Interactive Python Widget for Correcting WRF-Hydro Input Grids

Nicholas Elmer[1,2] and Andrew Molthan[1,3]

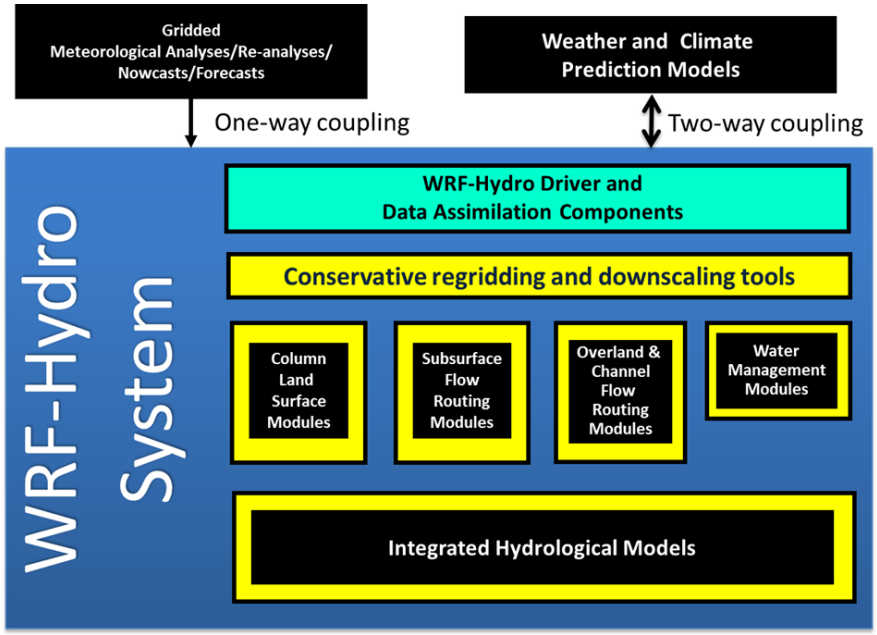
[1]NASA Short-term Prediction Research and Transition (SPoRT) Center, Huntsville, Ala.

[2]Department of Atmospheric Science, University of Alabama in Huntsville, Huntsville, Ala.

[3]Earth Science Office, NASA MSFC, Huntsville, Ala.

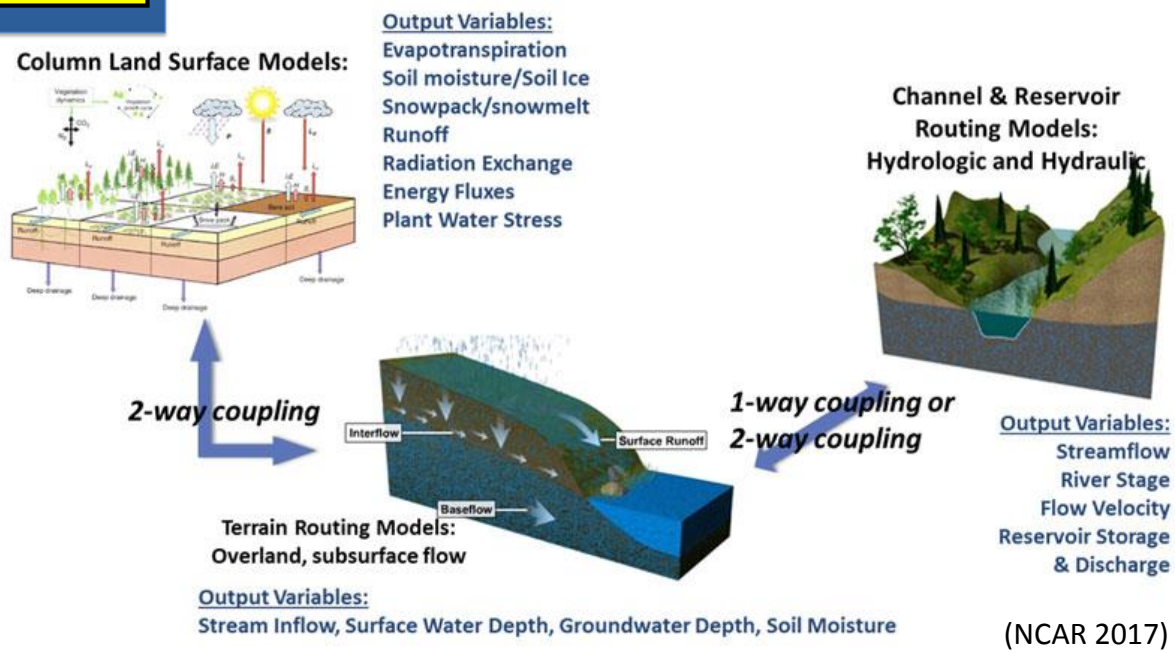
January 23, 2017
Seventh Symposium on Advances in Modeling and Analysis using Python
97th AMS Annual Meeting, Seattle, Wash.

WRF-Hydro System



(NCAR 2017)

- WRF-Hydro (Gochis et al. 2013) is WRF hydrological extension package
- Run in uncoupled or coupled mode
- Contains column land surface, terrain routing, and channel routing modules
- Supports multi-scale domains, so each module may require input grids at different spatial resolutions

