\`

The national institute Model for Lifetime INcome Distributional Analysis, LINDA

This manual provides practical guidance in the use of LINDA, a dynamic microsimulation model that projects a reference population cross-section through time, subject to endogenous savings and labour supply decisions.

User Manual version 3.30

J. van de Ven

19 January, 2022

The National Institute Lifetime INcome Distributional Analysis model LINDA

User Manual

Table of Contents

[1 Introduction 3](#_Toc93498882)

[2 Model set-up 5](#_Toc93498883)

[2.1 System requirements 5](#_Toc93498884)

[2.2 Loading the model onto a new computer 5](#_Toc93498885)

[2.3 Extracting base data from the Wealth and Assets Survey 6](#_Toc93498886)

[2.4 Using UKMOD to create a database for imputing tax and benefit payments 6](#_Toc93498887)

[2.5 Creating a simulation base 7](#_Toc93498888)

[3 Altering model parameters with Excel forms 8](#_Toc93498889)

[3.0 FORM 0: SIMULATION TASK 9](#_Toc93498890)

[3.1 FORM 1: NEW SIMULATION 10](#_Toc93498891)

[3.2 FORM 2: TAX AND BENEFIT OUTPUTS 12](#_Toc93498892)

[3.3 FORM 3: USER DEFINED ANALYSIS ROUTINES 14](#_Toc93498893)

[3.4 FORM 4: DECISION MAKING 15](#_Toc93498894)

[3.5 FORM 5: SELECTED EMPLOYMENT PARAMETERS 17](#_Toc93498895)

[3.6 FORM C1: STATE CONTRIBUTORY PENSIONS 18](#_Toc93498896)

[3.7 FORM C2: STATE PENSION AGE THRESHOLDS 19](#_Toc93498897)

[3.8 FORM C3: PRIVATE PENSIONS 20](#_Toc93498898)

[3.9 FORM C4: INDIRECT TAXATION 21](#_Toc93498899)

[3.10 FORM Z: RUN MODEL 22](#_Toc93498900)

[4 Altering Simulated Tax and Benefits Policy 23](#_Toc93498901)

[4.1 Transfer payments imputed from a reference database 23](#_Toc93498902)

[4.1.1 Setting-up a new reference database 23](#_Toc93498903)

[4.1.2 Reflecting trend variation of the policy environment 24](#_Toc93498904)

[4.1.3 Directing LINDA to reference a new database 24](#_Toc93498905)

[4.1.4 Output variables imputed from a reference database 24](#_Toc93498906)

[4.2 Programmed transfer payments 25](#_Toc93498907)

[5 Running the Model and Simulation Output 26](#_Toc93498908)

[5.1 Running the model 26](#_Toc93498909)

[5.1.1 Stacking simulation counterfactuals 26](#_Toc93498910)

[5.2 Simulation output 27](#_Toc93498911)

[5.2.1 Benefit unit level micro-data 27](#_Toc93498912)

[5.2.2 Post-simulation summary statistics 30](#_Toc93498913)

[6 Analysing Model Output 35](#_Toc93498914)

[6.1 Step 1: specify policy parameters 35](#_Toc93498915)

[6.1.1 Adjust LINDA parameters 35](#_Toc93498916)

[6.1.2 Inspect simulated tax and benefit payments 37](#_Toc93498917)

[6.1.3 Preparing the simulation for launch 37](#_Toc93498918)

[6.1.4 Inspecting the model parameters 38](#_Toc93498919)

[6.2 Step 2: run model and generate output 39](#_Toc93498920)

[6.3 Step 3: think about likely incentive effects 39](#_Toc93498921)

[6.4 Step 4: analyse model output 39](#_Toc93498922)

[6.4.1 Analysis of lifetime decile statistics 40](#_Toc93498923)

[6.4.2 Analysis of population aggregates 40](#_Toc93498924)

[6.5 Step 5: package results 40](#_Toc93498925)

[Appendix A: End User License 41](#_Toc93498926)

[Appendix B: The Utility Function 42](#_Toc93498927)

# Introduction

This manual describes use of the National Institute's *Lifetime INcome Distributional Analysis* model, *LINDA*, which is designed to explore the effects of changes to the tax and benefits structure on household circumstances through time. The model generates panel data for the evolving population cross-section, and a series of summary statistics for each considered policy environment.

LINDA is complementary to other analytical approaches that are now in use. Current large-scale static microsimulation models (including IGOTM, TAXBEN, and UKMOD) are able to provide detailed information regarding the immediate financial implications of policy change. However, such models are not well adapted to consider how savings, employment and consumption can be expected to adapt to altered financial incentives. Econometric analyses can go some way to filling in this missing detail, but not where uncertainty is likely to influence decision making. LINDA is specifically designed to explore savings and employment responses to policy change in context of important aspects of uncertainty that individuals face. The cost of this approach is that it is unable to reflect the degree of detail that is commonly taken into account by the two alternative analytical approaches (static microsimulation and econometric projections) that are referred to above. Thus, a thorough basis for balanced policy advice is best achieved by considering the same issue from alternative analytical perspectives, with LINDA providing one of these alternatives.

There are currently no tools that reliably forecast consumption or savings responses to the evolving policy context over any substantial time period (e.g. 5+ years). Nevertheless, it is often useful to consider how policy reforms could be expected to impact individuals at different times during the life course, and how such effects aggregate through time. LINDA stands at the current technological frontier for this purpose, projecting panel data that describe the the logical implications of alternative policy environments, given a set of explicit assumptions concerning how the future will evolve. While the projections derived from LINDA do not represent forecasts[[1]](#footnote-2), they can provide a unique insight into the influence that policy counterfactuals have on the evolving population, including incentive effects and associated behavioural responses.

LINDA starts from cross-sectional data for the benefit units of a sample of reference adults drawn from the Wealth and Assets Survey. As such, use of the model is limited to individuals who have been granted access to the Wealth and Assets Survey microdata.[[2]](#footnote-3) A benefit unit is defined as one or two adults (living together as a couple) plus any dependent children living with them. The user is first directed to run a simulation that projects the circumstances for the reference population cross-section back through time, and the evolving population cross-section forward through time. This base simulation allows the model to describe the complete life-history of each adult represented in the reference cross-section. The micro-data generated by the model are saved by the model, and used as the “base” data against which subsequent policy-specific projections can be compared.

Beyond this manual, a wide range of resources exist to aid in the use of the model. These resources, which include access to aspects of the underlying model code, video tutorials, and related research papers, can be accessed via a dedicated website at [simdynamics.org](http://www.simdynamics.org). A technical description is provided with the model, in addition to this user manual. Published journal articles also include van de Ven (2017a), and van de Ven (2017b).

This manual is divided into six sections. Section 2 describes how to set-up LINDA for analysing the effects of policy counterfactuals. Section 3 describes use of a set of Excel macro forms that have been designed to assist with alteration of selected model parameters and simulation tasks. Section 4 describes how to alter simulated tax and benefits policies. Section 5 describes how to run the model and the resulting output. Section 6 provides some advice concerning best-practice methods of analysis.

# Model set-up

## System requirements

LINDA is designed to operate on desktop workstations and the *Microsoft Windows* operating system. We recommend minimum system specifications of a 64-bit operating system, computing processor(s) with at least 6 physical cores, 32GB of RAM, and 1TB of hard disk space.[[3]](#footnote-4) *Microsoft Excel* is required to generate parameter alternatives and analyse summary statistics reported by the model. A computer screen not less than 17” in diagonal dimension is also required.

## Loading the model onto a new computer

The model is generally delivered as a single \*.msi file. This file extracts the files required to run the model, along with the associated directory structure. Double click on the \*.msi file, and work through a standard set of install options. Note that you may receive a security warning that the file is from an “unknown publisher” – this is because we have not obtained a (costly) software publisher’s certificate. The user license agreement is included in Appendix A to this document. Minor issues to note include:

* The model can be installed into any directory of your choosing, outside of the folders that are protected by the operating system (like the “Program Files” folder). As you will need to obtain ready access to the model folder, we suggest you save it in a directory that you access frequently. Also, the directory name should not include the character “.”
	+ The remaining discussion denotes this directory as “LINDA\”
* Model updates are delivered in the same fashion as the full model install. If an update is installed in the same directory as the original model, then it will over-write all pre-existing model files except those that have been generated by the user. This means that all of the simulations that you have generated via a previous version of the model will not be deleted when you up-date the model.

