An API For Microsimulation Models

Graham Stark

Social Work and Social Policy
University of Northumbria
Newcastle, UK
graham.stark@northumbria.ac.uk

Abstract—This note describes a simple general Application Programming Interface (API) for controlling microsimulation models.

Index Terms—Microsimulation, APIs

I. Introduction

Scotben [1] is a conventionally structured static microsimulation tax-benefit model, in the family of models branching out from the Institute for Fiscal Studies' TAXBEN, of which Graham Stark was one of the principal authors [2]. Scotben has been used in several projects at the University of Northumbria and elsewhere. With the exception of some survey data, Scotben is fully Open Source[3], and released under a permissive licence.

This note is intended as a 'warts and all' summary of some key aspects the models development, structure, novel features, uses to date, and strengths and weaknesses.

As with all models of this sort, ScotBen is continually being updated and developed, and so this note may not always be in sync with the latest version.

II. Scope

ScotBen's primary scope is the the Scottish fiscal system. The model covers taxes and benefits that individuals are directly liable for, such as income and spending taxes and cash benefits, though it also has some limited ability to capture the effects on individuals of e.g. Corporate Taxes and Green Levies. Both devolved taxes and benefits and those reserved to the UK government are included. Many hypothetical structural reforms can be modelled without requiring modifications to the code, including basic incomes, wealth taxes and various local taxation schemes. See the Section VII.A appendix for a full list of the taxes and benefits included.

III. Design and Implementation

At its heart, Scotben is a single period, static microsimulation model. It loosely follows the design of IFS's TAXBEN2, though ScotBen is programmed in a different language (Julia vs TAXBEN's Pascal/Delphi) and no TAXBEN code was referred to during development.

The model attempts to follow modern program development practices, with short, independently testable, functions, a comprehensive test suite[1], and readable code. In places,

---

[1]see: Section III.C below

however, the scale of the project means that these ideals are honoured in the breach.

Despite its fundamentally static nature, Scotben's modular design and clean interfaces make it easy to build simulations incorporating behavioural responses or long-term projections on top of the base model.

A. Julia

The model is written in the Julia programming language. As discussed in [4], Julia is intended to solve the "Two Language Problem": to be equally useful for conventional large-scale programming and for exploratory data science. Julia produces highly efficient code, comparable in speed with Fortran or C. Although not without its problems, this dual nature means that Julia has proven a good choice for microsimulation.

The model is organised as a Julia package - a bundle of code and data that can be automatically downloaded and run using standard Julia tools. All code and most data is stored and developed on the GitHub code sharing site [1].

Internally, the model is organised as a series of modules, arranged to minimise cross-dependencies. For example, there are modules that:

- capture the fiscal system parameters
- calculate income tax

Many other such modules are referred to below.

B. Data

ScotBen uses Family Resources Survey (FRS) [5] as its primary data. For Scottish runs, pooled Scottish FRS subsets from 2015-2022 are used (just over 17,000 households); UK-wide simulations presently use a single full FRS year. The HouseholdMappingFRS.jl package creates the main model dataset, and ModelHousehold.jl encapsulates the model's view of a household.

The model has a built in weighting system [6] using a native Julia implementation of standard survey data weighting algorithms[7]. As well as standard demographics, this allows us to weight to Scottish employment totals, disability benefit receipts, local authority-level populations and occupations. We can also use the inbuilt weighting system to use ScotBen to model individual Local Authorities.

Data is then uprated using a combination of ONS price and earnings data, supplemented by some OBR and SFC forecast data. Uprating is normally to 1 quarter behind the current quarter.

The FRS has very limited information on wealth and assets, and no information on consumption. It also has no

Local Authority identifiers. To allow modelling of (e.g.) a Wealth tax or indirect taxes such as Fuel Duty or VAT, the FRS data is supplemented by the Wealth and Assets Survey (WAS)[8], Living Costs and Food Survey (LCF)[9] and Scottish Household Survey (SHS) [10]. These datasets can either be matched in[11] - picking WAS/LCF/SHS records with similar characteristics to each FRS record, or imputed using linear regression (SHS is always data-matched).

C. Testing

ScotBen is developed Test First [12]. A large suite of Individual, Benefit Unit, and Household level example calculations for each tax and benefit was collected and expressed as unit-tests. Code was then written so all the tests pass. Sources used included official and semi-official online calculators[13], taxation textbooks[14], benefit manuals[15], and our own calculations. The test suite also contains numerous aggregate tests which check that complete model runs produce plausible aggregate values for revenues, expenditures and caseloads.