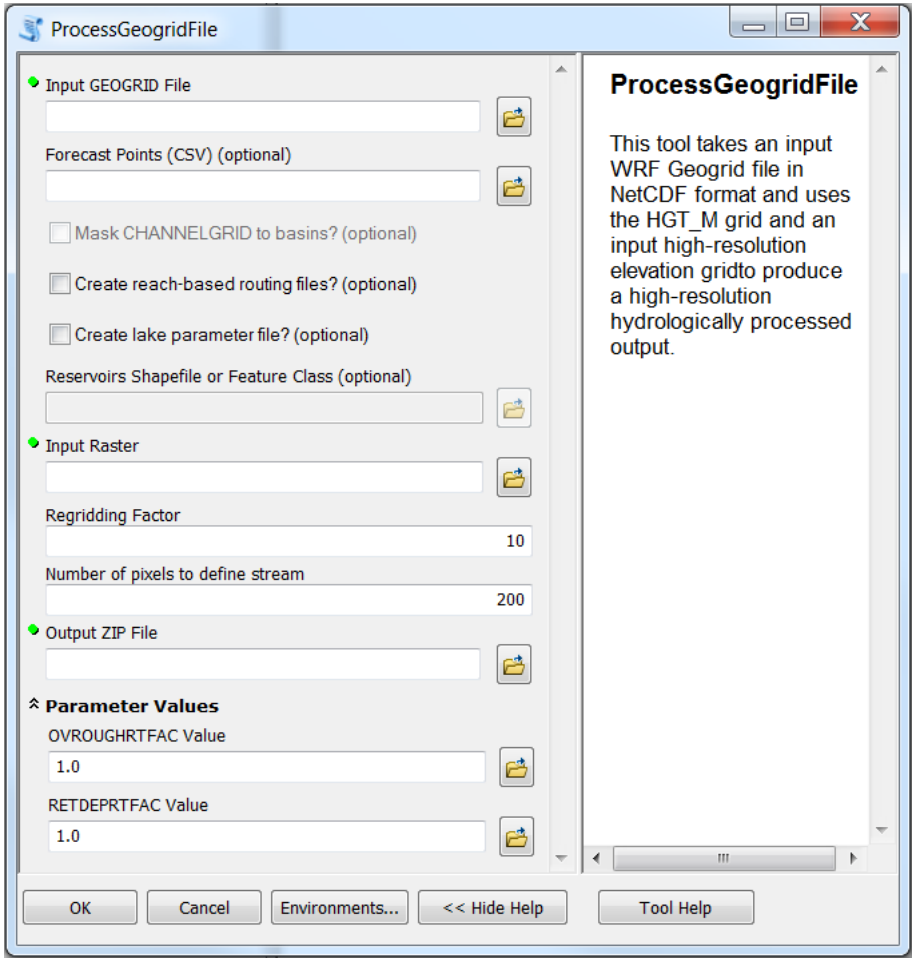


(NCAR 2017)

NASA SPORT

WRF-Hydro GIS Preprocessing Tool

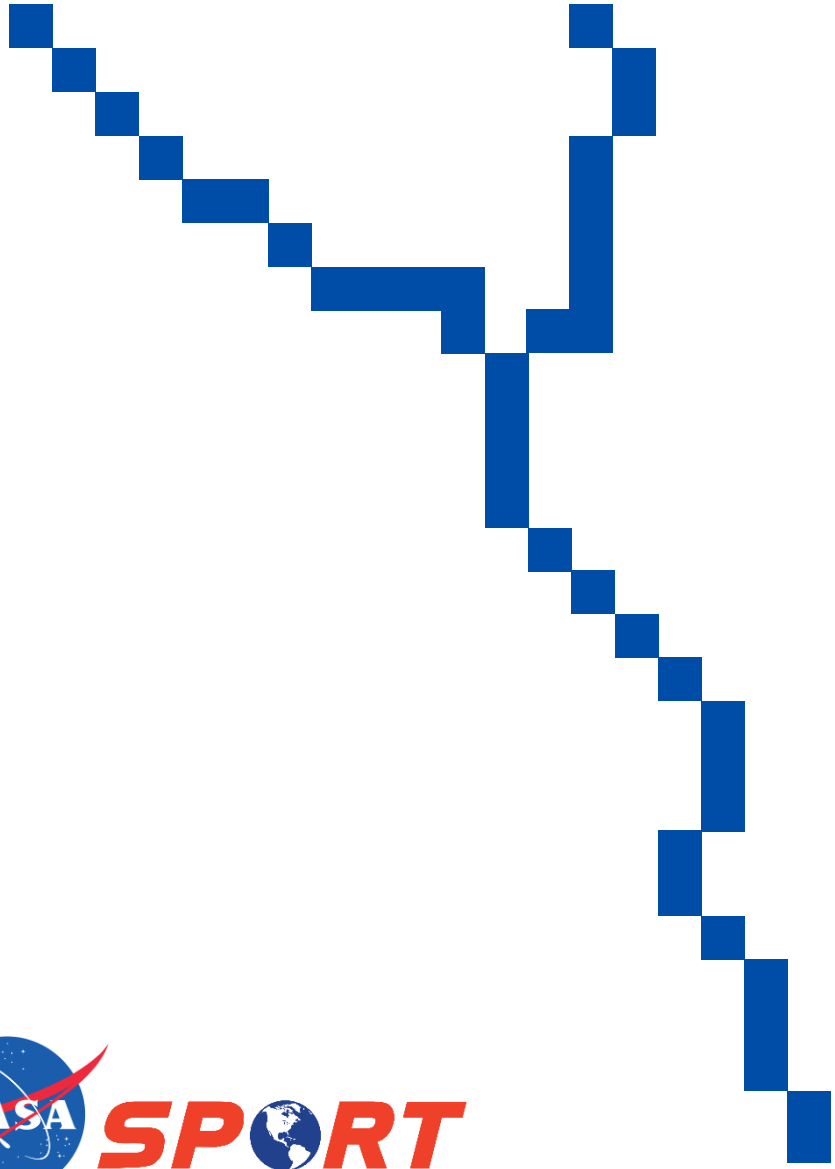
- Derives WRF-Hydro terrain routing and channel routing grids using ArcGIS hydrology tools
- Channel routing grids include channel grid and lake grid.
- Inputs:
  - WRF Preprocessing System (WPS) GEOGRID file
  - High-resolution Digital Elevation Model (DEM)
- NCAR working to make tool 100% open-source Python
- Preprocessing Tool Documentation: Sampson and Gochis 2015



