index recovery.
- Binary Index recovery.

5 Subjects Selection Window

In the Subjects Selection Window (see Figure 5.1), the user is required to select the analyzed subjects (one or multiple subjects can be chosen). When the choice is made, the sequence of windows continues.

6 Inconsistency Indices Settings Window

The software implements four types of inconsistency indices −number of violations (of some revealed preference axiom), the Afriat index, the Varian index and the Houtman-Maks index (for the definitions see Halevy et al. (2018) and the references therein, a detailed description of the algorithms implemented here is included in a forthcoming working paper). The inconsistency indices settings window (see Figure 6.1) allows the user the following actions:

- Revealed Preference Graphs
  - Preference Relation Hierarchy – if the subject's choices are consistent (satisfy GARP), the software will create a directed graph that represents the hierarchy of observations according to the revealed choices, from the most preferred to the least.
  - Preference Relations involved in cycles / minimal cycles – if the subject's choices are inconsistent (don't satisfy GARP), the software will create a directed graph that contains all revealed preference relations that are part of a cycle / minimal cycle.

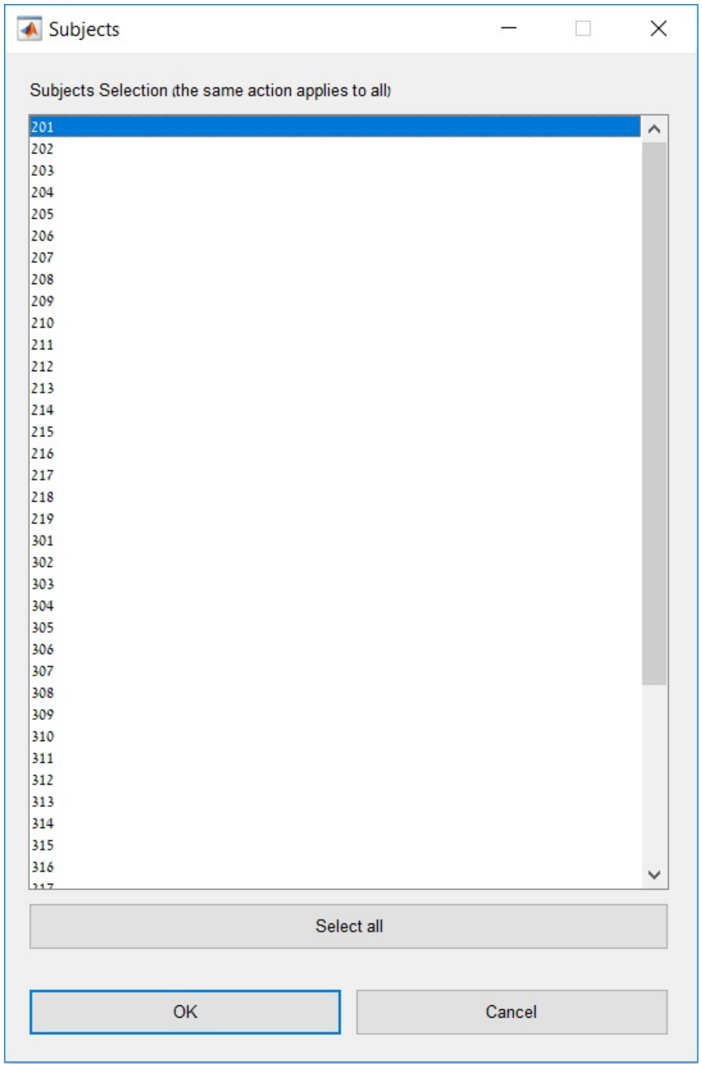


Figure 5.1: Subjects Selection Window

– Options – the graphs can be drawn without any labels on the edges (weak and strict relations will be drawn as thin and thick lines, respectively), or with weights that represent the endowment adjustment required to remove the relation.

