

Proposal

Wesley Bowman

May 29, 2014

1. Methodology: All programming will be done in Python, MatLab, or Fortran with it primarily being done in Python.

- Model Tuning:

I expect this be done by Andy Balzer. The Python code will be contributed by me as he needs it. It will use the Python UTide on any data, and output it in a format that Andy would like. I also contributed to setting up the FVCOM run file for spatially-varying bottom frictions.

These codes use the Python libraries netCDF4, numpy, pandas, and UTide primarily.

- Model Validation:

I have written UTide in Python, so that model validation is possible with the harmonic constituents.

- Data Accessibility:

This will be done using netCDF4 and pandas. There is a possibility of a GUI to be implemented here, upon request. This will not help on clusters though.

- Data Analysis:

Using mpi4py along with netCDF4 and pandas, harmonic analysis of the elevation and velocities can be done and saved to a netCDF4 file in parallel. It will be written in parallel since the grid sizes can be large, and running UTide is an expensive process that can take a noticeable time.

Interpolation can be done using Matplotlib, and pandas can be used to filter noise from the in-situ measurements.

- Format Conversion:

Pandas allows for data to be easily manipulated into many different output formats. Between numpy, scipy, netCDF4, and pandas, all file formats should be able to be read in, and manipulated to the user specific output.

- Plotting:

All plotting can be done using pandas with Matplotlib or just standalone matplotlib.

- GIS Interface:

This will be done with python, accessing all other python code with the GIS server sending info to python, and python sending the correct data back. This may require arcpy, or it may be doable with no special libraries besides the ones mentioned above already.

2. Role Definition: I will be lead programming and programming coordinator. I will take on the difficult programming task, as well as assist others in their programming duties. I will also organize the programs so that everyone will be able to know where code is and if it is up to date.

3. Detailed Time-line:

I need a little more specifics before I can contribute to a timeline. UTide needs to be finished first, which I plan on doing in the entirety of next week. After that, we should be able to move onto all the components that require UTide to be finish, and I can also send it out for beta testing then. After that, a lot of code is already done, and needs to be correctly organized. Code that Aidan left needs to be sorted through to find all the useful bits, and after talking with Karsten, we may get Aidan to come down and give us a brief tour of his code to make tha a bit faster. Data organization should take around 3 days I would imagine. After that, it will vary depending on the project. GIS interfacing will take a good while. The rest, each individual program should be given at least a 3 days to finish, that way the programmer is not rushed to finish and sends out an untested piece of code.