The Journal of Open Source Software

pyDeltaRCM: a flexible numerical delta model

Andrew J. Moodie[1], **Jayaram Hariharan**[1], **Eric Barefoot**[2], **and Paola Passalacqua**[1]

1 Department of Civil, Architectural, and Environmental Engineering, University of Texas at Austin, Austin, TX, USA **2** Department of Earth, Environmental and Planetary Sciences, Rice University, Houston, TX, USA

DOI: 10.21105/joss.03398

Software

- Review
- Repository
- Archive

Editor: Katy Barnhart

Reviewers:

- @zsylvester
- @jhnienhuis
- @salterg

Submitted: 08 June 2021
Published: 26 August 2021

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

River deltas provide many societal benefits, and sustainability of these landforms may be impacted by human modification and global climate change. Reduced-complexity numerical delta models incorporate limited physical processes, allowing researchers to assess the spatiotemporal evolution of landscape response to individual processes and environmental forcings. Isolating individual processes is useful to understand, for example, shifting delta morphology due to sea-level rise, changing vegetal cover, or flooding intensity. As a result, many numerical delta models have been proposed in the literature, and results from these studies are difficult to compare because of various design and implementation choices. *pyDeltaRCM* (`v2.0`) delivers a computationally efficient and easy-to-customize implementation of the DeltaRCM numerical model (Liang, Voller, et al., 2015), enabling comparison and reproducibility in studies of delta change due to various environmental forcings.

Statement of need

River deltas are societally important landforms because they provide arable land, deep inland ports, and are home to hundreds of millions of people globally (Edmonds et al., 2020). Existing at the interface between landmasses and water bodies, deltas are impacted by a multitude of processes arising in both of these domains. For example, changes in sediment input to the delta modulate the rate at which new land is built; similarly, rising water levels in the downstream basin create flooded land. In addition to natural processes, human landscape modification renders deltaic environments more sensitive to global climate change into the future (Paola et al., 2011). Demand to understand natural delta processes, and how these processes will respond to various environmental forcings, has led to a proliferation of numerical delta models in the literature (Overeem et al., 2005).

The DeltaRCM delta model (Liang, Voller, et al., 2015) has gained popularity among geomorphologists due to an attractive balance of computational cost, realism, and interpretability (Larsen et al., 2016). For example, studies have employed the DeltaRCM design to examine delta morphology and dynamism response to sea-level rise and regional subsidence (Liang, Van Dyk, et al., 2016; Liang, Kim, et al., 2016), as well as extended model design to simulate delta evolution with vegetation (Lauzon & Murray, 2018) and ice and permafrost (Lauzon et al., 2019; Piliouras et al., 2021). However, comparison among these studies is difficult, owing to disparate code bases, various implementation choices, lack of version control, and proprietary software dependencies.

Background

Here, version 2.0 of *pyDeltaRCM* is introduced; *pyDeltaRCM* is a computationally efficient, free and open source, and easy-to-customize numerical delta model based on the original DeltaRCM design. The original DeltaRCM framework is inspired by well-understood physical phenomena, and models mass movement as a probabilistic weighted random-walk process coupled with a set of hierarchical rules; the model is extensively described in Liang, Voller, et al. (2015) and Liang, Geleynse, et al. (2015).

This same framework is the basis for *pyDeltaRCM* v2.0, with a few modifications selected only to resolve known numerical instabilities, improve computational efficiency, and support reproducible simulations. *PyDeltaRCM* depends only on common Python packages `numpy` (Harris et al., 2020), `matplotlib` (Hunter, 2007), `scipy` (Virtanen et al., 2020), `netCDF4`, `pyyaml`, and `numba` (Lam et al., 2015).

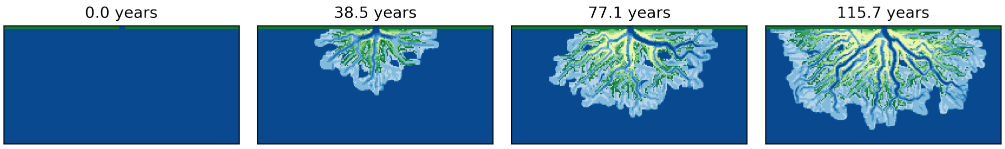

Figure 1: Simulation with *pyDeltaRCM* v2.0, default parameter set, and random `seed: 10151919`. Simulation was run for 4000 timesteps, and assumes 10 days of bankfull discharge per year; computational time was ~2 hours.

Flexible and easy to use

pyDeltaRCM is an object-oriented package, providing the central model class `DeltaModel`. By creating custom model behavior as subclasses of `DeltaModel`, researchers can easily add, subtract, and modify model components without altering code that is not pertinent to the science objective. Importantly, separating custom code from core model code makes clear how different studies can be compared. The *pyDeltaRCM* documentation provides several examples for how to implement custom model behavior on top of the core `DeltaModel` object.

pyDeltaRCM also provides infrastructure to accelerate scientific exploration, such as the ability to configure multiple simulations from a single file. Additionally, a preprocessor orchestrates `parallel` simulations for multi-core systems (optionally), and implements several tools to support simulations exploring a parameter space. For example, `matrix` expansion converts lists of parameters into an n-dimensional set of simulations. Similarly, replicate simulations can be created via an `ensemble` specification.