Screen capture of WRF-Hydro GIS Preprocessing Tool within ArcGIS

NASA SPORT

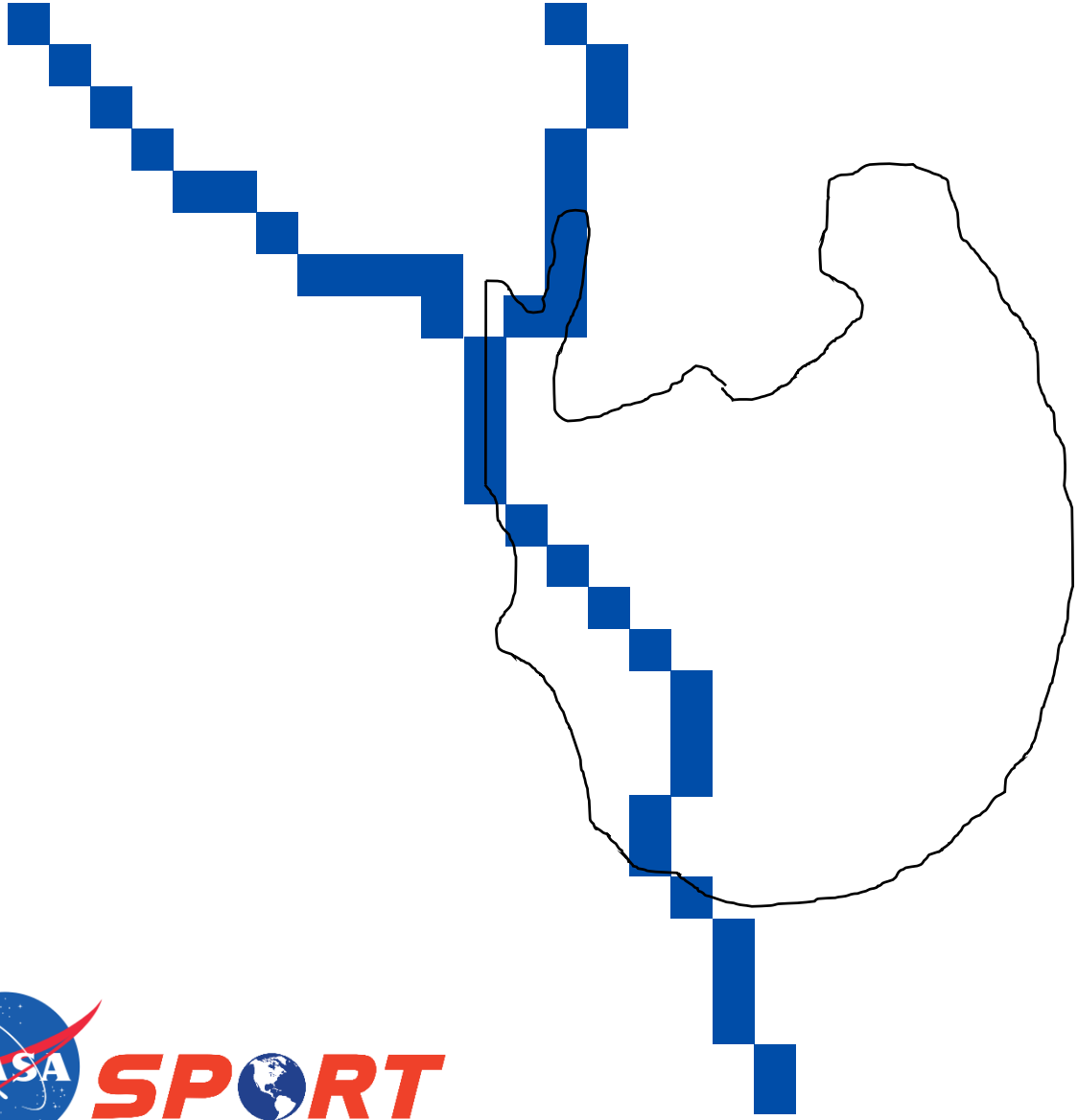
WRF-Hydro GIS Preprocessing Tool



- Channel grid derived from high-resolution DEM

(Sampson and Gochis 2015)

