An open source implementation of the Earth4All integrated assessment model

Pierluigi Crescenzi, Aurora Rossi, Emanuele Natale

An open source implementation of the Earth4All integrated assessment model

Pierluigi Crescenzi[1,†], **Aurora Rossi**[2,†,*], **and Emanuele Natale**[3]

[1]Gran Sasso Science Institute, L'Aquila, 67100, Italy
[2]I3S & INRIA d'Université Côte d'Azur, COATI Team, Sophia Antipolis, 06902, France
[*]corresponding author(s): Aurora Rossi (aurora.rossi@inria.fr)
[†]these authors contributed equally to this work

ABSTRACT

Integrated assessment models are critical tools for scientists to understand the complex interactions between human and natural systems, and for policymakers to design effective policies to tackle global societal and environmental challenges. However, the lack of transparency and reproducibility of the models hinders their use and trust in their results. In this paper, we present a new open-source implementation of the recent Earth4All integrated assessment model in the Julia programming language, which has been developed according to best coding practices using a modular modeling framework (WorldDynamics.jl), and is publicly available in a GitHub repository for community inspection and use under an open source license. We describe the model structure, data and parameters, and present a set of experiments to validate the new implementation and to show its capabilities. By providing an open-source, modular, well-documented implementation of the Earth4All model on a widely-used software development platform, we hope to foster the use of the model by the scientific community and facilitate the development of new integrated assessment models.

Background & Summary

Integrated assessment models (IAMs) are mathematical models that combine knowledge from different scientific disciplines to study the complex interactions between human and natural systems, and to support decision-making in the face of global societal and environmental challenges[1]. Computer models force humans to specify implicit assumptions in their mental models, and allow us to derive the precise consequences of models without being affected by the biases of the human mind; moreover, given two alternative models, they make their different assumptions explicit and allow objective comparison of their predictions[2]. IAMs of particular historical significance have been developed by the Club of Rome think tank, such as the World3 model[3] and subsequent revisions and more targeted models[4–7]. Recently, the Club of Rome released the Earth4All model[8], which provides a comprehensive and up-to-date representation of the global system, and supports the development of policies to tackle the Sustainable Development Goals established by the United Nations[9]. However, the usefulness of IAMs is often severely limited by technical issues, such as the accessibility of the code implementation, the lack of software documentation, and the fact that modern software development practices such as the use of a public development platform are not exploited[10]. We address these issues by providing a new implementation of the Earth4All model, which formerly relied on proprietary software Vensim[11] and Stella[12], using the scientific-computing oriented Julia programming language[13]. Per the requirement of modularity which is widely recommended for the scientific development of IAMs[10,14], we leverage the Julia ecosystem (in particular the ModelingToolkit.jl library[15]) to provide a modular implementation of the model in its main sectors. By isolating aspects of the model that are the domain of different scientific disciplines, the 12 modules (climate, demand, energy, finance, food-and-land, inventory, labor-market, output, population, public, well-being and other-performance-indicators) facilitate the specific development of different aspects of the model by different user communities, who can take advantage of the GitHub development platform to develop, review and publicly discuss the model. According to best modeling recommendations, we also provide detailed documentation which allows the user to reproduce the Earth4All model independently from ours and previous codebases[16]. Our

implementation is available on the GitHub software development platform under a free software license, allowing the scientific community to publicly report issues, openly discuss aspects of the model and possibly contribute to its further development.

Methods

Software Engineering Approaches to IAMs

The development of IAMs is a highly interdisciplinary endeavor that requires the close collaboration of scientists from different disciplines, such as economics, ecology, climatology, and computer science, among others. However, interaction among such different scientific communities has often been hindered by the fact that large IAM often appears as a monolithic black box whose understanding (let alone development) requires a discouraging effort. The nature of such a problem has classically been investigated within the field of software engineering itself, for the development of large, complex and long-lived software systems. Solutions include the use of a modular approach to software development and the use of software development platforms.

