SciPy 2022 Submission 203

Update information
Update authors
Add file

Withdraw

If you want to **change any information** about your paper, use links in the upper right corner.

For all questions related to processing your submission you should contact the conference organizers. Click here to see information about this conference.

The submission has been updated!

Submission 203	
Title	Python vs. the pandemic: a case study in high-stakes software development
Track	Machine Learning & Data Science
Author keywords	COVID-19 Epidemiology Mathematical modeling Agent-based models Numba Sciris Covasim
Abstract	Background: When it became clear in early 2020 that COVID-19 was going to be a major public health threat, politicians and public health officials turned to academic disease modelers like us for urgent guidance. Academic software development is typically a slow and/or haphazard process, and we realized that business-as-usual would not suffice for dealing with this crisis. We assembled a team to rapidly develop a COVID-19 model to (a) incorporate new data as soon as it became available, (b) explore policy scenarios, and (c) predict likely future epidemic trajectories. Here we describe the software engineering approach we took to address the greatest public health challenge of the last century (tinyurl.com/covasim-york). Methods: Drawing on a decade-plus of modeling and software development experience, we began developing several COVID models in parallel, in different languages (Python, R, and C++). We needed a highly detailed, agent-based model in order to capture the important nuances of COVID-19 (such as household and school network contact patterns, testing policies, superspreaders, etc.). But speed was also of the essence, both in terms of development time and model run time. It turned out that our Python model, called Covasim (covasim.org), best met these needs. Typically, agent-based models are simulated by looping over all agents and then looping over time, but this was too slow in Python to be practical. Instead, we use NumPy arrays to represent properties of each agent, which produced a roughly 30-fold performance gain. We used Numba and 32-bit arithmetic to achieve a further four-fold performance gain. Together, these optimizations resulted in a roughly 100-fold performance gain over typical Python simulations, meaning that policymakers can simulate realistic scenarios in just a couple minutes on their laptops. In addition to high performance, Covasim had to be easy both for users and developers. For users, we developed a web interface (app.covasim.org) built on Flask and Vue.js, and we followed the philosophy "Common tasks should be simple" to ensure that the library's API was as straightforward as possible. To ensure transparency and trust with developers, we made the code-open source from the very beginning (github.com/institutefordiseasemodeling/covasim). To further reduce development time, we

	relied heavily on Sciris (github.com/sciris/sciris), which is a library for scientific computing that provides additional flexibility and ease-of-use on top of NumPy, SciPy, and Matplotlib, including parallel computing, array operations, and high-performance data types. Results and conclusion: Using this approach, we were able to achieve near-C++ performance (7-million simulated person-days per second of CPU time), while retaining the flexibility of a modular, object-oriented Python codebase. We began informing policy in several locations (including Washington State and the United Kingdom) within several weeks of starting model development, an order of magnitude faster than a typical model development process. Covasim is now one of the most influential COVID models (paper.covasim.org), having been used by policymakers in over a dozen countries. While the need for COVID modeling is hopefully starting to decline, we and our collaborators are now applying the lessons we learned from developing Covasim to other areas, such as polio and family planning simulations.
Submitted	Feb 23, 01:33 GMT
Last update	Feb 23, 05:54 GMT
Short Summary	When COVID turned the world upside down in early 2020, health officials asked academic disease modelers like us for urgent guidance. Here we describe how we built Covasim (covasim.org), an agent-based COVID model, by using standard Python libraries like NumPy, Numba, and SciPy along with less common ones like Sciris (sciris.org). Covasim was created in a few weeks, an order of magnitude faster than the typical model development process, and achieves performance comparable to C++ despite being written in pure Python. It is now used by researchers and policymakers in more than a dozen countries.
Type of Submission	Talk
Are you (the submitting author) interested in serving on the program committee?	

Authors

first name	last name	email	country	affiliation	Web page	corresponding?
Cliff	Kerr	cliff@covasim.org	United States	Bill and Melinda Gates Foundation	http://covasim.org	✔
Robyn	Stuart	robyn@math.ku.dk	Denmark	University of Copenhagen		
Dina	Mistry	dina.c.mistry@gmail.com	United States	Twitter		
Romesh	Abeysuriya	romesh.abeysuriya@burnet.edu.au	Australia	Burnet Institute		
Jamie	Cohen	jamie.cohen@gatesfoundation.org	United States	Bill and Melinda Gates Foundation		
Lauren	George	lauren.george@live.com	United States	Microsoft		
Daniel	Klein	daniel.klein@gatesfoundation.org	United States	Bill and Melinda Gates Foundation		