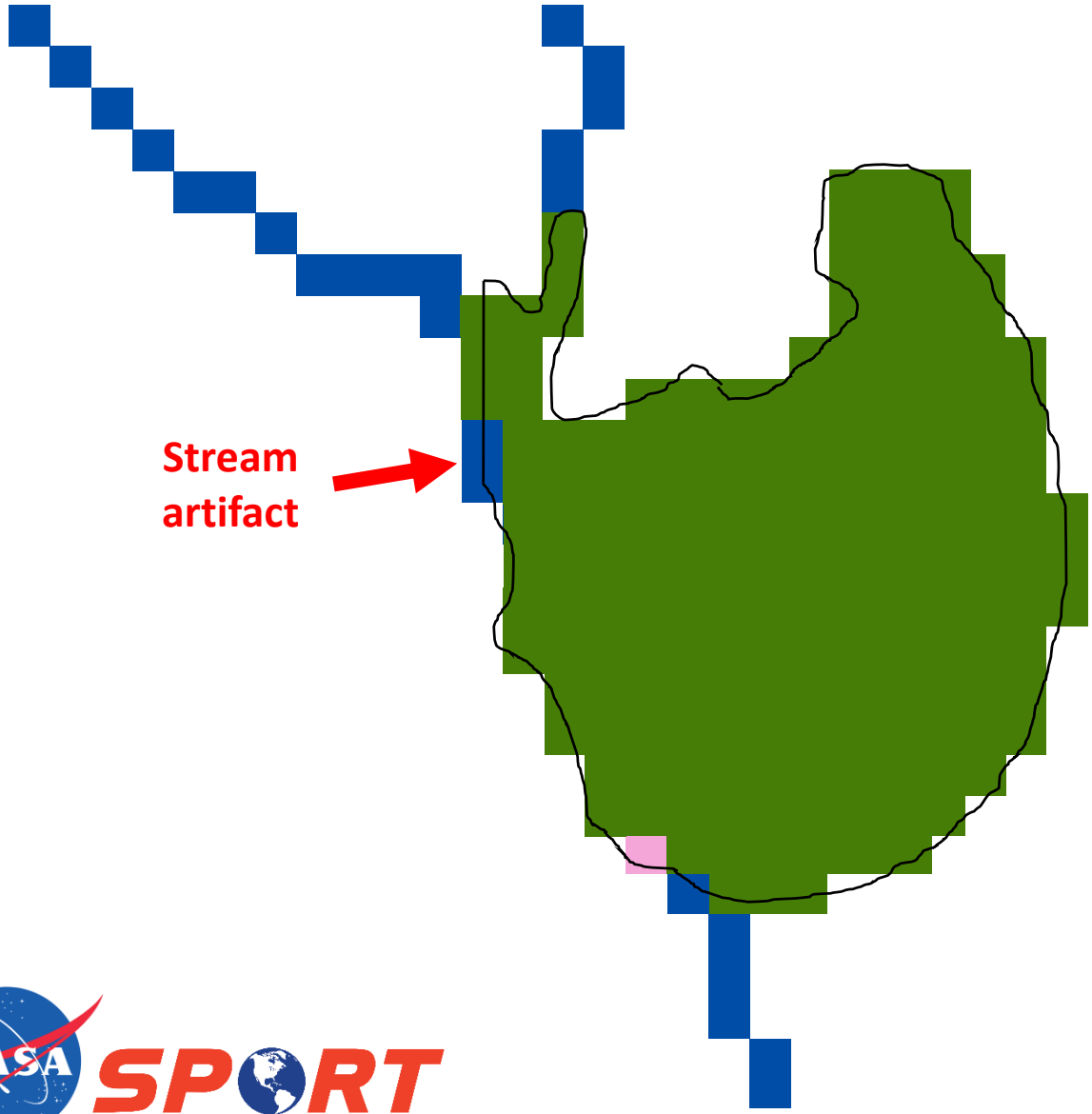
WRF-Hydro GIS Preprocessing Tool



- Channel grid derived from high-resolution DEM
- Insert reservoirs/lakes using lake shapefile

(Sampson and Gochis 2015)

WRF-Hydro GIS Preprocessing Tool

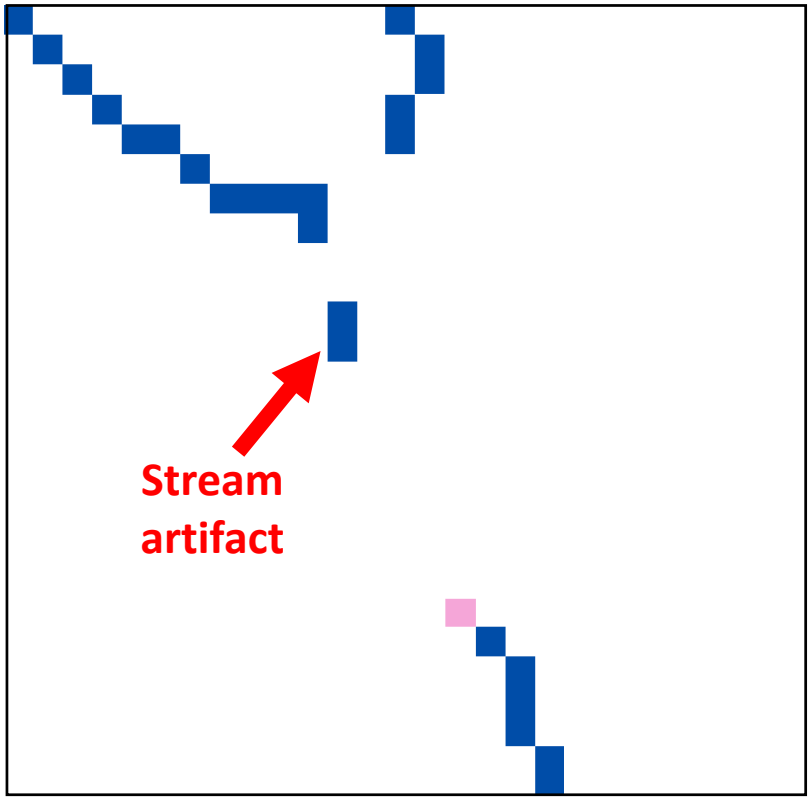


- Channel grid derived from high-resolution DEM
- Insert reservoirs/lakes using lake shapefile
- Lake polygon rasterized to mask channel grid
- Lakes numbered 1 to n
- Stream artifacts may result

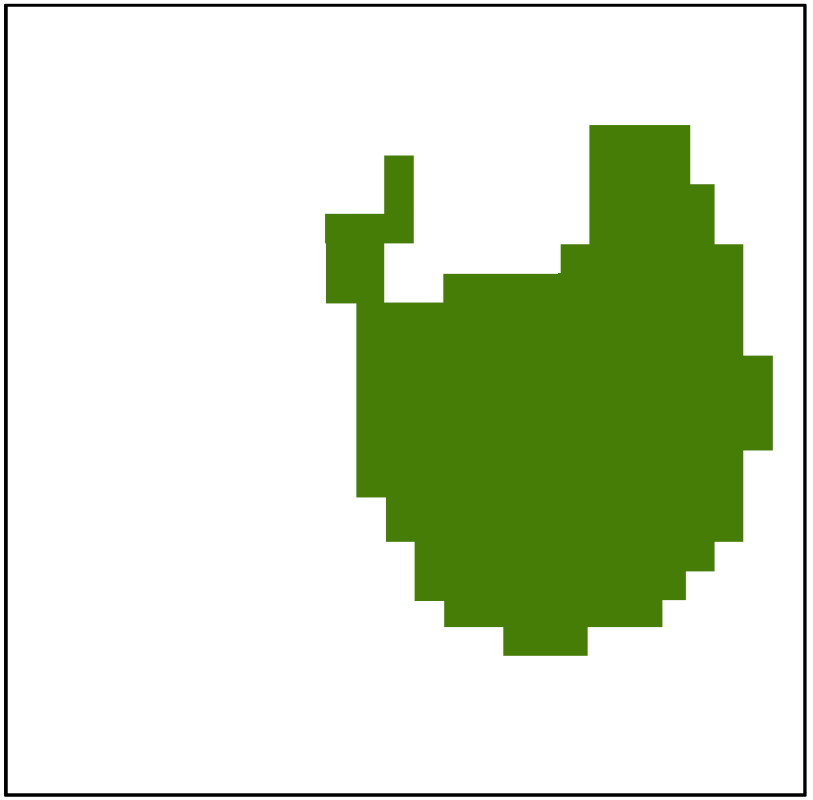
(Sampson and Gochis 2015)

