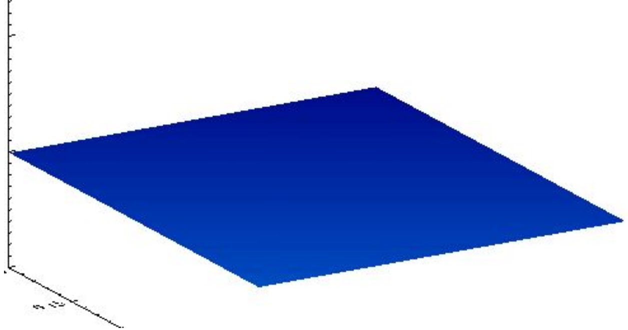
Using ANUGA

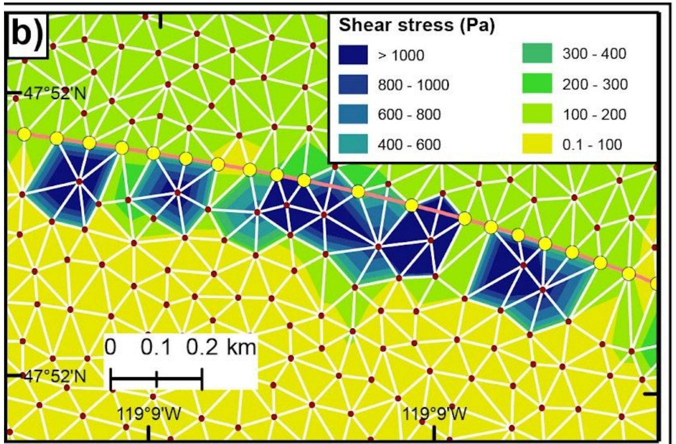
Intro to ANUGA: before you begin

- Download ANUGA, Python, text editor
  - Download ANUGA: https://github.com/GeoscienceAustralia/anuga_core (code → download zip)
  - ANUGA user manual: https://www.researchgate.net/publication/318511561_ANUGA_User_Manual_Release_20
  - Install Python (2.X, won't work with python 3!): https://www.python.org/downloads/release/python-2716/ (2.7.16)
  - Packages: numpy, netCDF
  - Nano text editor: https://www.nano-editor.org/download.php
- Optional: parallel processor
  - OpenMPI: https://www.open-mpi.org/software/ompi/v4.1/
- ArcGIS
- Other useful resources
  - ANUGA user community--great for troubleshooting! https://sourceforge.net/p/anuga/mailman/anuga-user/

Intro to ANUGA: what is it



- What's it doing?
  - Finite-volume method for solving the Shallow Water Wave Equation in 2D
  - Tracks water elevation, bed elevation, x-momentum, and y-momentum over time
  - Variable-size triangular mesh
  - Advantages over other models: numerical stability at high Froude numbers, resolve hydraulic jumps, wetting and drying capabilities
  - Limitations: only works with UTM projection systems, Mannings friction, slower than some other models, can't resolve vertical convection/turbulence
- What can you model?
  - Channel flow of known discharge/hydrograph*
  - Lake drains/dam bursts*
  - Coastal processes (storm surges, tsunamis)
  - Watershed-wide flow (rainfall, culverts)



Intro to ANUGA: inputs

- **DEM:** ascii file containing elevation data, can create using Raster to Ascii in Arc (should be a clean DEM--fill sinks and clip to domain before converting)
    - note! .prj projection file must be reformatted from the default created with Raster to Ascii
- **Domain boundary:** csv file containing x & y coordinates of outer domain limits (draw polygon → feature vertices to points → calculate geometry)
    - The fewer boundary segments, the better
- **Optional: other boundaries:** csv file(s) containing x & y coordinates for regions with different spatial resolution, non-spatially-uniform Mannings n, initial stage, breaklines, etc.
    - Must be COMPLETELY INSIDE domain polygon
- **Inlet:** csv file containing x & y coordinates for a line segment across which water enters the domain
    - Watch for overlap with channel banks
- **Gages:** csv file containing x & y coordinates for points to track stage over time
- **Time, space, and discharge range/resolution:** total model duration (seconds), time step (seconds), spatial resolution(s) as maximum triangle area (m^2), hydrograph for input discharge

Running ANUGA: hydrograph file

- Creates a .tms (time-series) file specifying the discharge over time
- Edit in txt2tms_example.py
  - dischargeTmsFile (line 8)
  - projectFileName (line 9)
  - time (line 16)
  - q (line 17)
- Run: in directory with ascii/prj/bnd/inlet/gage files, type:
  - python txt2tms_example.py

Running ANUGA: the model

- Establishes the domain, and evolves through time
- Edit in flood_anuga_example.py:
  - time (line 27)
  - root (lines 39 & 53)
  - pol_1 (line 103)
  - domain (line 109)
  - domain.set.quantity (lines 129-134)
  - bcline (line 147)
  - Q1 (line 148)
  - domain.set_boundary (line 161)
  - yieldstep (line 172)
- Run: in directory with ascii/prj/bnd/inlet/gage files (and after creating the .tms file) type:
  - python flood_anuga_example.py <spatial_res_default> <spatial_res_high> <folder_name> (For running on a single core)
  - mpirun -<number of cores> python flood_anuga_example.py <spatial_res_default> <spatial_res_high> <folder_name> (For running on # cores in parallel)

Processing ANUGA: outputs

- Output files:
  - example.sww
  - The .sww file contains stage, elevation, x-momentum, and y-momentum for each mesh vertex over all time steps in netCDF-readable format (can view in QGIS)
- Create ascii files of hydraulic parameters (processing_anuga_example_clean.py): edit
  - name (line 22)
  - Q (line 23)
  - n (line 26)
  - cell (line 27)
  - red (line 32)
  - write projection (lines 92-98) *change values to match example.prj
  - When finished, copy to folder containing sww file and type: python processing_anuga_example_clean.py
- Create gage files of stage (gauge_process.py): edit
  - sww_file (line 30)
  - gauge_in_file (line 31)
  - gauge_out (line 32)
  - When finished, copy to folder containing sww file and type: python sww2csv.py

Processing ANUGA: Displaying in ArcMap

- Ascii files can be added directly into ArcMap
- Convert to raster to work with other ArcMap tools (use Ascii to Raster, might need to run Define Projection as well)
- Arcpy script to go from ascii files to shapefiles of wetted extents (wetted_extents.py): edit
  - env.workspace (line 13)
  - outws (line 14)
  - discharges (line 17)
  - timesteps (line 18)
  - scenario (line 20)
  - inputDEMs_list (line 22)
  - DEMname (line 33)
- Arcpy script to calculate areas inside/outside high-water polygon from shapefiles of wetted extents (calculate_areas.py): edit
  - env.workspace (line 13)
  - outws (line 14)
  - discharges (line 17)
  - timesteps (line 18)
  - scenario (line 20)
  - hw_inner (line 23)
  - hw_outer (line 24)