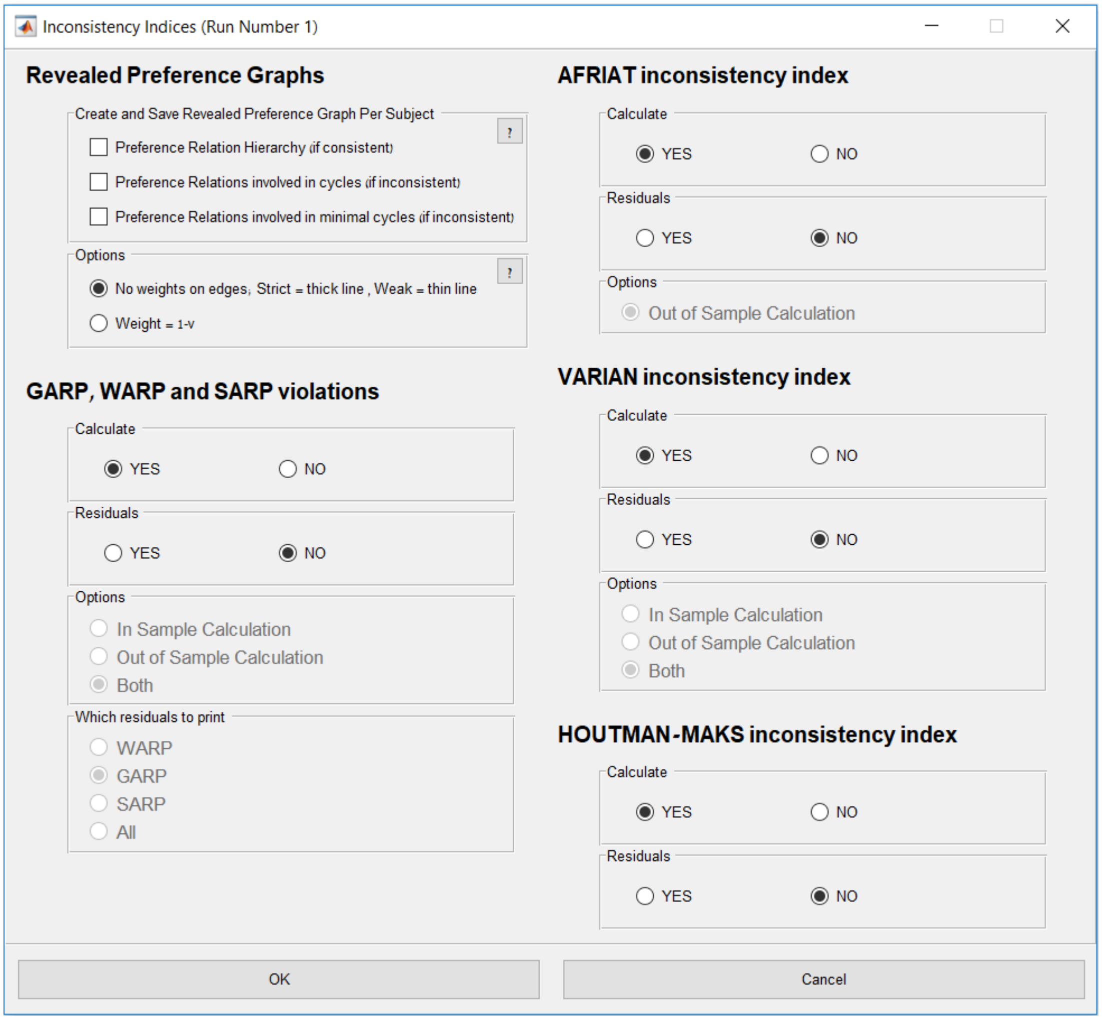


Figure 6.1: Inconsistency Indices Window

- GARP, WARP and SARP violations: [4]

[4]The software prints the following:
WARP Violations, GARP Violations, GARP Violation Pairs, SARP Violations, SARP Violation Pairs.
It does not print WARP Violation Pairs, since by definition: WARP Violations = 2 $\times$ WARP Violation Pairs. The definitions of violations are as follows:
We denote a directly revealed preferred to b by: $aR^0_D b$, strictly directly revealed preferred by: $aP^0_D b$, and revealed preferred (directly or indirectly) by: $aR_D b$.
A WARP Violation is an ordered pair (a, b) such that $aR^0_D b$ and $bR^0_D a$ and $a \neq b$.
A GARP Violation is an ordered pair (a, b) such that $aR_D b$ and $bP^0_D a$.
A SARP Violation is an ordered pair (a, b) such that $aR_D b$ and $bR^0_D a$ and $a \neq b$.
Violation Pairs count (a, b) and (b, a) as one pair that violates the axiom, regardless of whether only one of the two or both of them violate the relevant axiom.

- Calculate: The index is calculated only if YES is checked.
- Residuals: If YES is checked then we compute a residual for each observation. The residual is the number of violations that involve the observation (in-sample) or the number of violations lost if this observation is dropped (out-of-sample calculation). The user can require a calculation of the residual for a specific axiom or for all supported axioms simultaneously.