WRF-Hydro GIS Preprocessing Tool

- Preprocessing Tool does not remove stream artifacts
- Stream artifacts must be removed prior to running WRF-Hydro
  - Straightforward, programmatic method not available
  - Can be done interactively in Python using TkInter module



Channel grid



Lake grid

NASA SPORT

TkInter – A Python Graphical User Interface

- Provides basic Graphical User Interface (GUI) within Python
- Contains classes which allow display, positioning, and control of widgets
- Wrapper functions for Tcl/Tk
- Importing TkInter

```
import sys
if sys.version_info[0] < 3:
    from Tkinter import *
else:
    from tkinter import *
```

- Online documentation and resources:
  - https://wiki.python.org/moin/TkInter
  - http://tkinter.unpythonic.net/wiki/
  - http://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf

TkInter Widgets

Button
Canvas
Checkbutton
Entry
Frame
Label
LabelFrame
Listbox
OptionMenu
PanedWindow
Radiobutton
Scale
Scrollbar
Spinbox
Text
Tk
Toplevel

NASA SPORT

*Spyder 2.3.8, Python 2.7, TkInter 8.5

Applying TkInter – Interactive Python Widget

```
from Tkinter import *
class Tool(object):
    def _save(self):
        :
    def _quit(self, *kwargs):
        :
    def plot_subset(self, *kwargs):
        :
    def key(self, event):
        :
    def click(self, event):
        :
    def __init__(self, master, path,
        :

def main(path, *kwargs):
    root = Tk()
    tool = Tool(root, path, *kwargs)
    root.mainloop()

if __name__ == '__main__':
    #define parameters here
    path = "......."
    main(path, *kwargs)
```

```
def __init__(self, master, path, *kwargs):
    self.master = master
    #set up figure
    self.fig = mplfig.Figure()
    self.ax2 = self.fig.add_subplot(121)
    self.ax1 = self.fig.add_subplot(122)
    self.canvas = FigureCanvasTkAgg( \
        self.fig, master=self.master)

    #define event connections
    self.cid = self.canvas.mpl_connect( \
        'button_press_event', self.click)
    self.kid = self.canvas.mpl_connect( \
        'key_press_event', self.key)
    quit_but = Button(master=self.master, \
        text='Quit', command=self._quit)
    quit_but.pack(side=BOTTOM)

    #add event connections to canvas
    self.canvas.get_tk_widget().pack()

    #read WRF-Hydro input netCDF files here
    self.plot_subset(*kwargs)
```

