

Model Comparison - Original DeltaRCM and pyDeltaRCM

J. Hariharan

May 8, 2021

Contents

1	Background	2
2	Modeled Scenarios	2
2.1	Parameter Set	2
2.2	Parameter Differences	3
3	Topographies	3
4	Metrics	4
4.1	Surface Morphology	4
4.1.1	Planform Area	4
4.1.2	Channelized Fraction	5
4.1.3	Shoreline Roughness	6
4.2	Surface Dynamics	7
4.2.1	Channel Planform Overlap	7
4.3	Subsurface Composition	8
4.3.1	Net-to-Gross Ratio	8
5	Statistical Tests	8
5.1	Rank-Sum Tests	8
6	Conclusions	9
Appendices		10
A	Topography Figures	10
References		17

1 Background

DeltaRCM [3, 4] is a proven [3, 6] and popular [1, 2, 5, 8] numerical model used to simulate river delta growth and evolution. The original model was written in MATLAB [7], leaving room for improvement with regards to code accessibility, run-time, documentation, parallelization, and extensibility. With these needs in mind, a Python [9] implementation of the original DeltaRCM model was developed, dubbed *pyDeltaRCM*. This re-implementation of the DeltaRCM model includes minor modifications to the original DeltaRCM methods, and so there is a need to compare the new *pyDeltaRCM* model outputs to those from the original and validated DeltaRCM code. In this report, this model-to-model comparison is performed using a suite of 35 model runs from each version of DeltaRCM. The similarity of the two models is quantitatively assessed using a small collection of metrics.

2 Modeled Scenarios

7 scenario are modeled to simulate 200 years of river delta growth into an empty basin assuming 10 bankfull days of discharge per year. For each scenario, 5 replicate runs are conducted to capture the range of stochastic variability present in the random walk-based methodology.

2.1 Parameter Set

A comprehensive although non-exhaustive list of the model parameter values used for both the MATLAB and Python simulations is provided in Table 1.

Parameter	Value
dx	50 m
L0	150 m
Np_water	2000
Np_sed	2000
u0	1 m/s
N0	250 m
h0	5 m
H_SL	0 m
f_bedload	0.2
C0_percent	0.1
Csmooth	0.9
omega_sfc	0.1
omega_flow	0.9
Nsmooth	10
theta_water	1
coeff_theta_sand	2
coeff_theta_mud	1
beta	3
sed_lag (lambda)	1
coeff_U_dep_mud	0.3
coeff_U_ero_mud	1.5
coeff_U_ero_sand	1.05
alpha	0.1

Table 1: Model Parameters

The 7 modeled scenarios represent different levels of steady sea level rise (SLR). These 7 values are 0, 5, 10, 15, 20, 30, and 40 mm/yr. Rates of sea level are scaled temporally under the assumption that the model is simulating 10 bankfull days of discharge per year.

A low value for f_bedload is used in these simulations as low sand content generates highly cohesive and more variable simulation results. High input sand fraction values will result in simulations with smoother deltas. For the purposes of comparing models, it should be easier to detect differences between rougher planform shapes than smooth ones.

2.2 Parameter Differences

The interval at which model outputs are saved is different between the models. For the MATLAB code, saving occurs every 25 model timesteps, whereas in the Python code the model is saved every 35 timesteps. The saving interval does not impact the model behavior, but rather our ability to observe what the model is doing.

The modeled domain size is different between the two sets of runs performed, although in neither case do the simulated deltas reach the domain boundaries. In the MATLAB scenarios, the basin is 9 km long and 17 km wide. In the Python scenarios the basin is slightly with a length of 10 km and a width of 20 km.

Two model parameters are set to slightly different values in these sets of model runs. The reference slope, S_0 , is 1.4×10^{-4} for the MATLAB model runs, and 1.5×10^{-4} for the Python model runs. The number of times water parcel are iterated per timestep, `itermax`, differs between the runs. In the MATLAB models `itermax` = 1, as this is a computationally expensive step because the water parcels are routed in serial. The Python models use `itermax` = 3, as the new version of the model now routes all of the water parcels in parallel making this step much faster.

3 Topographies

Topography at the end of model runs (after 200 years), is shown in Figure 1. Break in colorbar to distinguish between water and land is made at 0.5 m below sea level [5, 6].

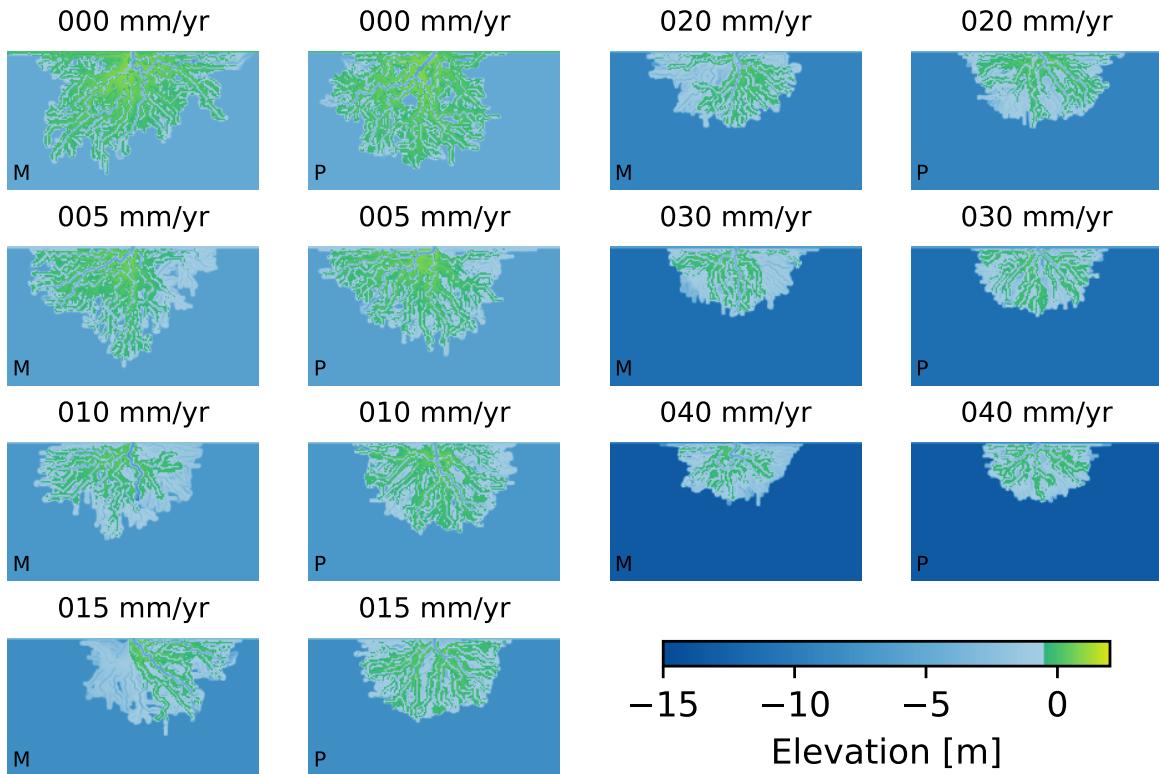


Figure 1: Single model realization final topographies. Numerical model used indicated by letter in bottom left corner of each plot, M for MATLAB and P for Python. Note that the domain has been clipped in all images.

