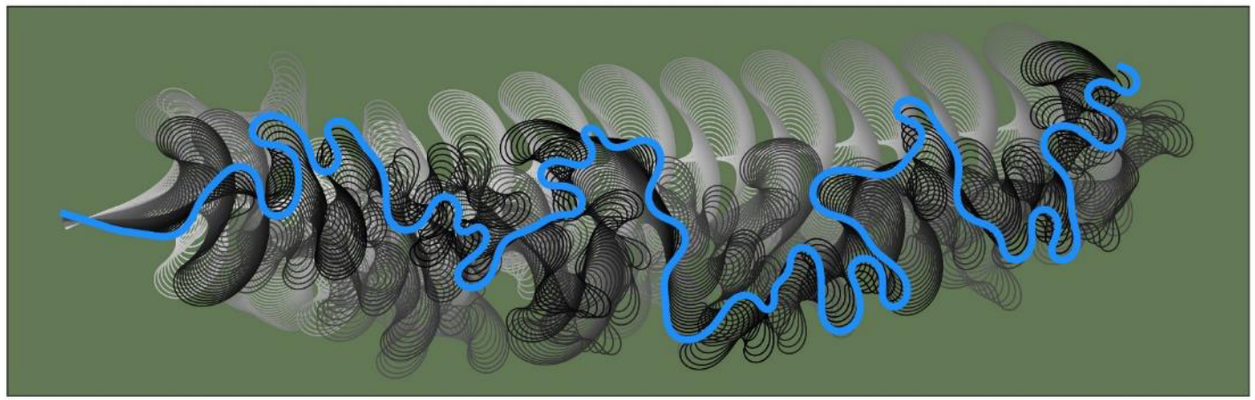
CSDMS Summer Science Series – June 30th, 2020

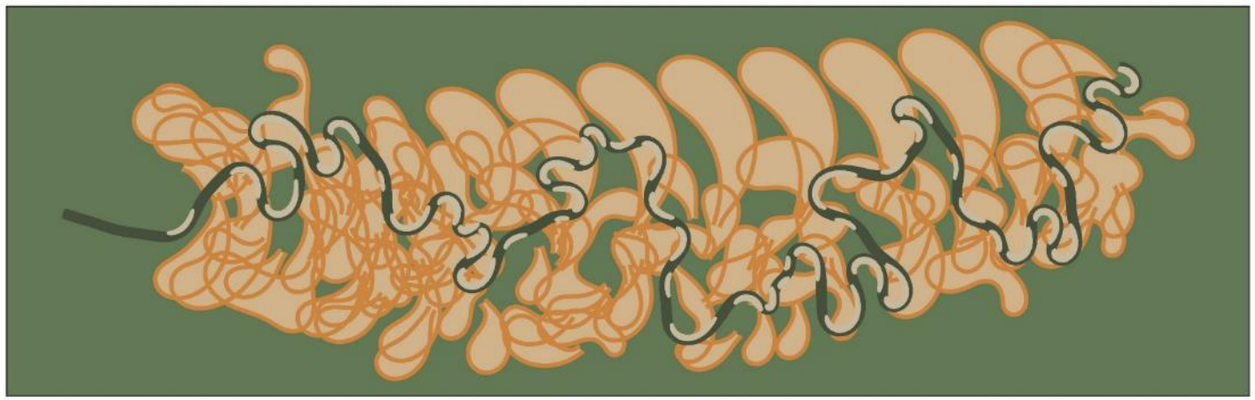
# pyRiverBed: A Python framework to generate synthetic riverbed topography of constant-width meandering rivers

(based on the manuscript recently submitted to *Computers & Geosciences*)

Zhi Li
Ph.D. candidate, Graduate Research Assistant
Ven Te Chow Hydrosystems Laboratory
Dept. of Civil and Environmental Engineering
Grainger College of Engineering
University of Illinois at Urbana-Champaign

