neworder: a dynamic microsimulation framework for Python

Andrew P Smith[1]

1 School of Law, University of Leeds, UK

Summary

Traditional microsimulation frameworks typically use a proprietary modelling language, often place restrictions on data formats, and vary in terms of efficiency or scalability. *neworder* provides an efficient, flexible, and scalable framework for implementing microsimulation models using standard Python code. Being a framework, it has been designed with reusability and extensibility as primary motivations.

It is predominantly implemented in C++ for maximal performance and supports both serial and parallel execution. Particular attention has been paid to the provision of powerful and flexible random number generation and timestepping functionality.

The package is extensively documented, including numerous detailed examples that showcase the functionality across a diverse range of applications including demography, finance, physics, and ecology.

It is available through the standard Python repositories (PyPI, conda-forge) and also as a Docker image.

Statement of Need

The *neworder* framework is designed to be as unrestrictive and flexible as possible, whilst still providing a solid foundation on which to implement a microsimulation or individual-based model. Being agnostic to data formats means that models can be easily integrated with other models and/or into workflows with rigid input and output data requirements.

It supports both serial and parallel execution modes, with the latter using MPI to distribute computations for large populations or to perform sensitivity or convergence analyses. *neworder* runs as happily on a desktop PC as it does on a HPC cluster.

neworder was inspired by MODGEN (Government of Canada, 2009c) and, to a lesser extent, the Python-based LIAM2 (*Liam2 Website*, n.d.) tool, and can be thought of as a powerful best-of-both-worlds hybrid of MODGEN and LIAM2.

Both MODGEN and LIAM2 require their models to be specified in proprietary languages (based on C++ and YAML, respectively), whereas our framework eliminates the extra learning curve as users simply define their models in standard Python code.

Whilst MODGEN supports parallel execution, LIAM2 does not. MODGEN is very restrictive with input data (which must be defined within the model code) and output data (which is a SQL database). *neworder* supports parallel execution, thus having the scalability of MODGEN, but without any restrictions on data sources or formats.

Both MODGEN and LIAM2 require manual installation and configuration of an environment in order to develop models; *neworder* and its dependencies can simply be installed with a single command.

The framework is comprehensively documented (Smith, 2021) and specifically provides detailed examples that are translations of MODGEN models from Bélanger & Sabourin (2017) and Statistics Canada Government of Canada (2009b), demonstrating how *neworder* implementations can be both simpler and more performant (see the Mortality example in the documentation).

Part of the design ethos is not to reinvent the wheel and to leverage the huge range of statistical functions in packages like *numpy* and *scipy*. However, functions are provided where there is a useful niche function or a major efficiency gain to be had. An example of the former are methods provided to sample extremely efficiently from non-homogeneous Poisson processes using the Lewis-Shedler algorithm (Lewis & Shedler, 1979), and the ability to perform Markov transitions *in situ* in a pandas dataframe, both of which result in at least a factor-of-ten performance gain.
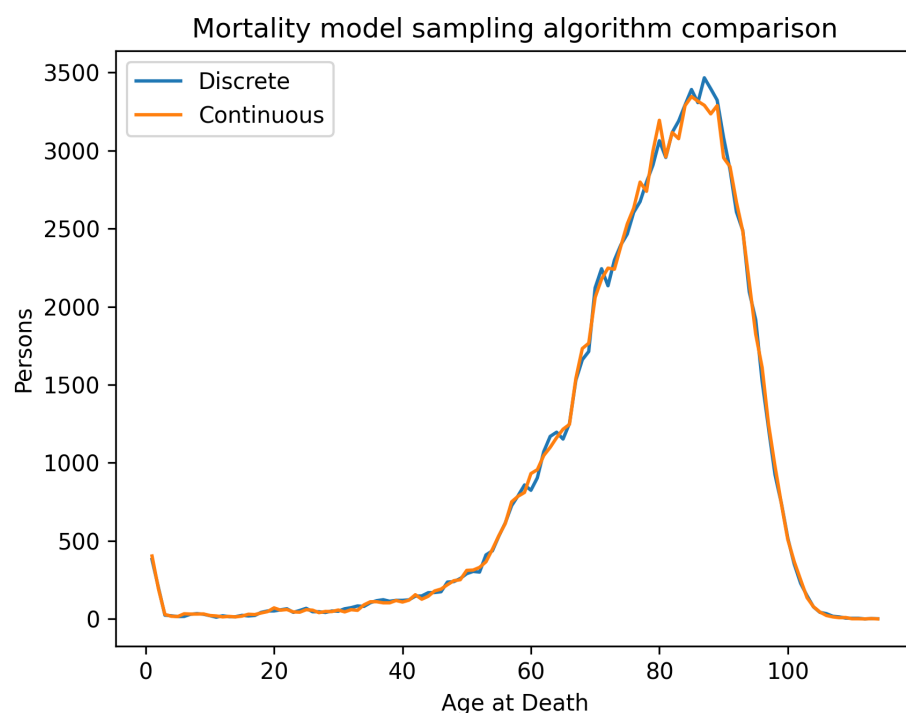

Figure 1: Sampling mortality: "Discrete" samples repeatedly at 1 year intervals, "Continuous" uses the Lewis-Shedler algorithm to sample the entire curve, with a tenfold performance improvement.

Another important consideration in *neworder*'s design is reproducibility, especially with regard to random number generators. Inbuilt extensible seeding strategies allow for fully deterministic execution and control over whether parallel processes should be correlated or uncorrelated, and users can implement their own custom strategies as necessary.

neworder is currently being used for a project developing an integrated supply-demand model of police and crime (M-O-P-D/Police-Supply-Demand, 2021): a microsimulation of crime at high spatial, temporal and categorical resolution drives an agent-based model of police resourcing (implemented in netlogo), which in turn can dynamically alter the microsimulation parameters according to how well it responds to the events generated.

Acknowledgements

References

Bélanger, A., & Sabourin, P. (2017). *Microsimulation and Population Dynamics: An Introduction to Modgen 12.* Springer International Publishing. https://doi.org/10.1007/978-3-319-44663-9

Government of Canada, S. C. (2009a). *General characteristics of Modgen applications–exploring the model RiskPaths.* https://www.statcan.gc.ca/eng/microsimulation/modgen/new/chap3/chap3

Government of Canada, S. C. (2009b). *Modgen and the application RiskPaths from the model developer's view.* https://www.statcan.gc.ca/eng/microsimulation/modgen/new/chap4/chap4

Government of Canada, S. C. (2009c). *Modgen (Model generator).* https://www.statcan.gc.ca/eng/microsimulation/modgen/modgen

Lewis, P. a. W., & Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly, 26*(3), 403–413. https://doi.org/10.1002/nav.3800260304

LIAM2 website. (n.d.). Retrieved May 8, 2021, from https://liam2.plan.be/pages/about.html

M-O-P-D/Police-Supply-Demand. (2021). M.O.P.D. https://github.com/M-O-P-D/Police-Supply-Demand

Smith, A. P. (2021). Neworder. In *neworder documentation.* https://neworder.readthedocs.io/en/stable/