4 Metrics

4.1 Surface Morphology

4.1.1 Planform Area

The area of the delta planform is identified using a topographic threshold of 0.5 m below the sea level [5, 6] and the opening angle method [11] with an opening angle of 75° .

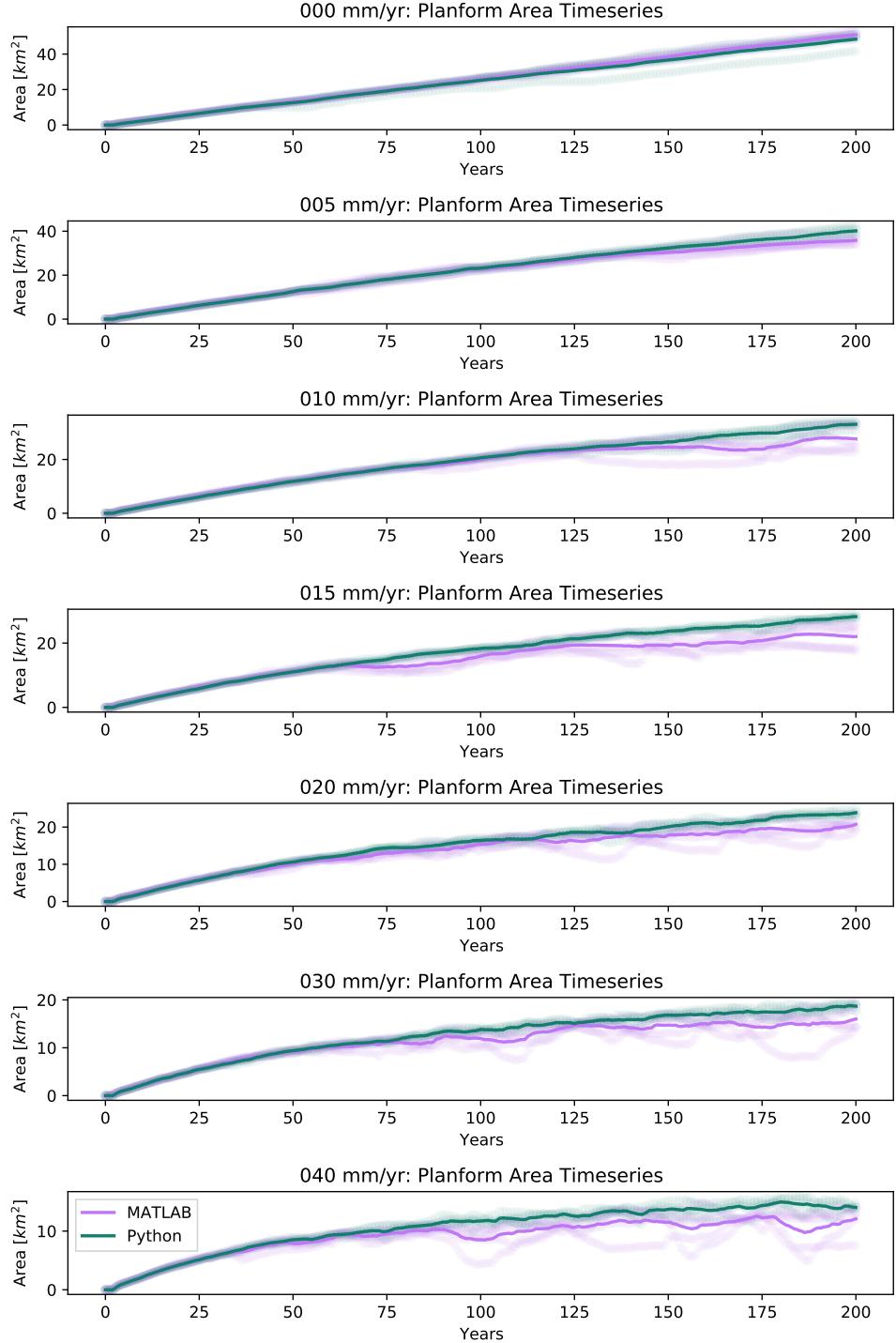


Figure 2: Planform area timeseries with average values shown as lines and individual run results shown as faint circles. MATLAB results shown in light purple, Python results in dark green.

4.1.2 Channelized Fraction

The channelized pixels on the delta topset are identified as locations where the water velocity exceeds the threshold for sediment mobilization (0.3 m/s) [1, 2, 5, 6]. Then the channelized fraction is computed by dividing the channelized area by the total area of the topset.

