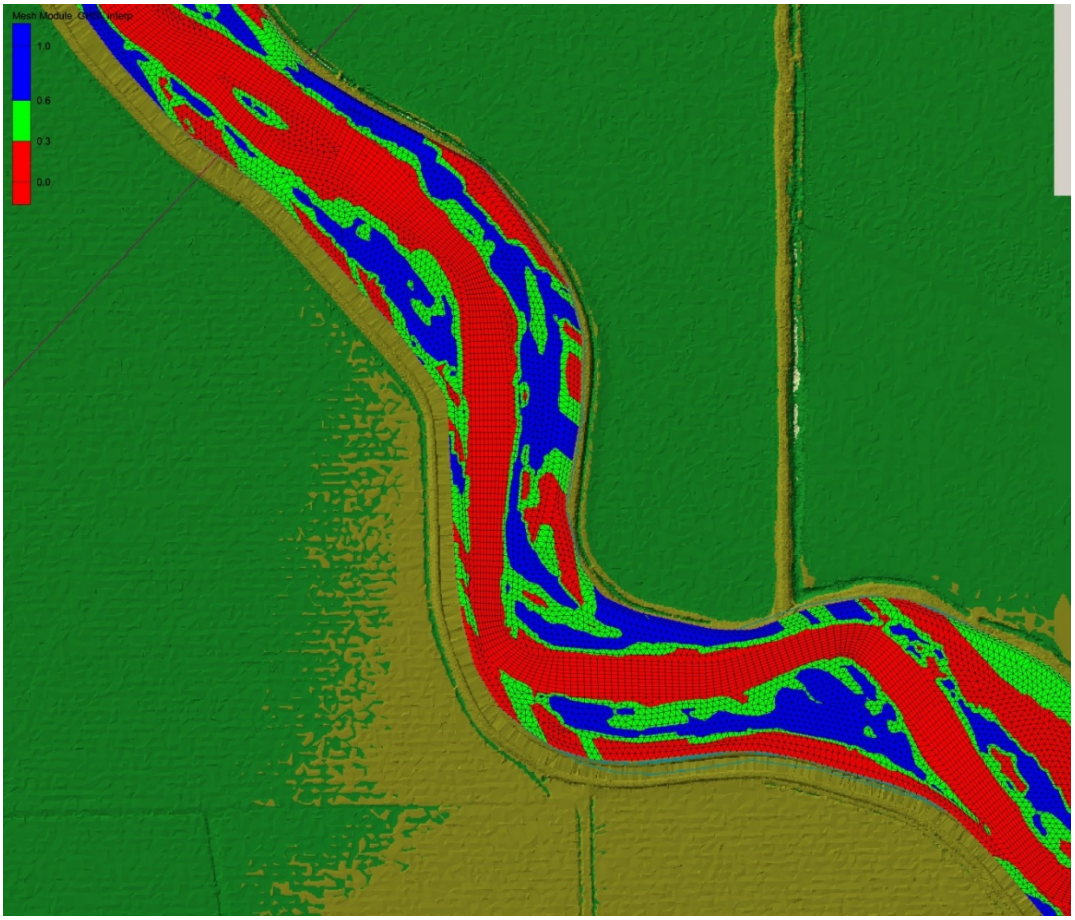
RECLAMATION
Managing Water in the West

Developing Tools for Efficient Handling of Data for Hydraulic Modeling and Habitat Analysis

Research and Development Office
Science and Technology Program
(Final Report) ST-2019-X9239-01



U.S. Department of the Interior
Bureau of Reclamation
Research and Development Office

October, 2019

Mission Statements

Disclaimer:
This document has been reviewed under the Research and Development Office Discretionary peer review process https://www.usbr.gov/research/peer_review.pdf consistent with Reclamation's Peer Review Policy CMP P14. It does not represent and should not be construed to represent Reclamation's determination, concurrence, or policy.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