The installer creates three subdirectories: CODE, DOCUMENTATION, and MODEL.

* The CODE subdirectory includes a Visual Studio project for adding tax and/or benefit schemes programmatically (LINDA\CODE\Taxes, see Section 4), and for writing your own post-simulation analysis routines (LINDA\CODE\UserAnalysis, see Section 3.3).
* The DOCUMENTATION subdirectory includes the model’s user manuals.
* The MODEL subdirectory includes files that are required to run the model.

The MODEL subdirectory contains four subdirectories in addition to a set of model files.  The subdirectory ANALYSIS\_FILES contains a set of Excel files that are used by the model’s in-built analysis routines, and an additional subdirectory that stores some related statistics. The subdirectory BASE\_FILES contains a separate subdirectory for each "base" specification that you create with the model, in which files that are required for the respective base specification are stored (which we return to discuss below).   The subdirectory SIMULATIONS will contain a separate subdirectory for each simulation that you run, in which are stored the panel data generated by the model, model parameters, and excel simulation output.   The TAX\_DATABASE directory will contain files describing the tax databases used by the model to project taxes and benefits.

## Extracting base data from the Wealth and Assets Survey

As noted in the introduction, the model starts with data reported by the Wealth and Assets Survey (WAS) for a population cross-section of reference adults. A Stata program (e.g. “derive WAS 2017.do”) is provided in the model subdirectory “LINDA\MODEL\BASE\_FILES\”, which can be used to extract and format the WAS data ready for importing into the model. Unless otherwise stated, the model has been specified to use the standard End User License version of the WAS available through the UK Data Archive. Please open the file “derive WAS 2017.do”, edit the directory locations as indicated by comments at the top of the file, and run it to generate the base dataset.

## Using UKMOD to create a database for imputing tax and benefit payments

LINDA is designed to use two complementary methods for imputing taxes and benefits, discussed at length in Section 4. The public download version of the model is set-up to impute taxes and benefits from a reference database generated by UKMOD. Here, we focus exclusively on the steps involved to obtain the reference database using UKMOD.

1. Request access to UKMOD via the following link: <https://www.microsimulation.ac.uk/ukmod/access/>
	1. You should follow the directions to request access to UKMOD and the Family Resources Survey data for 2017
2. Download the EUROMOD installer file from the following link: <https://www.microsimulation.ac.uk/euromod/access/>
3. Run the EUROMOD installer file, and accept all default settings
	1. Do not launch the program immediately.
4. When you are granted access to the UKMOD, you should receive credentials for accessing an online file repository, called Redmine.
5. Access Redmine, and use the site navigation (accessible from the top right drop-down box) to Download the UKMOD installation files and the FRS data files.
	1. Your access email should provide a link to the UKMOD installer files
	2. The FRS data are located under UKMOD > Data > Single wave UKMOD input data > UK\_2017\_latest\_release
6. Extract the UKMOD files
	1. UKMOD is an add-on to EUROMOD, delivered as a set of zip folders
	2. Extract the installer files to a folder of your choosing, which we denote by UKMOD\
	3. Extract the FRS data files to UKMOD\Input
		1. The input directory is packaged within the zip file: e.g. C:\LINDA\ UKMOD\_PUBLIC\_A2.51+\Input
7. Open EUROMOD
8. Set the EUROMOD “Project Folder” to UKMOD\
	1. e.g. C:\LINDA\UKMOD\UKMOD\_PUBLIC\_A2.51+
9. Select “UK” under “Countries”
10. Run UKMOD
11. Select country: UK; system: UK\_2017; dataset: UK\_2017\_a4
12. Run, and allow EUROMOD to finish
13. In Excel, open UKMOD\Output\uk\_2017\_std.txt
	1. In the import wizard use “Tab” delimiters
14. Save as LINDA\MODEL\TAX\_DATABASE\UKMOD\base2017.csv
	1. CSV (Comma delimited) (\*.csv) – note, **not UTF8 format**
	2. When prompted, “do you want to keep using that format?”, select “yes” and close Excel

## Creating a simulation base

As noted in the introduction, the model starts with data reported by the Wealth and Assets Survey for a population cross-section of reference adults. The model parameters have been calibrated to match the model to a wide range of summary statistics calculated from survey data sources, with the calibration structured around the year in which the reference population was observed (see the technical documentation supplied with the model for details). The model comes packaged, ready to project the circumstances of the population cross-section forward and backward through time, to build up a complete life history for each reference individual. It is recommended that this be done, and that the associated data should be defined as the “base” for subsequent simulations. This can be done by “double-clicking” the file “LINDA\MODEL\SIDD.exe”.

A window showing the model execution should open, and provide information to let you know what the model is doing. When the model simulation is complete, a total time for execution should be reported, and you should see a cursor indicating that the window is ready for a new command.

1. Close the execution window
2. Open the MODEL subdirectory, and then open “Job File.xls”
	1. note that a “Security Warning” may appear in Excel if macros have been disabled
3. Ensure that you allow macros to work in Excel
	1. please ask your system administrator if you require assistance with this
4. Press ALT+F8
5. Select “SIDD”, and press the “Run” button
6. Enter “base2017” into the text-box with the title “name of run to adopt as new base”
7. Enter “base2017” into the text-box with the title “directory name for new base”
8. Press the “CONVERT RUN TO NEW BASE” button

Excel will then work away for a short while, after which you should receive a message confirming that the new base has been created. Please note that the macro may look as though it has frozen at this stage – please give it some time to complete.  If you look in the BASE\_FILES subdirectory, you should now see a new subdirectory with the name “base2017”, which includes all of the files defining the base simulation specification.

# Altering model parameters with Excel forms

Most of LINDA’s key parameters are stored in the spread sheet “LINDA\MODEL\Job File.xls”. Altering this file name may prevent the main executable file SIDD.exe from locating the model parameters (see Section 4.1.1 for discussion). “Job File.xls” is comprised of a number of worksheets. The parameters which drive the model are displayed in the worksheet *input*. Parameter values of the base simulation are stored in the sheet *inputA.* Expert users can make changes directly to the parameters described in the *input* worksheet. Alternatively, a system of Excel macro forms is included with “Job File.xls” to assist alteration of selected parameters. Differences between the data stored in the *input* and *inputA* worksheets are identified by 1s in the *check* sheet. Any other sheets included in “Job File.xls” are beyond the scope of this manual, and should not be altered by the user.

To use the front-end, users should open “Job File.xls” and run the macro “*SIDD”*, visible in the Tools/Macros window (Alt+F8). This displays Form 0.

## 3.0 FORM 0: SIMULATION TASK



**TIP: Explanatory notes can be found by pressing the “?” buttons.**

Form 0 offers the user a series of alternative options for running LINDA.

*New Simulation*

Pressing the “NEW SIMULATION” button will lead to user forms that are designed to guide adjustment for a selected set of model parameters governing a new simulation. The related forms are returned to below, starting with “Form 1”.

*Start From Existing Job File*

Forms can be initialised using parameters described in the existing *input* sheet of Job File.xls by clicking the “START FROM EXISTING JOB FILE” button.

*Analyse Existing Simulation*

The model permits a series of supplementary analyses to be run after a simulation is complete. Pressing the “ANALYSE EXISTING SIMULATION” button will open a new form that allows the user to choose which additional analyses will be performed. This alternative is returned to under “Form 2” below.

*Convert Run to New Base*

As discussed in the introduction, the model projects a population through time assuming a series of “base” parameters. The base from which model projections are made can be re-specified to reflect any simulation that the user has previously run by listing the simulation name in the “NAME OF EXISING RUN TO ADOPT AS BASE” text box, and pressing the “CONVERT RUN TO NEW BASE” button. A name for the new base must also be entered, and all associated files will subsequently be stored under the given name in the subdirectory “base\_files” of the main model directory. The model will subsequently use the specified simulation as its base for simulating a population forward through time.