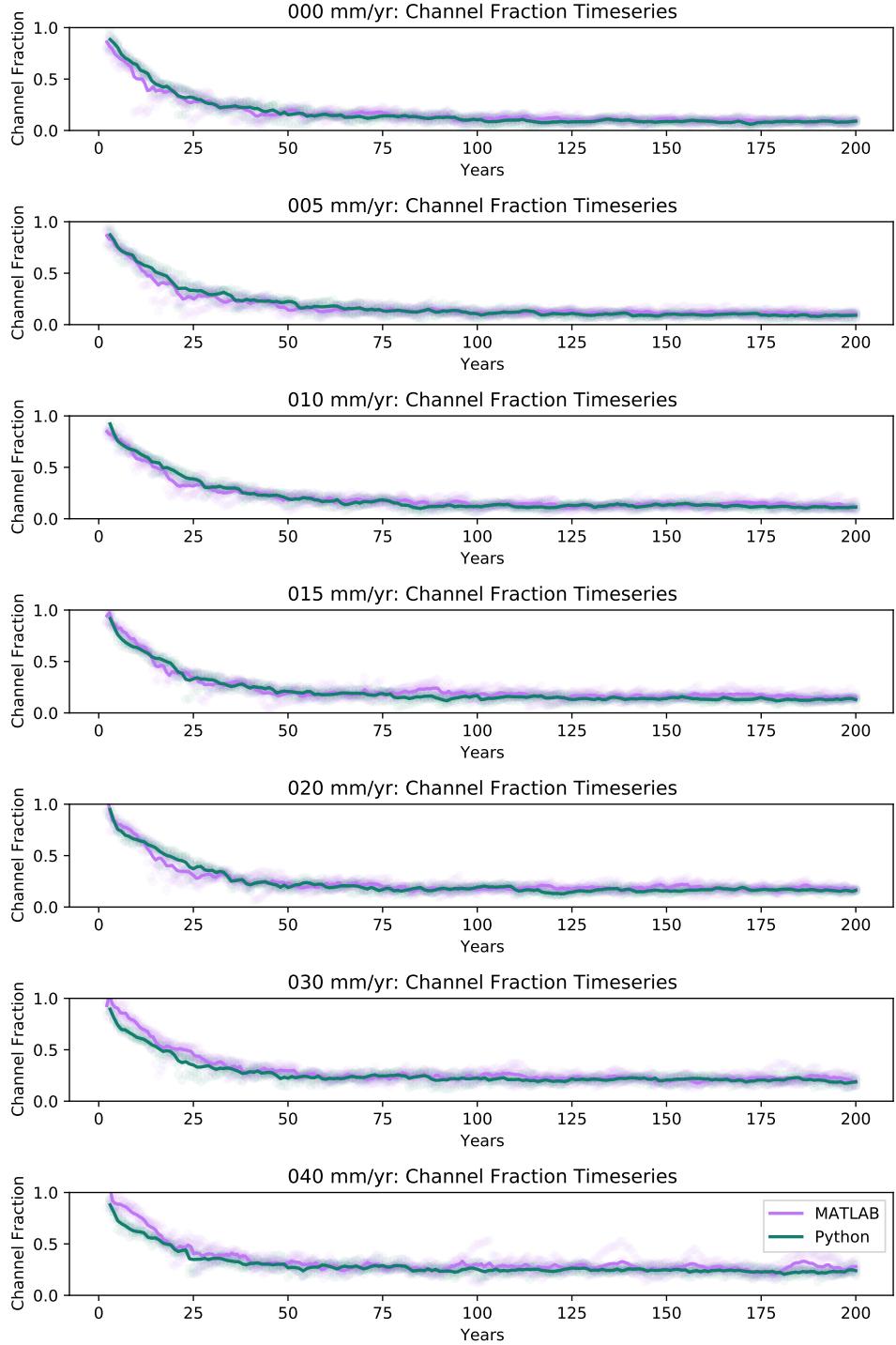


Figure 3: Channelized fraction timeseries with average values shown as lines and individual run results shown as faint circles. MATLAB results shown in light purple, Python results in dark green.

4.1.3 Shoreline Roughness

Shoreline roughness is calculated as the length of the shoreline divided by the square root of the planform area. The shoreline is derived from the land map of the delta planform by performing morphological operations using the scikit-image Python library [12]. First edge detection [10] is performed, and then the resulting edge is made binary and skeletonized [15] to create the shoreline image.

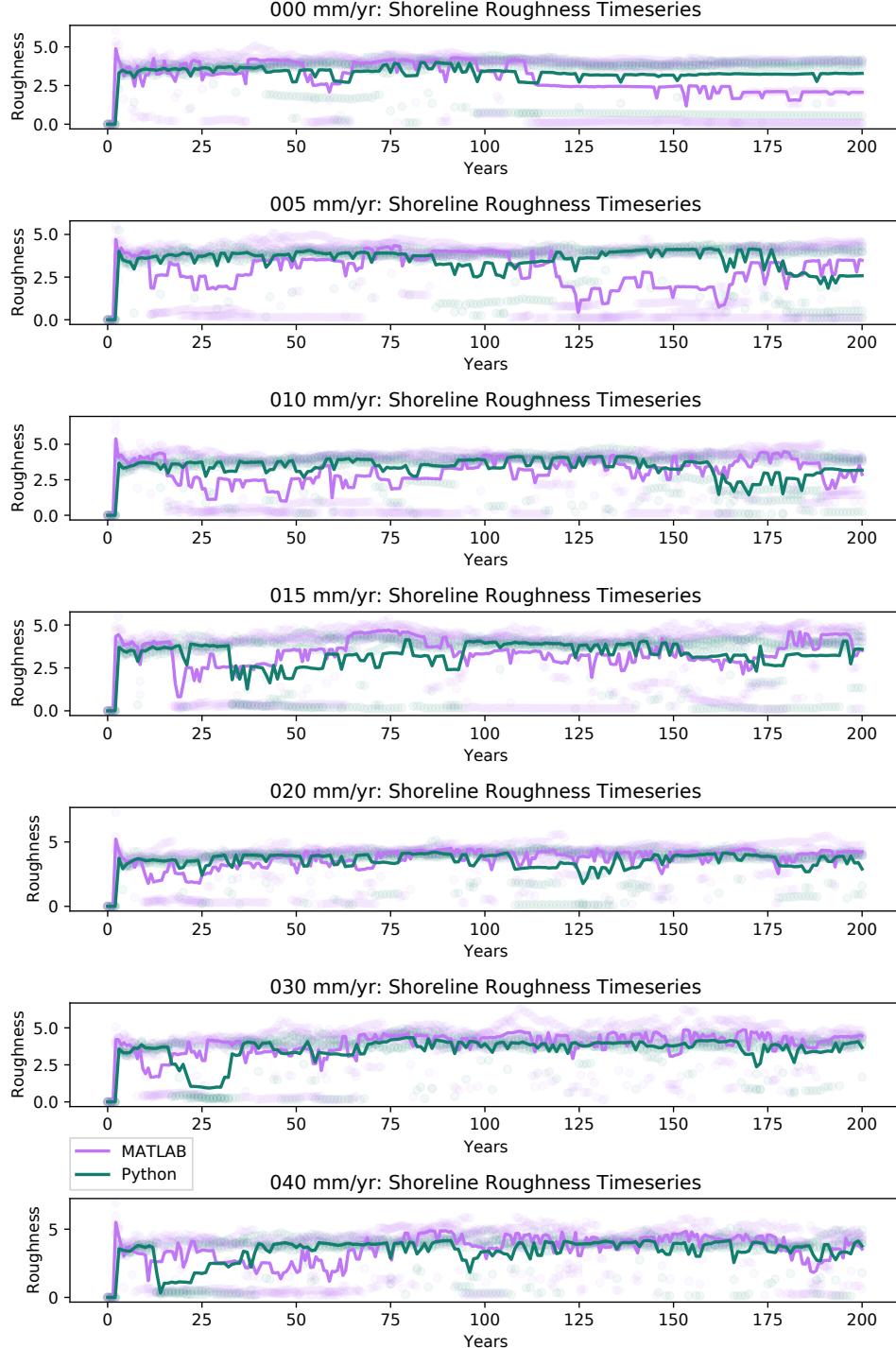


Figure 4: Shoreline roughness timeseries with average values shown as lines and individual run results shown as faint circles. MATLAB results shown in light purple, Python results in dark green.

4.2 Surface Dynamics

4.2.1 Channel Planform Overlap

The channel planform overlap metric [13] measures the loss of similarity in channel locations over time. It is common for the normalized overlap value, O_Φ , to be used to fit an exponential decay equation

$$O_\Phi = (a_M - p_M) e^{-Mt} + p_M, \quad (1)$$

where M , the decay constant (units of inverse time), is a mobility parameter describing how quickly the channel system is reworking itself. For these 200 year simulations, the channel planform overlap is evaluated using base channel maps from recorded outputs in the interval [100, 200] years. These base maps are compared to those within 50 years, meaning the transient maps are sourced from the interval (100, 200] years.

SLR [mm/yr]	a_M	$M [\text{yr}^{-1}]$	p_M
000	0.839 ± 0.013	0.059 ± 0.003	0.122 ± 0.008
005	0.873 ± 0.012	0.078 ± 0.003	0.197 ± 0.005
010	0.859 ± 0.012	0.064 ± 0.002	0.135 ± 0.007
015	0.845 ± 0.016	0.075 ± 0.003	0.116 ± 0.007
020	0.832 ± 0.014	0.064 ± 0.003	0.09 ± 0.008
030	0.853 ± 0.016	0.089 ± 0.004	0.105 ± 0.006
040	0.866 ± 0.017	0.104 ± 0.004	0.127 ± 0.005

Table 2: MATLAB planform overlap exponential fit parameters. Average parameter values for the set of 5 replicate runs reported \pm the average standard deviation of the individual parameter estimates.