T1. REPORT DATE: OCTOBER 2019	**T2. REPORT TYPE:** RESEARCH	**T3. DATES COVERED**

T4. TITLE AND SUBTITLE Developing Tools for Efficient Handling of Data for Hydraulic Modeling and Habitat Analysis	**5a. CONTRACT NUMBER** 19XR0680A1-RY15412019EN19239
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 1541 (S&T)
6. AUTHOR(S) Daniel Dombroski, ddombroski@usbr.gov	**5d. PROJECT NUMBER** ST-2019-X9239-01
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 86-68240
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Daniel Dombroski, Bureau of Reclamation, Denver Technical Service Center, 6th Ave & Kipling St., PO Box 25007, Denver, CO 80225	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Research and Development Office U.S. Department of the Interior, Bureau of Reclamation, PO Box 25007, Denver CO 80225-0007	**10. SPONSOR/MONITOR'S ACRONYM(S)** **R&D:** Research and Development Office **BOR/USBR:** Bureau of Reclamation DOI: Department of the Interior
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) ST-2019-X9239-01

12. DISTRIBUTION / AVAILABILITY STATEMENT
Final report can be downloaded from Reclamation's website: https://www.usbr.gov/research/

13. SUPPLEMENTARY NOTES

14. ABSTRACT Hydraulic and sediment modeling as well as habitat analysis are increasingly becoming a part of routine restoration project work at Reclamation. There is a demonstrated need for more proficient ways to handle complex data sets. Reclamation needs to invest in tool development that will enable practitioners to conduct analyses efficiently and effectively. The focus of this research proposal is in scoping a path forward for developing a streamlined habitat analysis workflow. Such a workflow would be greatly aided by the availability of user-friendly tools that can be used to conduct habitat analyses on a variety of projects in any region.

15. SUBJECT TERMS Hydraulic Modeling, Inundation Mapping, Habitat analysis, habitat suitability, habitat restoration

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT** U	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Daniel Dombroski
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U			**19b. TELEPHONE NUMBER** 303-819-6017

BUREAU OF RECLAMATION

Research and Development Office
Science and Technology Program

Sedimentation and River Hydraulics Group, Denver Technical Service Center, 86-68240

(Final Report) ST- 2019-X9239-01

Developing Tools for Efficient Handling of Data for Hydraulic Modeling and Habitat Analysis

DANIEL DOMBROSKI Digitally signed by DANIEL DOMBROSKI Date: 2019.10.07 20:02:54 -06'00'

Prepared by: Daniel Dombroski, Ph.D., P.E.
Hydraulic Engineer, Sedimentation and River Hydraulics Group, 86-68240

D. BRADLEY Digitally signed by D. BRADLEY Date: 2019.10.10 09:19:19 -06'00'

Peer Review: D. Nathan Bradley, Ph.D.
Hydraulic Engineer, Sedimentation and River Hydraulics Group, 86-68240

For Reclamation disseminated reports, a disclaimer is required for final reports and other research products, this language can be found in the peer review policy:
This document has been reviewed under the Research and Development Office Discretionary peer review process https://www.usbr.gov/research/peer_review.pdf consistent with Reclamation's Peer Review Policy CMP P14. It does not represent and should not be construed to represent Reclamation's determination, concurrence, or policy.

Executive Summary

River restoration projects are often motivated by a need to promote biological, geomorphic, and hydraulic processes that are important to the local and regional ecosystem. Increasing population numbers of a single (or several) species is often a primary driver for restoration projects, although the path to achieving end goals is often complicated by the constraints imposed by managed systems and heavily allocated resources. Due to the challenges and costs associated with such projects, stakeholders and administrators are increasingly expecting quantitative analysis be performed to project the value and effectiveness of restoration alternatives relative to existing conditions. Multi-dimensional hydraulic, sediment, and habitat modeling are now routinely requested by project offices in order to meet these demands. However, the workflow of processing the data into meaningful results remains tedious, often requiring advanced geospatial analysis and programming skills. Project work has supported the development of a suite of tools here at the TSC that are now available to process hydraulic model results and conduct habitat analyses. The tools are in the form of computer scripts, functions, and modules written in Python and Matlab programming languages, requiring users to be proficient in computer programming. The focus of this report is in technology transfer; specifically, evaluating proficient ways to make these tools available to a greater user base within Reclamation and abroad without requiring the users to hold advanced computer programming and geoprocessing skills. The outcome of the work is a summary of the current state of work and how it can be used to efficiently meet project objectives.