A modular approach allows to split the system into smaller, more manageable subsystems, and the development and study of each subsystem in a relatively isolated fashion from the others[17,18]. Modularization has rigorously been adopted in the development of early IAMs such as World3[2], it has been recommended by the National Academy of Science report on the Social Cost of Carbon Dioxide[14], and it has been a main motivation behind the development of the IAM framework Mimi.jl for the estimation of the social cost of carbon dioxide emissions[10,19]. While we preserve the division of the Earth4All model in sectors as provided in the Vensim implementation of the original authors, our new implementation leverages the graph-based composable functionality of the Julia ModelingToolkit.jl library[15]. ModelingToolkit.jl internally represents system variables and equations as a bipartite dependency graph that allows the optimization of the solution of the system via symbolic manipulation of the equations. On the other hand, by leveraging the graph representation of the system we can provide additional functionalities that allow the user to automatically integrate the software modules corresponding to the various subsystems, such as the automatic calculations of the equations that identify variables at the intersection of different subsystems (see Figure 1 and function VARIABLE_CONNECTIONS in the Data Records section). Alternatively, the user can choose to avoid common variables among subsystems and to integrate them by manually providing the equations corresponding to their dependencies; however, we believe the user would soon see the advantage of the reduced complexity implied by our approach.

As for the use of software development platforms such as GitHub, together with control versioning systems such as Git, it is becoming nowadays a widespread practice in all scientific communities, allowing to develop software collaboratively, to publicly discuss issues, and to transparently track the development of the software[20].

Overview

Our implementation uses version 1.9 of the Julia programming language[13], and extensively leverages consolidated packages of the Julia ecosystem such as DifferentialEquations.jl[21] for the solution of the system of differential equations. While the Vensim implementation of the Earth4All model only provides the possibility to solve the system using Euler integration and Runge-Kutta integration of order at most four, our implementation allows using a variety of solvers, including the high-order Rosenbrock methods and the implicit extrapolation methods which are particularly suited for stiff systems of differential equations, among others (we defer the reader to the DifferentialEquations.jl documentation for an up-to-date list of solvers: https://docs.sciml.ai/DiffEqDocs/stable/solvers/ode_solve/). Importantly, our implementation allows the use of parallel methods such as multithreading, distributed methods, and GPU-accelerated methods for the solution of ensembles of systems, none of which are supported by Vensim. Earth4All.jl thus allows to readily speed up the solution of Monte Carlo simulations for uncertainty estimation and parameter search by orders of magnitude.

[1] The indices corresponding to variables can be displayed by running `println.(enumerate(states(sys))," -> ",getdescription.(states(sys)))`.