SLR [mm/yr]	a_M	$M [\text{yr}^{-1}]$	p_M
000	0.871 ± 0.014	0.081 ± 0.004	0.186 ± 0.009
005	0.886 ± 0.013	0.079 ± 0.003	0.175 ± 0.008
010	0.869 ± 0.014	0.071 ± 0.004	0.135 ± 0.011
015	0.867 ± 0.014	0.078 ± 0.004	0.122 ± 0.01
020	0.864 ± 0.015	0.082 ± 0.004	0.128 ± 0.009
030	0.859 ± 0.017	0.094 ± 0.005	0.142 ± 0.009
040	0.876 ± 0.015	0.102 ± 0.004	0.119 ± 0.007

Table 3: Python planform overlap exponential fit parameters. Average parameter values for the set of 5 replicate runs reported \pm the average standard deviation of the individual parameter estimates.

4.3 Subsurface Composition

4.3.1 Net-to-Gross Ratio

The net-to-gross ratio of the total deposit is calculated as the sum of sand fraction values in the preserved stratigraphy divided by the total number of voxels in the stratigraphy. Stratigraphy in both models is computed at a vertical resolution of 5 cm.

SLR [mm/yr]	Run 1	Run 2	Run 3	Run 4	Run 5	Avg.	Std. Dev.
000	0.1909	0.1889	0.1924	0.1841	0.1918	0.1896	0.003
005	0.2236	0.2005	0.1939	0.2024	0.216	0.2073	0.0108
010	0.2611	0.2451	0.2199	0.2673	0.2584	0.2504	0.0169
015	0.2679	0.279	0.2577	0.2686	0.2661	0.2678	0.0068
020	0.2679	0.2763	0.2643	0.2628	0.2688	0.268	0.0047
030	0.2693	0.2644	0.2708	0.2633	0.2714	0.2679	0.0033
040	0.274	0.2702	0.2691	0.2682	0.279	0.2721	0.004

Table 4: MATLAB net-to-gross values

SLR [mm/yr]	Run 1	Run 2	Run 3	Run 4	Run 5	Avg.	Std. Dev.
000	0.2282	0.2225	0.2282	0.2285	0.254	0.2323	0.0111
005	0.2173	0.2114	0.2223	0.2186	0.2203	0.218	0.0037
010	0.2018	0.1928	0.2041	0.196	0.2108	0.2011	0.0063
015	0.1976	0.1945	0.1882	0.1911	0.1964	0.1936	0.0035
020	0.193	0.1826	0.1774	0.1905	0.2008	0.1889	0.0082
030	0.1782	0.1858	0.1851	0.1894	0.186	0.1849	0.0037
040	0.1878	0.1851	0.1893	0.1824	0.1814	0.1852	0.003

Table 5: Python net-to-gross values

5 Statistical Tests

5.1 Rank-Sum Tests

Results for the channelized fraction and shoreline roughness metrics are qualitatively stationary in time after the initial channel network is formed. The DeltaRCM channel network may form in as few as 75 years [1, 2, 6, 8]. Therefore, metric values between years 100 and 200 reflect properties of the fully developed distributary network. Values from the 5 replicates per SLR scenario will be lumped together for each model (MATLAB and Python), and used to perform a rank-sum test [14] with a null hypothesis, H_0 : the two sets of values are drawn from the same distribution.

SLR [mm/yr]	p-value
000	4.91e-23
005	4.37e-21
010	1.85e-08
015	2.00e-39
020	4.61e-28
030	1.38e-10
040	1.57e-24

Table 6: Channel Fraction Rank-Sum Results

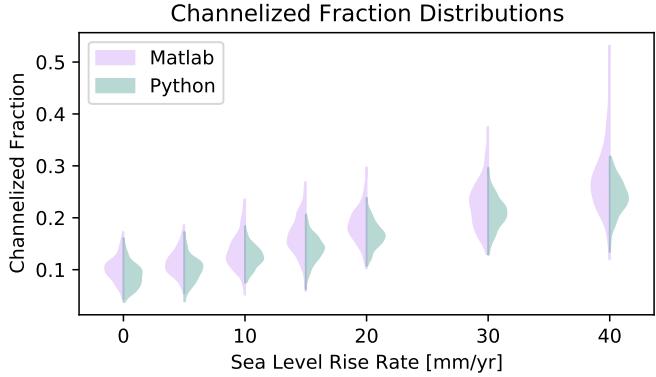


Figure 5: Smoothed distributions of channel fraction values (years 100-200).

SLR [mm/yr]	p-value
000	9.32e-05
005	4.96e-08
010	2.37e-09
015	4.26e-17
020	8.17e-23
030	4.03e-55
040	5.47e-33

Table 7: Shoreline Roughness Rank-Sum Results

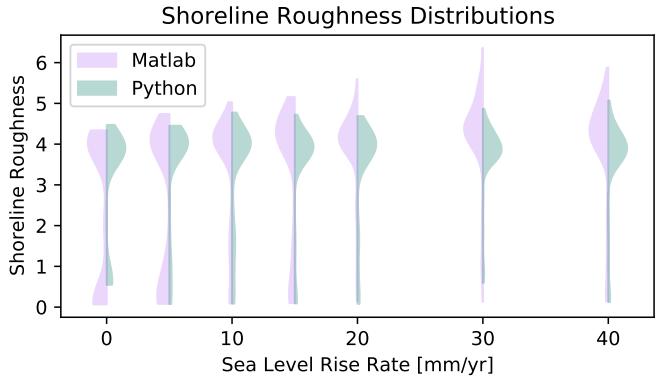


Figure 6: Smoothed distributions of shoreline roughness values (years 100-200).

6 Conclusions

Although model results look qualitatively very similar, there are some slight differences between the MATLAB and Python outputs. It is hard to know whether or not these differences are due to the differences in parameter values for `S0` and `itermax`, or whether they are due to actual code differences. It is harder still to know whether or not these differences matter and which model is generally more representative of true systems and processes.

On the surface, the channelized fraction (Figure 3) and shoreline roughness (Figure 4) appear to be similar between the models, however a rank-sum test performed comparing the values from years 100-200 (Tables 6 & 7), suggest that the underlying distributions are indeed different.

The surface dynamics, as measured using the channel planform overlap metric, provide evidence of similar trends between the two models (increasing mobility with increasing sea level rise rates). However, the python implementation appears to be more mobile than the MATLAB version (Tables 2 & 3, third column). This trend in particular may be influenced by the frequency at which model outputs were obtained, the MATLAB code output information about the system every 25 timesteps (~ 0.7 years) whereas the Python code was set to output information every 35 timesteps (~ 1 years).

Net-to-gross properties of the preserved deposits also present slightly perplexing results. The MATLAB results show a deposit that tends to be sandier than the input bedload fraction, with increasing sandiness with higher sea level rise rates (Table 4). On the other hand, the Python results show an opposite trend, with decreasing sandiness as the sea level rise rate is increased, and overall net-to-gross values at or slightly below the input bedload fraction (Table 4).