Contents

Background

Reclamation is at the forefront of many large-scale hydraulic, sediment, and habitat studies of river and reservoir environments throughout the Western States. As certain types of analyses become standard practice among these projects, Reclamation has a responsibility to operate efficiently and effectively. Without continued efforts to evaluate ways to improve workflow efficiency, valuable financial resources will be wasted.

In order to meet the need for hydraulic and sediment modeling and habitat assessment as a part of project scope, restoration offices are often budgeting a significant amount of resources to be spent on analyses that rely largely on methodology that is consistent from project to project. There are very few readily developed tools available to directly aid the habitat analysis process. Those that are available, such as the North Arrow Research Habitat Model (http://habitat.northarrowresearch.com/) have been cumbersome to use with tight constraints on input data. Other solutions, such as using geospatial tools like ArcGIS directly, offer great flexibility in analysis, but are slow and impractical for batch processing of large numbers of data sets. The result is that restoration practitioners are required to become highly proficient in geospatial software and computer programming, or engage an engineer that is, in order to handle processing of hydraulic and sediment data or conduct habitat analyses. The workflow demands many hours of labor, particularly in cases where many variables (e.g., design alternatives) or environmental conditions (e.g., restoration flow rates) are being evaluated. Because the underlying methodology of typical habitat analyses are analogous from project-to-project, the workflow would be more efficient with standardized procedures.

The benefit is in finding ways to dramatically improve the efficiency of workflow processes, thereby decreasing costs associated with common procedures that share relevancy across a broad range of project types. It is anticipated that using specially developed analysis tools for handling hydraulic and habitat data will cut labor costs by approximately 50-80% relative to either writing computer code or using ArcGIS spatial capabilities.

Summary of Work Outside Reclamation

According to the review by Melcher et al. (2018), the principles of habitat modeling can generally be divided into two categorizations:

1. Empirically-based habitat suitability models, which link descriptions of the abiotic conditions to availability of habitat for biotic systems
2. Process-based population or bioenergetics models, which describe biological processes that relate population and energy dynamics to functions such as feeding and growth.

There may exist further differentiating features between habitat modeling approaches, such as those that are tasked with predicting habitat needs to sustain a prescribed population (e.g., ESHE, described below). Herein is a brief overview of modeling and processing tools developed outside of Bureau of Reclamation, some of which are readily available.

The following italicized descriptions are excerpts taken directly from documentation provided by the developers and publishers of the respective utilities.

Habitat Modeling Toolkits

Emigrating Salmonid Habitat Estimation (ESHE)

We describe the creation of a deterministic simulation model that overlays modeled river hydrology with simulated cohorts of juvenile Chinook salmon, allowing for calculation of benefits provided by seasonally-inundated floodplain habitats. By tracking the movement, growth, and survival of juvenile Chinook salmon in a simulated system, we can provide estimates of rearing habitat needs in time and space, helping prioritize floodplain restoration locations.

The ESHE model used to calculate required Suitable Habitat simulates stationary growth (rearing) and downstream movement (emigration) of individual daily groups (cohorts) of juvenile spring-run and fall-run Chinook salmon (Oncorhynchus tschawytscha). The model tracks their numbers (abundance), average speed, size, the amount of territory needed per fish (territory size), and ultimately the amount of suitable habitat required to sustain the number of juvenile salmon present within a model reach. Model outputs provide daily estimates of the number of juvenile spring-run and fall-run Chinook salmon present in each model reach and the required area of suitable habitat needed to support them throughout the rearing and emigration period.