## FORM 1: NEW SIMULATION



*Simulation Name*

The user must provide name for each simulated scenario. The name can be flexibly defined, but should not include any of the following characters: < > : “ / \ | ? \*. The results of the simulation are stored in a sub-directory with the supplied name. We also recommend avoiding very long simulation names.

*Start year of simulation*

The model will project forward, starting from data described for the year defined in this form.

*End year of simulation, and maximum population size*

The model is designed to project the evolving population cross-section forward through time. Projecting a population cross-section forward through time involves expanding the simulated population size, to accommodate entry into the sample of immigrants and the maturation of dependent children. The model will augment the simulated population in every year that it projects forward until it encounters either of the following conditions: (i) it reaches the end of the simulated time horizon, or (ii) it reaches the maximum population size. Thereafter, no additional individual is added to the simulated population.

*Price reference year for simulated output*

The simulated output includes a wide range of financial statistics, all of which are defined as real (inflation adjusted) relative to a given reference year. The reference year that the model uses to generate these statistics is defined here.

*Tax database for simulation*

LINDA can use two methods to simulate taxes and benefits: draws from a reference database, and functional programming. These methods are described at greater length in Section 4. The model is directed to use a database for projecting tax and benefit payments by supplying a set of names describing the files that store the necessary data. These terms can be provided via the current form.

Up to two databases can be supplied for each simulation, with differences between them used by the model to reflect policy change. A *reference year* should be supplied for each database, defining the year in which the respective database should be assumed to apply. A state pension age should also be supplied for each database, defining the age from which state pensions are assumed to be payable within the respective database. Note that the state pension age(s) defined here used only for referencing the database; the state pension age assumed for projecting the population is defined in Form C2 (below).

A *subdirectory name* should be provided for each database, along with up to three file names that store the base data. At load, the model will search for each file name in the subdirectory: “LINDA\MODEL\tax\_database\XXX\” where “XXX” is the subdirectory name defined in this form, and will provide a warning if the respective data could not be found. If no reference database should be used for the analysis, then directory and file names described in this form should be left blank.

## FORM 2: TAX AND BENEFIT OUTPUTS



*Variables imputed from reference database(s)*

Up to five variable names recorded by the database files used to impute taxes and benefits can be requested for inclusion in the set of data output by the model. All variables should be limited to reporting numerical data, and each should be identified as a tax, benefit, or “other” variable. Tax and benefit variables are imputed by LINDA in the same way as described for disposable income under Section 5.1. If a reference database is used to impute transfer payments, then the model will report any variable listed under this form, using the variable name with the “.csv” extension. The model will also report a tax residual under the name “taxdb\_residual.csv”. The tax residual is equal to taxable income[[4]](#footnote-5) plus the sum of any requested intermediate benefits imputed from a reference database, less the sum of any requested database taxes, less imputed disposable income.

*User programmed taxes and benefits*

If taxes and benefits are evaluated by user programmed functions, then the *number of taxes* and the *number of benefits* reported by the user defined functions should be provided at the bottom of this form. Otherwise, the values of each of these text boxes should be set to zero.

*Analyse Taxes*

It is sometimes useful to explore what taxes and benefits the model will project given alternative benefit unit characteristics, which is possible using the “ANALYSE TAXES” button. Note that this option is exclusive, in the sense that a bespoke test simulation will be run that focusses exclusively on analysing the tax and benefits system.

## FORM 3: USER DEFINED ANALYSIS ROUTINES



*Population summary statistics*

This post-simulation routine produces summary statistics designed to reflect the cross-sectional state of the economy. Results are saved in Excel files are saved in the respective simulation subdirectory (defined in Form 0), with name(s) 'hi\_level\_statsYYYY.xls' where YYYY denotes the considered year.

*Statistics for equivalised income deciles*

The model can be directed to produce simulated averages for population deciles, specified by equivalised disposable benefit unit income. The revised OECD equivalence scale is used to adjust disposable incomes. If the model is directed to analyse data for a population cross-section, then the relevant year should be included in the form as directed. Otherwise, the birth year of the cohort of interest should be provided. Associated statistics are saved in the respective simulation subdirectory under the file name “analysis\_dec.xls”.

*Bespoke code for post-simulation summary statistics*

It is possible to program your own routines for generating post-simulation statistics. The entry point for the related code (in C) is found in the directory “CODE/UserAnalysis”.

## FORM 4: DECISION MAKING



Projections made by LINDA typically proceed in two discrete stages. In the first stage, the model evaluates a look-up table that describes utility maximising decisions for any feasible combination of individual specific characteristics (the model state-space). In the second stage, LINDA projects panel data at annual intervals for the evolving population cross-section. This two-stage procedure is necessary to capture ‘rational’ decision making, but can lead to simulation run-times that limit flexibility of the analyst. The model has consequently been designed to allow the analyst to project decisions using a look-up table saved by an existing simulation, which can be selected via the options at the top of this form.

*Allow for discrete change between birth cohorts*

Forward looking policy variation is typically managed by interpolating between the circumstances of birth cohorts born at different times. This approach provides a reasonable approximation so long as policy change influences the broad population cross-section; for example, a change in income tax rates. The approach is less adequate, however, where a policy intervention affects discrete birth cohorts; for example, a forward change in state pension age. It is possible to focus on behavioural responses to a shift in policy affecting discrete birth cohorts by supply a birth year defining the threshold over which the change should be assumed to take place.

*Preference Parameters*

The preference relation assumed for the model is described in Appendix B of this manual. Key parameters governing the nature of the preference relation can be altered within this sheet. Please note that the preference parameters that are supplied with the model are the product of a detailed (and complex) calibration process (see the technical report accompanying the model, and van de Ven, 2017b, for detail). It is advised that these parameters should be altered only for the purpose of exploring model sensitivity to alternative assumptions*.*

*Option to not save simulated panel data*

LINDA generates a large amount of output, which can take a substantial amount of time to save. Exploratory analyses often involve repeated simulations where the focus of interest is exclusively on a small set of post-simulation summary statistics. In this case, simulation run-times can be shortened by opting to not save simulated panel data, via the checkbox at the bottom of the current form.

## FORM 5: SELECTED EMPLOYMENT PARAMETERS



*Employment options*

Here the user can choose between alternative specifications for the simulated labour supply decision of each adult: full-time/not employed; full-time/part-time/not employed; and an option that allows the user to define an arbitrary number of labour supply alternatives. When the third of these options is chosen, then the model assumes that the same hourly wage rate applies to all labour alternatives. If the full-time/part-time option is chosen, then hourly wage rates may vary by the labour decision.

*Minimum hourly wages*

Tick this box to impose minimum hourly wage rates. Minimum hourly wage rates can be defined to vary over four age bands, and values should be set for the year from which the respective minimum applies. The model applies the NMW by assuming that any individual who’s underlying productivity implies a lower hourly wage rate at a given employment option (e.g. full-time / part-time) than the prevailing minimum wage cannot find work at the respective employment option. The temporal growth rates to assume for minimum wage rates in forward projections are also be defined here

## FORM C1: STATE CONTRIBUTORY PENSIONS



Form C1 allows two forms of State contributory pension to be included in the analysis. Contributory Pension 1 offers a flat-rate increase in the pension payable from state pension age for each year that contributions are accredited during the working lifetime, and is designed to reflect the basic State Pension. Contributory Pension 2 provides pension benefits from state pension age that can increase with earnings during the working lifetime, and is designed to reflect the State Second Pension.

## FORM C2: STATE PENSION AGE THRESHOLDS



This is the age from which state pensions are taken. State pension age is also used as one of the matching criteria for imputing taxes and benefits from a reference database, and can be referenced by functional tax routines programmed for the model.

## FORM C3: PRIVATE PENSIONS



Form C3 allows the parameters of personal pension schemes available in the model to be set. A single personal pension has been included in the model parameterisation to limit associated computation time. Pension contributions are specified as a percentage of labour income, and only if labour income exceeds the lower threshold defined in this form. Caps on contributions can be administed within three mutually exclusive age bands, and the tax relief given to contributions can also be subject to limits. Private pension wealth is assumed to attract a fixed rate of return, and a fixed share of private pension wealth is assumed to be taken as a lump-sum, with the remainder used to purchase a life annuity.

