

User guide – modelling Basic Income in UKMOD

Variables and constants to set up

- i) Various variables for the BI system lines are defined in 1.18 (in the comments it is defined which variable is necessary for each BI system)
 - a. The two loop variables `i_BI` and `i_t` are intermediate variables but their initial values need to be set in 1.18.1 and 1.18.2
 - b. `buc_s`, `pens_s`, `taxinc_s`, `threshhigh_s` and `netBI_s` are simulated variables that will appear in the output (for all versions)
 - c. the three `dpost` variables are used in the break condition for the loop systems and also appear in the output in all versions.

NB all variables not yet set are put to 0, rather than n/a, because they are used in all systems later and so need to be defined, even if not set initially.

- ii) Various constants needed are defined in 3.23 (again noted for which particular BI systems)
 - a. Variables setting net levels of working-age BI, pensioner's BI, the proportion of working-age BI that children get, and the level of tax increases, for the systems in which they are set initially and remain constant
 - b. For the systems where an increment for each loop is needed, the size of each increment and the threshold of revenue neutrality to stop the loop (see below under loops).

NB some constants are set to n/a when not needed in particular versions because they are not used at all in that system.

Interaction with other benefits

- iii) `buc_s` is a simulated non-means tested benefit so we have added it to:
 - `lls_bennt` (line 6.6.29) the list of non-means-tested benefits
 - `lls_bensim` (line 6.16.26) the list of simulated benefits.
- iv) Because we will give all children a BI, we have made changes to Child Benefit:
 - a. We have removed the existing level of child benefit and set `$CBFirst` line 3.6.1 and `$CBOther` line 3.6.2 both to zero.
- v) `buc_s` needs to be added to the list of any mean-tested benefits for which we want it to enter their means test. By default we have added `buc_s` to the means-test for:
 - Income Support (`il_IS_means`) line 8.3.38
 - Guarantee Credit (`il_GC_means`) line 8.4.36
 - HB means applying if benefit unit receives SC but not GC (`il_HBCTB_meansSC`) line 8.11.39
 - HB and CTB means applying if benefit unit does not receive PC and all aged<60 (`il_HBCTB_meansIS`) line 8.12.41
 - HB and CTB means applying if benefit unit does not receive PC and one aged>=60 (`il_HBCTB_meansGC`) line 8.13.38
 - Tax credits (`il_TC_means`) 8.16.23
 - Universal Credits as unearned income (`il_UC_means_unearned`) 8.27.19.

If you do not want BI to enter these means tests, buc_s can be removed from any of these by changing + to n/a.

Interaction with (personal income) tax system

- vi) Do you still want a personal allowance for income tax – and if so how much? Go to line 3.4.1 to change it and also alter the income limit for having a personal allowances at 3.4.10 accordingly
By default we have set the personal allowance to 0 at line 3.4.1 and abolished its income limit by raising the limit to an impossibly high level at line 3.4.10.
- vii) Do you want to alter income tax bands and other allowances of the tax system? This involves altering other constants under 3.4.
By default, we have left all these unchanged.
- viii) Do you want to alter the NI system? Go to 3.5 to alter:
 - a. primary threshold (\$Nic1PT line 3.5.1), small earnings exception (\$Nic2Thresh line 3.5.4) and Class4 Lower profits limit (\$Nic4LPL line 3.5.5), all of which we have set at 0 by default
 - b. Upper Earnings Limit (\$Nic1 UEL line 3.5.2) and Upper Profits Limit (\$Nic4 UPL line 3.5.6)
both of which we have abolished by default (by setting their thresholds at impossibly high levels)
 - c. You could also alter the rates at which Nics are payable under 26. By default, we have left all these unchanged.
- ix) Do you want your BI to be taxable?
If so it needs to enter the tax base il_tinty at line 8.18.23 and il_tinty_NSND at 8.22.22
– we have decided yes by default so have added buc_s to both.
- x) Increment in tax rates (policy 33)
To allow for an increment to tax rates, a simple increment of i_t is added to all tax rates in 33.14 S&D income tax schedule with additions, 33.24 Income tax schedule since 2008/09 with possible additions and 33.27 Scottish income tax schedule since 2016/17 with possible additions (all systems)
The size of this increment is set in different ways for different systems in policy 29
BEN:Basic Income

Setting up Basic Income (Policy 29 BEN: Basic income)