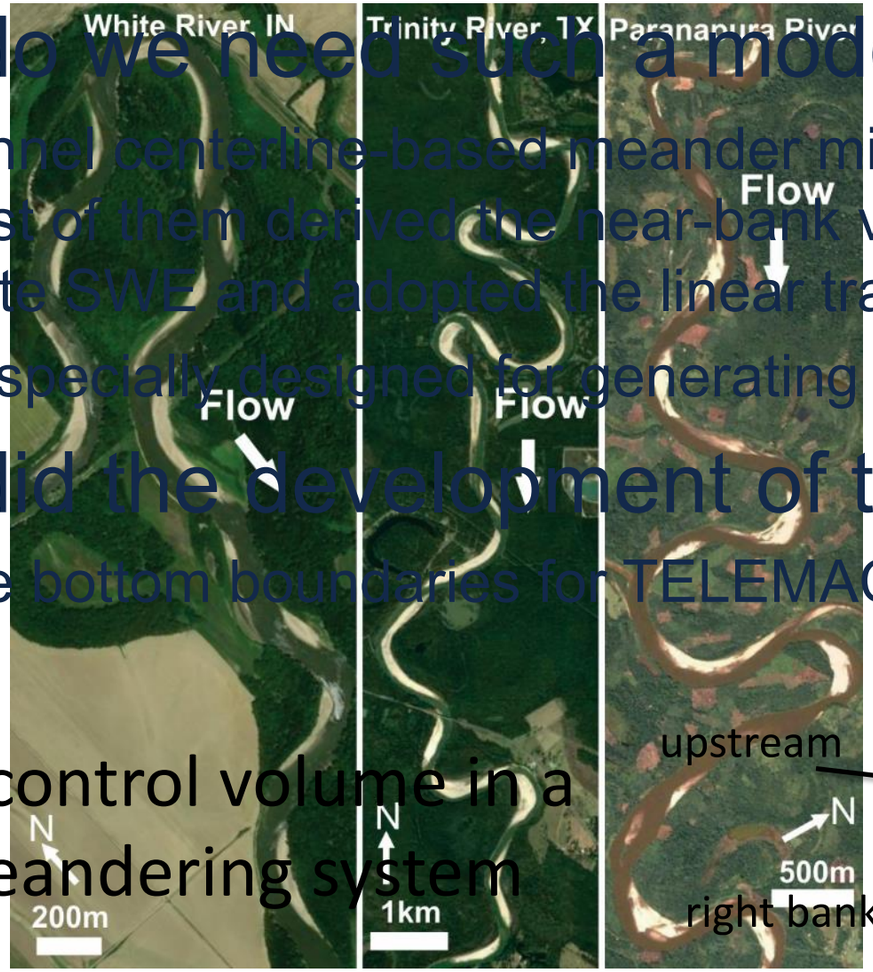
Advisor: Prof. Marcelo H. García



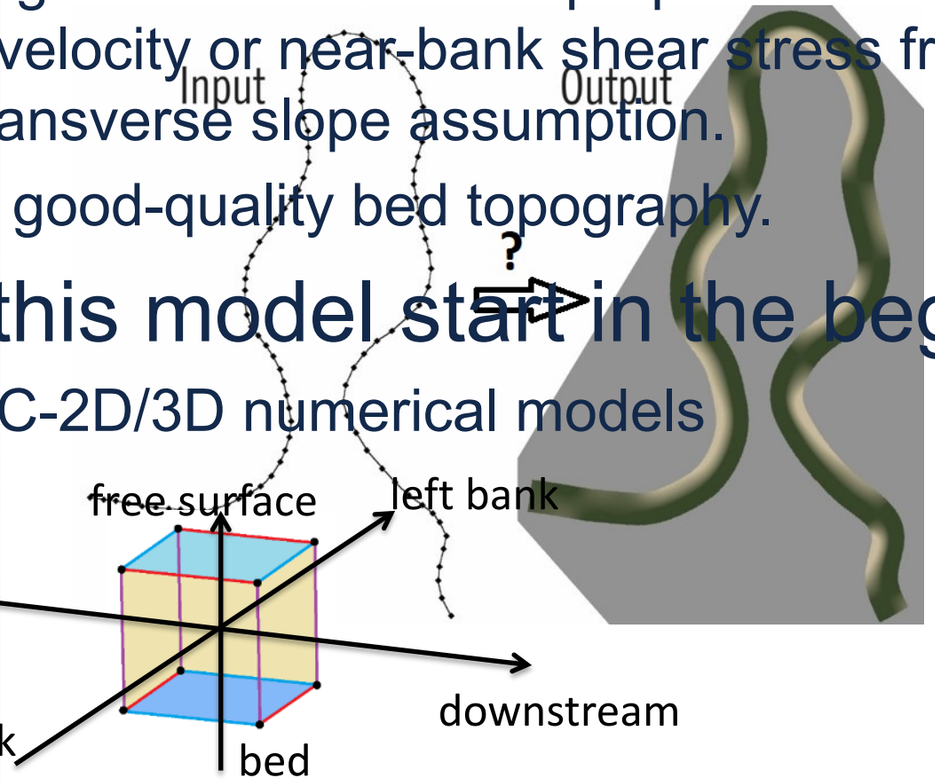


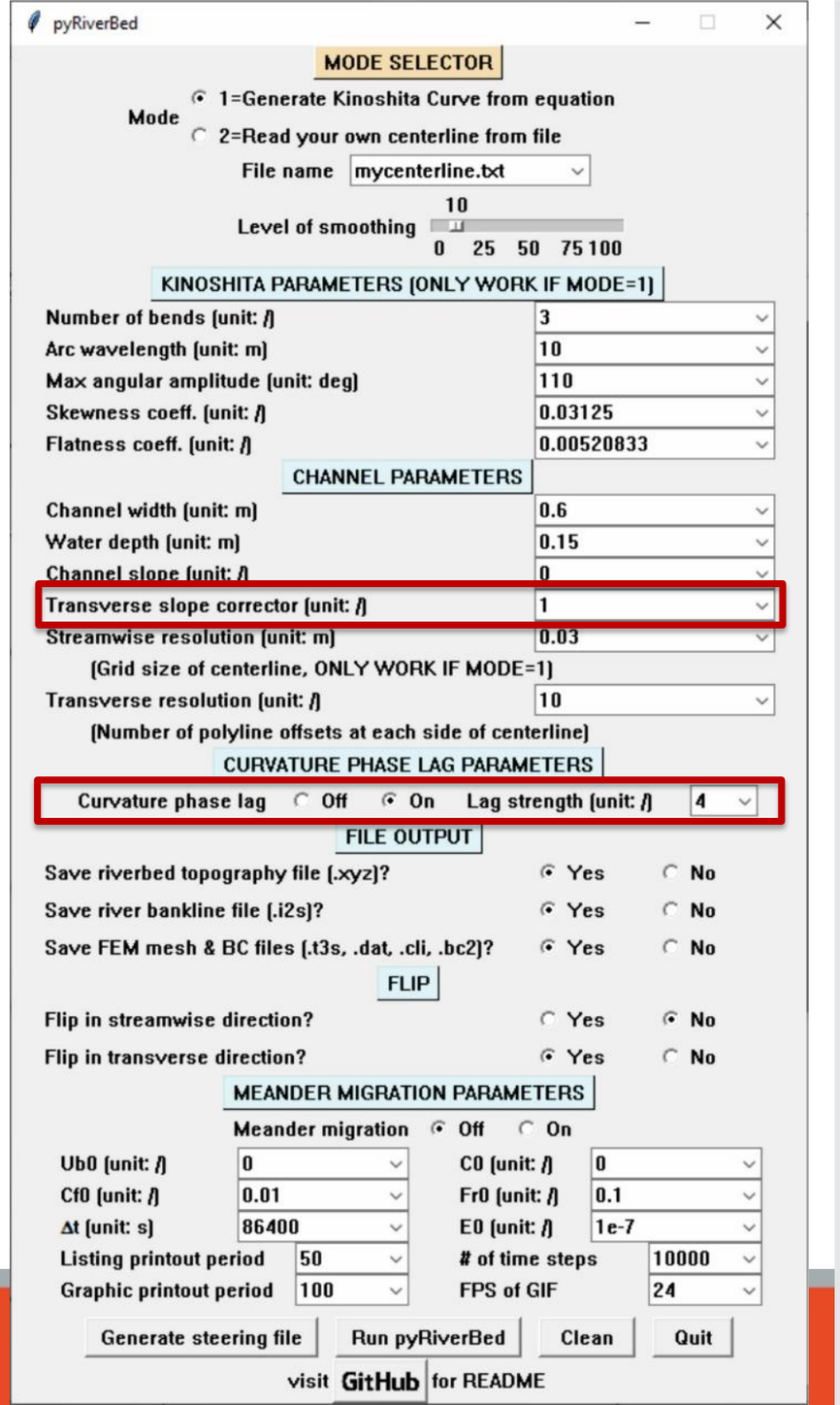
# Motivations

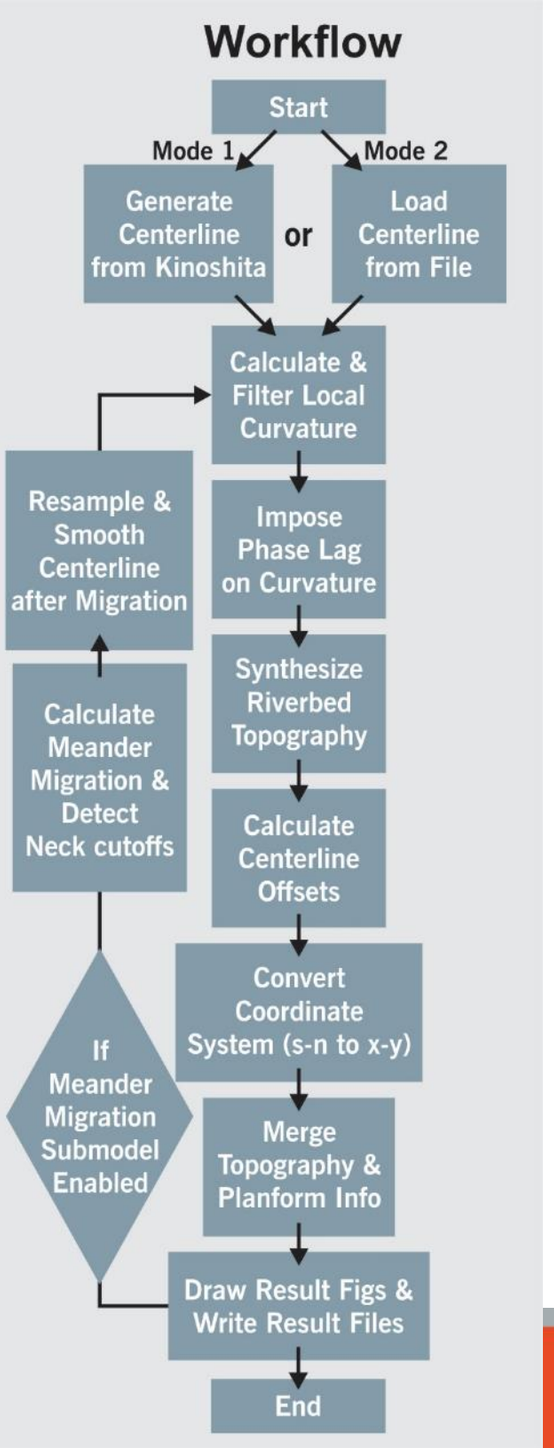
- Why do we need such a model
  - Many channel centerline-based meander migration models were proposed in the past 50 years. Most of them derived the near-bank velocity or near-bank shear stress from the steady state SWE and adopted the linear transverse slope assumption.
  - No one is specially designed for generating good-quality bed topography.
- Why did the development of this model start in the beginning
  - To create bottom boundaries for TELEMAC-2D/3D numerical models