## FORM C4: INDIRECT TAXATION



LINDA can be directed to include an allowance for indirect taxes in the simulated analysis. In this case, the model uses reduced form regression equations to disaggregate aggregate consumption (which is simulated endogenously) into the consumption categories that are subject to alternative tax rates. The model therefore accounts for income effects associated with indirect taxes (ie the reduction in aggregate purchasing power), but not price effects (ie the influence of indirect taxes on relative prices of alternative consumption subgroups). Please contact the NIESR for further details.

## FORM Z: RUN MODEL



This final form allows the user to go back to Form C4, to run the model, or exit the Excel forms. Three options are also provided for running the model. Running the “course model” achieves a fast run-time, but at the cost of numerical accuracy. This option should be used for exploratory analyses only. At the other end of the scale, running the “fine model” implies a relatively long run-time, to obtain a high degree of numerical accuracy. An intermediate option between these two extremes is also provided, and we recommend using the “medium model” specification in general.[[5]](#footnote-6)

The base specification has been set up to balance the competing objectives of accuracy with computation time. It is recommended that this base specification be maintained, and that the alternative model specifications be used for the purpose of sensitivity analysis only.

Finally, the user can direct the model to adjust capital and labour prices to reflect a General Equilibrium in a closed economy. This option is discussed at further length in Section 6, under the subheading “High-level summary statistics”.

##

# Altering Simulated Tax and Benefits Policy

LINDA is fundamentally designed to explore the effects of policy counterfactuals. This is done by comparing projections under a counterfactual policy context against those of a “base” context. In this section we describe how to define a policy counterfactual.

LINDA uses two complementary methods to simulates taxes and benefits. The first method is to project transfer payments based on a reference database generated by a third-party tax-benefit-calculator. The second method is to project transfer payments using programming code that is integrated with LINDA. These two methods can each be used in isolation or in concert.

The default parameterisation that is delivered with LINDA uses only calls to a reference database, with the entry point for programming code to simulate taxes and benefits left empty for user adaptation. Matching methods are used to impute taxes and benefits for simulated benefit units from data described by the reference database. These matching methods refer to the age of the eldest unit member, relationship status, the number and age of dependent children, labour status of each adult, and taxable income; see van de Ven (2021), Section 8.1 for technical details.

Consider a policy reform that could be captured by the set of variables included in LINDA. Then the decision to model the reform by altering the reference database or writing bespoke programming code depends on whether the reform can also be captured by the tax-benefit-calculator, and be reflected adequately by the set of matching characteristics referenced by LINDA for imputations. If so, then adaptation of the tax-benefit-calculator may be preferred. Alternatively, writing bespoke programming code should be applied. The remainder of this section discusses each of these alternatives in turn.

## Transfer payments imputed from a reference database

LINDA is designed to integrate flexibly with third-party (static) tax and benefit calculators. Third-party calculators of this sort can now be found for many countries, and include EUROMOD maintained by Eurostat for European Union countries and the UK (see also UKMOD), the Inter-Governmental Tax Model (IGOTM) maintained by UK Treasury, and TAXBEN maintained by the Institute for Fiscal Studies.

Each third-party tax and benefit calculator has its own user interface, and the associated documentation should be consulted to determine how to evaluate a policy context of interest. By default, LINDA is supplied assuming a database derived from UKMOD, as discussed in Section 2.4. The current discussion focusses on how to set-up LINDA to use a new reference database for imputing taxes and benefits, and the associated options that are available. For details on use of UKMOD, on-line training resources are provided at: [www.microsimulation.ac.uk/ukmod/resources](http://www.microsimulation.ac.uk/ukmod/resources)

### Setting-up a new reference database

Data from the tax-benefit-calculator should be saved in comma-separated-variable format. The first row of data of each file should provide variable names, and all variables should be limited to numerical information. Please delete any non-numerical variables generated by the calculator before attempting to load data into LINDA. Data for the reference database can be uploaded to LINDA in up to three separate files, and files can differ in terms of their respective levels of aggregation (e.g. household, benefit unit, individual). LINDA will integrate the data from the alternative files, and report any errors that it encounters.

The model parameterisation that is delivered with the installer file is designed to integrate with a database derived from a specific tax-benefit-calculator. Changing between database derived from alternative calculators involves amending LINDA’s parameters to reference alternative database variable names. These parameters are defined in column E of “job file.xls”, sheet “inputA”; please contact NIESR for support if you would like further details.

### Reflecting trend variation of the policy environment

Wage growth is a key feature that distinguishes the lifetime labour market opportunities of alternative birth cohorts. Assuming that tax and benefits policy remains time invariant (in real terms) in context of real wages growth can lead to widespread tax-bracket creep and marginalisation of welfare payments. The model has consequently been designed to permit the analyst to off-set both of these effects (see van de Ven 2021, Section 8.1.1 for technical details).

First, and most simply, it is possible to define an average annual threshold growth rate to off-set tax-bracket-creep, and another growth rate to off-set marginalisation of (low income) welfare payments. In case of the former, the model will discount private incomes before matching to the database. In case of the latter, the model will adjust disposable incomes, for a range of low private incomes to reflect the assumed growth rate.

Furthermore, trend variation of the policy environment can be captured by specifying two alternative databases, each of which is associated with a separate simulation year. In this case, the underlying data reported by each database should be identical, and only the simulated taxes and benefit values should vary. LINDA will then use geometric interpolation between the years specified for the two databases to impute tax and benefit values for any given simulation year. This second approach is very flexible, accommodating trend variation of individual tax and benefit rates and thresholds, but is more challenging for the user to define as it requires specification of a counterfactual transfer system that reflects trend policy variation to some alternative (future) year.

### Directing LINDA to reference a new database

The files describing a new policy context, obtained and organised as described in Section 4.1.1, should be saved in the model subdirectory “LINDA\MODEL\tax\_database\XXX”, where “XXX” is a subdirectory name of your choosing. LINDA is directed to reference the new database by amending model parameters that describe the associated file names and locations. These model parameters can be defined using Form 1 of the Excel front-end, as described in Section 3.1.

### Output variables imputed from a reference database

In addition to the standard panel data generated by LINDA following a simulation (see Section 5.2), the model can be directed to report selected variables described by a reference database. Up to five variables reported by the reference database can be included in the model output, which can be defined using Form 3 of the Excel front-end (see Section 3.2).

Where variables describing selected taxes and benefits are requested for output, then LINDA will impute these financial variables in the same way as it imputes disposable income. The model will then output each variable in comma-separated-variable format, along with the remainder of the simulated panel data, with file names set to the respective variable names (as defined in Form 3).

In all cases where LINDA imputes taxes and benefits using a reference database, the model will also report a variable named “taxdb\_residual.csv”, representing the residual net tax payment. This residual will be negative where a net benefit is projected to be received by the model and is calculated as:

*taxdb\_residual* = private income - <imputed disposable income> - sum(<imputed taxes>) + sum(<imputed benefits>)

Here, all terms in <> are imputed from the reference database, and private income is pre-tax and benefit taxable income of the benefit unit, sometimes referred to as original income. This is equal to the sum of employment income, private pension income, state pension income, (positive) investment income, less private pension contributions.

LINDA reports two additional variables whenever taxes and benefits are imputed from a reference database. *taxdb\_idX*, where X takes the value 1, 2, or 3, report observation identifiers from the tax database. These variables allow the user to identify which observations reported by the tax database were used to impute transfer payments for each simulated benefit unit in each projected year. Furthermore, *taxdb\_match* reports a statistic that describes the type of match obtained for each individual. This statistic takes a two-digit numerical value. The first digit takes a value equal to 1, 2, or 3, with lower values defining a more precise match. The second digit takes a value between 1 and 9, describing the number of candidate observations reported by the tax database, from which a match was selected.

## Programmed transfer payments

As discussed in Section 2.2, LINDA comes packaged with a CODE subdirectory, in which are stored a series of program files to facilitate integration of programmed tax and benefit payments. The files are organised for integration with the Microsoft Visual Studio Integrated Development Environment (IDE), and can be used to write C code to evaluate bespoke tax and benefit payments. Although use of these files is beyond the scope of this user manual, a dedicated user tutorial exists that provides a detailed walk-through of a simple policy reform. This tutorial can be found at: [www.simdynamics.org/tutorials.htm](http://www.simdynamics.org/tutorials.htm).