• Afriat Inconsistency Index and Houtman-Maks Inconsistency Index:

- Calculate: The index is calculated only if YES is checked.
- Residuals: If YES is checked then we compute a residual for each observation −the difference between the index as calculated on the whole dataset and the index as calculated on the dataset when this observation is dropped (out-of-sample calculation).

• Varian Inconsistency Index:[5]

- Calculate: The index is calculated only if YES is checked.
- Residuals: If YES is checked then we compute one to three different residuals for each observation (denote the disaggregated vector of adjustments that minimizes the index by $\mathbf{v} = (v^1, \dots, v^n)$ and the aggregator function by f):
  * In-sample component − The value of the observation in the disaggregated vector that minimizes the index, v^i.
  * In-sample difference − Leave one out procedure without re-calculation, $f(\mathbf{v}) - f(\mathbf{v}_{-i})$.

[5] Varian Index is usually calculated accurately, in which case the columns "VARIAN Index Max", "VARIAN Index Mean" and "VARIAN Index AVG(SSQ)" in the results file will contain the index's value. However, if the Varian Index Mean for example was approximated, the "VARIAN Index Mean" column will remain empty, and the columns "VARIAN Index Mean Lower Bound", "VARIAN Index Mean Approximate" and "VARIAN Index Mean Upper Bound" will contain the lower bound, the approximation, and the upper bound for the Varian Index with mean aggregator function, respectively. See Appendix for advanced options regarding the level of approximation.

* Out-of-sample − Leave one out procedure with re-calculation, denote the index calculated on the dataset without observation i by $I_{V(D_{-i})}$, then this residual is $f(\mathbf{v}) - I_{V(D_{-i})}$.

When OK is clicked the Running Plan Window pops up.

7 Functional Form Settings Window

The Functional Form Settings window can take two forms depending on the object of choice.

7.1 Risk Preferences

First, we describe the window that appears when analyzing a data set collected in an experiment where bundles are portfolios and the goods are Arrow securities (see Figure 7.1). In this window the user is required to address the following issues:

1. **Functional Forms** −Within the Disappointment Aversion framework, the user must select the functional form of the VNM utility function. The user options are the well known CRRA or CARA.

2. **Solution Options** −The user must choose the computational approach towards basic calculations (optimal choices in the NLLS case and optimal budget line adjustments in the MMI and BI case). One option is the numeric approach which is based on MATLAB's optimization procedures. The other option is the analytic approach which is based on a pre-solevd Utility Maximization Problem (for the NLLS) and Expenditure Minimization Problem (for the MMI and BI). The calculations are documented in the file "DA2-analytic.pdf" included in the package. **The analytic approach is much faster and therefore recommended for a user that uses the functional forms provided in the package.**