The testsuite currently contains over 1,000 individual tests and 10,000 lines of code. In addition there are over 150 runtime consistency assertions in the main body of code - these will halt a simulation if any abnormal condition is encountered.

The suite is used as a 'continuous integration' tool [16]: part or all of the tests are run before every change to the model is committed. This helps prevent errors being introduced during development.

IV. Applications And Novel features

Scotben's clean, modular, design makes it easy to incorporate the code into specialised applications. Examples to date are:

A. Budget Constraints

This uses the a simple algorithm[17], [18] to draw exact budget constraints - the relationship between net income and earnings for individuals with differing earnings capacity and family circumstances. Often this relationship is startlingly non-linear. These budget constraints are the best foundation for dynamic labour supply models.

An online Budget Constraint Generator is available at: https://stb.virtual-worlds.scot/bcd/

B. Local Taxation

The weighting system discussed above can be used to weight the FRS dataset so that the sample grosses up to the population of a local authority. Data is weighted to 2022 Census data on occupation, tenure, accommodation type, employment status, age, sex and household size. See the module LocalWeightGeneration.jl

C. The Public Preference Calculator (TriplePC)

The Public Policy Preference Calculator (TriplePC)[19] is an adaption of the model that extends the microsimulation art in two ways:

First, as well as modelling the outcomes of a policy in the conventional way, our model uses Conjoint Analysis of public acceptability data to give an indication of the policy's popularity. This is novel and important. There are measures that might actually be popular with the electorate, but which policymakers have been unwilling to touch because of uncertainty about their electoral consequences[20].

Second, we integrate health outcomes into the model[21]. We built a model relating SF-12 scores to income and demographic characteristics; SF-12 is a widely used measure of an individual's health-related quality of life, with two summary scores: the Physical Component Summary (PCS-12) and the Mental Component Summary (MCS-12). The health model is estimated over 12 waves (2009/11-2020/22) of Understanding Society [22] panel data and implemented in the HealthRegressions.jl module, which maps the regression coefficients to the ScotBen dataset.

This version uses the UK-wide FRS sample. An online version is available at: https://triplepc.northumbria.ac.uk.

D. Legal Aid

Scotben has recently been used to build a model of Legal Aid entitlement and costs for the Scottish Legal Aid Board (SLAB).

The module LegalAidCalculations.jl encapsulates the means-test rules. Entitlements are estimated by mapping SLAB provided administrative data - the LegalAidData.jl module is used for this. The model is actively being used to design reforms to the Legal Aid means-tests.

V. Interfaces

By design, the ScotBen model package has no user interface code. This actually makes it **easier** to build user interfaces, since there are no clashing dependencies.

Several model Web interfaces have been built (not all may be active).

- https://scotben.virtual-worlds.scot/ - A simple budget simulator, implemented using the Genie web package;
- https://ubi.virtual-worlds.scot/ - A basic income sim, implemented with Dash
- https://stb.virtual-worlds.scot/bcd/ - The Budget Constraint generator discussed above, also implemented with Dash;
- https://triplepc.northumbria.ac.uk - TriplePC, also implemented with Gini.

The Legal Aid model also has a web interface, though this is not currently public.

In addition, the model can be integrated into Pluto and Jupyter notebooks - work on this is ongoing with a view to using ScotBen in teaching.

ScotBen can also be run from conventional command-line scripts with a few lines of code - there are multiple examples of this in the test suite discussed above.

VI. TO-dos

ScotBen is under active development. As of January 2025, tasks include:

- *Synthetic Datasets* since the main datasets cannot be included with the open-source distribution, work is

underway om creating synthetic datasets with the same properties;

- *Local Modelling* - the local re-weighting scheme described above is still being refined;
- *Code cleanups and reorganisation* - code 'TODO's that are being worked on include clearer and more consistent names fpr variables and functions, removing duplicated and unused code, and and simplifications of some modules, notably the (https://github.com/grahamstark/ScottishTaxBenefitModel.jl/blob/master/src/STBIncomes.jl)[income handling module].
- *Automatic updating* presently, updating model parameters, inflation rates and grossing up targets takes at least one working week, but much of the needed material is available via public APIs, so the intention is to use those APIs to largely automate the updating process;
- *verificaton against other microsimulation models* - work is now underway to verify ScotBan's individual-level calculations against those of Policy Engine[23];
- *benefit takeup and tax avoidance/evasion* - work is beginning on creating takeup correction routines[24] and revenue responsiveness to taxes on income[25].