A control volume in a meandering system






```
+-----------------------------------------------------------+
                    - pyRiverBed -
 Generate Synthetic Riverbed Topography for Meandering Rivers
                        MIT License
          Zhi Li, Univ. of Illinois Urbana-Champaign
                    zhil2[at]illinois[dot]edu
+-----------------------------------------------------------+

   MODE 1: GENERATE KINOSHITA CURVE FROM EQUATION
   MODE 2: READ YOUR OWN RIVER CENTERLINE FROM FILE

+> Trying to read steering file... [done]
MODE 1: GENERATE KINOSHITA CURVE FROM EQUATION is selected
Kinoshita Curve parameters are read from steering file:
Eqn: θ=1.919862*sin(2πs/10.0)
      +7.076364*[0.03125*cos(6πs/10.0)-0.005208*sin(6πs/10.0)]
+----------------------+---------+--------+
|            Parameter |   Value |   Unit |
|----------------------+---------+--------|
|      Number of bends |       3 |      / |
|                Width |     0.6 |      m |
|                Depth |    0.15 |      m |
|               Length |      40 |      m |
|        Arc wavelength |      10 |      m |
|                Slope |       0 |      / |
| Streamwise resolution |    0.03 |      m |
| Transverse resolution |    0.03 |      m |
|   Streamwise # of pts |    1333 |      / |
|   Transverse # of pts |      21 |      / |
+----------------------+---------+--------+
+> Calculating Kinoshita Curve... [done]
+> Extending centerline to have straight channels at both ends..
+> Filtering curvature (5-pt Savitzky-Golay + 5-pt Moving Averag
+> Adding phase lag to curvature... [done]
+> Calculating synthetic riverbed topography... [done]
+> Offsetting Polyline #10 & #12...(innermost) [done]
+> Offsetting Polyline #9 & #13... [done]
+> Offsetting Polyline #8 & #14... [done]
+> Offsetting Polyline #7 & #15... [done]
+> Offsetting Polyline #6 & #16... [done]
+> Offsetting Polyline #5 & #17... [done]
+> Offsetting Polyline #4 & #18... [done]
+> Offsetting Polyline #3 & #19... [done]
+> Offsetting Polyline #2 & #20... [done]
+> Offsetting Polyline #1 & #21...(outermost) [done]
   * Note: Polyline #11 is centerline
+> Saving riverbed topography file... [done]
+> Saving finite element mesh files... [done]
+> Plotting... [done]
+> My job is done
```

# Inputs

## Mode 1

File needed:

*none*

All parameters specified in the GUI

## Mode 2

File needed:

*[An ASCII file with centerline coordinates in 2-column* **`XY`** *format]*

*Example:*

| | | |
|---|---|---|
| *line1* | *678.25* | *2221.07* |
| *line2* | *678.54* | *2220.13* |
| *line3* | *678.88* | *2219.15* |
| *line4* | *679.29* | *2218.21* |
| *line5* | *679.76* | *2217.26* |
| | ⋮ | ⋮ |
| *lastline* | *682.24* | *2210.21* |

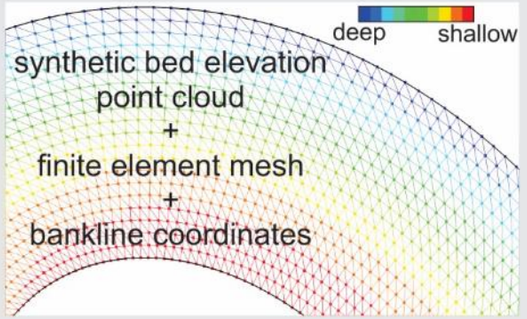
All other parameters specified in the GUI

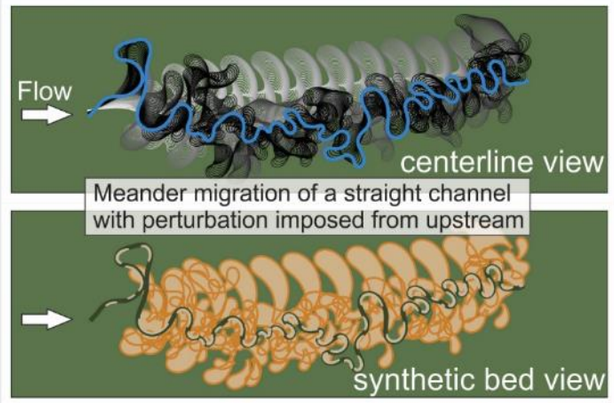
# Outputs

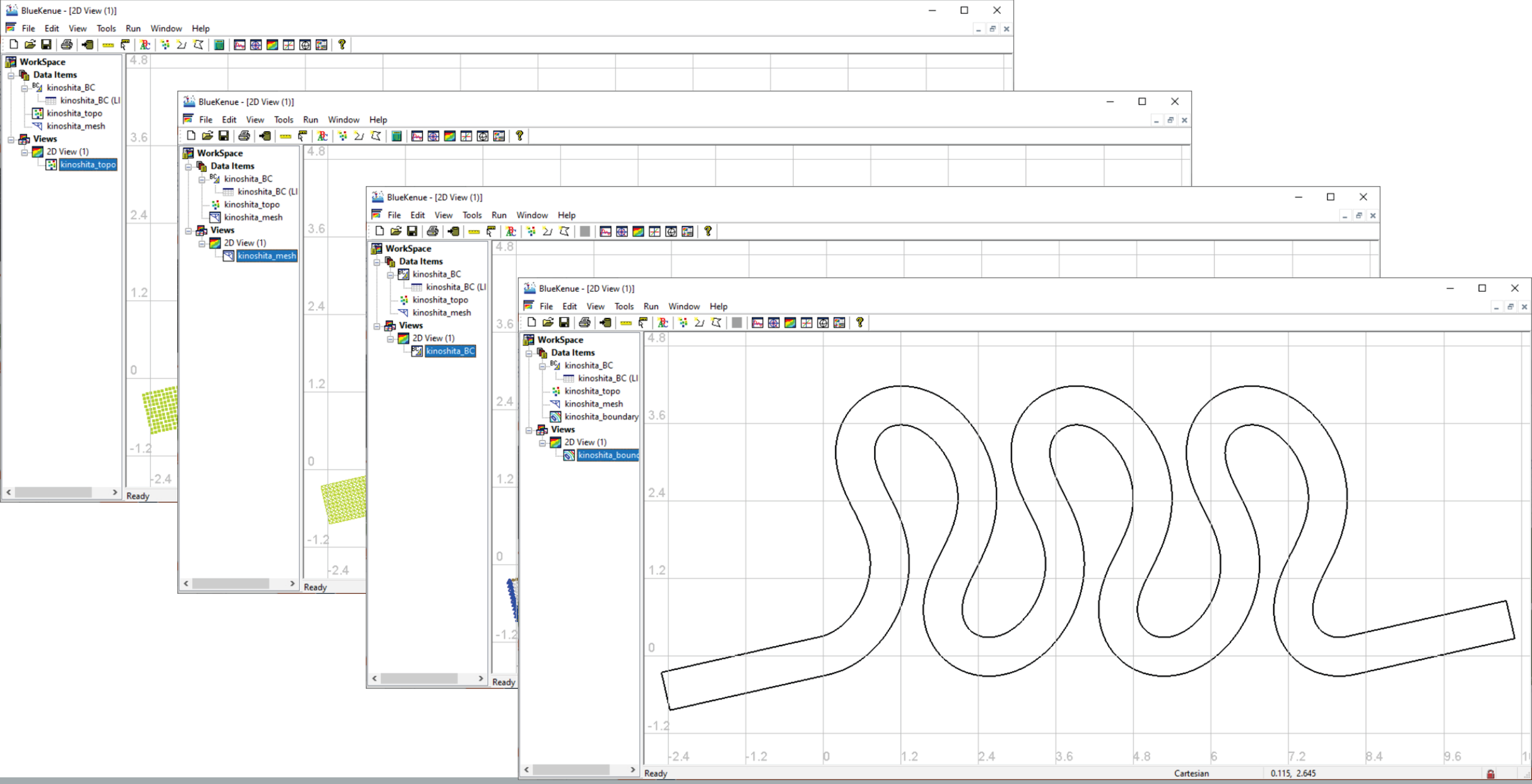
- Result figure of synthetic bed in x-y & s-n coordinates and local & lagged curvature in **`JPG`** & **`PDF`** format.
- Synthetic bed elev. point cloud in **`XYZ`** format.
- River bankline file in **`I2S`** format.
- Finite element mesh file in **`T3S`** format.
- Boundary condition file in **`CLI`** format.

