PyEscape: A narrow escape problem simulator package for python

Nathan Hughes[1], **Richard J Morris**[1], and **Melissa Tomkins**[1]

1 John Innes Centre, Norwich, UK

DOI: 10.21105/joss.02072

Software

- Review
- Repository
- Archive

Editor: Vincent Knight

Reviewers:

- @pdebuyl
- @markgalassi

Submitted: 27 January 2020
Published: 31 March 2020

Summary

In biology many research questions focus on uncovering the mechanisms which allow particles (molecules, proteins etc.) to move from one location to another. Often these movements are from one domain to another and these domains are in some way contained. The "narrow escape problem" is a bio-physics problem where the solution would provide the average time required for a Brownian particle to escape a bounded domain through a particular opening. Originally proposed by (Holcman & Schuss, 2004), solutions to this problem have been given and refined over the years (Schuss, Singer, & Holcman, 2007).

Recently, solutions have been given (Kaye & Greengard, 2020) for more complex narrow escape problems, such as arbitrary escape pore patterning and size variation. However, these are provided without easily accessible implementations and confine the problem to a single container shape.

Here, we present a novel Python library which enables stochastic simulations to be ran in order to approximate the narrow escape problem for unique scenarios. The mathematical models provided are implementations of random-walks in 3-dimensions, (Codling, Plank, & Benhamou, 2008). With our models we show that they are good approximations for analytical solutions (example notebook), and that they can be scaled to many custom problems.

Through this library we provide functionality for both cube and spherical shaped domains. We enable a broad range of simulation variables to control, in the most simple case a user will select the volume and shape they wish to act as their container, the number and size of escape pores on the container's surface and the diffusion coefficient of the particle of interest. Additionally, we give an implementation of Fibonacci spheres which allows for the fast placement of escape pores pseudo-evenly spaced on the surface of a sphere, this is often useful in experiments to test how number of escapes relates to mean escape time.

External libraries used

The models given are implemented through NumPy (van der Walt, Colbert, & Varoquaux, 2011)], results are visualised through Matplotlib (Hunter, 2007)] and documentation has been implemented using Jupyter-notebooks (Kluyver et al., 2016).

Acknowledgements

We acknowledge the Norwich Research Park Doctoral Training programme (NRDTP), European Research Council and the Biotechnology and Biological Sciences Research Council (BBSRC) for their support and funding.

References

Codling, E. A., Plank, M. J., & Benhamou, S. (2008). Random walk models in biology. *Journal of The Royal Society Interface*, *5*(25), 813–834. doi:10.1098/rsif.2008.0014

Holcman, D., & Schuss, Z. (2004). Escape Through a Small Opening: Receptor Trafficking in a Synaptic Membrane. *Journal of Statistical Physics*, *117*(5), 975–1014. doi:10.1007/s10955-004-5712-8

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Kaye, J., & Greengard, L. (2020). A fast solver for the narrow capture and narrow escape problems in the sphere. *Journal of Computational Physics: X*, *5*, 100047. doi:10.1016/j.jcpx.2019.100047

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., et al. (2016). Jupyter notebooks – a publishing format for reproducible computational workflows. (F. Loizides & B. Schmidt, Eds.). IOS Press.

Schuss, Z., Singer, A., & Holcman, D. (2007). The narrow escape problem for diffusion in cellular microdomains. *Proceedings of the National Academy of Sciences*, *104*(41), 16098–16103. doi:10.1073/pnas.0706599104

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*, *13*(2), 22–30. doi:10.1109/MCSE.2011.37