VII. More Information

- A blog about the model
- Video presentation from the Juliacon 2022 conference;
- Powerpoint Presentation from the 2024 International Microsimulation (IMA) Conference
- Video from the 2022 IMA conference

Last updated: 2025-10-20.

A. Appendix: Modelled Taxes And Benefits

TABLE I: MODELLED TAXES

Tax	Code Module	Notes
Income Tax	`IncomeTaxCalculations.jl`	*Scottish and reserved UK*
National Insurance	`NationalInsuranceCalculations.jl`	*Employees, Self Employed and Employers (though this needs more thought on incidence)*
Council Tax	`LocalLevelCalculations.jl`	*plus some simple modelling of local income taxes and domestic rates*
Wealth Taxes	`OtherTaxes.jl`	*using WAS data*
VAT and excise duties	`IndirectTaxes.jl`	*using LCF data; incomplete module*
incidence of essentially any tax incident on wages	`OtherTaxes.jl`	

a) Modelled benefits:

TABLE II: MODELLED NON MEANS-TESTED BENEFITS

Benefit	Code Module	rUK equivalent
Pension Age Disability Payment	`NonMeansTestedBenefits.jl`	*Attendance Allowance*
Child Benefit	"	
Adult Disability Payment	"	*Disability Living Allowance (DLA)*
Carer Support Payment	"	*Carer's Benefit*
Adult Disability Payment	"	*Personal Independence Payment (PIP)*
Contributory Employment Support Allowance	"	*reserved benefit*
Contributory Job Seeker's Allowance	"	""
Old/New State Pensions	"	""
Bereavement Support Payment (and predecessors)	"	""
Maternity Allowance	"	""
Statutory Maternity Pay	"	""

note: The Scottish disability benefits are modelled as being equivalent to the corresponding rUK benefits, though a mechanism exists to make the disability tests more or less generous.

TABLE III: MODELLED MEANS-TESTED BENEFITS

Benefit	Code Module	Notes
Scottish Child Payment	`ScottishBenefits.jl`	*Scottish-specific; passported*
Universal Credit	`UniversalCredit.jl`	*UK reserved, though ScotGov aspire to remove the 2-child limit*
Savings Credit/ Pension Credit	`LegacyMeansTestedBenefits.jl`	
Council Tax Reductions	`LegacyMeansTestedBenefits.jl`	
Housing Benefit	""	*Being phased out for working-age families; see UCTransition.jl*
Working Tax Credit	""	""
Child Tax Credit	""	""
Income Support	""	""
Non-Contributory Employment Support Allowance	""	""
Non-Contributory Job Seeker's Allowance	""	""

TABLE IV: OTHERS

Benefit	Code Module	Notes
Minimum Wages	`HouseholdAdjuster.jl`	
Scottish Civil Legal Aid	`LegalAidCalculations.jl`	
Basic Incomes	`UBI.jl`	
Wealth Taxes	`OtherTaxes.jl`	
Various Local Taxation schemes	`LocalLevelCalculations.jl`	
The Benefit Cap	`BenefitCap.jl`	
Bedroom Tax/ Discretionary Housing Allowance	`ScottishBenefits.jl`	*Scottish-specific*

b) Not currently modelled:
- Any form of Student Support;
- Student loans and repayments (working on repayments ATM)
- Food banks or similar;
- Foster Care payments
- Scottish Best Start payments
- Child Winter Heating Payment
- Winter Heating Payment
- Funeral Support Payment
- Job Start Payment
- any local authority-specific payments
- Young Carer Grant

References

[1] G. Stark, "Scottish Tax Benefit Model." Accessed: Jul. 19, 2024. [Online]. Available: https://github.com/grahamstark/ScottishTaxBenefitModel.jl

[2] P. Johnson, S. Webb, and G. Stark, "TAXBEN2: The New IFS tax and Benefit Model," May 1990.

[3] O. S. Initiative, "The Open Source Definition | Open Source Initiative." Accessed: Nov. 21, 2012. [Online]. Available: http://opensource.org/osd

[4] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017, doi: 10.1137/141000671.