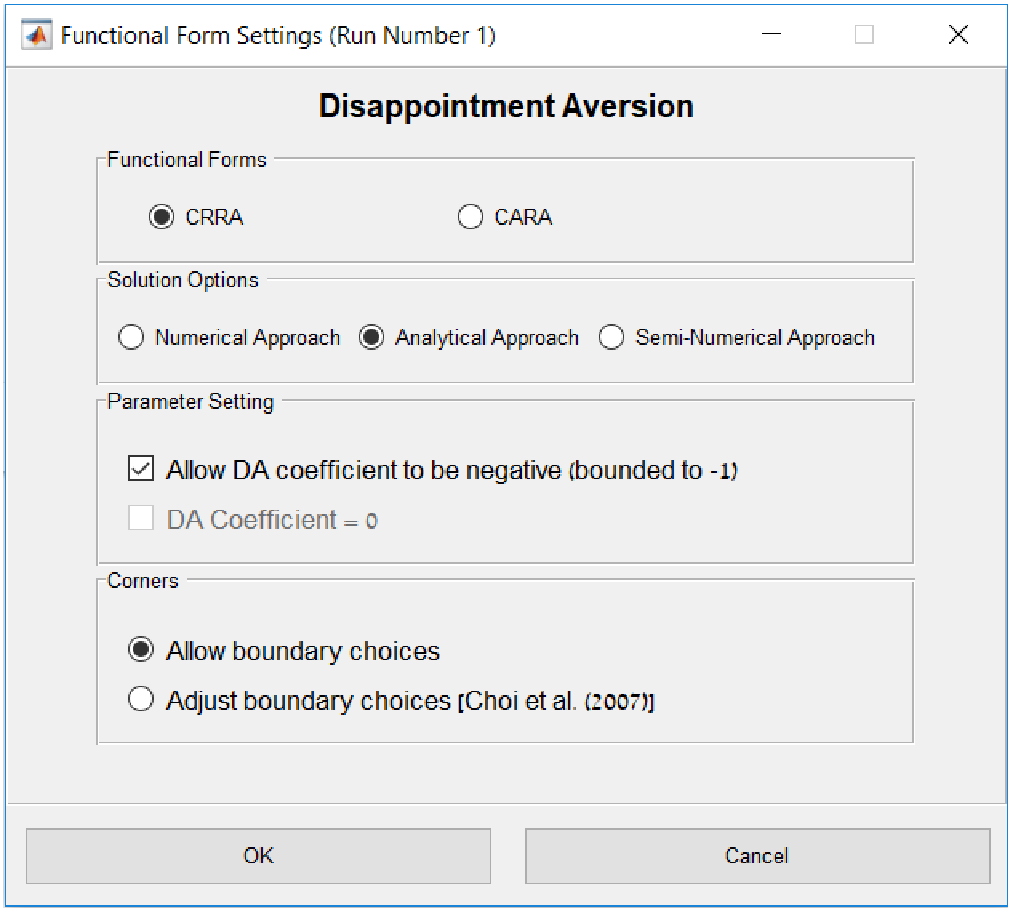


Figure 7.1: Functional Form Settings Window - Risk Preferences

In case the researcher wishes to introduce **a new functional form** we propose the numeric approach which requires minimal adjustments to the code and no analytic calculations. In addition, for the MMI and BI, we propose also a semi-numeric approach where the researcher is required to pre-solve only the Utility Maximization Problem. This may be useful since the Expenditure Minimization Problem may be hard to calculate analytically and implement computationally. For technical details please consult "Code Map and Program Guide.pdf".

3. **Parameter Setting** – The user has three options here:

   - Checking only the upper box allows for $\beta \geq -1$.

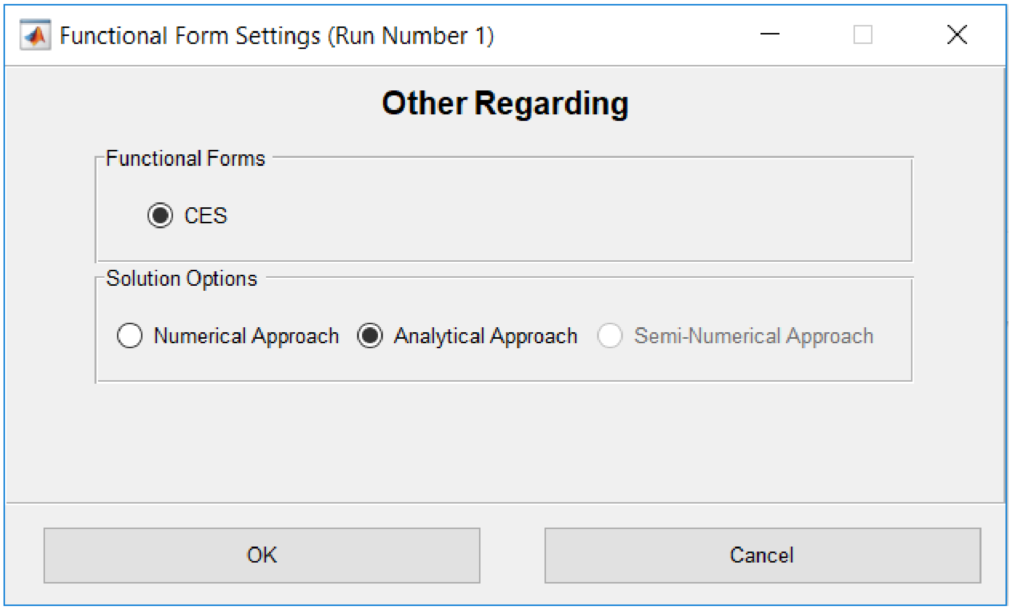


Figure 7.2: Functional Form Settings Window - Other Regarding Preferences

- Checking only the bottom box fixes β to be zero.
- Leaving both boxes unchecked restricts β to be non-negative ($\beta \geq 0$).

4. **Corners** – This issue is relevant only to the case where CRRA is the chosen functional form. In this case boundary choices may become problematic, and the user is suggested with two options:
    - Do nothing.
    - Adjust the corner choices so that the dataset will include only internal choices (the method applied follows Choi et al. (2007)).

7.2 Other Regarding Preferences[6]

Next, we describe the window that appears when analyzing a data set collected in an experiment where bundles are token allocations and the goods are tokens

[6] Currently we add no datasets of other-regarding preferences. A suitable example will be added once the relevant paper will be published.