If meander migration enabled:

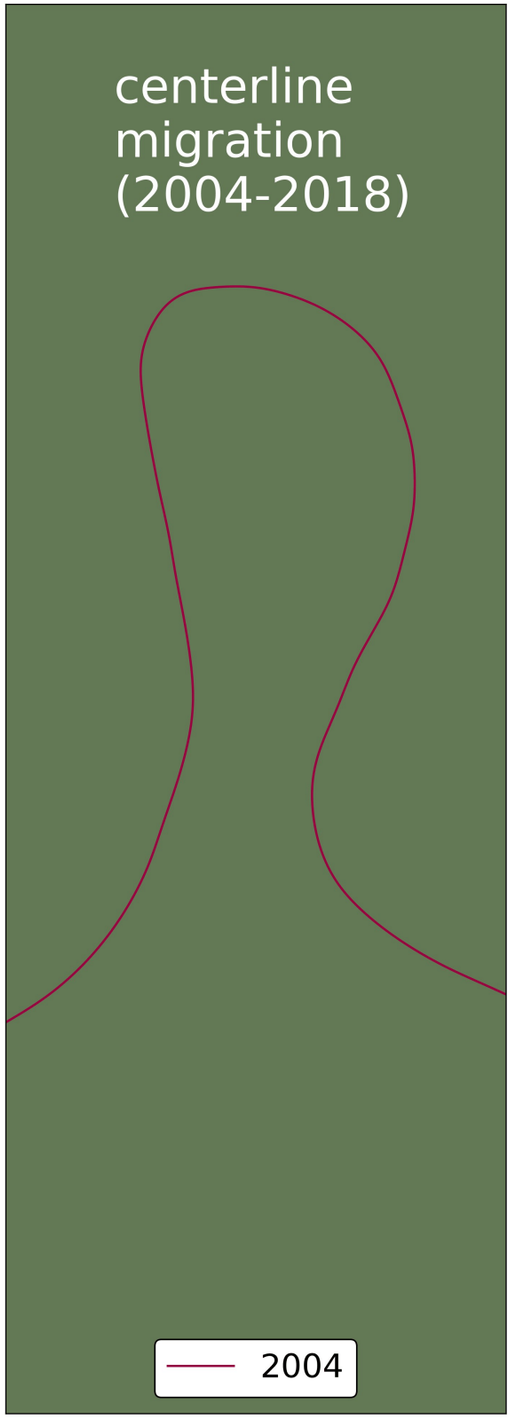
- Result figures of each snapshot during migration and animation files in **`GIF`** format.

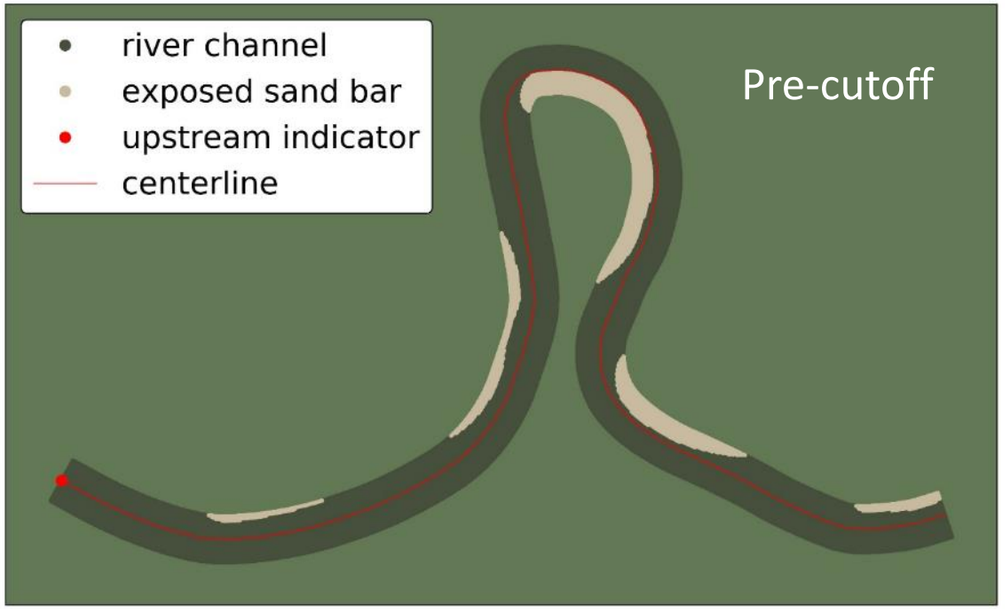


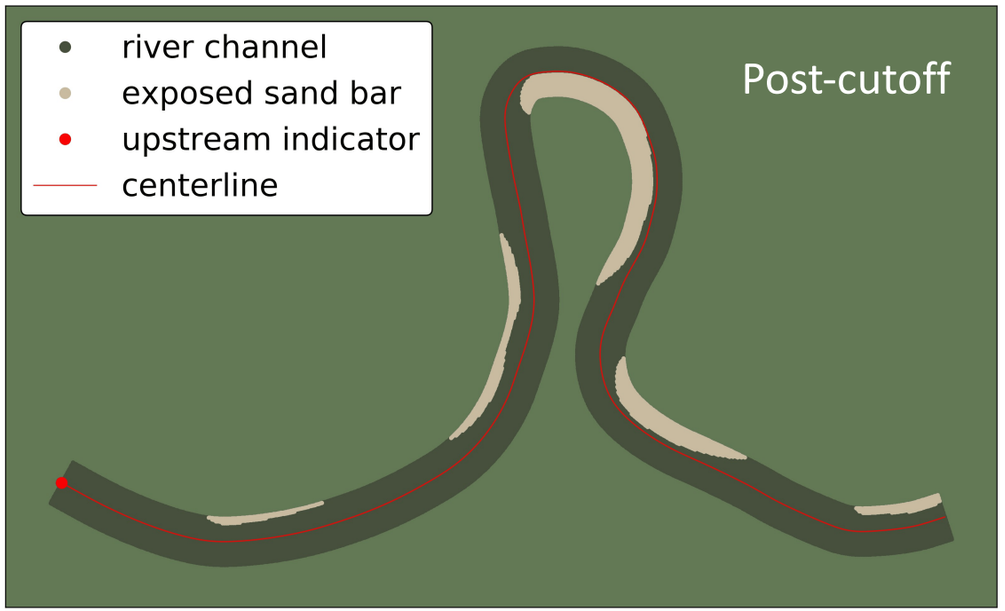




BlueKenue - [2D View (1)]
File Edit View Tools Run Window Help
WorkSpace
Data Items
kinoshita_BC
kinoshita_BC (LI
kinoshita_topo
kinoshita_mesh
kinoshita_boundary
Views
2D View (1)
Ready
Cartesian
0.115, 2.645