# Running the Model and Simulation Output

## Running the model

Having specified model parameters (see Sections 3 and 4), it is time to set the model running. This can be done either through the model Excel front-end – see Form Z in Section 3 – or by launching the SIDD.exe executable program manually. This second option can be followed, by double-clicking on the associated file through File explorer as displayed below:

 

### Stacking simulation counterfactuals

It is possible to set the model up so that it will automatically work consecutively through a list of “job file” specifications.

The Excel front-end works on the assumption that a single simulation will be run, with parameters stored in the default file “Job File.xls”. When the model executable file SIDD.exe first starts, however, it looks in the base model subdirectory (where SIDD.exe is saved) for a file named “job file1.xls”. If it finds a file with this name, then it will load in the parameters saved in that file and perform the associated simulation. When that simulation is complete, the model will jettison the parameters described by “job file1.xls”, and search for a file named “job file2.xls”. If it finds “job file2.xls”, then it will load in the parameters saved in that file and perform the associated simulation. The model will then continue working in a similar way, searching for a file named “job file<X+1>.xls”, after completing the simulation described by file “job file<X>.xls”, until no consecutively higher numbered file is found, at which time the model will stop. If the model fails to find a file named “job file1.xls”, then it will revert to the default file “Job File.xls”, and will stop after the single simulation is complete.

To set-up a series of model simulations to be run in consecutive order, it is possible to work through the Excel front-end described in Section 3 for each simulation in turn, remembering to adopt a different “simulation name” for each specification in FORM 1, and press “EXIT” at FORM Z. Then save the associated excel file with the name “job fileX.xls”, setting X to 1 for the first simulation, 2 to the second, and so on. When all of the desired files have been set-up, double-click on SIDD.exe as described above, and the model will work its way through the simulations in turn. ***Please remember to delete the numbered files “job fileX.xls” when the stacked simulations have completed.*** Failure to do this will mean that the model will perform undesired simulations when used on subsequent occasions.

## Simulation output

The model produces, for each simulated adult, panel data for a range of characteristics over the life-course, and saves associated results in the subdirectory “LINDA\MODEL\simulations\XXX”, where XXX is the name given to the respective simulation (see Form 1). These “micro-data” are reported in a standard format (csv) that is accessible by widely available statistical packages, including Excel. The model can also be directed to generate a series of post-simulation summary statistics, which will be reported in associated Excel files also saved to the subdirectory “LINDA\MODEL\simulations\XXX”. Each of these respective model outputs is discussed separately below.

### Benefit unit level micro-data

The model can be directed to save all of the simulated characteristics projected by each simulation in “Comma Separated Variable” format (CSV). Each simulated characteristic is saved with a separate file name. The data in each file are stored as a two-dimensional table, with each row representing a single simulated adult, and each column a single simulated year. The model is designed to economise the use of disk space by only saving those variables that are actually generated by a simulation (\* indicates variable saved optionally). The file names that may be produced by a simulation, and a description of the associated data, are listed below.

age: age of reference adult

alc\_dy\*: alcohol duty paid (£ per week)

bcohort: birth cohort

benagg: aggregate state welfare benefits (excl. state pensions, £ per week)

ben\_eeX\*: benefit X calculated by C function, where X is order in output vector

ben\_unit: benefit unit identifier, equal to the person identifier (psnno) of the reference adult

carer\*: one adult carer in the benefit unit

comexcc: non-discretionary childcare expenditure (£ per week)

comexhl\*: non-discretionary healthcare expenditure (£ per week)

comexhs: non-discretionary expenditure on rent and mortgage interest payments (£ per week)

comexot: other non-discretionary expenditure (£ per week)

comextt: total non-discretionary expenditure (£ per week)

cons: non-durable discretionary consumption of benefit unit (£ per week)

cons\_ind\*: non-durable consumption of simulated individual (includes both cons and non-discretionary expenditure, £ per week)

cp1\*: number of contribution years for rights to the first state contributory pension (BSP)

cp2\*: second state contributory pension (S2P) (£ per week)

cpinc\*: state pension income (cp1/2/proxy) received (£ per week)

death\_age: age at which reference adult dies

donor: psnno of adult used to initialise characteristics[[6]](#footnote-7)

dppart\*: flag = 1 if default is to participate in personal pensions / 0 to not participate

education\*: highest education level achieved during the simulated lifetime (1 = lowest qualification, n = highest), in order of attainment (right columns come later – interpret with reference to *educndage*)

educndage\*: ages at which education level changes

emp1/2: employment status of benefit unit reference adult / spouse (0 indicates non-employment, and *n* indicates full-employment, where *n* = the total number of possible labour states, minus 1; intermediate numbers indicate intermediate employment states).

emig\_age: age emigrated out of simulated population

eqs: equivalence scale (modified OECD scale)

ful\_dy\*: fuel duty paid (£ per week)

hlthstate\*: indicator for health / disability states – poorer states take higher values

hsgmd: housing mortgage debt (£)

hsgmr: housing mortgage interest (£ per week)

hsgret: capital gain on housing wealth (£ per week)

hsgw: gross wealth held in housing (£)

humcap: benefit unit human capital (wage potential) (£ per week)

ihr\_dy\*: insurance, higher rate duty paid (£ per week)

immig\_age: age immigrated into model population

immig\_type\*: immigrant region of origin (where accommodated)

inherit\_age: age at which inheritance received

inherit\_pno: person number from whom inheritance received (forward simulations)

inherit\_val: value of inheritance (£)

isa\*: wealth held in Individual Savings Accounts (ISAs, £)

isa\_cont\*: contributions to ISAs (£ per week)

isr\_dy\*: insurance, standard rate duty paid (£ per week)

labinc: labour income of benefit unit (£ per week)

leis: proportion of time spent in leisure

na: number of adults in benefit unit (1=singles 2=couples)

net: benefit unit net (disposable) income (£ per week), excluding returns to housing

ninvinc: net investment income on generic “other” wealth (w – hsggw + hsgmd)

nk: aggregate number of dependent children

nk\_allX\*: number of children, in birth age X

nk\_psnno\*: psnno code of matured dependent children

nkd\_allX\*: indicator that one child from birth age X is disabled

parttr1/2\*: participation tax rates of reference person / spouse

peninc: private pension (PP+OP) income received (£ per week)

ppc: private pension contributions (£ p.w.)

PPcont\*: aggregate contributions to private pension (including employer contribution) (£ p.w.)

PPcr\*: type of private pension to which benefit unit may contribute (integer = 1,…, n where n = total number of private pension schemes considered; only reported if more than 1 scheme available).

PPpen\*: aggregate accrued rights to private pension, defined as an annuity stream (£ p.w.)

PPpenb\*: aggregate accrued rights to private pension, defined as a wealth equivalent (£)

prec: age at which pension income first received

prett: pre-tax and benefit benefit unit income (£ per week)

psnno: unique person identifier

region: internal geographic region

 1= North East; 2=North West; 3=Yorkshire and Humber;

 4=East Midlands; 5=West Midlands; 6=East;

 7=London; 8=South East; 9=South West;

 10=Wales; 11=Scotland; 12=Northern Ireland;

ret: whether benefit unit defined as retired for pension purposes

student\*: student status (0 = non-student, 1= type 1 student, 2=type 2 student)

taxdb\_idX: tax database identifier X, where X rises with the detail of the micro-unit (Section 3.1)

taxdb\_match: matching statistic for imputed transfer payments from reference database (see below)

taxdb\_residual: the net impact of transfers imputed from a database (£ per week)

taxagg: aggregate benefit unit tax burden (£ per week)

tax\_eeX\*: tax X calculated by C function, where X is order in output vector

tbc\_dy\*: tobacco duty paid (£ per week)

user\_taxX\*: user defined output X from tax routines

val: measure of expected lifetime utility

vat\_rr\*: reduced rate VAT paid (£ per week)

vat\_sr\*: standard rate VAT paid (£ per week)

vatdy\*: total consumption taxes (VAT) and duties (£ per week)

w: liquid wealth of benefit unit (£)

w2: aggregate benefit unit net worth (£)

wage\_offer\*: flag = 1 if reference adult receives wage offer, 0 if they do not

wage\_offer2\*: flag = 1 if spouse receives wage offer, 0 if they do not

yrenter: year entered simulation sample – sample ordered from earliest to latest year