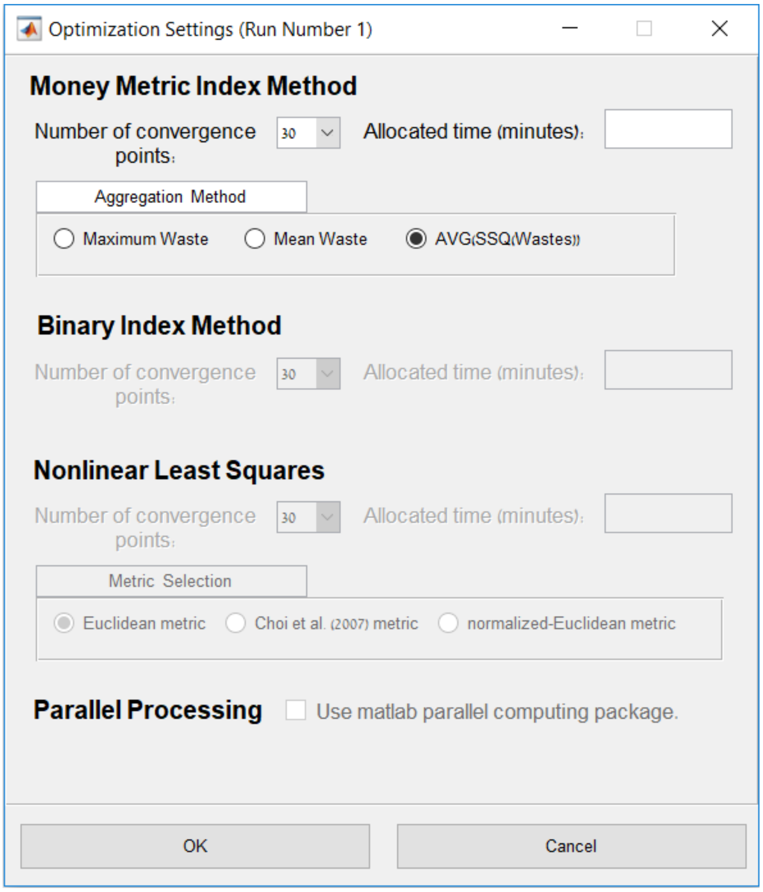


Figure 8.1: Optimization Settings Window

for Self and tokens for Other (see Figure 7.2). In this window the user is required to address the following issues:

1. **Functional Forms** – Currently only the CES functional form is implemented.

2. **Solution Options** – See description for the risk preferences case (the calculations of the first order conditions for the analytic approach are documented in the file "CES-analytic.pdf" included in the package).

8 Optimization Settings Window

In the Optimization Settings window (see Figure 8.1) the user is required to specify the details of the optimization process on the parameters space. For all three methods the user is required to specify:

- **Number of convergence points** – The main part of the optimization process over the parameters space runs MATLAB's optimization routine multiple times, each time with a different set of initial parameters. The chosen number of convergence points specifies the number of times the optimization process requires the "best" result to be replicated in order to terminate. Hence, the process may terminate either on account of "enough" replications of the "best" result (between 3 and 100 in analytic calculation and between 3 and 30 in numeric calculation) or on account of utilizing the whole set of initial points (100 in analytic calculation and 30 in numeric calculation). See Appendix for advanced options to control the numbers mentioned in this paragraph.

- **Allocated time** – The user has an option to add time limit (in minutes) as a third stopping rule for the optimization process. To avoid time limits the user should leave this field empty.

- **Parallel Processing** – If the numerical approach is selected, the user has the option to use the parallel processing package of the MATLAB. This requires to terminate other processes, since it uses all the CPU cores. This option is irrelevant for the analytical approach.

For the Money Metric method, the user is required also to choose the aggregation method:

- **Aggregation Method** – As described in Halevy et al. (2018), the MMI method can aggregate the observation level adjustments using various aggregators. The user's options are (let n be the number of observations and let v^i be the adjustment for the i^{th} observation):

  - Maximum Waste – $1 - \min_{i \in \{1,\ldots,n\}} v^i$.

- Mean Waste – $\frac{1}{n}\sum_{i=1}^{n}\left(1-v^{i}\right)$.
- AVG(SSQ(Wastes)) – $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(1-v^{i}\right)^{2}}$.