- Sabine River

In Beck (1988):

$$\eta = H - (1 - \frac{h_c}{S_t n})\|S_t n, 0\| - \frac{h_c}{S_t n} exp(-S_t \frac{n}{H})\|S_t n, 0\| \quad (2.2)$$

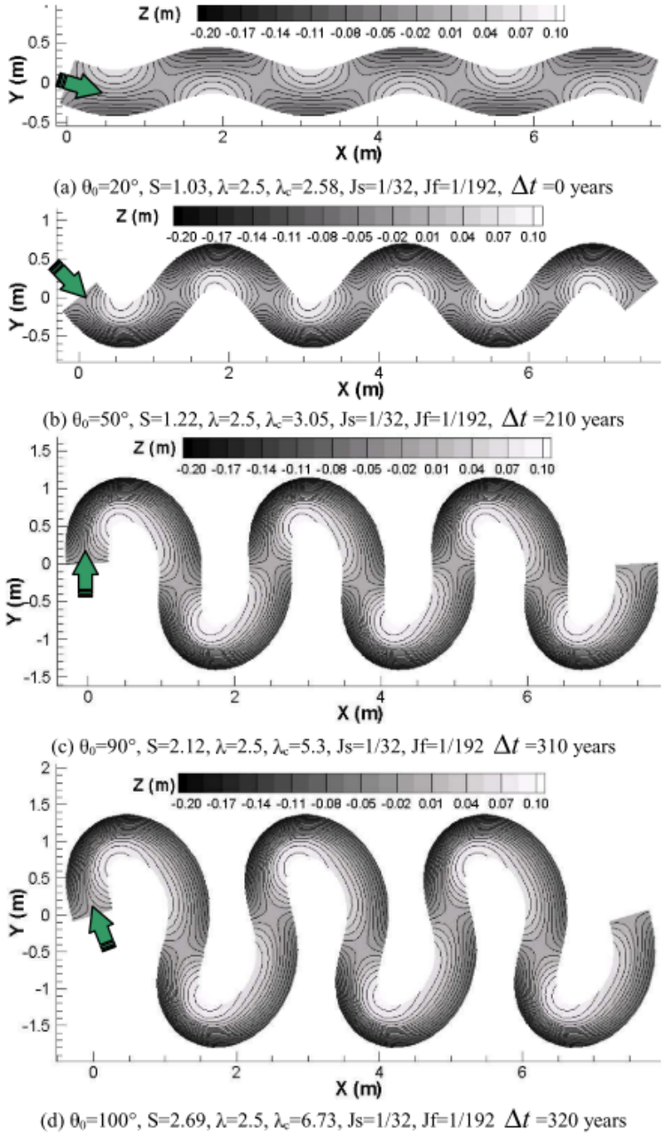
$$h_c = \frac{4BH|S_t| - S_t^2 B^2}{2B|S_t| + 2Hexp(-|S_t|\frac{B}{H})} \quad (2.3)$$

$$S_t = AHC \quad (2.4)$$

$$A = 3.8\left[1 + \frac{\beta}{6.96} exp\left(\frac{-6.96}{\beta}\right)\right] \quad (2.5)$$

where $\eta$ is the bed elevation, $H$ is the mean water depth, $h_c$ is the water depth at river centerline, $S_t$ is the transverse slope, $\|X, Y\|$ is the larger value of $X$ and $Y$, $B$ is the channel half-width, $A$ is the scour factor and $C$ is the local curvature. Equation 2.2 can be re-written as $\eta = f(H, B, C)$, which indicates the bed elevation can be a function of three parameters: water depth, channel width and channel curvature.

Abad & Garcia, 2005



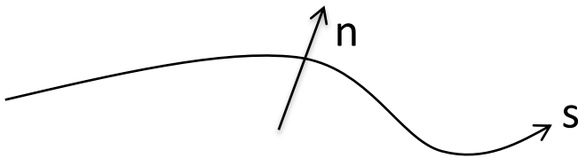
(a) $\theta_0$=20°, S=1.03, $\lambda$=2.5, $\lambda_c$=2.58, Js=1/32, Jf=1/192, $\Delta t$ =0 years

(b) $\theta_0$=50°, S=1.22, $\lambda$=2.5, $\lambda_c$=3.05, Js=1/32, Jf=1/192, $\Delta t$ =210 years

(c) $\theta_0$=90°, S=2.12, $\lambda$=2.5, $\lambda_c$=5.3, Js=1/32, Jf=1/192 $\Delta t$ =310 years

(d) $\theta_0$=100°, S=2.69, $\lambda$=2.5, $\lambda_c$=6.73, Js=1/32, Jf=1/192 $\Delta t$ =320 years

Can we simply replace the local curvature C
in Eqn 2.4 with cumulative (lagged) curvature?




Starting from SWE and after doing linear stability analysis, we get near-bank perturbation velocity as

$$u_b(s) = a_1 e^{-a_2 s} + a_3 C(s) + a_4 e^{-a_2 s} \int_0^s C(s') e^{a_2 s'} ds' \\ = a_1 e^{-a_2 s} + a_3 C(s) + a_4 \left[ C(s') * e^{-a_2 (s - s')} \right](s) \quad (2.40)$$

where * denotes for convolution integral

$$a_1 = u_b(0) + \chi C(0) \quad (2.36)$$

$$a_2 = 2 C_{f0} \beta \chi \quad (2.37)$$

$$a_3 = -\chi \quad (2.38)$$

$$a_4 = C_{f0} \beta \chi^2 \left[ \left( 1 + 5\sqrt{C_{f0}} \right) \left( A + \chi^3 F_0^2 \right) + 1 \right] \quad (2.39)$$

Evaluate the RHS term by term:

- The first term depends on ub(0) and C(0), which are initial conditions. The example on previous page gave a temporal varying non-zero ub(0).
- The second term implies the near-bank excess velocity is negative proportional to the local curvature, because chi > 0 leads to a3 < 0.
- The third term, the convolution integral is equivalent to imposing a lag to the local curvature. And the a4 wraps scour factor A, friction coeff. Cf0, half-width to depth ratio beta, Froude number F0 and sinuosity to the power of -1/3 into it.