URL: https://www.fishsciences.net/

Producer: Cramer Fish Sciences

Reference: Hinkelman, T., Pergman, P., Merz, J., Emigrating Salmonid Habitat Estimation (ESHE): A Web-Based Tool for Estimating Habitat Needs for Outmigrating Juvenile Salmonids, AFS Annual Meeting, Aug 16-20, 2015, Portland, OR.

Target Species: Chinook Salmon

Download Available: No

Arc Habitat Suitability Index (ArcHSI)

ArcGIS-based HSI

Arc Habitat Suitability Index (ArcHSI) is a geographical information system (GIS) model that estimates the ability of an area to meet the food and cover requirements of an animal species. The components and parameters of the model occur in tables and can be easily edited or otherwise modified. ArcHSI runs on personal computers with the full installation of ArcGIS (Version 8.2+). ArcHSI is a tool intended to inform planners of the probable impacts on wildlife for the alternatives they develop, and it augments, rather than supplants, the expertise of resource specialists with site-specific knowledge. System requirements are platform specific, but a typical system requires 512 megabytes RAM and a 1 gHz CPU. A Unix ArcINFO version is also available. We have included a demonstration that provides a step-by-step tour of the software, including setting up the first run; interpreting, formatting, and analyzing output; and identifying and solving the most common problems experienced by new users.

URL: https://www.fs.fed.us/rm/forest-grassland-lab/products/arc-habitat-suitability/

Producer: US Forest Service

Target Species: Terrestrial Elk and Deer

Dependencies: ArcGIS

Inputs: Species and vegetation info files, ArcINFO GIS coverages

Outputs: ArcINFO coverages, summary table

CASiMiR Model

Fuzzy logic model

CASiMiR is unique in that it makes use of physical and biological parameters through the application of expert knowledge using a fuzzy logic based rule system (JORDE et al. 2000, SCHNEIDER 2001), but can also use the standard preference function method as well. The main advantages of using habitat simulation models are that:

- *The ecological condition of an aquatic ecosystem is directly coupled with the living conditions of the typical resident species.*
- *The use of habitat models allows for the effects of changing flow rates and structural characteristics to be account for, and to some extent, can be used to predict their impacts.*
- *Changes to the flow rate result primarily in impacting the water depth, flow velocity, and substrate conditions, all of which are major factors in determining the habitat suitability and can be directly evaluated with numerical models.*
- *Due to the direct relationship between habitat conditions and flow rate, a quantitative basis can be established whereby an overall ecological assessment of the habitat requirements can be performed.*

URL: http://www.casimir-software.de/ENG/aufbau_eng.html

Producer: SJE Ecological Engineering, University of Stuttgart, and Umwelt Buro

Target Species: Fish, Benthos, Vegetation

Inputs: flow velocity, water depth, substrate, submergence time, changes in water surface, species-specific habitat requirements

inSTREAM Model

Agent-based model

inSTREAM is an individual-based model (IBM) of trout in a stream environment; it predicts how trout populations respond to many kinds of environmental and biological change. The simulated *environment includes spatially and temporally varying in hydraulic conditions (depth, velocity, cover providing velocity shelter), temperature, turbidity, and food availability. The model trout adapt to changing conditions mainly by selecting which habitat to use, making a tradeoff between growth rate and mortality risk. Trout feed and grow, experience various kinds of mortality, and reproduce.*

URL: http://www2.humboldt.edu/ecomodel/instream.htm

Producer: Humboldt State University

Target Species: Trout

Inputs: time-series of flow, temperature, and turbidity; barriers to passage

Physical HABitat SIMulation (PHABSIM)

HSI-based model

The purpose of the Physical Habitat Simulation System (PHABSIM) is to simulate a relationship between streamflow and physical habitat for various life stages of a species of fish or a recreational activity. The basic objective of physical habitat simulation is to obtain a representation of the physical stream so that the stream may be linked, through biological considerations, to the social, political, and economic world.

