2. Methods

2.1. Model design

We modeled the vertical accretion of a tidal platform ($d\eta/dt$) using a zero-dimensional mass balance approach initially described by Krone (1987) and validated by subsequent studies (Allen, 1990; French, 1993; Temmerman et al., 2003, 2004). The rate of vertical accretion is described as

$$\frac{d\eta(t)}{dt} = \frac{dS_m(t)}{dt} + \frac{dS_o(t)}{dt} + \frac{dP(t)}{dt} + \frac{dM(t)}{dt}, \tag{1}$$

where $dS_m(t)/dt$ is the rate of mineral sedimentation, $dS_o(t)/dt$ is the rate of organic matter sedimentation, $dP(t)/dt$ is the rate of compaction of the deposited sediment, and $dM(t)/dt$ is the rate of tectonic subsidence. Studies have shown only ~3 % of bulk sedimentation in the region comes from organic matter and therefore we chose to set $dS_o(t)/dt$ to zero. We also effectively internalized compaction by using dry bulk density within the mineral sedimentation term so we also set $dP(t)/dt$ to zero. Lastly, we set the $dM(t)/dt$ to 6 mm which is consistent with measured rates of tectonic subsidence for the region Higgins et al. (2014) . The remaining mineral sedimentation terms varies within a tidal cycle and requires additional steps.

Need ref. Could also use a proportion of total sediment accumulation. Maybe unncessary. It likely gets internalized after model calibration. Maybe just combine S_m and S_o?

In order to solve $dS_m(t)/dt$, we began by conceptualizing a tidal platform periodically inundated by cyclical tides. We first defined depth to be

$$h(t) = \zeta(t) - \eta(t), \tag{2}$$

where $\zeta(t)$ is the water-surface elevation and $\eta(t)$ is the sediment-surface elevation which also implies that

$$\frac{dh(t)}{dt} = \frac{d\zeta(t)}{dt} - \frac{d\eta(t)}{dt}. \tag{3}$$

Independently, we assume when $h(t) > 0$, the rate of mineral sedimentation is

$$\frac{dS_m(t)}{dt} = \frac{w_s C(t)}{\rho_b}, \tag{4}$$

where w_s is the nominal settling velocity of a sediment grain, $C(t)$ is the depth-averaged suspended sediment concentration (SSC) in the water column, and ρ_b is the bulk density of the sediment. We assumed no resuspension of mineral sediment which is practical and consistent with previous studies (Krone, 1987; Allen, 1990; French, 1993; Temmerman et al., 2003, 2004). We used Stoke's law to determine w_s. Stoke's law assumes unhindered settling which likely overestimates actual settling rates and, therefore, mineral sedimentation rates. However, we only considered settling for a singular, median grain size which likely underestimated mineral sedimentation rates from coarser grains. Model calibration further corrected for these errors. Thus, the w_s given by Stoke's law should be considered an imprecise, but reasonable approximation.

Need to investigate further. Look into Steckler papers from 2010 and 2013.

Need ref.

In order to solve for $C(t)$ in eq. (4), we first defined a mass balance of sediment within the water column as

$$\frac{d}{dt}[h(t)C(t)] = -w_s C(t) + C_b \frac{dh(t)}{dt}, \tag{5}$$

which can be expanded and rerranged as

$$\frac{dC(t)}{dt} = -\frac{w_s C(t)}{h(t)} - \frac{1}{h(t)}[C(t) - C_b]\frac{dh(t)}{dt}. \tag{6}$$

The mass flux from the boundary term was constrained to only occur during flood tide ($d\zeta/dt > 0$) which is consistent with previous studies (Krone, 1987; Allen, 1990; French, 1993; Temmerman et al., 2003, 2004). We formalized this mathematically using a Heaviside function which serves as a binary switch controlling the flux of sediment from the boundary. The Heaviside function is given as

$$S = \frac{d\zeta}{dt}, H(S) = \begin{cases} 0 & \text{if } S < 0 \\ 1 & \text{if } S \geq 0. \end{cases} \tag{7}$$

Eq. (6) then becomes

$$\frac{dC(t)}{dt} = -\frac{w_s C(t)}{h(t)} - \frac{H(S)}{h(t)}[C(t) - C_b]\frac{dh(t)}{dt}. \tag{8}$$

Finally, we solved eqs. (8), (4) and (1) in that order. Our apporoach differs from previous studies in that we resolve each equation at every timestep. We do this by using a computationally efficient Runge-Kutta method which allows for a variable time step. The system of equations was solved using an explicit Runge-Kutta method of order 5(4) (Dormand and Prunce, 1980) and implemented in Python using SciPy (Virtanen et al., 2020).

Software and/or data availability

Acknowledgements

References

Allen, J.R.L., 1990. Salt-marsh growth and stratification: A numerical model with special reference to the Severn Estuary, southwest Britain. Marine Geology 95, 77–96. doi:10/dpcnnc.

Dormand, J., Prunce, P., 1980. A family of embedded Runge-Kutta formulae. Journal of Computational and Applied Mathematics 6, 19–26. doi:10/cfw5fc.

French, J.R., 1993. Numerical simulation of vertical marsh growth and adjustment to accelerated sea-level rise, North Norfolk, U.K. Earth Surface Processes and Landforms 18, 63–81. doi:10/drw2f8.

Higgins, S.A., Overeem, I., Steckler, M.S., Syvitski, J.P.M., Seeber, L., Akhter, S.H., 2014. InSAR measurements of compaction and subsidence in the Ganges-Brahmaputra Delta, Bangladesh. Journal of Geophysical Research: Earth Surface 119, 1768–1781. doi:10/ghjtd3.

Krone, R., 1987. A Method for Simulating Marsh Elevations, in: Coastal Sediments, American Society of Civil Engineers, New Orleans, Louisiana. pp. 316–323.

Temmerman, S., Govers, G., Meire, P., Wartel, S., 2003. Modelling long-term tidal marsh growth under changing tidal conditions and suspended sediment concentrations, Scheldt estuary, Belgium. Marine Geology 193, 151–169. doi:10/dqg2zs.

Temmerman, S., Govers, G., Wartel, S., Meire, P., 2004. Modelling estuarine variations in tidal marsh sedimentation: Response to changing sea level and suspended sediment concentrations. Marine Geology 212, 1–19. doi:10/dsjcj6.

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., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. doi:10/ggj45f.