As proposed by Beck (1988), $A$ has the following relation with $\beta$:

$$A = 3.8 \left[ 1 + \frac{\beta}{6.96} exp\left( \frac{-6.96}{\beta} \right) \right] \quad (2.41)$$

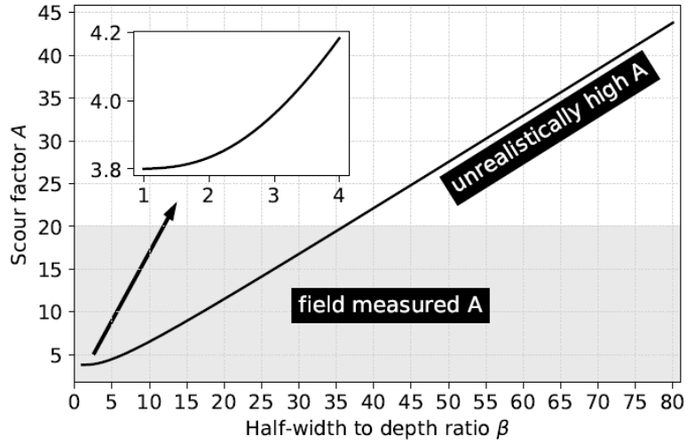



Table 2.2: Values of the scour factor $A$ in earlier studies.

| Studies | A values |
|---|---|
| Parker and Andrews (1986) | 2-15 |
| Johannesson and Parker (1989) | 2.5-6 |
| Hasegawa (1989) | 2-8 |
| Garcia et al. (1994) | 6 |
| Chen and Duan (2006) | 0-2 |
| Guneralp and Rhoads (2010) | 10 |
| Ottevanger et al. (2012) | 0.2 |
| Ottevanger et al. (2013) | 0.1, 0.35, 2, 7, 10, 11, 21 (summarized other works) |
| Motta et al. (2014) | 4, 5 and 6 |

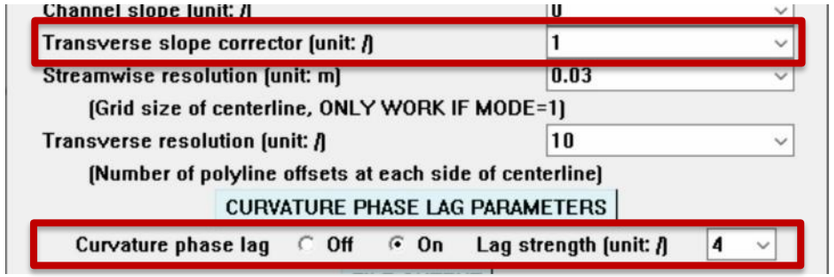
$$a_4 = C_{f0}\beta\chi^2\Big[\Big(1+5\sqrt{C_{f0}}\Big)\Big(3.8\big(1+\frac{\beta}{6.96}exp(\frac{-6.96}{\beta})\big) + \chi^3 F_0^2\Big)+1\Big] = f(C_{f0},\beta,\chi,F_0) \quad (2.42)$$

Now we work on replacing the convolution integral $\big[C(s')*e^{-a_2(s-s')}\big](s)$. We re-write Equation (40) as (assuming the first term is zero),

$$u_b(s) = a_3 C(s) + a_4\mathbb{F}(C(s)) \approx a_3 C(s) + a_4\mathbb{G}(C(s)) \quad (2.43)$$

where $\mathbb{F} = \big[C(s')*e^{-a_2(s-s')}\big](s)$ is the convolution integral of $C(s)$ and $\mathbb{G}$ is the upstream-ward weighted moving average of $C(s)$. The macro effect of $\mathbb{G}$ is same as $\mathbb{F}$, imposing a phase lag to the local curvature $C(s)$. Note that moving average is also a type of convolution, mathematically.

$\mathbb{F}$ to $\mathbb{G}$,
Phase shift magnitude
& Amplitude changed?
Here's a solution




$$S_t = \alpha \cdot A\mathbb{G}H \quad (2.45)$$

where $\alpha$ is the transverse slope corrector to 1) compensate and correct the neglect of $a_3$ term and 2) calibrate and compare against field measured bed transverse profile. The $\alpha$ can be defined in the GUI of pyRiverBed. Figure 2.5 gives a sample of three different ways to compute the transverse slope $S_t$. The one with $\alpha$ is exactly what pyRiverBed uses.

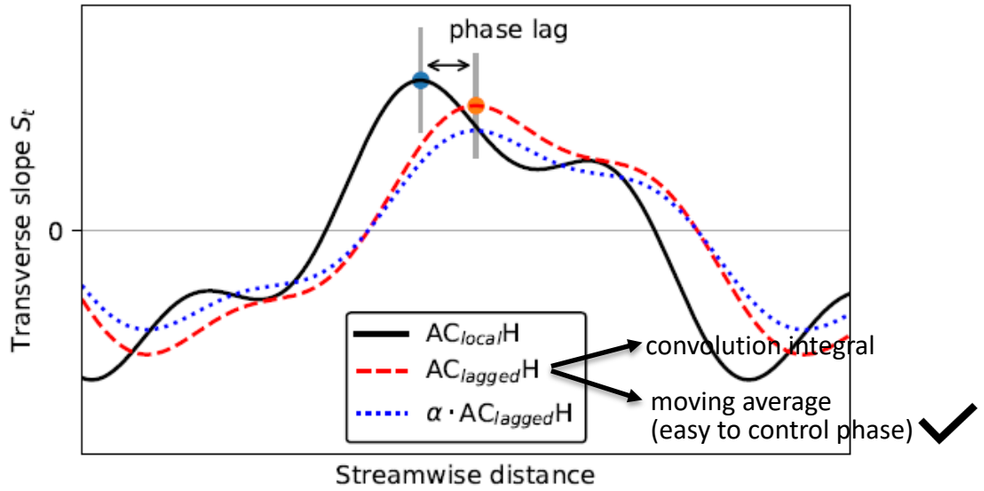



Figure 2.5: The schematic diagram showing how phase lag and amplitude of the transverse slope controlled in the proposed method.

# Implementation

Python is a lot slower than C/Fortran in sense of computing.