*Spyder 2.3.8, Python 2.7, TkInter 8.5

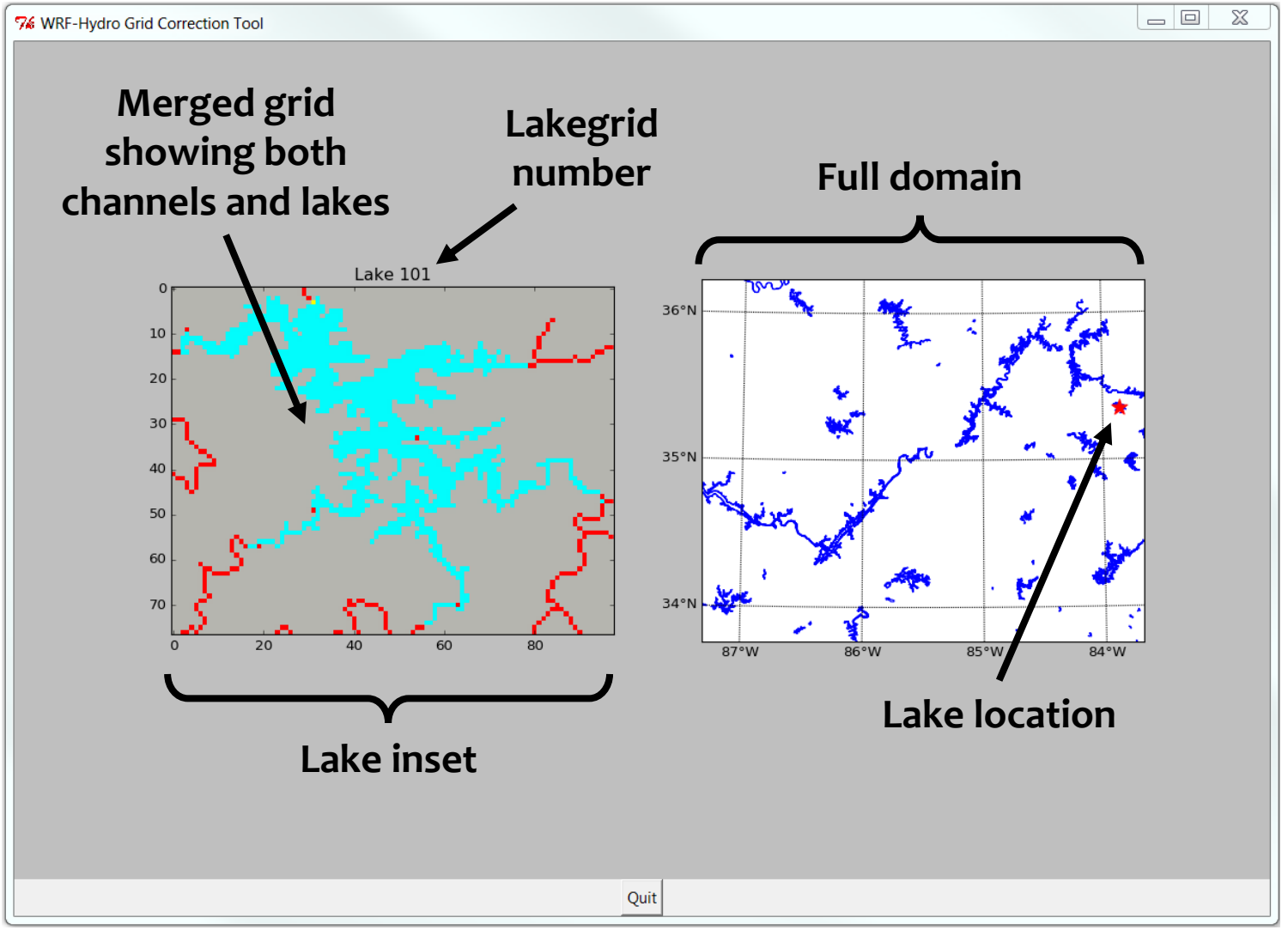
Interactive Python Widget

Lake

Channel

Node / outlet

Land



Interactive Python Widget

```
class Tool(object):
    def _save(self):
        :
    def _quit(self, *kwargs):
        :
    def plot_subset(self, *kwargs):
        :
    def key(self, event):
        :
    def click(self, event):
        :
    def __init__(self, master, path,
        :

def main(path, *kwargs):
    root = Tk()
    tool = Tool(root, path, *kwargs)
    root.mainloop()

if __name__ == '__main__':
    #define parameters here
    path = "......."
    main(path, *kwargs)
```

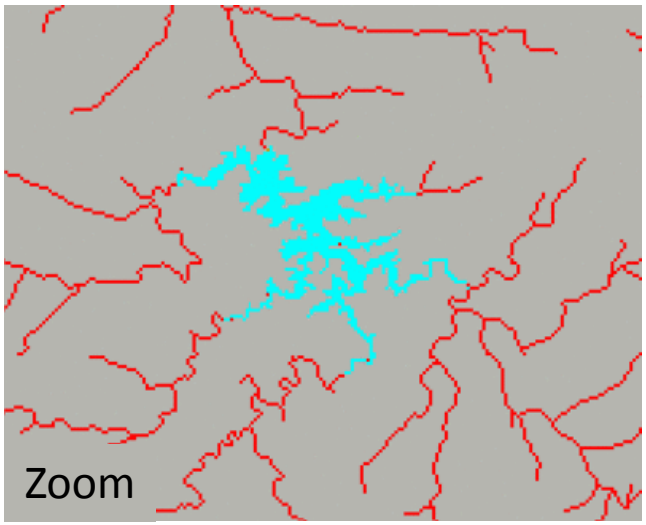
```
def _quit(self, *kwargs):
  self._save()
  #disconnect event connections
  self.fig.canvas.mpl_disconnect(self.cid)
  self.fig.canvas.mpl_disconnect(self.kid)
  self.master.quit()     # stops mainloop
  self.master.destroy()  # this is necessary
  return
```

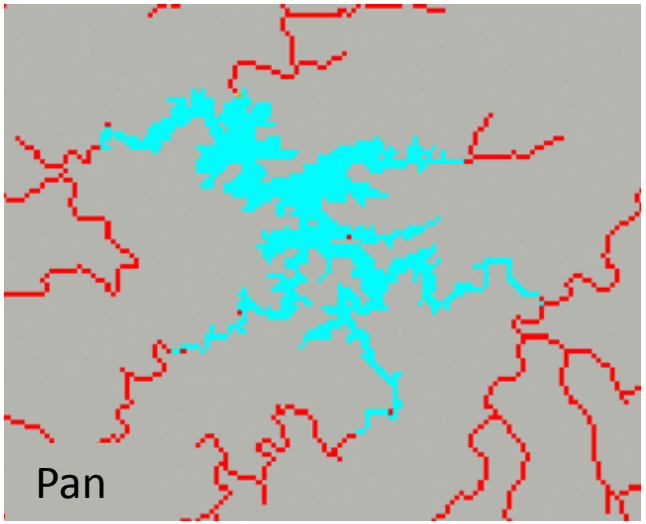
```
def key(self, event):
  if event.key == 'q': # quit
      self._quit()
  elif event.key == 's': # save
      self._save()
  elif event.key == 'up' or event.key == 'down' \
      or event.key == 'left' \
      or event.key == 'right' : # pan
      :
  elif event.key == 'z': # zoom
      :
  else: print 'The %s key does nothing...' \
      %repr(event.key)
  return
```

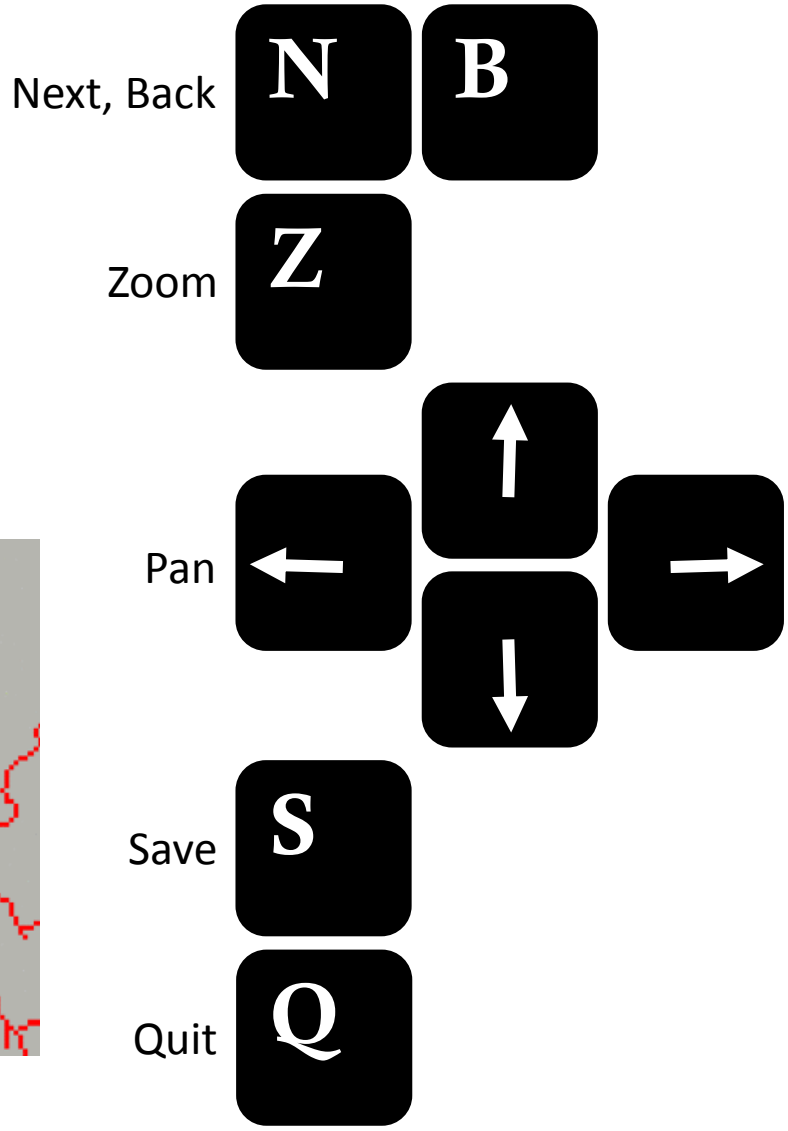
```
def click(self, event):
  if event.button == 1:
      print event.xdata, event.ydata
      :
  elif event.button == 2:
      :
  else: print 'Try Again...'
  return
```