Total benefit unit net worth is given by:

*w2t = wt + PPbenbt + isat*

The following relationship describes the evolution of liquid wealth in most periods:

*wt+1 = wt + nett + inheritt – const – comexttt­ – isa\_contt –* vatdy *t* (1)

The exceptions are when relationship status changes, when a benefit unit first accesses private pension wealth, or when either the upper or lower bound on *w* is encountered. In the last of these cases, *w* is constrained to remain within the considered bounds.[[7]](#footnote-8)

Net income is given by:

*nett = prettt + benaggt + cpinct – taxaggt – ppct* (2)

*prettt = labinct + peninct + hsgret t + ninvinc t* (3)

Total committed expenditure is given by:

*comexttt­ = comexcct + comexhst + comexott + comexhlt* (4)

Total indirect taxes and duties is given by:

*vatdyt­ = vat\_rrt + vat\_srt + alc\_dyt + ful\_dyt + ihr\_dyt + isr\_dyt + tbc\_dyt* (5)

The variable *net* represents the model’s projection for disposable income before housing costs, as used in the DWP’s HBAI publication. Disposable income after housing costs can be evaluated as: *net* - *comexhs*. Similarly, disposable income net of mortgage interest payments is obtained by: *net – hsgmr*.

The variable *taxdb\_match* takes numerical values equal to XYZ. Here, X takes integer values of 1, 2, or 3, indicating the matching criterion used (1 being finest, and 3 coarsest). Y takes integer values between 1 and 9, indicating the number of candidates from which a donor was selected to impute tax and benefit payments (censored at 9 or more). Z takes values of 0 or 1, where 0 indicates a unique candidate selected for imputation, and 1 indicating that two candidates selected for imputation.[[8]](#footnote-9) Where the second of these applies, then the identification variables taxdb\_idX refer only to the second candidate considered.

### Post-simulation summary statistics

Two key sets of post-simulation summary statistics can be requested using Form 3 of the Excel front-end (Section 3.3).

#### Population summary statistics for selective years

LINDA can be asked to report high-level statistics for up to six selected years following a simulation. These summary statistics fall into two broad categories, which are each represented by a separate table. Simulated population averages for key benefit unit income, consumption, and balance sheet items are reported in one table, and statistics relating to the macro-economy are reported in the other. The population averages that are reported in the first table are reasonably self-explanatory. The statistics relating to the macro-economy, however, warrant further comment.

Economic analyses of the type for which LINDA has been devised can typically be distinguished in relation to their treatment of factor prices. On the one hand, *partial equilibrium* analyses assume that factor prices are fixed; this is also commonly referred to as the small open economy assumption. On the other hand, the alternative assumption is that factor prices adjust endogenously to reflect changes in demand and supply. In the case of LINDA, the analyst can choose between these two basic frameworks, as noted in relation to Form Z. Here, directing the model to generate results for the General Equilibrium in a closed economy results in the model adjusting interest rates and wage rates to reflect changes in the supply of capital and labour that are implied by the respective simulation.

The basic idea is that, if saving increases, then this raises the supply of capital in the economy. Increasing the supply of capital in the economy, all else held fixed, should reduce the rate of return paid to capital. This is simulated by LINDA on the assumption that aggregate production, *Y*, combines aggregate capital, *K*, and labour, *L*, in the form of a CES function:



Assuming that factor markets are perfectly competitive implies that capital and labour are paid their respective marginal products, so that the interest rate, *r*, and wage rate, *w*, are given by:



where ** is the rate of depreciation of capital. The aggregate capital stock is set equal to aggregate benefit unit wealth less government debt. Similarly, aggregate labour supply is set equal to the aggregate wage bill, and these two (capital and labour) are combined to give aggregate production as described by the CES function referred to above. Investment is *I* = (** + *g*).*K*, where *g* is the assumed growth of the economy. Government consumption, *G*, is then calculated to equate aggregate income to expenditure:



where *C* is aggregate benefit unit consumption. It should be noted that we do not consider the issue of the government budget balance because we do not cover all of the forms of taxation that are applied in practice (for example, LINDA currently omits taxation of firms).

When directed to generate results for the General Equilibrium in a closed economy, the model iteratively adjusts interest rates, *r*, and wage rates, *w*, until these are consistent with the associated aggregates generated for output, capital and labour supply. Note, however, there is no guarantee that the model will converge. The analyst is given a warning if the model has failed to find a solution after searching over 25 alternative parameter combinations, and can choose to continue with the analysis or cancel out in that case.

#### Decile-level summary statistics for population cross-sections and birth cohorts

A workbook analysis\_dec.xls is generated by the programme and is placed in the sub-directory given by the simulation name. Users are likely to be interested in both the absolute values and the differences from the base, which are each reported in separate worksheets.

The decile groups reported in the spreadsheet are each comprised of one tenth of the weighted sample population that survives to a given period, where lower deciles describe the circumstances of benefit units with lower lifetime income *as observed in the prevailing policy counterfactual*. Lifetime income is calculated as the present discounted value of the net (disposable) income stream, averaged over the period of survival. Survival probabilities in the model depend only on age and birth year. This means that, when considering evolving characteristics for a given birth cohort, then each benefit unit will (approximately) contribute to the same lifetime income decile throughout its life-course.

The difference statistics hold all factors other than the assumed policy parameters constant between the base and counterfactual simulation populations. Hence, each benefit unit is subject to the same random draws in the two sets of simulations, and each “decile” reported in the spreadsheet is comprised of the same sample of benefit units at all points in time. This is crucially important to distinguishing the simulated responses to changes in the policy environment from other modelling considerations. A useful check that this aspect of the model is working as intended is to run a counterfactual simulation that holds all parameters fixed as the same parameters as for the simulation base. Difference statistics calculated between this test simulation and the base simulation should be very close to zero (subject to some rounding errors).

The statistics reported in the spreadsheet are defined in Table 1. As is implicit in the above discussion, the sheet shows arithmetic differences for all variables except the value function. Differences in expected lifetime utility between a given simulation and the associated simulation base are expressed either as compensating variations (£ of liquid wealth equivalents) or percentage differences, as little meaning can be given to absolute differences.

**Table 1 Data reported in analysis\_dec.xls (see Section 5.2.1 for variable definitions)**

|  |  |
| --- | --- |
| **Variable** | **Notes and Definitions.**  |
| number of benefit units contributing to statistic | Provides a measure of the un-weighted sample size contributing to the reported statistics – this provides an indication of the statistical reliability of the reported statistics (noting that each decile will be comprised of approximately 1/10 of the total population sample, and means based on fewer than 20 observations should be treated as approximate only). |
| number of adults | Describes number of adults in benefit unit. Evaluated using variable na.csv |
| consumption | Consumption, including non-discretionary expenditure (£ per week). Evaluated using variable cons.csv |
| leisure | This is the proportion of the leisure available to someone who does not work. Someone who works full-time is defined as having leisure of 0.5. The ratio of the part-time working week to the full-time working week is user-determined in Form 1 as Part-time Ratio (1) for single people and the Lab Ratio variables for couples. Evaluated using variable leis.csv |
| proportion of households with some employment | Proportion of households with some employment, where households are defined as benefit units (see introduction for definition). Evaluated using variables emp1.csv and emp2.csv |
| proportion of adults employed | Proportion of simulated adults employed. Evaluated using variables emp1.csv and emp2.csv |
| gross labour income | Average income from employment (£ per week). Evaluated using variable labinc.csv |
| average contribution to private pension | Average employee and employer contributions to private pensions over decile (£ per week). Evaluated using variable PPcont.csv |
| propn decile contributing to private pension | Proportion of decile making any contributions to private pensions. Evaluated using variable PPcont.csv |
| private income | Private (pre-tax and benefit) income (£ per week). Evaluated using model variable prett.csv |
| disposable income | Disposable (post-tax and benefit) income (£ per week). Evaluated using model variable net.csv |
| unsecured debt | Average value of unsecured debts by population decile (£). Evaluated using model variable w.csv |
| propn of population with unsecured debt | Proportion of decile with negative net liquid wealth (cash on hand). Evaluated using model variable w.csv |
| net (non-pension) assets | Average value of net non-pension assets by population decile (£). Based only on data for benefit units without unsecured debt, using model variable w.csv |
| pension wealth | Average value of assets held in pensions (£). Evaluated using model variables w.csv and w2.csv |
| total wealth | Average total wealth (£). Evaluated using model variable w2.csv |
| lifetime discounted equivalised net income | Sum of disposable (post-tax and benefit) income, adjusted by the revised OECD scale and price inflation. Evaluated using model variables net.csv and eqs.csv |
| value function | Remaining life-time welfare. Utility Units. Percentage rather than absolute differences should be considered. Evaluated using variable V.csv |
| upper threshold for decile | Threshold used to allocate benefit units to deciles as considered in relevant worksheet, defined in terms of average discounted equivalised net income earned over simulated lifetime (£ per week). Upper threshold of “lifetime discounted equivalised net income” described above. |

# Analysing Model Output

This section provides a brief step-by-step guide concerning how the model should be used to explore the effects of policy alternatives. The departure point for discussion follows the model set-up, as described in Section 2.