But…

Numba

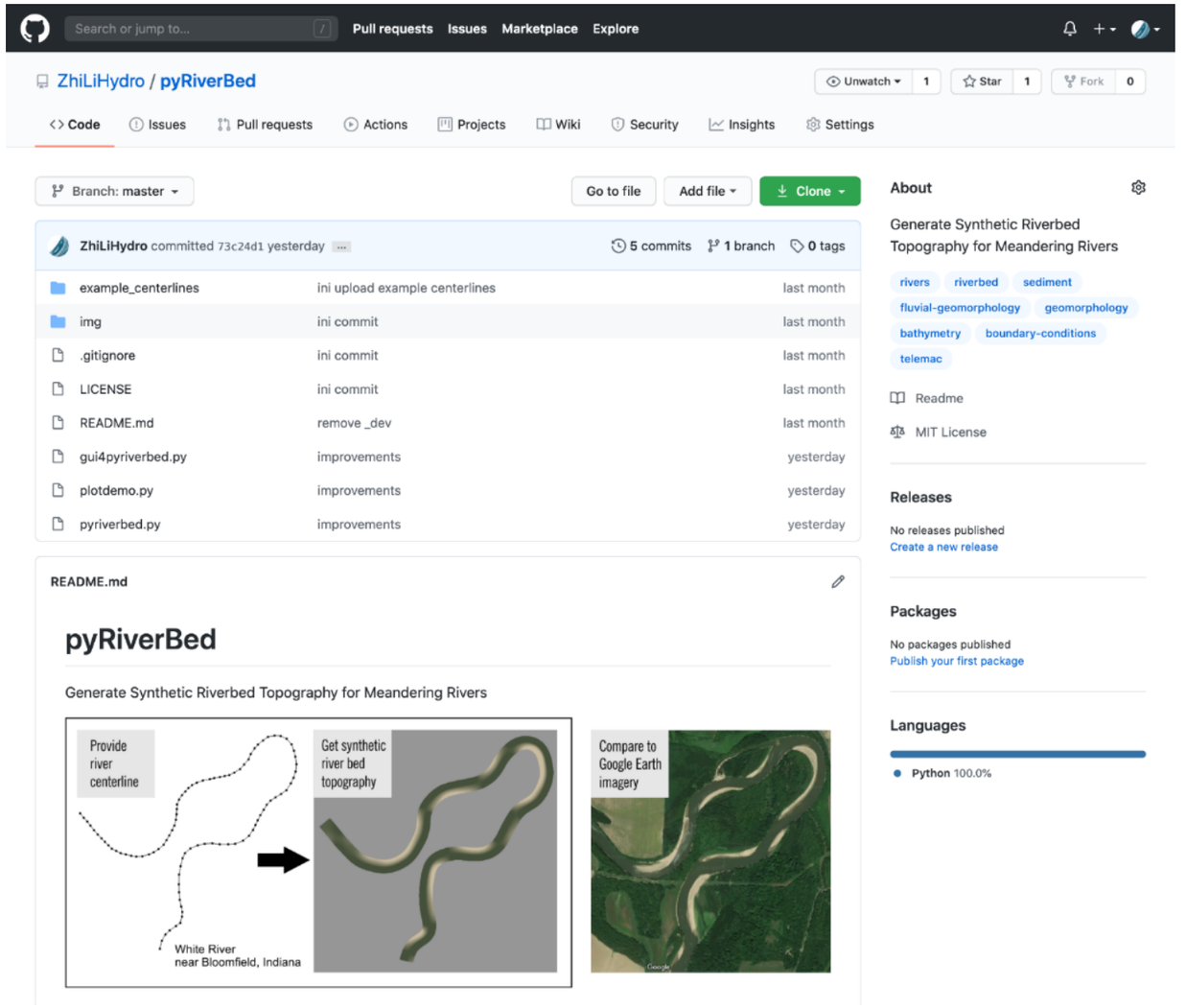
Numba makes Python code fast

Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.

## `tkinter` — Python interface to Tcl/Tk

**Source code:** Lib/tkinter/__init__.py

The `tkinter` package ("Tk interface") is the standard Python interface to the Tk GUI toolkit. Both Tk and `tkinter` are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)