The two basic components of PHABSIM are the hydraulic and habitat simulations of a stream reach using defined hydraulic parameters and habitat suitability criteria. Hydraulic simulation is used to describe the area of a stream having various combinations of depth, velocity, and channel index as a function of flow. This information is used to calculate a habitat measure called Weighted Usable Area for the steam segment from suitability information based on field sampling of the various species of interest.

URL: https://www.usgs.gov/software/physical-habitat-simulation-phabsim-software-windows

Producer: USGS

Target Species: Varietal fish species

Inputs: Flow and survey data

Outputs: Habitat vs discharge functions for each target species and life stage

Download Available: Yes

Diversity based HABitat SIMulation (DHABSIM)

Eco-Environmental Diversity (EED) index

DHABSIM (Diversity based HABitat SIMulation) is an easy-to-use two-dimensional habitat calculator develop by Professor Masahiko Sekine at Yamaguchi University. The solver imports geometry, velocity, and depth from other flow calculation solvers implemented in iRIC, and then *uses that information to calculate habitat conditions for small to medium sized rivers without using Habitat Suitability Indices (HSI) for specific fish species. Based on the diversity-based algorithm and the physical flow inputs, the solver outputs a habitat distribution map. The solver can be used to compare a variety of flows, or to project the impacts of potential changes in river discharge or other physical variables.*

URL: https://i-ric.org/en/solvers/dhabsim/

Producer: Professor Masahiko Sekine / Yamaguchi University

Target Species: Varietal fish species

Inputs: Hydraulic modeling results, Vegetation mapping

Outputs: Map of EED

Download Available: Yes

Habitat Model Software

HSI-based model

The Habitat Model Software is an open, flexible ecological simulation modeling environment. It calculates the area of usable habitat for a particular organism based on a set of physical relationships. Originally developed for modeling anadromous fish habitat, the software is entirely generic and can be used for any species and lifestage.

The software takes either tabular or spatially explicit inputs that represent physical conditions (e.g. depth, velocity and substrate) and combines them using user-defined habitat suitability curves. Typically these curves are obtained from empirical studies published in the literature and take the form of continuous or categorical utilization curves. A rich user interface allows users to create and manage the curves, link the curves to their data and then batch run simulations.

The core habitat model is free and open source under the GPL license. It uses the open source Geospatial Data Abstraction libray (GDAL) for spatial operations.

URL: http://habitat.northarrowresearch.com/

Producer: North Arrow Research

Target Species: Varietal fish species

Inputs: Orthogonal rasters of hydraulic and physical conditions, tabular habitat suitability criteria

Outputs: Maps of combined habitat suitability index

Download Available: Yes

Riverine Community Habitat Assessment and Restoration Concept (RCHARC)

Comparative Habitat Distribution Model

RCHARC compares underlying patterns of depths and velocities for alternative scenarios to an ideal reference system and uses the results as the basis of the community-level impact analysis. The relative value of the habitat for each species can be determined by how much of the frequency distribution falls within its suitability curve, or restated, the composition of this hypothetical group of species is at least partially determined by the distribution of depth and velocity. The degree of impact is approximated by the degree to which the physical habitat changes between the target and standard systems.