The guide has been written so that it will be applicable to a wide range of alternative subjects of interest, from studies concerned solely with the distributional implications of policy, to those that focus on behavioural responses to policy counterfactuals. Each step of a stylised analytical problem is described under a separate section heading, in approximate chronological order. At the end of each step, we provide a practical example of the tasks involved with reference to a policy counterfactual in which all rates of income tax increase by 10 percentage points. This policy counterfactual is also one of those explored as part of the online tutorial series at: [www.simdynamics.org/tutorials.htm](http://www.simdynamics.org/tutorials.htm)

**TIP: It is important that you allow ample time to conduct your analysis.**

## Step 1: specify policy parameters

The available methods for altering tax and benefits policy parameters are described in Section 4. Here, we assume that you have access to the respective tax-benefit-calculator, and have used the calculator to generate three files that conform to the policy counterfactual of interest, and which are suitable for importing into LINDA.

### Adjust LINDA parameters

Suppose that you have three files describing the database, named “one.csv”, “two.csv”, and “three.csv”, and have saved these files into the subdirectory “LINDA\MODEL\tax\_database\10pp\_increase”. LINDA is then directed to draw on these files for imputing taxes and benefits via the Excel front-end:

 

 

Here we have added a new simulation to be named “10pp\_sim”, which will reference the new database files.

### Inspect simulated tax and benefit payments

From Form 1 discussed above, you proceed to Form 2. Here you may choose to add variables that are present in the reference database, and which describe margins of interest in relation to the considered reform. In the current example, you might opt here to ask LINDA to report measures of income tax that are described as part of the reference database.



Having defined any variables of interest, you then have two options. One is to proceed with the model simulations. Alternatively, it is often useful – especially when first starting to use the model – to “eyeball” the taxes and benefits that the prevailing model specification will generate for benefit units with selected characteristics. This pre-simulation analysis can be undertaken by pressing the ANALYSE TAXES button.

After pressing the ANALYSE TAXES button you will receive a message asking you to amend the file LINDA\MODEL\analysis\_files\tax\_analysis2.xls, to describe benefit unit characteristics that LINDA will generate taxes and benefits for. Once you are done, save the file, and double-click on LINDA\MODEL\SIDD.exe, to launch LINDA. The model will take a short while to evaluate taxes and benefits for each of the examples described in the Excel file, after which it will report results to LINDA\MODEL\simulations\<sim name>\tax\_analysis2.xls, where <sim name> is “10pp\_sim” in this example. Check the results, and revise the tax and benefit specification as necessary.

### Preparing the simulation for launch

Having prepared the tax and benefit structure as set out above, access Form 3 of the Excel front-end by pressing ENTER in Form 2. If you analysed taxes as suggested under Section 6.1.3, this will involve opening Job File.xls, pressing the START FROM EXISTING JOB FILE button in Form 0, and then accepting default values by pressing ENTER to Form 3.



Form 3 allows you to request LINDA to evaluate a series of post-simulation statistics. All statistics will be reported to Excel files, saved in subdirectory LINDA\MODEL\simulations\10pp\_sim – see Section 3.3 for details. If there are other terms of the simulation that you would like to alter through the Excel front-end, the press ENTER. Otherwise, press the ENTER and RUN button to go directly to the final user form. You can then press LAUNCH MODEL to run the model directly from the front-end, or EXIT. It is a good idea to press EXIT in most cases, which will give you a chance to inspect the model specification before running the simulation.

### Inspecting the model parameters

The “check” sheet of “Job File.xls” is provided to assist you in identifying all of the parameter differences assumed for a counterfactual, relative to the model base specification. In the current example, the sheet should look something like this:



The total number of parameter differences with the model’s base simulation are reported in cell A1 of this sheet (11). Looking across row 1 of the sheet will indicate which columns include altered parameters. In this case, we see that there are 4 differences in column E, and 7 differences in column BQ. Comparing the respective columns in the “input” sheet against “inputA” sheet should verify that all differences either relate to the change in reference database that were implemented in Form 1 above (column E), or the post-simulation analysis routines requested in Form 3 (column BQ).

## Step 2: run model and generate output

Having verified the model parameters for the new simulation correspond to those that we are interested in, we can launch the model program, as described under Section 5.1

## Step 3: think about likely incentive effects

An important feature of the model is that savings and labour supply decisions are endogenous. This means that these decisions react to the incentives that are embodied by the assumed policy environment. Rationalising results that are reported by the model consequently depends in part on understanding the drivers underlying simulated behaviour. It is useful to think about what behaviour you expect to see before the model completes its analysis, as this will help you to identify where to begin looking once the simulated results are complete. It is easiest to think about incentives relative to the base simulation, as the model can be asked to generate comparisons with the base simulation by default.

In the current example, we have raised the marginal income tax rates by 10 percentage points. This can be expected to have a large number of effects on the simulated population. Some of the most important are listed here in bullet point form:

* + Higher tax rates imply two key effects:
		1. Affected benefit units will have lower lifetime income in the absence of any behavioural responses – this consideration will motivate responses that are sometimes referred to as a “wealth” or “income” effects.
		2. Affected benefit units will be subject to higher effective marginal tax rates on labour income, and on investment income that is not tax-shielded – this consideration will motivate responses that are sometimes referred to as “price” or “substitution” effects.
	+ Likely wealth effects:
		1. Lower consumption
		2. Increased employment
		3. Lower saving
	+ Likely substitution effects:
		1. Higher consumption, as effective post-tax returns to saving decrease
		2. Lower employment, as the marginal returns to labour supply have fallen
		3. Lower saving in assets that are subject to taxes
		4. Increased saving in assets that are tax shielded.

## Step 4: analyse model output

This stage of the analysis involves comparing the model output against the expectations that you formed in Step 3 (described above). It is not uncommon to find that behaviour deviates from what you had initially expected, as the incentives embodied by policy can often be difficult to anticipate (which is one motivation for using LINDA).

### Analysis of lifetime decile statistics

Inspection of lifetime decile statistics present one of the most accessible approaches to infer the drivers of outputs from LINDA. In the current example, we directed LINDA to produce these statistics as part of its post-simulation analysis in Form 3 (see Section 6.1.3). Lifetime decile statistics are informative, because they follow the same group of simulated individuals through their respective life-courses, thereby generating a coherent narrative. These statistics would be reported in file LINDA\MODEL\simulations\10pp\_sim\dec\_analysis.xls in the current example.

A useful place to start when inspecting these statistics are the population average differences with the simulation base, which are reported in the right-most columns of the “differences” sheet. Having rationalised the narrative described reported for population averages, an appreciation of the associated distributional effects can be obtained by comparing averages against the decile level statistics. Taken together, these statistics provide a powerful tool for inferring incentive effects of policy counterfactuals and associated behavioural responses. From there, it is possible to consider associated implications for population aggregates.