Similarly, for the Non-Linear Least Squares method, the user is required to choose the aggregation method:

- **Aggregation Method** – The NLLS method can aggregate the observation level distances between the predicted bundles and the observed bundles using various aggregators. The user's options are (let n be the number of observations, x^i the observed bundle, $\hat{x^i}$ the predicted bundle and p^i the corresponding price vector):
  - Euclidean Metric – $\min_{\beta,\rho}\sum_{i=1}^{n}\left\|x^{i}-\arg\max_{x:p^{i}x\leq p^{i}x^{i}}\left(u\left(x;\beta,\rho\right)\right)\right\|$ where $\|\cdot\|$ is the Euclidean norm.
  - normalized-Euclidean Metric – $\min_{\beta,\rho}\sum_{i=1}^{n}\frac{\left\|x^{i}-\arg\max_{x:p^{i}x\leq p^{i}x^{i}}\left(u\left(x;\beta,\rho\right)\right)\right\|}{\left\|\left(\frac{m}{p_1^i},0\right)-\left(0,\frac{m}{p_2^i}\right)\right\|}$ where $\|\cdot\|$ is the Euclidean norm.
  - Choi et al. (2007) Metric – $\min_{\beta,\rho}\sum_{i=1}^{n}\left(\ln\frac{x_2^i}{x_1^i}-\ln\frac{\hat{x_2^i}}{\hat{x_1^i}}\right)^{2}$ where $\hat{x^i}=\arg\max_{x:p^{i}x\leq p^{i}x^{i}}\left(u\left(x;\beta,\rho\right)\right)$.

9 Output File Format Window

The Output File Format window (see Figure 9.1) enables the user to partially customize the output file.

- **Select the Values Needed for the Output File** – the value of the chosen loss function always accompanies every set of parameters that is reported in the output file. The user can choose to calculate the value of other loss functions given the reported set of parameters. The optional loss functions are (the chosen criterion is grayed):
  - NLLS value based on Euclidean Metric.
  - NLLS value based on CFGK Metric.

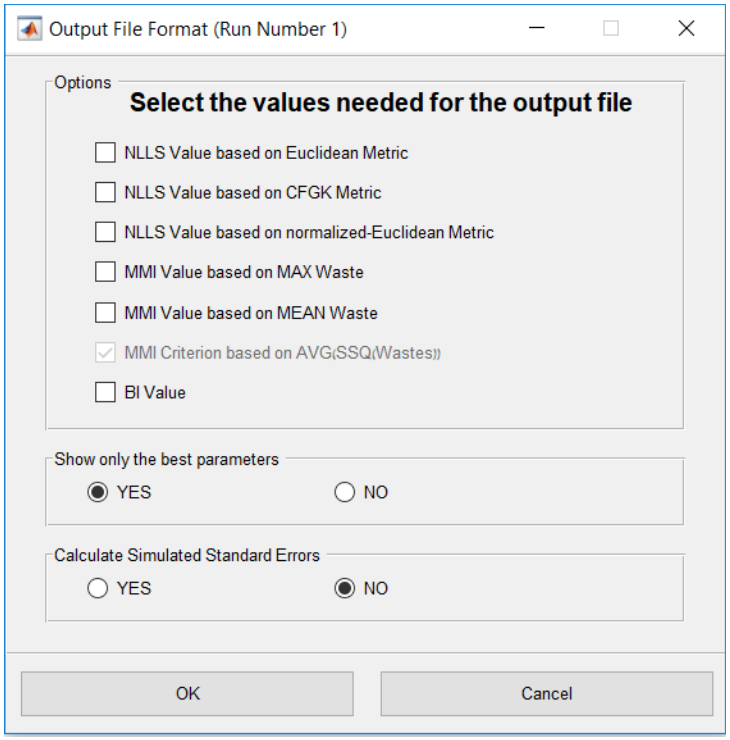


Figure 9.1: Output File Format Window

- NLLS value based on normalized-Euclidean Metric.
- MME value based on MAX Waste.
- MME value based on MEAN Waste.
- MME value based on AVG(SSQ(Wastes)).
- BI Value.

- **Show Only the Best Parameters** – The user can choose the output file to include only the best set of parameters for each subject (by choosing YES) or to include every set of parameters that the code considered as "close" to the optimal set (by choosing NO).