NASA SPORT

Interactive Python Widget







NASA SPORT

Interactive Python Widget

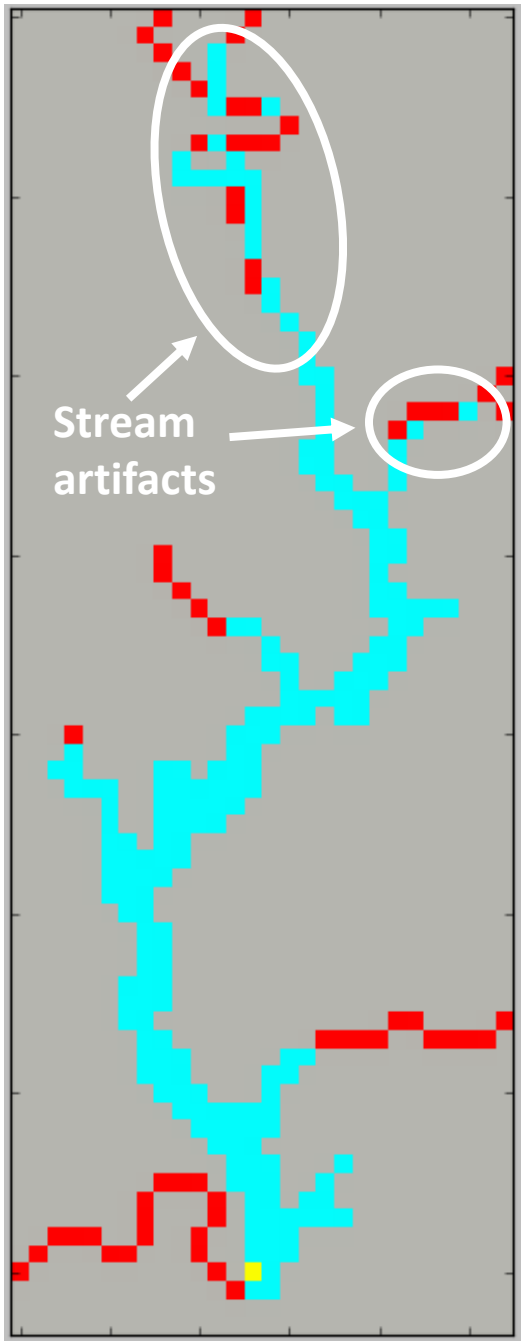
Lake

Channel

Node / outlet

Land

Lake Catoma, Ala. (Google Maps)

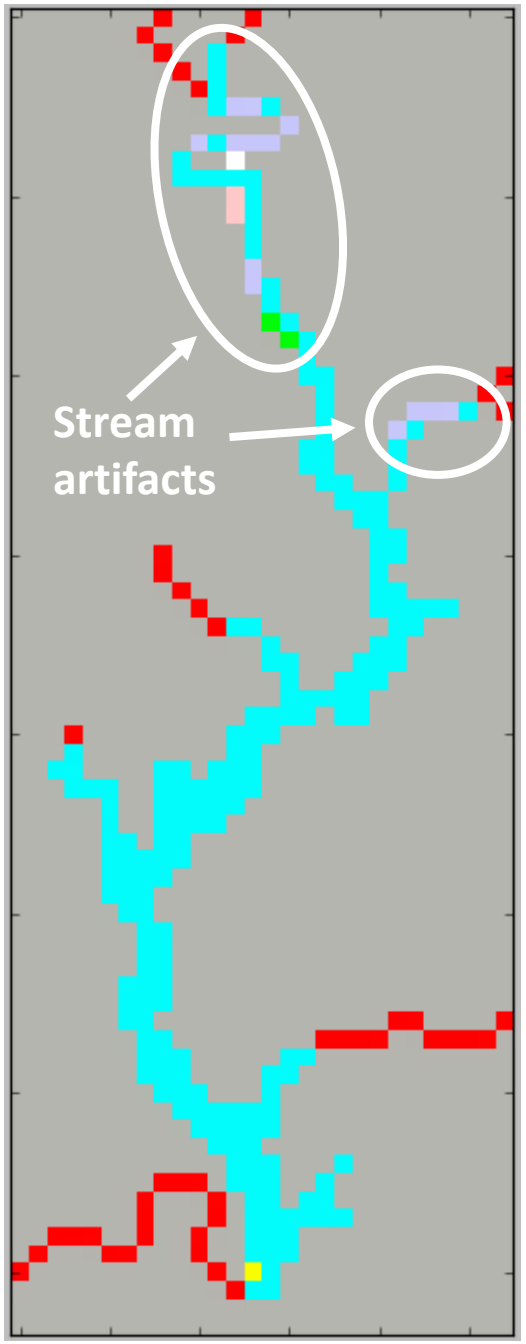


NASA SPORT

Interactive Python Widget

- Lake
- Channel
- Node / outlet
- Land
- Channel → Lake
- Channel → Land
- Lake → Land
- Land → Lake

