

# 1 *pyDeltaRCM*: a flexible numerical delta model

2 **Andrew J. Moodie<sup>1</sup>, Jayaram Hariharan<sup>1</sup>, Eric Barefoot<sup>2</sup>, and Paola**  
3 **Passalacqua<sup>1</sup>**

4 **1** Department of Civil, Architectural, and Environmental Engineering, University of Texas at Austin,  
5 Austin, TX, USA **2** Department of Earth, Environmental and Planetary Sciences, Rice University,  
6 Houston, TX, USA

DOI: [10.21105/joss.03345](https://doi.org/10.21105/joss.03345)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Pending Editor](#) ↗

Submitted: 08 June 2021

Published: 08 June 2021

## License

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

## 7 Summary

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

## 19 Statement of need

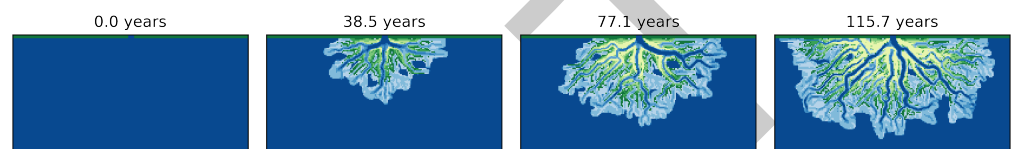
20 River deltas are societally important landforms because they provide arable land, deep inland  
21 ports, and are home to hundreds of millions of people globally ([Edmonds et al., 2020](#)). Existing  
22 at the interface between landmasses and water bodies, deltas are impacted by a multitude  
23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

30 The DeltaRCM delta model ([M. Liang, Voller, et al., 2015](#)) has gained popularity among geo-  
31 morphologists due to an attractive balance of computational cost, realism, and interpretability  
32 ([Larsen et al., 2016](#)). For example, studies have employed the DeltaRCM design to exam-  
33 ine delta morphology and dynamism response to sea-level rise and regional subsidence ([Man](#)  
34 [Liang, Van Dyk, et al., 2016](#); [Man Liang, Kim, et al., 2016](#)), as well as extended model design  
35 to simulate delta evolution with vegetation ([Lauzon & Murray, 2018](#)) and ice and permafrost  
36 ([Lauzon et al., 2019](#); [Piliouras et al., 2021](#)). However, comparison among these studies is dif-  
37 ficult, owing to disparate code bases, various implementation choices, lack of version control,  
38 and proprietary software dependencies.

## Background

Here, version 2.x 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 [M. Liang, Voller, et al. \(2015\)](#) and [M. Liang, Geleynse, et al. \(2015\)](#).

This same framework is the basis for *pyDeltaRCM* v2.x, 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.x, default parameter set, and random seed: 10151919. Simulation was run for 4000 timesteps, and assumes 10 days of bankfull discharge per year.

## 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. 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](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, N., 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, Man, 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, Man, 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>

- 119 Liang, M., Voller, V. R., & Paola, C. (2015). A reduced-complexity model for river delta  
120 formation – Part 1: Modeling deltas with channel dynamics. *Earth Surface Dynamics*,  
121 3(1), 67–86. <https://doi.org/10.5194/esurf-3-67-2015>
- 122 Overeem, I., Syvitski, J. P. M., & Hutton, E. W. H. (2005). Three-Dimensional Numerical  
123 Modeling of Deltas. In *River Deltas—Concepts, Models, and Examples*. SEPM Society for  
124 Sedimentary Geology. <https://doi.org/10.2110/pec.05.83.0011>
- 125 Paola, C., Twilley, R. R., Edmonds, D. A., Kim, W., Mohrig, D., Parker, G., Viparelli, E.,  
126 & Voller, V. R. (2011). Natural Processes in Delta Restoration: Application to the Mis-  
127 sissippi Delta. *Annual Review of Marine Science*, 3(1), 67–91. [https://doi.org/10.1146/](https://doi.org/10.1146/annurev-marine-120709-142856)  
128 [annurev-marine-120709-142856](https://doi.org/10.1146/annurev-marine-120709-142856)
- 129 Piliouras, A., Lauzon, R., & Rowland, J. C. (2021). Unraveling the combined effects of ice  
130 and permafrost on arctic delta morphodynamics. *Journal of Geophysical Research: Earth*  
131 *Surface*, 126(4). [https://doi.org/https://doi.org/10.1029/2020JF005706](https://doi.org/10.1029/2020JF005706)
- 132 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,  
133 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,  
134 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson,  
135 E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scien-  
136 tific Computing in Python. *Nature Methods*, 17, 261–272. [https://doi.org/10.1038/](https://doi.org/10.1038/s41592-019-0686-2)  
137 [s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2)