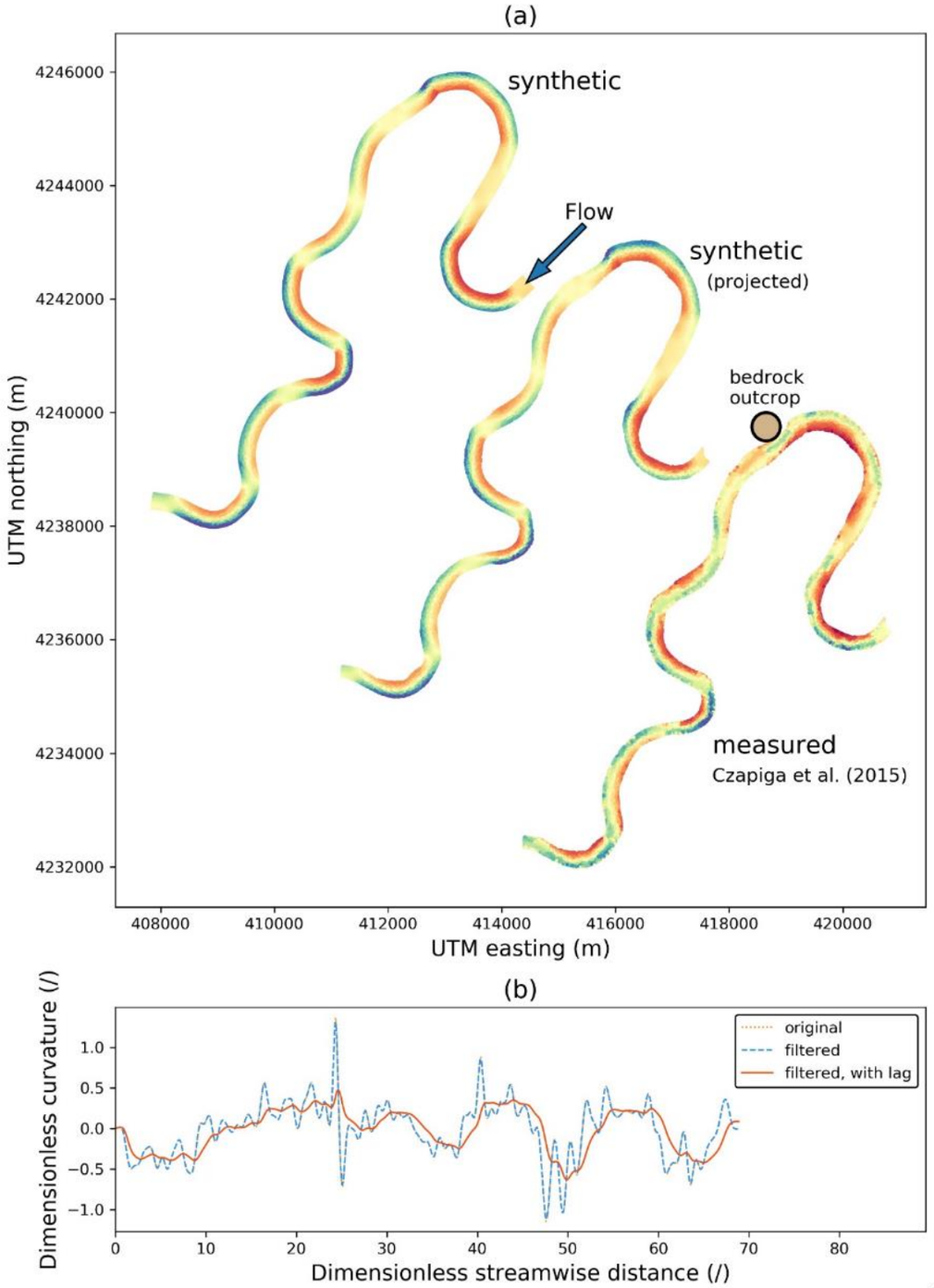

https://github.com/ZhiLiHydro/pyRiverBed

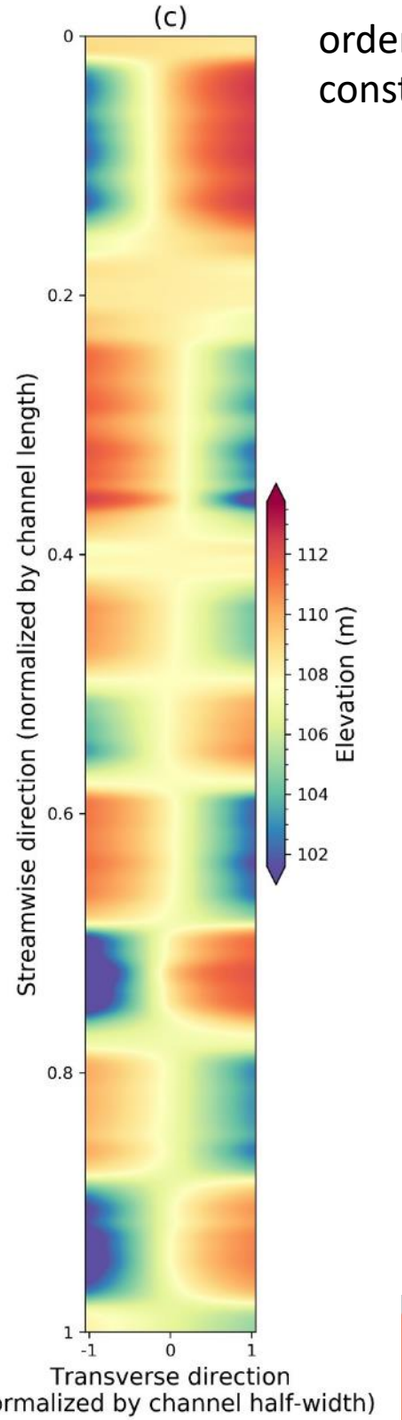
# Results

# Wabash River - Maier Bend

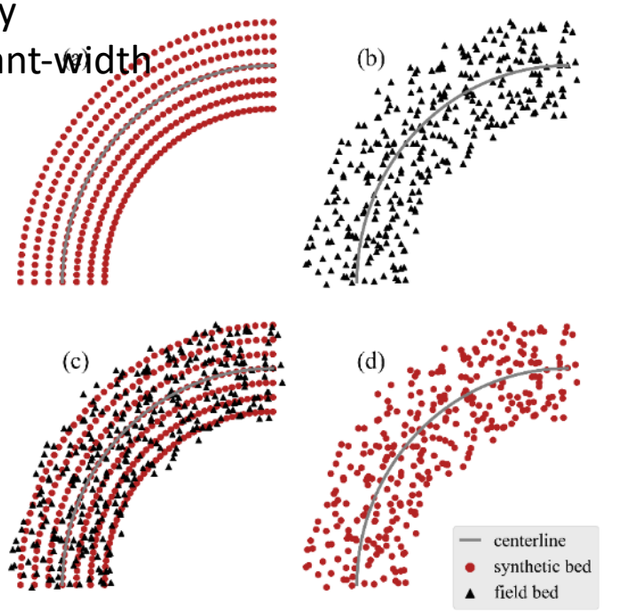
Width=260m
Depth=5m
Slope=1.2E-4

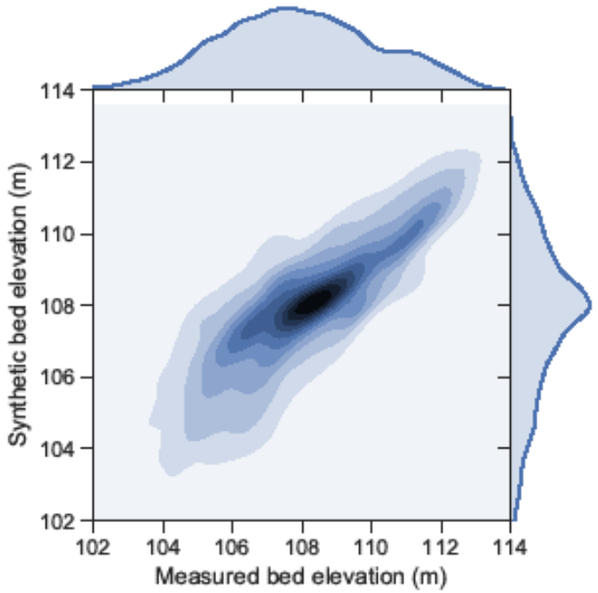
alpha = 0.7
Phase lag
Strength = 4



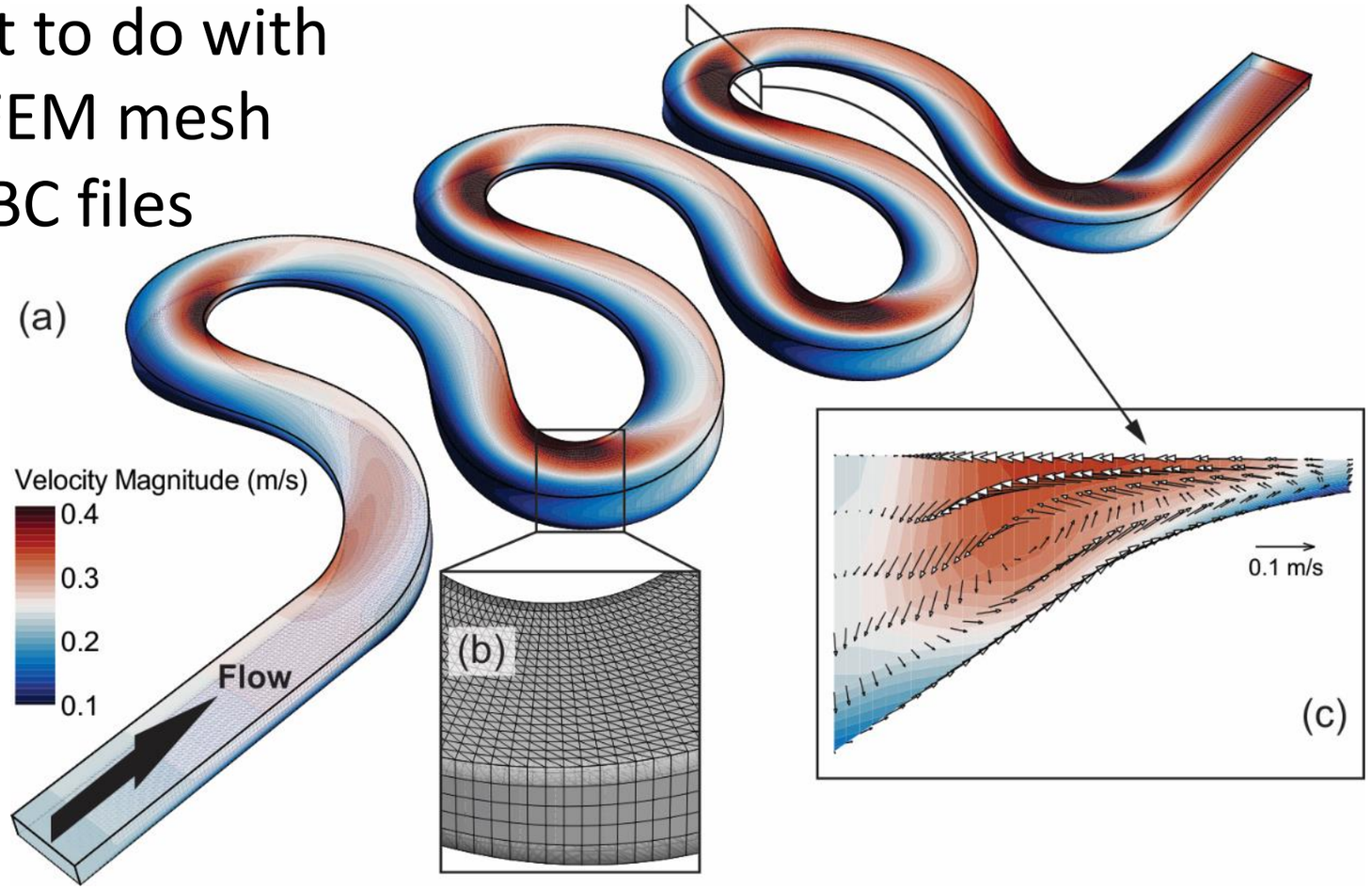




orderly
constant-width



# What to do with the FEM mesh and BC files




The major motivation of making FEM meshes and BC files right after the production of synthetic bed is to offer a convenient follow-up step to perform CFD modeling using TELEMAC or other solvers.

An example workflow of meandering river modeling could be: RivMAP/PyRIS → pyRiverBed → TELEMAC

Schwenk et al (2017)
Moneaglia et al (2018)

# Highlighted features of pyRiverBed:

- Provides two modes:
  - > Making synthetic meandering rivers via the built-in Kinoshita Curve calculator.
  - > Reading users' own real river centerlines.
- Expands 1D centerline to 2D river channel by AutoCAD-like polyline offsetting.
- Calculates riverbed topography through an analytical method.
- Makes FEM mesh files and boundary condition files for TELEMAC numerical modeling usage.
- Models meander migration and outputs synthetic riverbed topography for each snapshot during channel migration.
- Uses a tkinter-based graphical user interface as front end.
- Cross-platform, runs on macOS, Linux, Windows machines.

# Thank you!