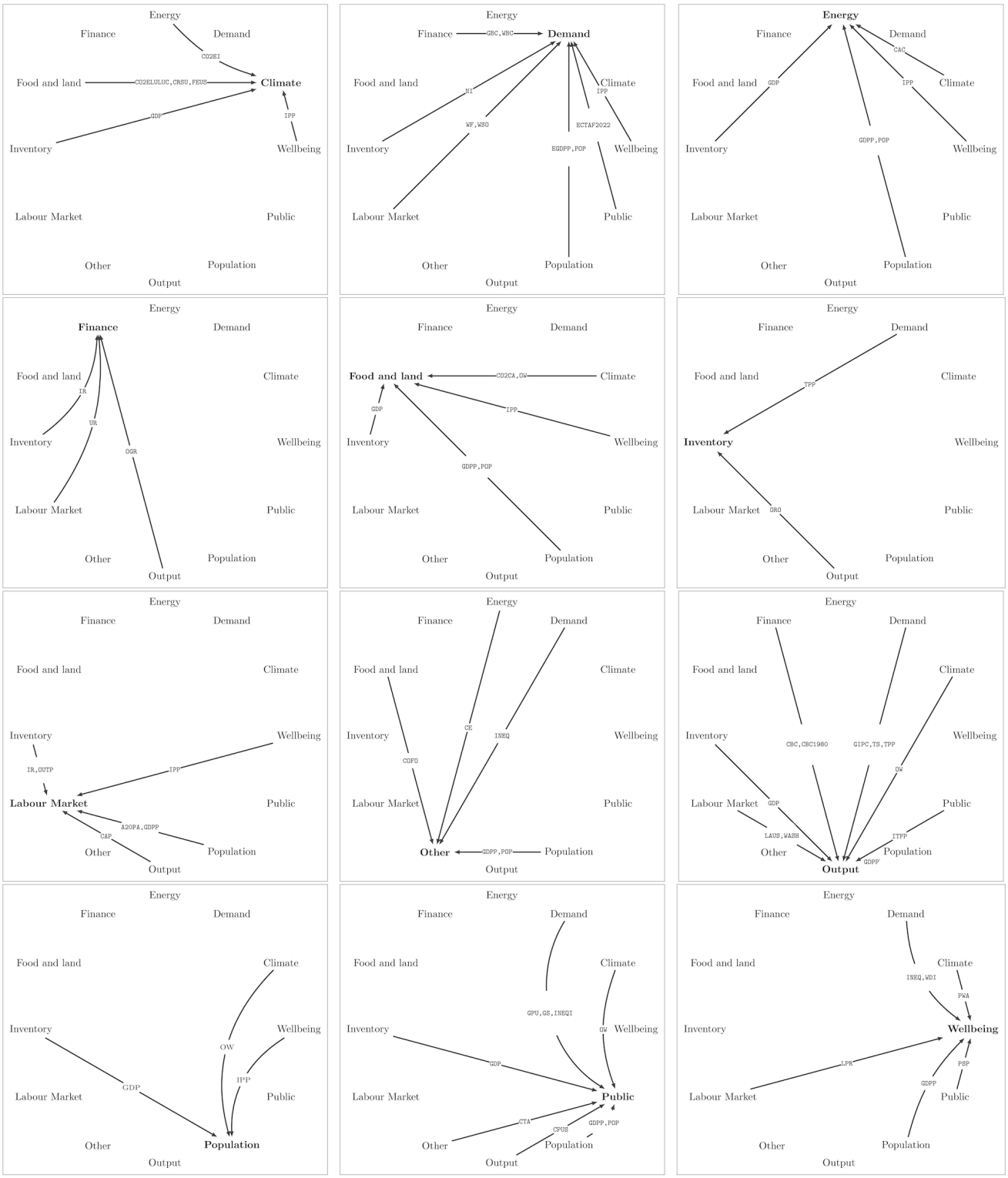

Figure 1. Diagrams representing the different subsystems of the Earth4All model and the dependencies among them resulting from common variables. For each subsystem (in boldface), the edges in the corresponding diagram show which variables of other subsystems it depends on.

Listing 1. Source code for generating Figure 2. `prob_func` is the function that is called to perturb the initial conditions. In this case, we are adding a small amount of normally distributed noise to the initial condition. The trajectory of any system variable can be shown by providing the respective index[1].

```
include("Earth4All.jl")
using ModelingToolkit, DifferentialEquations, Plots
tltl = Earth4All.run_tltl()
sys = structural_simplify(tltl)
prob = ODEProblem(sys, [], (1980, 2100))
function prob_func(prob, i, repeat)
    remake(prob, u0 = prob.u0 .* (1 .+ 0.1*randn(length(prob.u0))))
end
ensemble_prob = EnsembleProblem(prob, prob_func = prob_func)
sol = solve(ensemble_prob, Euler(), EnsembleThreads(),
            trajectories=100; dt=0.01, dtmax=0.01)
summ = EnsembleSummary(sol)
plot(summ, fillalpha=.5, trajectories = 26)
```

A simple example is illustrated in Figure 2, which is produced by the code in Listing 1.

As discussed in the Software Engineering Approaches to IAMs section, our implementation leverages the graph-based composable functionality of the Julia ModelingToolkit.jl library[15] to provide a modular implementation of the model in its main sectors. Additionally, ModelingToolkit allows the user to annotate variables with additional metadata, which we used to provide shorter acronyms for the model variables to produce more readable equations, while still providing the original name and additional information as associated variable metadata.

Our code also imports the WorldDynamics.jl package[22], which implements previous IAM from the Club of Rome and allows to compare the predictions of the Earth4All model with older models.

Interestingly, we found that the DELAY-N function, a classical function in System Dynamics theory that is widely used in IAMs, behaves differently in the Vensim and Stella software. Our implementation, which is consistent with the one provided in standard textbooks of System Dynamics[23], is consistent with the Stella implementation, thus accounting for a slight discrepancy between the original Vensim implementation and our Julia version of the Earth4All model.

All code and data for the replication of the present work are publicly available (see Code Availability section). The code organization is detailed below in the Data Records section.

Further details and additional tutorials for the use of the Earth4All.jl model can be found in the model documentation of the Earth4All.jl GitHub repository.