- xi) 29.1 sets i_t as a variable equal to the fixed increase in tax rates (for the BI loop and no loop, for tax loop that is set in 1.18 as starting value for the loop)
- xii) 29.2 gives the tax increase a name (taxinc_s) that will appear in the output as a simulated variable for all BI versions
- xiii) 29.3 calculates the gross BI payment needed to give the fixed net level of BI (for the tax loop and no loop, for BI loop that is set in 1.18 as starting value for the loop)

- xiv) 29.4 allows for the possibility that the gross BI payment could exceed the higher rate threshold and raises the threshold to the level of the gross BI payment in that case, so that only basic tax is payable on any BI payment (all systems but only needed for tax loop)
- xv) 29.5 records what the higher rate threshold now is
- xvi) 29.6 calculates and records working-age net BI
- xvii) 29.7 allocates adult's and child's gross BI
- xviii) 29.8 calculates the pensioners gross BI needed to give them the proposed level of pensioner's net BI
- xix) 29.9 gives all over 65 the gross payment needed to top up any existing state pension to pensioner's gross BI.

Loops for fiscal neutrality

- xx) The new policy 46 fiscaltotals_uk calculates benefit spending and tax and sic revenues and compares the fiscal balance to that of the baseline system (in this case UK_2020pre covid19). It also produces output with prefix "dpost", that is useful for checking convergence of the loop based systems (see later).
- xxi) The new policy 47 loopend_uk runs the BI and tax loops; it is only needed for systems with such loops. It:
 - a. Sets up an increment for each loop of size set in the constants at 1.18 for BI loop in 47.1 and tax loop at 47.2. These should each be set to n/a except for systems needing that particular loop
 - b. Then 47.3 defines the start, end and stopping condition for the loop (both loops) with
 - i) 47.3.1 starts loop at policy 33 buc_uk where the basic income policy is set
 - ii) 47.3.2 ends it at this policy itself 47 loopend_uk
 - iii) 47.3.3 makes the number of iterations appear in the output as loopend_revneutral
 - iv) 4.7.4 makes the break condition for the loop be that dpost_fiscalratio is close to (within threshold% of 1).
- xxii) In Output_std_uk the number of decimal points in which output is given was changed from 2 to 3 at 48.1.32 (necessary to get correct picture of tax rate changes in tax loop versions).

Getting convergence with loops

- xxiii) Getting convergence can be a lengthy process in UKMOD but there are ways to speed this up. The way to get convergence is:
 - a. To guess an initially starting point and set it under 1.18.1 or 1.18.2
 - b. To set a relatively large threshold under 1.19.7
 - c. Run the system and provided it seems to have converged then use the outcome of that run as starting point of new run with a smaller threshold.
- xxiv) The only way to be sure if a run has converged is to check in the output file that loopcount_revneutral (ie the number of runs) is less than 20, but one can often guess

from the Stats Presenter output, by running it as a reform scenario with Baseline uk_2020preCovid19. You can check if there has been convergence by looking at the *Difference to base* column the numbers for *Government revenue through taxes and National Insurance contributions* and for *Government expenditure on benefits and tax credits* are close to each other.

xxv) If no convergence you can try the following:

- i. If the increase in revenue is higher than the increase in expenditure,
 1. In a BI loop, BI needs to be higher, so use a starting value for *i_BI* greater than or equal to *buc_s* in the final run last time (NB output of *buc_s* is per month so multiply by 12)
 2. In a tax loop, the tax rate needs to be lower, so use a starting value for *i_t* lower than that of the last run
- ii. If the increase in expenditure is higher than the increase in revenue,
 1. In a BI loop, BI needs to be lower, so use a starting value for *i_BI* lower than that of the last run
 2. In a tax loop, the tax rate needs to be higher, so use a starting value for *i_t* greater than or equal to *taxinc_s* in final run last time

xxvi) Another way to achieve convergence is to experiment to get a good starting point by switching off 47 *loopend_uk* and trying some starting values in 1.18. Then looking at *dpost_fiscalratio* in the output is useful, since whether it is greater or less than 1 can explain whether higher or lower levels of BI/taxation are needed (depending on loop).

User guide written by Jerome De Henau, Sue Himmelweit and Sara Reis.

Latest version May 2021

For more information see <https://www.microsimulation.ac.uk/>