#### What to do when results do not conform to expectations

If the decile level statistics do not present a rationalizable narrative, then it is often useful to focus upon the micro-data generated for individual benefit units. Suppose, for example, that you cannot understand why labour supply falls under the policy environment of interest, relative to the base simulation. In this case, you could look for units that are simulated to reduce their labour supply substantively under the new policy environment. Bearing in mind that the model is designed so that the only differences between alternative simulations are due to the considered policy environments, this approach will hopefully provide clues as to the incentives that underlie the unanticipated behavioural responses. As a last resort, staff at the NIESR can be contacted for technical support (subject to availability).

### Analysis of population aggregates

In the current example, post-simulation aggregate statistics were requested for five population cross-sections under Form 3 in Section 6.1.3, corresponding to simulation years 2017, 2027, 2047, and 2067. In contrast to the decile level statistics discussed in Section 6.4.1, the statistics reported by these post-simulation routines are calculated on data reported for all simulated adults present in each considered year. As in the case of the decile level statistics discussed above, we recommend that the user focus on population aggregate statistics that compare a policy counterfactual against the respective base simulation. Given the narrative identified from decile level statistics as discussed above, the population aggregates are usually easy to rationalise, and can provide a useful indication of the implications of reform from an economy wide perspective.

## Step 5: package results

Having identified a behavioural “story” underlying simulated results, the packaging exercise is usually fairly straight-forward. As this exercise will vary, depending upon the subject of concern, we do not discuss it further here.

# Appendix A: End User License

The computational engine upon which LINDA is based is referred to as the Simulator of Individual Dynamic Decisions, SIDD. LINDA is essentially a parameterisation and the policy environment that NIESR has implemented to permit SIDD to consider implications for the United Kingdom. The user license is for use of the SIDD engine. Model parameters assumed by LINDA are either high-level summary statistics that NIESR has calculated, and which are a matter of public record, or micro-data (i.e. the Wealth and Assets Survey) that must be accessed via third-party agreements not covered here.

**PLEASE READ CAREFULLY BEFORE INSTALLING**

**Licence Agreement**

This license agreement ("Licence") is a legal agreement between you (Licensee, or you) and the National Institute of Economic and Social Research, whose address is at 2 Dean Trench Street, Smith Square, London, SW1P 3HE, UK (Licensor, us or we) for the Simulator of Individual Dynamic Decisions Model software ("SIDD").

SIDD is a microsimulation model of tax and benefit systems, including the documentation (printed and electronic). For the avoidance of doubt, SIDD and this agreement does not include source input micro-data from surveys and administrative data sources.

We licence use of SIDD to you on the terms of this Licence Agreement. We do not sell the SIDD or accompanying documentation to you. We remain the owners of SIDD at all times.

1. Licence

1.1. In consideration of the mutual promises contained herein the Licensor hereby grants a non-exclusive, non-transferable, non-sublicensable, royalty-free, licence to the Licensee to use SIDD.

1.2. The Licensee shall not without the written permission of the Licensor make available SIDD, either as an integral part of any tool or software or independently, to any third party.

1.3. There are no implied Licences granted by this Licence

2. Acknowledgement: The Licensee shall acknowledge the Licensor in any publication or materials resulting from the use of the SIDD.

3. Fees: SIDD shall be free of charge to the Licensee subject to the terms and conditions of this Licence.

4. Intellectual Property: All intellectual property rights in SIDD will remain vested in the Licensor.

5. Warranties

5.1. SIDD is provided "as is". The Licensor shall have no obligation to support or otherwise maintain SIDD. The Licensor makes no warranties, express or implied with respect to SIDD including any warranty of merchantability, fitness for a particular purpose or non-infringement of intellectual property. The Licensor does not warrant that SIDD will be bug free, operate without interruption, or meet the Licensee's specific requirements.

5.2. All other conditions, warranties or other terms which might have effect between the Parties or be implied or incorporated into this Licence or any collateral contract, whether by statute, common law or otherwise, are hereby excluded, including the implied conditions, warranties or other terms as to satisfactory quality or the use of reasonable skill and care.

6. Termination

6.1. The Licensor reserves the right to terminate with immediate effect the Licence granted hereunder for any reason by notice in writing.

6.2. Without prejudice to any other rights or remedy it may have hereunder, the Licensor may terminate this Licence immediately upon notice in writing to the Licensee in the event that the Licensee:

6.2.1. materially breaches this Licence and fails to remedy such breach (if capable of remedy) within thirty (30) working days from the date of receipt of written notice from the Licensor specifying the breach and requiring its remedy;

6.2.2. is declared insolvent by competent regulatory authority.

7. Liability: Nothing in this Licence is intended to exclude or limit the liability of either Party in respect of:

(a) death or personal injury caused by its negligence;

(b) fraudulent misrepresentation;

(c) deliberate repudiatory breach of this Licence; or

(d) any other liability which cannot be so excluded or limited under applicable law.

8. General

8.1. This Licence contains the entire understanding between the Parties in relation to the subject matter of this Licence and supersedes all other agreements, negotiations, representations and undertakings, whether written or oral of prior date between the Parties.

8.2. If any provision of this Licence is held to be void or unenforceable by applicable law, it shall be severed therefrom and the remaining provisions of the Licence shall, to the extent possible, remain valid and fully enforceable.

8.3. Nothing in this Licence shall be construed as creating a joint venture, partnership, contract of employment or relationship of principal and agent between the Parties.

8.4. No third party shall have any right to enforce any term of this Licence.

# Appendix B: The Utility Function

This has two components to it. Within-period utility *u* is a function of total benefit unit consumption *ci,t* adjusted for effective benefit unit size *i,t* and leisure time represented by *li,t*.

 represents the consumption-equivalent of leisure and ** the elasticity of substitution between consumption and leisure.

 ()

Within-period utility enters into an intertemporal utility function in the manner represented below. Intertemporal discounting takes a quasi-hyperbolic form, where ** is the long-run discount factor, and ** is the excess short-run discount factor. When ** = 1, preferences are time consistent, which implies that – for any given set of circumstances – the same decisions will maximise expected lifetime utility, regardless of when the decisions are made. That is, if an individual could commit to savings and employment decisions that take their evolving circumstances into account for any future age, then they will make the same decisions regardless of their current age. With 0 < ** < 1, intertemporal preferences exhibit myopia, which means that people would like to be more patient in the future than will actually be the case. The model assumes that people are ‘sophisticatedly’ myopic, in the sense that they are aware of their own self-control problems and react to them. This can result, for example, in a preference to lock savings away in a pension rather than a bank account, to avoid the temptation of spending the savings prematurely.

** is relative risk aversion, and *j-t,t* is the probability of surviving *j* years, given survival to age *t. a* and *b* represent the warm glow utility derived from leaving a positive bequest *w+i,t+1*.

()

1. LINDA does not provide empirically derived probabilities associated with any projected outcome. [↑](#footnote-ref-2)
2. See the UK Data Service, Catalogue number 7215; “Wealth and Assets Survey, Waves 1-5, 2006-2016”. [↑](#footnote-ref-3)
3. The model is delivered on the assumption that you have a 64-bit operating system. Please contact the NIESR if you require files to run on a 32-bit system. Note also that these minimum system requirements are for the base model specification, and are not sufficient for every feasible model specification. Please contact NIESR for further assistance. [↑](#footnote-ref-4)
4. Taxable incomes are defined in the model as the sum of benefit unit labour income, private pension income, state pension income, and investment income (restricted to be non-negative), less concessional private pension contributions. [↑](#footnote-ref-5)
5. The coarse and medium model specifications both use linear interpolation and Brent’s method for solving in one dimension or Powell’s method for solving in multiple dimensions. These two specifications differ only in the number of “grid points” that are used to approximate the state space. The fine model specification uses cubic interpolation, and then cycles between Brent’s/Powell’s method and a Simplex search routine. The fine model specification also divides the state space into appreciably more points than the coarse and medium model specifications. [↑](#footnote-ref-6)
6. Donors of adults in data for reference cross-section refer to imputed relationships for receipt of inheritances. [↑](#footnote-ref-7)
7. There may also be some sporadic departures from equation (1) in backward simulations due to imperfections in the search routines used to ensure incentive compatibility. [↑](#footnote-ref-8)
8. 2 candidates are taken when the random draw used to randomly select a candidate is very close to the boundary used to distinguish the two candidates assumed for imputation. This is done to avoid anomalies that might otherwise arise due to machine error of the programming code. [↑](#footnote-ref-9)