Data Records

The source code of the Earth4All.jl model is contained in the SRC folder of the provided repository. The file FUNCTIONS.JL contains utility functions, including the VARIABLE_CONNECTIONS function and System Dynamics functions such as the DELAY_N function described in the Overview section, and other functions for model validation such as those for the calculation of the relative error with respect to the Vensim implementation. The SRC contains a file for each subsystem of the model, namely CLIMATE.JL, DEMAND.JL, ENERGY.JL, FINANCE.JL, FOODLAND.JL, INVENTORY.JL, LABOURMARKET.JL, OTHER.JL, OUTPUT.JL, POPULATION.JL, PUBLIC.JL and WELLBEING.JL (see also Fig. 1 for a graphical representation of the shared variables among the different subsystems). Each of the previous files imports the files PARAMETERS.JL, INITIALISATIONS.JL, SUBSYSTEMS.JL and PLOTS.JL, which are contained in a subfolder with the same name of the file. The file PLOTS.JL produces a plot showing the main variables of the subsystem. The file SUBSYSTEMS.JL contains the variables and the equations defining the subsystem, while INITIALISATIONS.JL and PARAMETERS.JL contain the initial values and the parameters of the subsystem, respectively, as obtained from the model source code and technical report[24]. The file EARTH4ALL.JL imports all the above files, and by running it in the Julia REPL the user can run the model scenarios and leverage all functionalities

described in this paper.

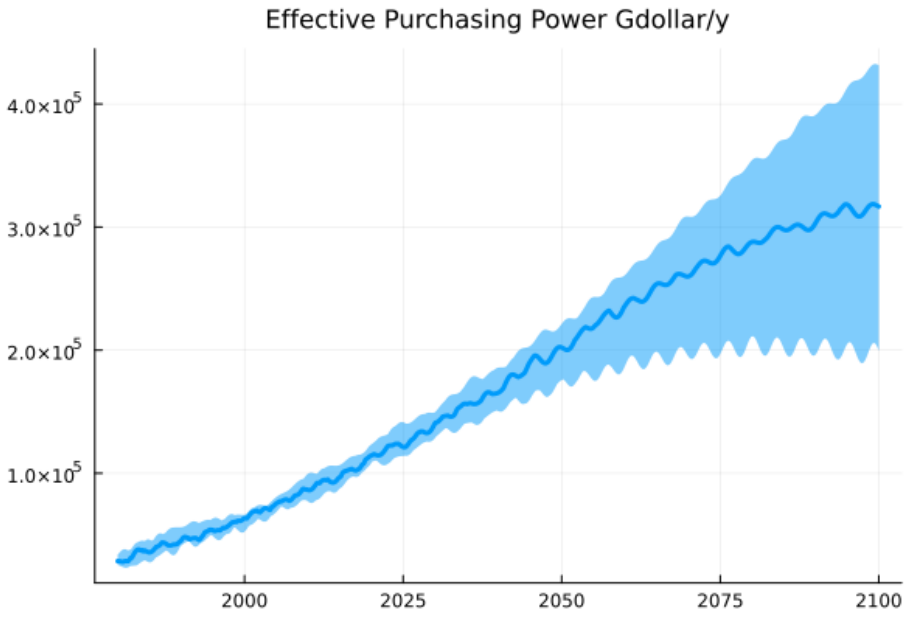

Figure 2. A simple example of usage of the ensemble method functionalities provided by DIFFERENTIALEQUATIONS.JL: each initial value u_0 is perturbed by a relative factor of the form $1 + Z_{u_0}/10$ where Z_{u_0} is a standard Gaussian random variable. The shadow in the plot shows the interval between the first and third quartile of the values of the annual Effective Purchasing Power, for each time value of ten independently perturbed runs. The latter values are automatically calculated by the ENSEMBLESUMMARY function.