Reproducibility and computational efficiency were important priorities in *pyDeltaRCM* development. For example, to-disk logging records all parameters, system-level and version data, and random-seed information to ensure that all runs can be recreated. Additionally, "checkpoint" infrastructure has been added to the model, which records simulation progress during computation and can later resume model runs for further simulation. Finally, *pyDeltaRCM* uses `numba` for computational optimization (Lam et al., 2015), and does not depend on any proprietary software.

pyDeltaRCM component units and integrations are thoroughly documented and tested. Component-level documentation describes implementation notes, whereas narratives in "Guide" and "Example" documentation describes high-level model design and best practices for model use and development. *pyDeltaRCM* also couples with other numerical models via the CSDMS Basic Model Interface 2.0 (DeltaRCM Team, n.d.; Hutton et al., 2020).

Acknowledgments

We gratefully acknowledge Rebecca Lauzon and Mariela Perignon for developing an implementation of *DeltaRCM* in Python that was the basis for *pyDeltaRCM*. We also thank the National Science Foundation for supporting us in developing this software, by way of a Postdoctoral Fellowship to A.M. (EAR 1952772) and a grant to J.H. and P.P. (EAR 1719670).

References

DeltaRCM Team. (n.d.). *Basic modeling interface (BMI) wrapper to the pyDeltaRCM model*. https://github.com/DeltaRCM/BMI_pyDeltaRCM

Edmonds, D. A., Caldwell, R. L., Brondizio, E. S., & Siani, S. M. O. (2020). Coastal flooding will disproportionately impact people on river deltas. *Nature Communications*, *11*. https://doi.org/10.1038/s41467-020-18531-4

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Hutton, E. W. H., Piper, M. D., & Tucker, G. E. (2020). The basic model interface 2.0: A standard interface for coupling numerical models in the geosciences. *Journal of Open Source Software*, *5*(51), 2317. https://doi.org/10.21105/joss.02317

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. https://doi.org/10.1145/2833157.2833162

Larsen, L. G., Eppinga, M. B., Passalacqua, P., Getz, W. M., Rose, K. A., & Liang, M. (2016). Appropriate complexity landscape modeling. *Earth-Science Reviews*, *160*, 111–130. https://doi.org/10.1016/j.earscirev.2016.06.016

Lauzon, R., & Murray, A. B. (2018). Comparing the cohesive effects of mud and vegetation on delta evolution. *Geophysical Research Letters*, *45*(19), 10, 437–410, 445. https://doi.org/10.1029/2018GL079405

Lauzon, R., Piliouras, A., & Rowland, J. C. (2019). Ice and permafrost effects on delta morphology and channel dynamics. *Geophysical Research Letters*, *46*(12), 6574–6582. https://doi.org/10.1029/2019GL082792

Liang, M., Geleynse, Nathanael., Edmonds, D. A., & Passalacqua, P. (2015). A reduced-complexity model for river delta formation – Part 2: Assessment of the flow routing scheme. *Earth Surface Dynamics*, *3*(1), 87–104. https://doi.org/10.5194/esurf-3-87-2015

Liang, M., Kim, W., & Passalacqua, P. (2016). How much subsidence is enough to change the morphology of river deltas? *Geophysical Research Letters*, *43*(19), 10, 266–210, 276. https://doi.org/10.1002/2016GL070519

Liang, M., Van Dyk, C., & Passalacqua, P. (2016). Quantifying the patterns and dynamics of river deltas under conditions of steady forcing and relative sea level rise. *Journal of Geophysical Research: Earth Surface*, *121*(2), 465–496. https://doi.org/10.1002/2015JF003653

Liang, M., Voller, V. R., & Paola, C. (2015). A reduced-complexity model for river delta formation – Part 1: Modeling deltas with channel dynamics. *Earth Surface Dynamics*, *3*(1), 67–86. https://doi.org/10.5194/esurf-3-67-2015

Overeem, I., Syvitski, J. P. M., & Hutton, E. W. H. (2005). Three-Dimensional Numerical Modeling of Deltas. In *River Deltas–Concepts, Models, and Examples*. SEPM Society for Sedimentary Geology. https://doi.org/10.2110/pec.05.83.0011

Paola, C., Twilley, R. R., Edmonds, D. A., Kim, W., Mohrig, D., Parker, G., Viparelli, E., & Voller, V. R. (2011). Natural Processes in Delta Restoration: Application to the Mississippi Delta. *Annual Review of Marine Science*, *3*(1), 67–91. https://doi.org/10.1146/annurev-marine-120709-142856

Piliouras, A., Lauzon, R., & Rowland, J. C. (2021). Unraveling the combined effects of ice and permafrost on arctic delta morphodynamics. *Journal of Geophysical Research: Earth Surface*, *126*(4). https://doi.org/10.1029/2020JF005706

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2