Appendices

A Topography Figures

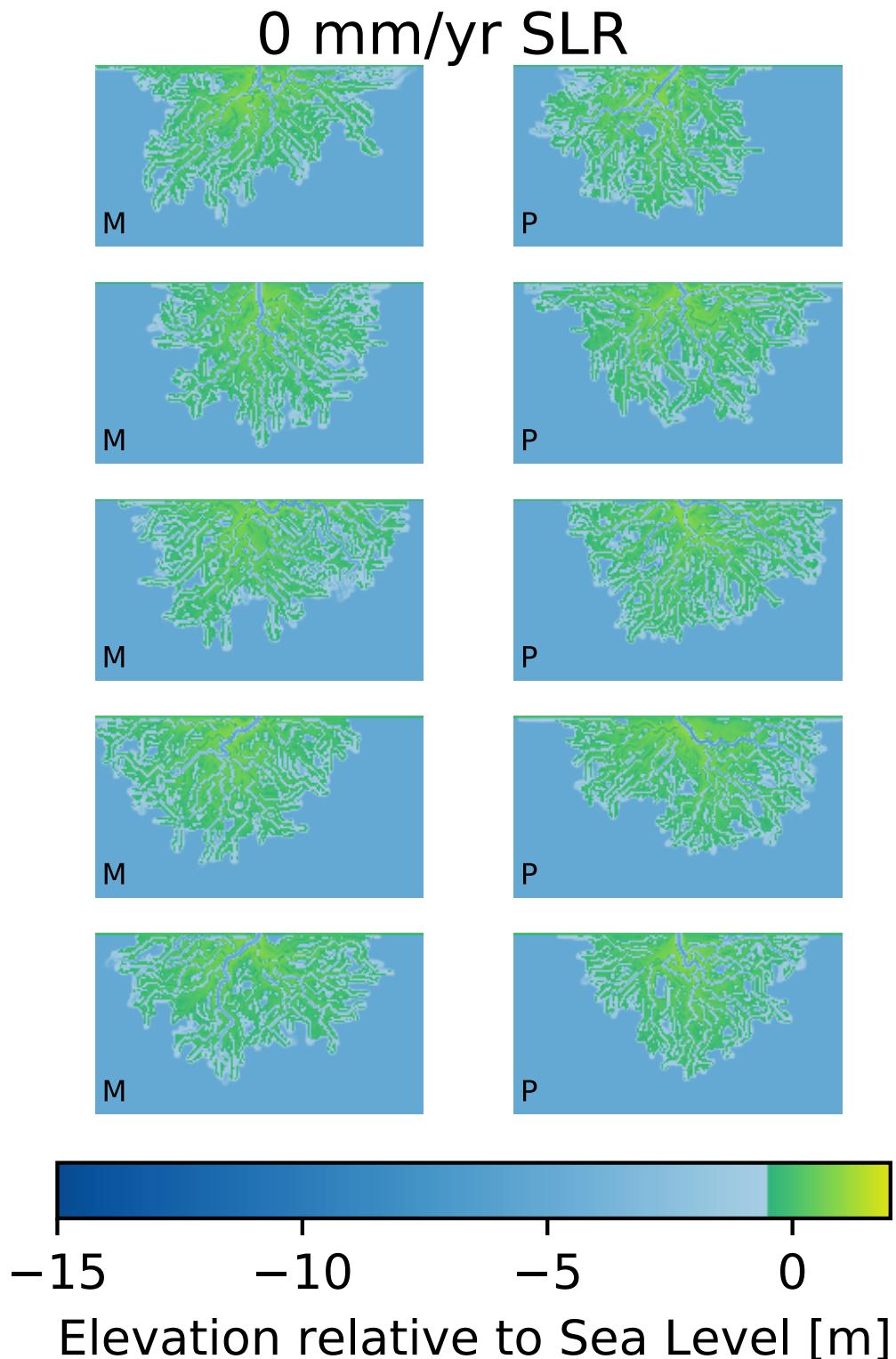


Figure 7: Final topographies for 0 mm/yr replicates. Domain has been clipped.

5 mm/yr SLR

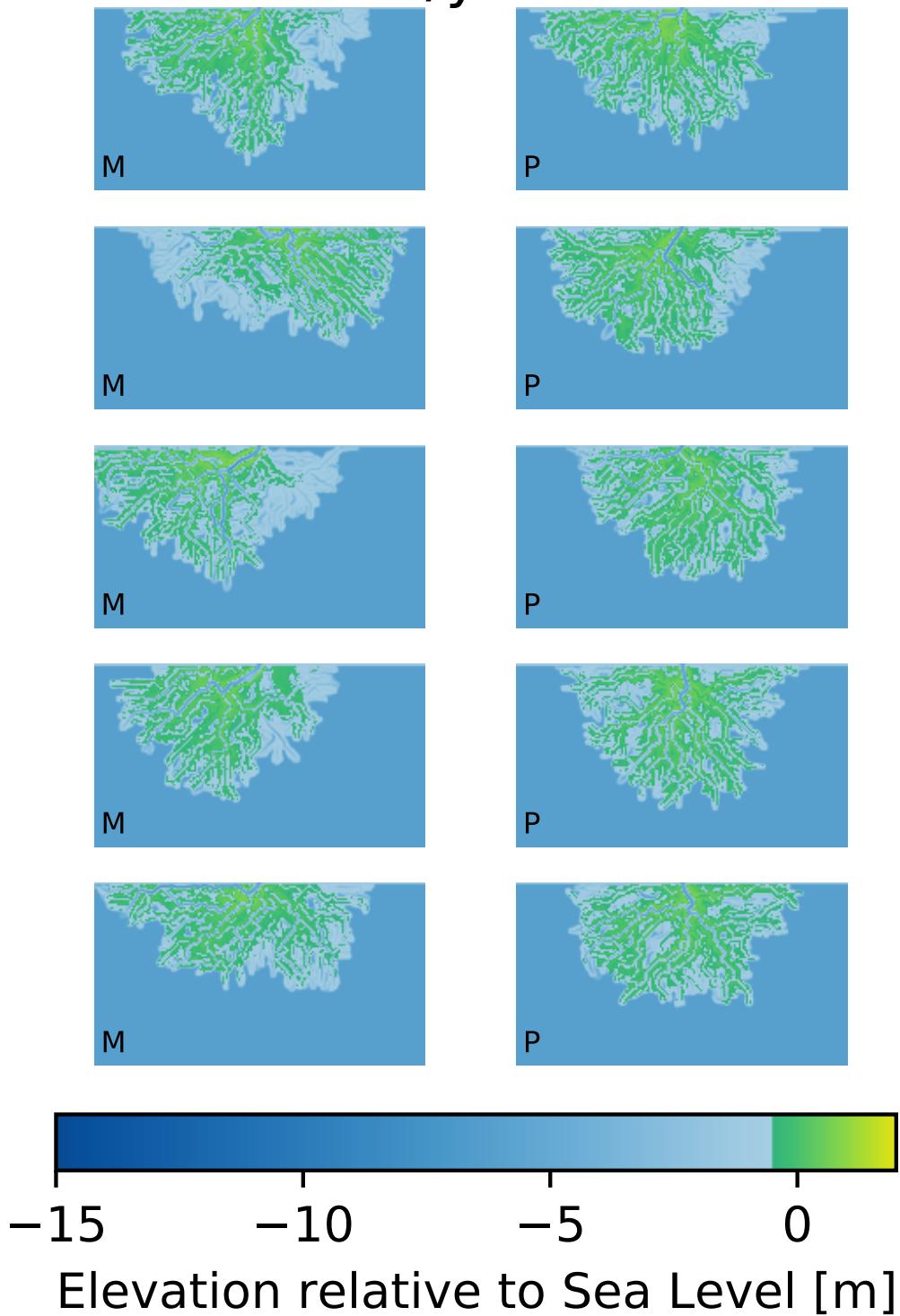


Figure 8: Final topographies for 5 mm/yr replicates. Domain has been clipped.