[5] DWP, *The Family Resources Survey*. 2012. Accessed: Nov. 16, 2012. [Online]. Available: http://research.dwp.gov.uk/asd/frs/

[6] J. Creedy, "Survey Reweighting for Tax Microsimulation Modelling," Sep. 2003. [Online]. Available: http://ideas.repec.org/p/nzt/nztwps/03-17.html

[7] G. Stark, "grahamstark/SurveyDataWeighting.jl." Accessed: May 31, 2024. [Online]. Available: https://github.com/grahamstark/SurveyData Weighting.jl

[8] O. f. N. Statistics, *Wealth and Assets Survey*. UK Data Service, 2019. doi: 10.5255/UKDA-Series-2000056.

[9] O. f. N. Statistics, *Living Costs and Food Survey, 2017-2018*. UK Data Service, 2019.

[10] S. A. H. Scottish Government, "Scottish Household Survey - Publication Summary - Methodology and Fieldwork Outcomes." Accessed: Dec. 13, 2018. [Online]. Available: http://www2.gov.scot/Topics/Statistics/16002/PublicationMethodology

[11] A. LEULESCU and M. AGAFIȚEI, "Statistical matching: a model based approachfor data integration." [Online]. Available: https://ec.europa.eu/eurostat/documents/3888793/5855821/KS-RA-13-020-EN.PDF/477dd541-92ee-4259-95d4-1c42fcf2ef34?version=1.0

[12] V. One, "Test-Driven Development: Test-First Agile Programming." Accessed: Feb. 06, 2018. [Online]. Available: https://www.versionone.com/agile-101/agile-software-programming-best-practices/test-first-programming/

[13] P. I. Practice, "Better Off Calculator from Policy in Practice." Accessed: May 31, 2024. [Online]. Available: https://policyinpractice.co.uk/software/

[14] A. Melville, "Taxation Finance Act 2023." Accessed: May 31, 2024. [Online]. Available: https://www.pearson.com/en-gb/subject-catalog/p/taxation-finance-act-2023/P200000011257/9781292729275

[15] CPAG, "Welfare Benefits and Tax Credits Handbook 2017/18." Accessed: Feb. 06, 2018. [Online]. Available: https://www.shop.cpag.org.uk/welfare-benefits-and-tax-credits-handbook-2017-2018

[16] I. Google, "Introducing "Testing on the Toilet"." Accessed: Nov. 14, 2021. [Online]. Available: https://testing.googleblog.com/2007/01/introducing-testing-on-toilet.html

[17] A. Duncan and G. Stark, "A recursive algorithm to generate piecewise linear budget constraints," May 2000, doi: 10.1920/wp.ifs.2000.0011.

[18] G. Stark, "grahamstark/BudgetConstraints.jl." Accessed: May 31, 2024. [Online]. Available: https://github.com/grahamstark/BudgetConstraints.jl

[19] G. Stark, E. Johnson, M. Johnson, D. Nettle, and H. Reed, "The Public Policy Preference Calculator (TriplePC): Developing a comprehensive welfare policy microsimulation.," *International Journal of Microsimulation.*, 2024, [Online]. Available: https://doi.org/10.17605/OSF.IO/SKPYB

[20] D. Nettle, M. T. Johnson, J. Chrisp, and E. Johnson, "SP2.1 Conjoint experiment on welfare scheme preferences and predictive tool," 2023, doi: 10.17605/OSF.IO/HTSQC.

[21] H. Reed *et al.*, "Examining the relationship between income and health among adults in the UK Preprint 2024.04.03.pdf," Apr. 2024, Accessed: May 29, 2024. [Online]. Available: https://osf.io/https://osf.io/ngtxs

[22] I. for Social and Economic Research, "Understanding Society – The UK Household Longitudinal Study." Accessed: Feb. 08, 2018. [Online]. Available: https://www.understandingsociety.ac.uk/

[23] N. Woodruff and M. Ghenis, "PolicyEngine UK." Accessed: May 29, 2024. [Online]. Available: https://policyengine.org/uk/

[24] V. Fry and G. Stark, *The take-up of means-tested benefits, 1984-90*. 1993. doi: 10.1920/re.ifs.1993.0041.

[25] S. F. Commission, "How we forecast behavioural responses to income tax policy March 2018." Accessed: Mar. 11, 2018. [Online]. Available: http://www.fiscalcommission.scot/publications/occasional-papers/how-we-forecast-behavioural-responses-to-income-tax-policy-march-2018/