Tidalbar.jl

T. M. Hepkema

January 14, 2021

1 Introduction

Below follows a short description of the TidalBar code. See Chapter 5 of my PhD thesis for the model equations, parameters and discretizations. The code is written in pure Julia. The model is a local morphodynamic model to simulate the dynamics of patterns emerging in a sandy bottom of a tidal channel when the tides runs through it.
It provides:

1. Hydrodynamical time integrator. That is, given a bottom pattern, it calculates the currents, sediment transport and free surface elevation during one (or more) tidal cycles. Time integrators can be selected from DifferentialEquations.jl. Default is Runge-Kutta 4 with fixed time steps of 5 seconds.
2. Morphodynamical time integrator. Using the hydrodynamical time integrator it integrates the bottom in time. Time integrators can again be chosen from DifferentialEquations.jl. Default is Euler Forward with time steps of 1 week.
3. Morphodynamic equilibrium finder using the Newton-Raphson method (BifurcationKit.jl)
4. Jacobian at equilibria using Automatic Differentiation (ForwardDiff.jl)

GitHub repositories, documentations and tutorials of the different packages:

- DifferentialEquations.jl:
  - `https://github.com/SciML/DifferentialEquations.jl`
  - `https://diffeq.sciml.ai/stable/`
- BifurcationKit.jl:
  - `https://github.com/rveltz/BifurcationKit.jl`
  - `https://rveltz.github.io/BifurcationKit.jl/dev/`
- ForwardDiff.jl:
  - `https://github.com/JuliaDiff/ForwardDiff.jl`
  - `http://www.juliadiff.org/ForwardDiff.jl/stable/`
- Parameters.jl:
  - `https://github.com/mauro3/Parameters.jl`
  - `https://mauro3.github.io/Parameters.jl/stable/`

2 Installation

Assuming Julia is installed. Go to the TidalBar folder and:

- in pkg: `activate .` (incl. the 'dot' referring to the current directory)
- in pkg: `instantiate`
- in pkg: `precompile` (takes a while)

If you don't know what happens here, read:
`https://docs.julialang.org/en/v1.0/stdlib/Pkg/`
The code makes use of unicode characters. If they are not properly displayed in your set-up,

- use VScode (`https://code.visualstudio.com/download`) as editor (with the Julia extension)
- install/set JuliaMono (`https://github.com/cormullion/juliamono`) as font.

3 Overview of files

3.1 src/

swec.jl

Contains the right hand side for the ODEs for ζ, u, v, and C and the function `swec()` that does the time integration.

morphodynamics.jl

Contains the right hand side for the ODEs for h and the function `bed_evolution()` that does the time integration.

hydrodynamical_terms.jl

Contains functions to calculate the terms in right hand side of the ODEs of ζ, u and v.

sediment_terms.jl

Contains functions to calculate the terms in right hand side of the ODEs of C and additional (bed load) sediment transport terms.

params.jl

Contains struct (and constructor) with model parameters, check this one for the defaults and the different options.

grid.jl

Contains struct (and constructor) for the grid.

initial_bottomheights.jl

Contains functions to initialize the bottom topography with, for example, a gaussian bump or $\cos(kx)\cos(ny)$ structure.

allocate_arrays.jl

Struct (and constructor) with all the allocated arrays.

utils.jl

Contains helper functions, e.g., calculating derivatives at different grids or printing the progress.

read_output.jl

Contains functions to read output.

write_output.jl

Contains functions to write output.

3.2 examples/

main_hydro.jl

Example file of how to run a hydrodynamical run given a certain bottom pattern.

main_morpho.jl

Example file of how to run a morphodynamical time integration run.

visualize_swec.jl

Example file of how to read output of hydrodynamical run.

continuation_in_B.jl

Messy, buggy and not optimized example of how to do a continuation in channel width. I advise to use this only as inspiration/proof of concept and write you're own optimized version. It also provides a hint at how to use the pseudo-arclength continuation from BifurcationKit.jl