10 mm/yr SLR

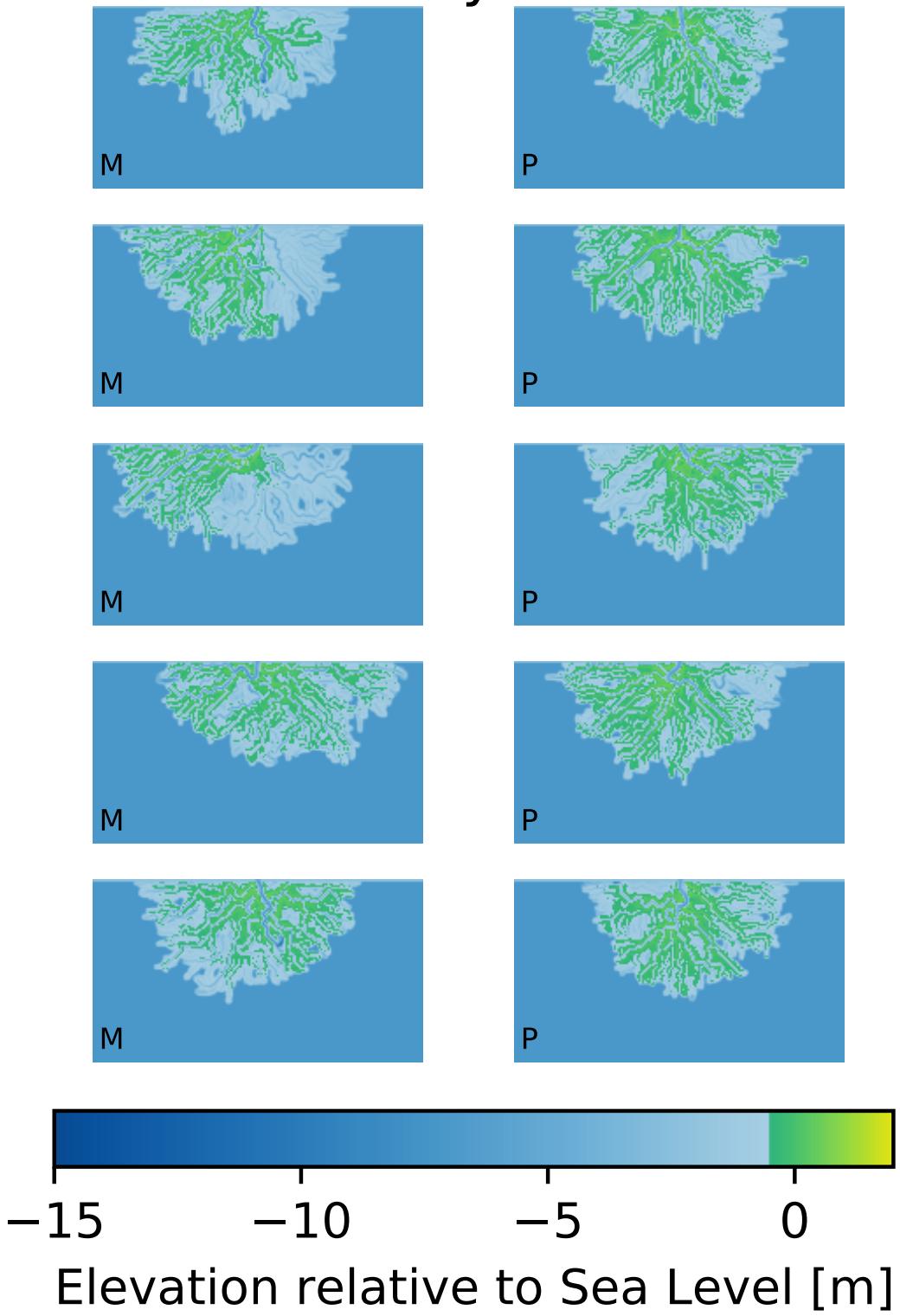


Figure 9: Final topographies for 10 mm/yr replicates. Domain has been clipped.

15 mm/yr SLR

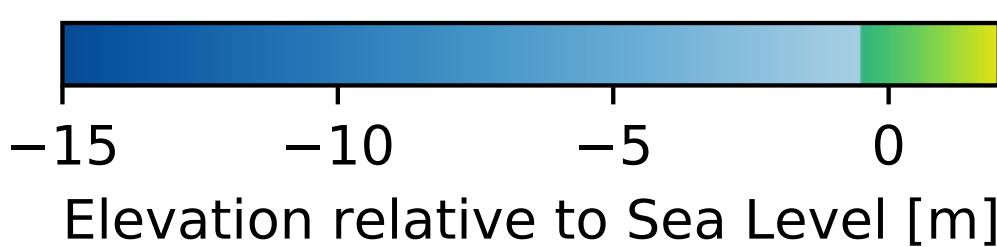
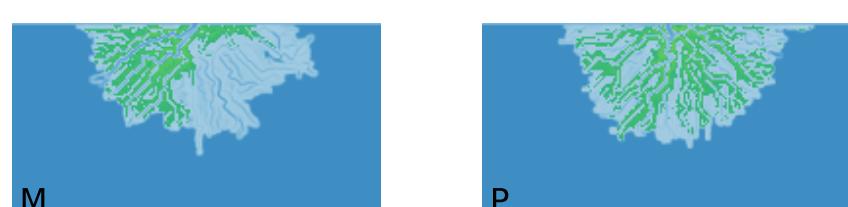
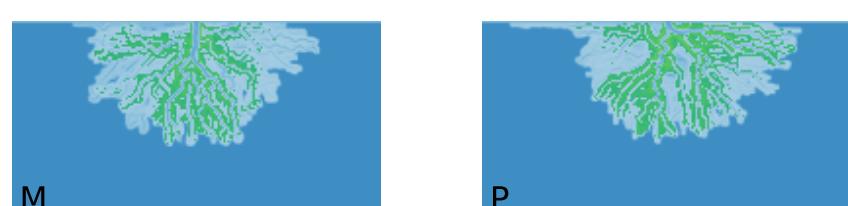
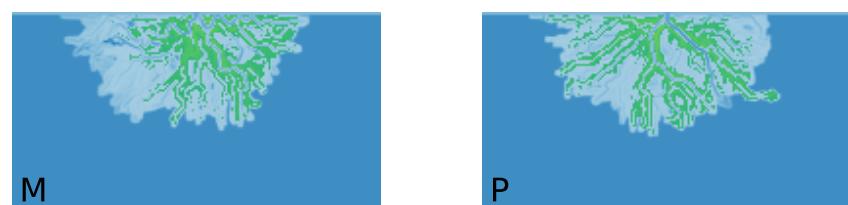
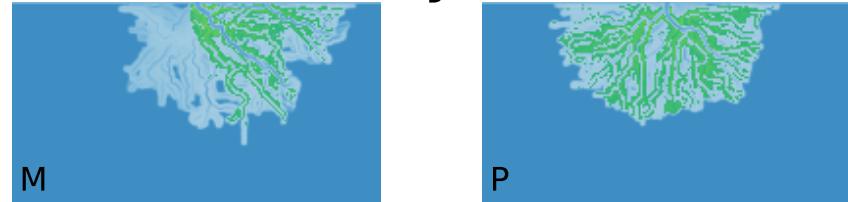


Figure 10: Final topographies for 15 mm/yr replicates. Domain has been clipped.