Technical Validation

We have validated our implementation of the Earth4All model by comparing the results of the model with the results of the Vensim implementation of the original authors. Each variable of the model has been compared with the corresponding variable in the Vensim implementation, and the relative error has been calculated as the absolute value of the difference between the two implementations divided by the value of the Vensim implementation. Except for the variables which take the value zero in the Vensim implementation, the relative error is compatible with approximation errors due to floating point arithmetic, even without taking into account the discrepancy with the Vensim implementation of the DELAY-N function mentioned in the Overview section. Additionally, a visual comparison of the two implementations shows their accurate coincidence. To this aim, the repository documentation contains a detailed tutorial on how to compare the Earth4All.jl output against the Vensim implementation; for user convenience, all variable values of the Vensim output are provided in the VENSIMOUTPUT folder of the repository, and utility functions are provided to load the data and produce the plot of the overlaid trajectories (see e.g. Fig. 3). For producing plots, we make use of the PLOTLYJS.JL charting library which allows the user to interactively explore the data by zooming in and out, and by hovering over the data points to see their values.

Code and documentation availability

Code for Earth4All.jl is available on the EARTH4ALL.JL public Github repository at `https://github.com/worlddynamics/Earth4All.jl` under the free MIT license. The associated documentation is available at `https://worlddynamics.github.io/Earth4All.jl/`. The documentation includes exhaustive tables of the system variables, their defining equations, their initial values and dependencies on system parameters and other variables.

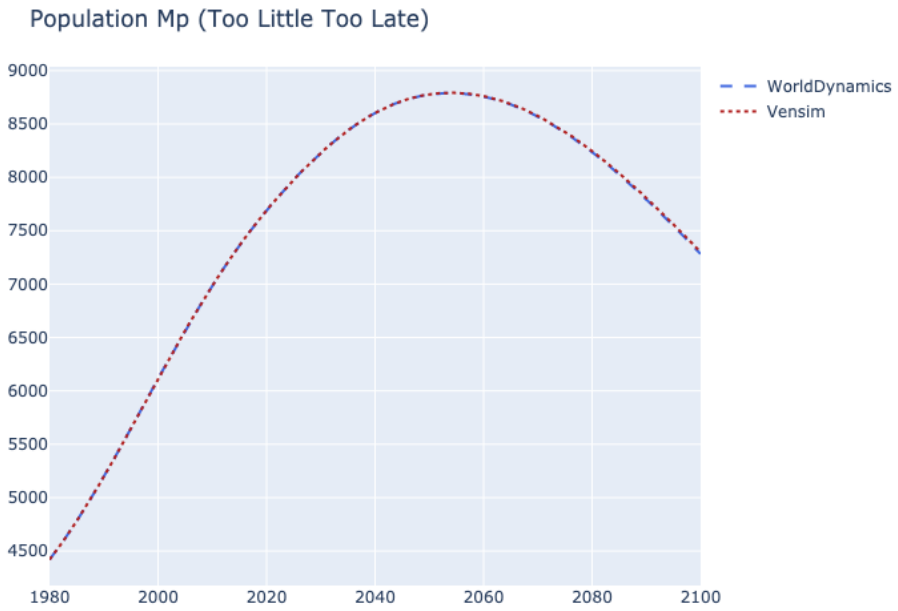

Figure 3. Example plot automatically generated by the PLOT_AND_COMPARE function for the population variable in the TLTL scenario. We can see that the output of the Vensim and Julia implementations of the Earth4All model are visually identical.

Acknowledgements

We thank Lucia Nasti for her help with the implementation of the Inventory subsystem. This work has been supported by the French government, through the UCAJEDI and UCA DS4H Investments in the Future projects managed by the National Research Agency (ANR) with the reference number ANR-15-IDEX-0001 and ANR-17-EURE-0004. The authors are grateful to the OPAL infrastructure from Université Côte d'Azur for providing resources and support.

Author contributions statement

P. C. and A. R. wrote most of the code. All authors contributed to the design and the technical aspects of the implementation. P. C. and E. N. conceived the original idea and supervised the work. E. N. wrote the manuscript in consultation with the other authors.

References

1. Costanza, R., Leemans, R., Boumans, R. M. J. & Gaddis, E. Integrated Global Models. In Costanza, R., Graumlich, L. J. & Steffen, W. (eds.) *Sustainability or Collapse?*, An Integrated History and Future of People on Earth, 417–445 (The MIT Press, 2007). j.ctt5hhhsc.26.
2. Forrester, J. W. *World Dynamics* (Cambridge, Mass. : Wright-Allen Press, 1973).
3. Meadows Donella H. *The Limits to Growth* (Universe Books, 1972), brown spots on pages edition edn.
4. Meadows, D. H., Meadows, D. L. & Randers, J. *Beyond the Limits: Confronting Global Collapse, Envisioning a Sustainable Future* (Chelsea Green Pub Co, Vermont, 1993).
5. Meadows, D. H., Randers, J. & Meadows, D. L. *Limits to Growth: The 30-Year Update* (Chelsea Green Publishing, White River Junction, Vt, 2004), illustrated edition edn.