- **Calculate Simulated Standard Errors** – The user can choose to generate a distribution of recovered parameters from 1000 re-samples of the data set. In case YES is chosen, the output file will include, for every

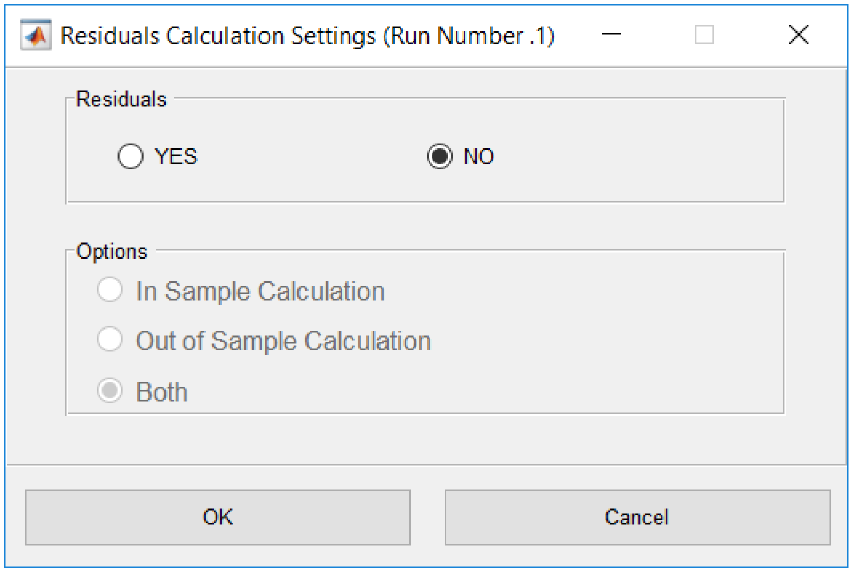


Figure 10.1: Residuals Calculation Settings

subject, the mean, standard deviation, 5% percentile and 95% percentile of this distribution. See Appendix for advanced options to control the significance level and the number of samples.

10 Residuals Calculation Settings

The final window, the Residuals Calculation Settings (see Figure 10.1), enables the user to generate an additional output file that reports the observation level residuals of the recovered set of parameters in terms of adjustments (in the cases of the MMI and the BI recovery methods) or in terms of Euclidean distance (in the NLLS recovery method case). The In Sample Calculation reports the residuals relative to the set of parameters recovered using the complete sample, while the Out of Sample Calculation reports the residuals relative to the set of parameters recovered when the relevant observation is dropped from the sample (and therefore the Out of Sample Calculation is significantly slower).

To avoid the residuals calculation the user should choose NO in the Residuals frame. Otherwise, the user may choose the type of calculation in the

Options frame. When OK is clicked the Running Plan Window pops up.

11 Final Comments

This user manual is meant for users who wish to use the package as is. However the code can be easily altered to support additional functionalities - additional types of preferences, additional functional forms, improved diagnostics, etc. In fact, the code is designed to accommodate some of these additions without the need to be acquainted with the whole package. For advice and support please consult "Code Map and Program Guide.pdf". For any question please contact Dotan Persitz.

References

Choi, Syngjoo, Raymond Fisman, Douglas Gale, and Shachar Kariv (2007) "Consistency and Heterogeneity of Individual Behavior under Uncertainty," *American Economic Review*, Vol. 97, No. 5, pp. 1921–1938, December.

Halevy, Yoram, Dotan Persitz, and Lanny Zrill (2018) "Parametric recoverability of preferences," *Journal of Political Economy*, Vol. 126, No. 4, pp. 1558–1593.

Appendix

A Advanced Options

The Advanced Options window (see Figure A.1) can be accessed using the button at the bottom of the Dataset Selection window (Figure 3.1a).

This window includes various advanced options that some users may find useful. To change any value in this window, the respective Edit button should be clicked first.