20 mm/yr SLR

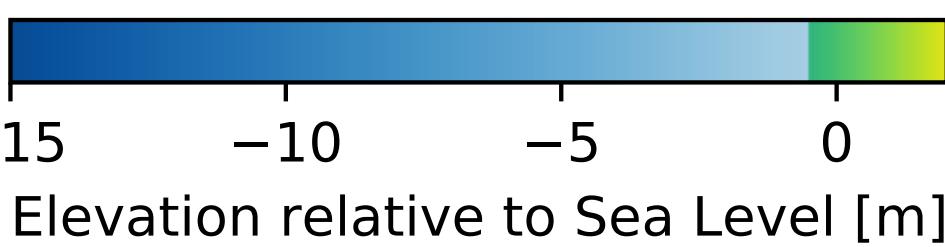
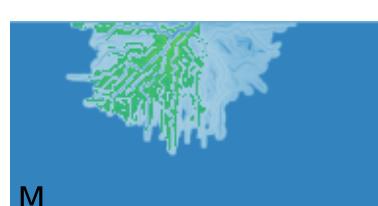
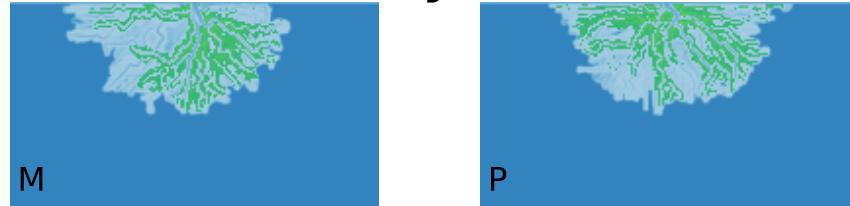


Figure 11: Final topographies for 20 mm/yr replicates. Domain has been clipped.

30 mm/yr SLR

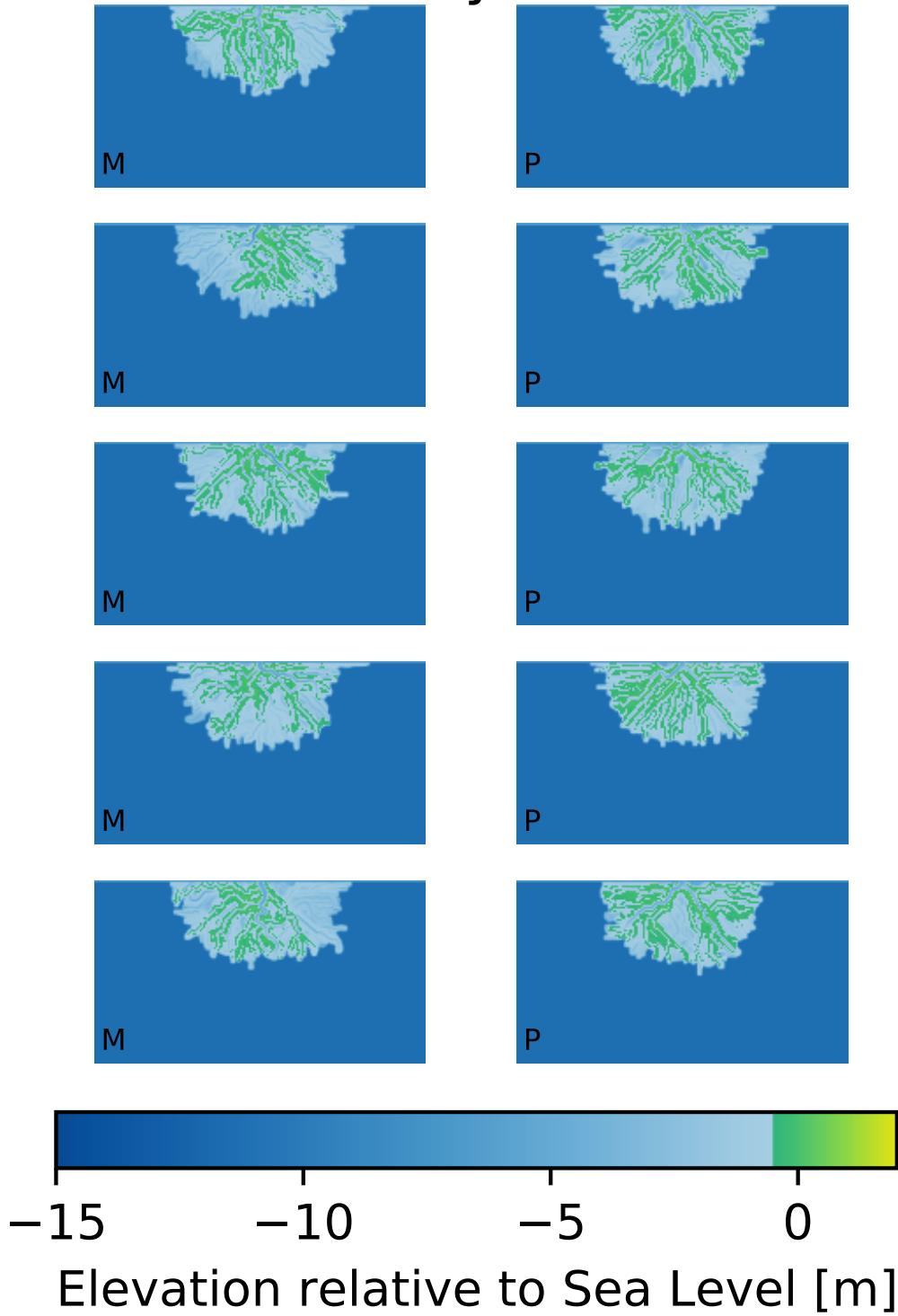


Figure 12: Final topographies for 30 mm/yr replicates. Domain has been clipped.

