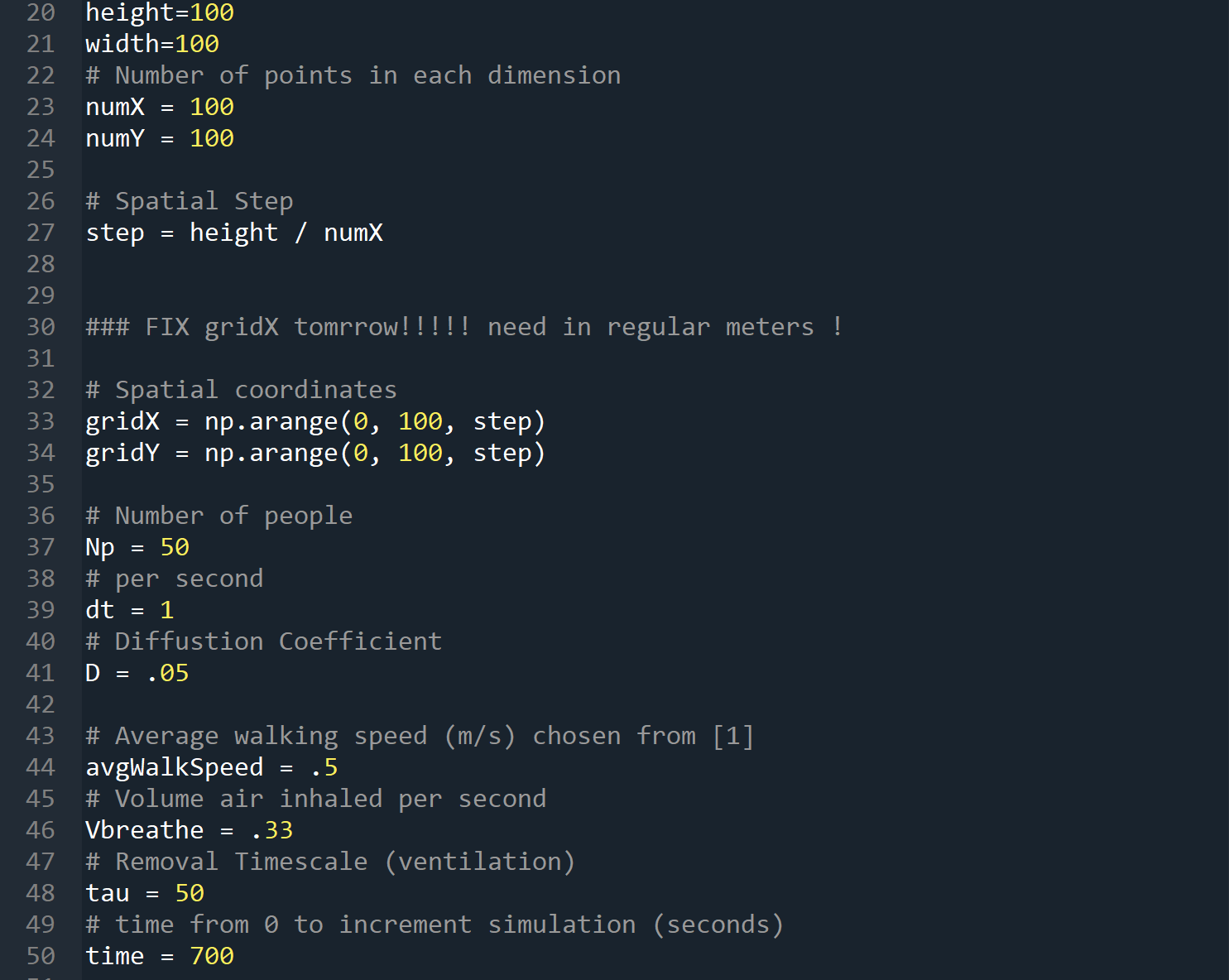
Jack Sullivan

Monte Carlo Simulation of SARS-CoV-2 Aerosol Transport via

Finite Difference Schemes

Vuorinen and an extensive team of researchers conducted several different types of simulations of SARS-CoV-2 aersol transport during the current pandemic. Many of these models were based on computational fluid dynamics softwares and too complex to replicate in a short timeframe. Here the Monte-Carlo modeling technique they used to study the spread of the virus based on parameters attained from CFD simulations were replicated in Python.



The spatial grid was constructed to be quantized, but that was eventually not used for the spatial coordinates of the “people” themselves, but to instead build the concentration field that the diffusion equation would manipulate.

The time and Np parameters were set arbitrarily, as they effected the outcome of the simulation but not the concentration of the viral particles in the air. Many of the other parameters were known from other simulations carried out by Vuorinen et al. The diffusion coefficient and timestep (dt), for example, had to be consistent with the known values. All values were measured in terms of meters and seconds, unlesss otherwise specified.

Walk speed: Vuorinen et al

Vbreathe

Removal timescale

The output on line 64 shows the energy level, the calculated solution of the weak value, the eigenvalue for the approximation, and the error between those two values. I was uncertain exactly how the coefficient (eigenvalue) was related to the solution map.

Text

Description automatically generated

In the variables dictionary, psi is configured as the unknown field and v as the test parameter.

Text

Description automatically generated

The single Direchlet boundary coundition is located on line 106, seeming to specify that the value of psi at the surface is zero.

Text

Description automatically generated

The solving method used, from scipy, found the eigenvalues of the system of linear equations that the differential equation was cast into to solve for psi.

Graphical user interface, application

Description automatically generated

After running the post-processing script, the output is displayed, presumably as a probability map for the electron structure of Hydrogen at different energy levels. The solutions are given by the terminal:

Text

Description automatically generated

Where the values in the FEM column are the values of the coefficients in the Ritz-Galerkin approximation.

The solver function was changed to a Newton-Raphson method from a different code, but the result seemed meaningless compared to the earlier one:

Shape

Description automatically generated

It did not produce results for the different Eigenvalues, and it was unclear how to adjust it to the quantum well code or if this was possible.

# Sources

Moehlis, J. M. (2001, October 24). *Solution of the Diffusion Equation by Finite Differences.* Retrieved from me.ucsb.edu: https://sites.me.ucsb.edu/~moehlis/APC591/tutorials/tutorial5/node3.html

*The two-dimensional diffusion equation.* (n.d.). Retrieved from scipython.com: https://scipython.com/book/chapter-7-matplotlib/examples/the-two-dimensional-diffusion-equation/

Ville Vuorinen, M. A. (2020). Modelling aerosol transport and virus exposure with numerical simulations in relation to SARS-CoV-2 transmission by inhalation indoors. *Safety Science, 130*. Retrieved from http://www.sciencedirect.com/science/article/pii/S0925753520302630