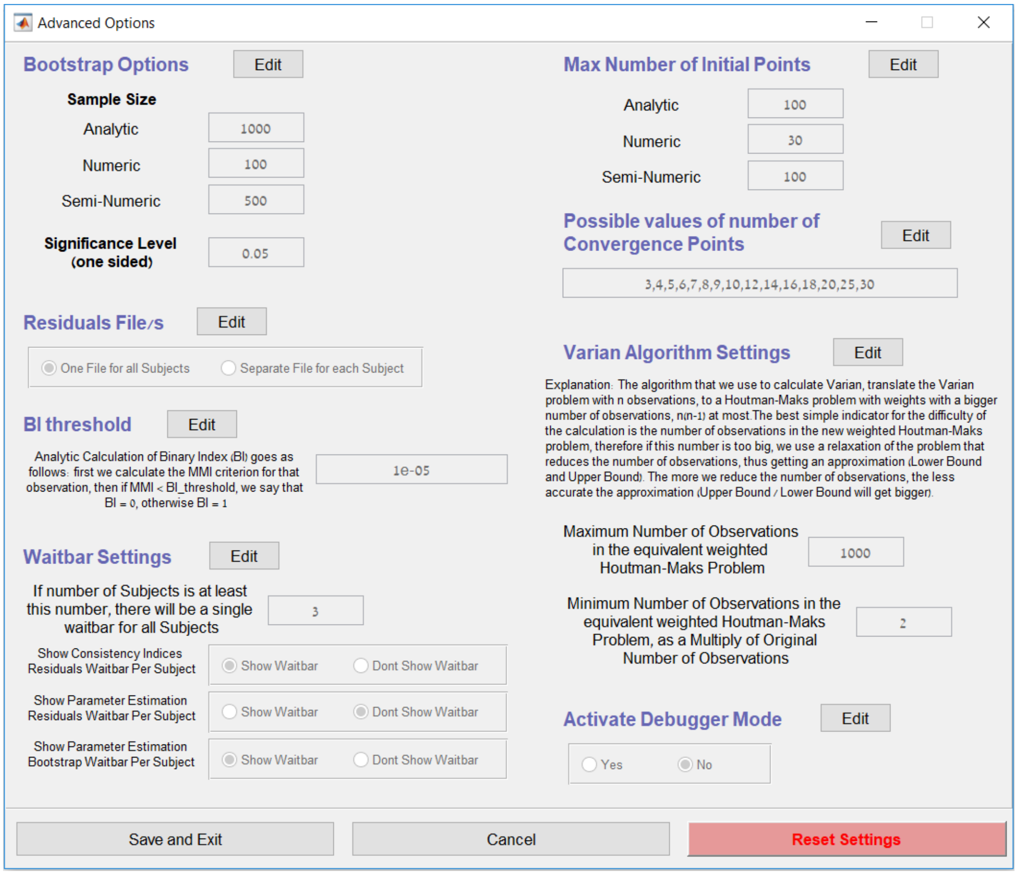


Figure A.1: Advanced Options Window

- Bootstrap Options

  The bootstrap in the parametric estimation (see "Calculate Simulated Standard Errors" in section 9) is set to have 5% one-sided significance level and pre-defined sample sizes (for analytic, numeric and semi-numeric). Here the user can edit these default values.

- Residuals File/s

  By default when residuals are calculated (either for inconsistency indices or for parametric estimation), the results are printed to one file with a separate sheet for each subject. The user can have the residuals analysis of each subject to be printed in a separate file.

- BI threshold
  The Binary Index (see Halevy et al. (2018)) is calculated per observation by first calculating the MMI for the observation given some parameters, and then if the MMI for the observation is more than 0 – it is assigned a BI of 1, and 0 otherwise. Due to computational limitations we give a threshold, 10^{-5} by default, such that observations with MMI value smaller than this threshold will be considered as having a BI of 0 and not of 1. Here, the user can adjust this threshold.

- Waitbar Settings
  By default, if the software was asked to perform calculations on only 1 or 2 subjects, each will have a separate waitbar, and if 3 or more – there will be a single waitbar that shows how many subjects have already been calculated and how many remain.
  In addition, even if there is a single waitbar for all subjects, when consistency indices are calculated with out-of-sample residuals, a second waitbar will appear for each subject showing the progress of the out-of-sample calculation over the subject's observations. Similarly, when parametric estimation takes place with bootstrap, a second waitbar will appear per subject showing the progress over the bootstrap's iterations.

- Max Number of Initial Points and Possible values of number of Convergence Points – See "Number of convergence points" in section 8.

- Varian Algorithm Settings
  The software usually calculates the Varian Inconsistency Index accurately, but at times, especially when the data contain a large number of observations per subject, the calculation becomes too difficult, and an approximation is used.[7]
  This number (1000 by default) determines the level of approximation – the lower the number, the more the software will tend to use an ap-

[7]Our experience shows that for datasets of less than 50 observations no such issues arise, for datasets of 50-100 observations they are unlikely to arise, and for larger datasets approximation is sometimes needed when the dataset is very inconsistent.

proximation, and will also use a more crude approximation (higher ratio between the upper and lower bound of the Varian Index in the results file). We recommend that if for some subjects the computation is too long, try to reduce this number (e.g. 800 or 700) while if the computation is fast but too inaccurate, try increasing this number (e.g. 1200 or 1500).

- Activate Debugger Mode
  When debugger mode is active, a warnings file that contains warnings about criterion values (NLLS, MMI) that do not make sense will be created if such will appear throughout the calculations.