40 mm/yr SLR

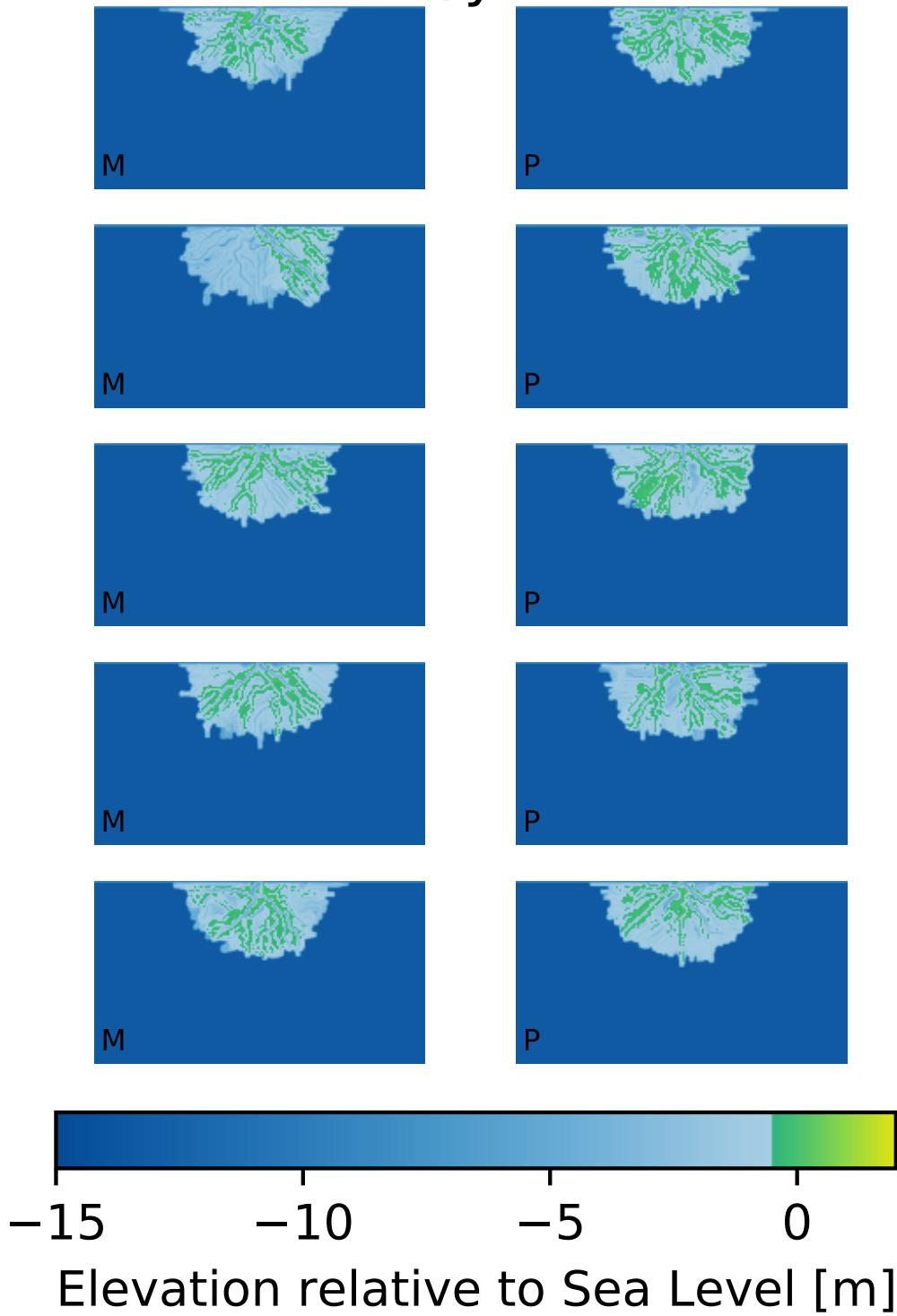


Figure 13: Final topographies for 40 mm/yr replicates. Domain has been clipped.

References

- [1] R. Lauzon and A. B. Murray. Comparing the Cohesive Effects of Mud and Vegetation on Delta Evolution. *Geophysical Research Letters*, 45(19):10,437–10,445, oct 2018. ISSN 19448007. doi: 10.1029/2018GL079405. URL <https://onlinelibrary.wiley.com/doi/abs/10.1029/2018GL079405>.
- [2] R. Lauzon, A. Piliouras, and J. C. Rowland. Ice and Permafrost Effects on Delta Morphology and Channel Dynamics. *Geophysical Research Letters*, 46(12):6574–6582, jun 2019. ISSN 19448007. doi: 10.1029/2019GL082792. URL <https://onlinelibrary.wiley.com/doi/abs/10.1029/2019GL082792>.
- [3] M. Liang, N. Geleynse, D. A. Edmonds, and P. Passalacqua. A reduced-complexity model for river delta formation - Part 2: Assessment of the flow routing scheme. *Earth Surface Dynamics*, 3(1):87–104, 2015. ISSN 2196632X. doi: 10.5194/esurf-3-87-2015. URL www.earth-surf-dynam.net/3/87/2015/.
- [4] M. Liang, V. R. Voller, and C. Paola. A reduced-complexity model for river delta formation - Part 1: Modeling deltas with channel dynamics. *Earth Surface Dynamics*, 3(1):67–86, 2015. ISSN 2196632X. doi: 10.5194/esurf-3-67-2015. URL www.earth-surf-dynam.net/3/67/2015/.
- [5] M. Liang, W. Kim, and P. Passalacqua. How much subsidence is enough to change the morphology of river deltas? *Geophysical Research Letters*, 43(19):10,266–10,276, oct 2016. ISSN 19448007. doi: 10.1002/2016GL070519. URL <http://doi.wiley.com/10.1002/2016GL070519>.
- [6] M. Liang, C. Van Dyk, and P. Passalacqua. 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, feb 2016. ISSN 21699003. doi: 10.1002/2015JF003653. URL <http://doi.wiley.com/10.1002/2015JF003653>.
- [7] MATLAB version 9.7.0.1296695 (R2019b) Update 4. The Mathworks, Inc., Natick, Massachusetts, 2019.
- [8] A. Piliouras, R. Lauzon, and J. C. Rowland. Unraveling the Combined Effects of Ice and Permafrost on Arctic Delta Morphodynamics. *Journal of Geophysical Research: Earth Surface*, 126(4):e2020JF005706, apr 2021. ISSN 2169-9003. doi: 10.1029/2020jf005706. URL <https://onlinelibrary.wiley.com/doi/10.1029/2020JF005706>.
- [9] Python Software Foundation. Welcome to Python.org, 2016. URL <https://www.python.org/>.
- [10] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [11] J. B. Shaw, M. A. Wolinsky, C. Paola, and V. R. Voller. An image-based method for shoreline mapping on complex coasts. *Geophysical Research Letters*, 35(12):n/a–n/a, jun 2008. ISSN 00948276. doi: 10.1029/2008GL033963. URL <http://doi.wiley.com/10.1029/2008GL033963>.
- [12] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL <https://doi.org/10.7717/peerj.453>.
- [13] A. D. Wickert, J. M. Martin, M. Tal, W. Kim, B. Sheets, and C. Paola. River channel lateral mobility: Metrics, time scales, and controls. *Journal of Geophysical Research: Earth Surface*, 118(2):396–412, apr 2013. ISSN 21699011. doi: 10.1029/2012JF002386. URL <https://doi.org/10.1029/2012JF002386>.
- [14] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. ISSN 00994987. URL <http://www.jstor.org/stable/3001968>.
- [15] T. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984.