6. Sverdrup, H. U., Olafsdottir, A. H. & Ragnarsdottir, K. V. Modelling Global Wolfram Mining, Secondary Extraction, Supply, Stocks-in-Society, Recycling, Market Price and Resources, Using the WORLD6 System Dynamics Model. *BioPhysical Econ. Resour. Qual.* **2**, 11, 10.1007/s41247-017-0028-x (2017).

7. Sverdrup, H. U., Olafsdottir, A. H. & Ragnarsdottir, K. V. Development of a Biophysical Economics Module for the Global Integrated Assessment Model WORLD7. In Cavana, R. Y., Dangerfield, B. C., Pavlov, O. V., Radzicki, M. J. & Wheat, I. D. (eds.) *Feedback Economics: Economic Modeling with System Dynamics*, Contemporary Systems Thinking, 247–283, 10.1007/978-3-030-67190-7_10 (Springer International Publishing, Cham, 2021).

8. Dixson-Decleve, S. *et al.* *Earth for All: A Survival Guide for Humanity* (New Society Publishers, Gabriola Island, British Columbia, Canada, 2022).

9. Biermann, F., Kanie, N. & Kim, R. E. Global governance by goal-setting: The novel approach of the UN Sustainable Development Goals. *Curr. Opin. Environ. Sustain.* **26–27**, 26–31, 10.1016/j.cosust.2017.01.010 (2017).

10. Moore, F. C. *et al.* Mimi-PAGE, an open-source implementation of the PAGE09 integrated assessment model. *Sci. Data* **5**, 180187, 10.1038/sdata.2018.187 (2018).

11. Jones, L. Vensim and the development of system dynamics. In *Discrete-Event Simulation and System Dynamics for Management Decision Making*, chap. 11, 215–247, 10.1002/9781118762745.ch11 (John Wiley & Sons, Ltd, 2014).

12. Lindfield, G. The role of the Stella software package in simulation and modelling. *Int. J. Math. Educ. Sci. Technol.* **23**, 865–880, 10.1080/0020739920230605 (1992).

13. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Rev.* **59**, 65–98, 10.1137/141000671 (2017).

14. Valuing Climate Damages: Updating Estimation of the Social Cost of Carbon Dioxide (2017). Report by the Committee on Assessing Approaches to Updating the Social Cost of Carbon, Board on Environmental Change, Division of Behavioral and Social Sciences, National Academies of Sciences, Engineering and Medicine.

15. Ma, Y. *et al.* ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling. *arXiv:2103.05244 [cs]* (2021). 2103.05244.

16. Nordhaus, W. Integrated Economic and Climate Modeling. Handbook of Computable General Equilibrium Modeling, Elsevier (2013).

17. Gauthier, R. & Ponto, S. *Designing Systems Programs* (Prentice Hall Press, Englewood Cliffs, N.J, 1970).

18. Parnas, D. L. On the criteria to be used in decomposing systems into modules. *Commun. ACM* **15**, 1053–1058 (1972).

19. Rennert, K. *et al.* Comprehensive evidence implies a higher social cost of CO2. *Nature* **610**, 687–692, 10.1038/s41586-022-05224-9 (2022).

20. Wilson, G. *et al.* Good enough practices in scientific computing. *PLOS Comput. Biol.* **13**, e1005510, 10.1371/journal.pcbi.1005510 (2017).

21. Rackauckas, C. & Nie, Q. Differentialequations. jl–a performant and feature-rich ecosystem for solving differential equations in julia. *J. open research software* **5** (2017).

22. Crescenzi, P., Natale, E. & Serafim, P. B. Worlddynamics.jl: v0.1.0, 10.5281/zenodo.7093581 (2022).

23. Richardson, G. P., III Pugh, A. L. & III, A. L. P. *Introduction to System Dynamics Modeling With Dynamo* (Productivity Pr, Portland, Or, 1981).

24. Randers, J. & Collste, D. The Earth4All model of human wellbeing on a finite planet towards 2100 (2023).