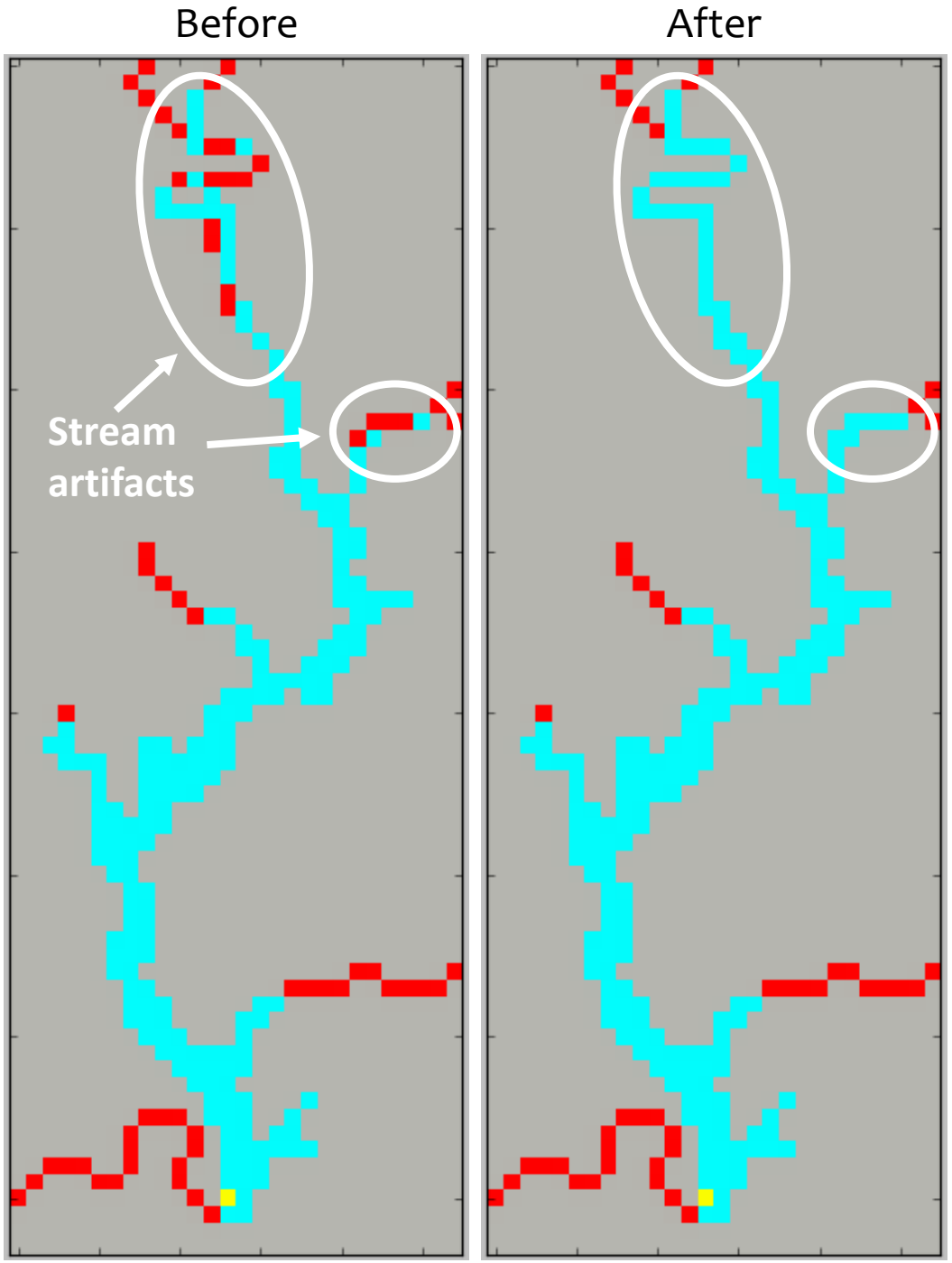
Lake Catoma, Ala. (Google Maps)



NASA SPORT

Interactive Python Widget

- Stream artifacts removed, resulting in continuous channel/lake grid
- Widget saves updated channel and lake grids for use in WRF-Hydro
- Process only takes a few minutes



NASA SPORT

Summary

- TkInter is standard Python GUI package
- For many applications requiring array manipulations, programmatic methods using Python or netCDF operators (NCO) are preferred
- However, TkInter widgets can be beneficial when a straightforward, programmatic methods are not available, thereby requiring manual modifications
- Interactive Python Widget
  - Modifies grids created by WRF-Hydro GIS Preprocessing Tool
  - Independent of ArcGIS; 100% open-source Python
  - Enables array to be visualized and modified concurrently
  - Can be extended/modified for many other applications

NASA SPORT

Nicholas Elmer
nicholas.j.elmer@nasa.gov

NASA SPoRT

Webpage: http://weather.msfc.nasa.gov/sport/

Blog: https://nasasport.wordpress.com/

Facebook: NASA SPoRT Center

Twitter: @NASA_SPoRT

References

Gochis, D., W. Yu, and D. Yates, 2013: The NCAR WRF-Hydro technical description and user's guide: version 1.0, 120 pp., http://www.ral.ucar.edu/projects/wrf_hydro/images/ WRF_Hydro_Technical_Description_and%20User_Guide_v1.0.pdf.

NCAR, 2017: WRF-Hydro Modeling System. Research Applications Library, https://www.ral.ucar.edu/sites/default/files/public/projects/wrf_hydro/.

Sampson, K., and D. Gochis, 2015: WRF Hydro GIS Pre-Processing Tools: Version 2.2 Documentation. National Center for Atmospheric Research, Research Applications Laboratory, Boulder, Colo., 39 pp.

Shipman, J. W., 2013: Tkinter 8.5 reference: a GUI for Python. New Mexico Tech Computer Center, 168 pp., http://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf.

NASA SPORT