URL: N/A

Producer: US Army Corps of Engineers (USACE)

Reference: Nestler, J. M., Schneider, L. T., Latka, D., and Johnson, P. N. (1995). "Physical habitat analysis using the Riverine Community Habitat Assessment and Restoration Assessment and Restoration Concept (RCHARC): Missouri River case history," Technical Report EL-95-18, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS. https://www.researchgate.net/publication/279425014_Physical_Habitat_Analysis_Using_the_Riverine_Community_Habitat_Assessment_and_Restoration_Concept_RCHARC_Missouri_River_Case_History

Target Species: Varietal fish species

Inputs: Cross-sectional based velocity and depth distributions

Outputs: Ranking of habitat alternative results

Download Available: No

Habitat Modeling Software (HaMoSoFt)

Pseudo-3D HSI Model

HaMoSoFt attempts to remedy some drawbacks of commonly used weighted usable area approaches to habitat classification by considering the three-dimensionality of flow environments. The development of the model was motivated by increasing acknowledgement that ecological needs of fish are a function of flow velocity at varying depths. The model calculates Usable Volume of habitat for the species of interest.

URL: N/A

Producer: University of Natural Resources and Applied Life Sciences, Vienna

Reference: Mouton, A.; Meixner, H.; Goethals, P.; De Pauw, N.; Mader, H., 2007. Concept and application of the usable volume for modeling the physical habitat of riverine organisms. River. Res. Applic. 23: 545-558. https://onlinelibrary.wiley.com/doi/pdf/10.1002/rra.998

Target Species: Varietal fish species

Download Available: No

Programming Resources

Swarm Simulation System

Toolkit for building multi-agent simulations

Swarm is a multi-agent software platform for the simulation of complex adaptive systems. In the Swarm system the basic unit of simulation is the swarm, a collection of agents executing a schedule of actions. Swarm supports hierarchical modeling approaches whereby agents can be composed of swarms of other agents in nested structures. Swarm provides object-oriented libraries of reusable components for building models and analyzing, displaying, and controlling experiments on those models. Swarm is currently available as a beta version in full, free source code form. It requires the GNU C Compiler, Unix, and X Windows. More information about Swarm can be obtained from our web pages, https://www.santafe.edu/research/results/working-papers/the-swarm-simulation-system-a-toolkit-for-building.

SRH-2D Development and Analysis Tools

ArcGIS Tools

D. Nathan Bradley
Sedimentation and River Hydraulics Group
Bureau of Reclamation Technical Service Center

Over the course of 2019, D. Nathan Bradley have developed a suite of ArcGIS tools written in Python to develop and analyze SRH-2D hydraulic models. The suite is divided into three ArcGIS Toolboxes, the “2dm Mesh Toolbox,”, the “SRH Results Toolbox,” and the “Trinity Habitat Calculations Toolbox,” which contains habitat analysis tools that are specific to the Trinity River in Northern California. The tools run within ArcMap, which provide a graphical user interface to the Python code. A “command line” Python script interface to each tool is also provided, which can be adapted by technical users to automate input to the tools. Processing large numbers of SRH-2D models results may be more efficient using this method.

The 2dm Mesh Toolbox contains tools intended for use during the development of an SRH-2D hydraulic model that uses a .2dm file to define the model mesh. These tools are not applicable to SRH-2D simulations that are developed and executed within the Aquaveo SMS software. The toolbox contains the following tools:

- Assign Elevations to Mesh – Assigns elevations from an elevation model in the form of a DEM, TIN, or ArcGIS Terrain to a model mesh. This tool is intended to replace the SMS Z Solver tool and it is at least 100 times faster than the Z Solver tool when assigning elevations to large meshs.

- Assign Material IDs to Mesh – Maps material IDs stored in a polygon shapefile to the elements of a model mesh. Material IDs are used to assign hydraulic roughness (Manning’s n) in the SRH-2D .SIF file.
- Compute Mesh Connectivity – This tool is a pre-processing step needed for an SRH Results Toolbox tool that computes gradients across the mesh from hydraulic model results.
- Convert Mesh to Shapefile – Creates an ArcGIS polygon shapefile representing the model mesh. This is useful for sharing the model mesh with people who do not use SMS and is the basis of the SRH-2D model results display and analysis tools.

The SRH Results Toolbox contains tools for calibrating, visualizing, and analyzing SRH-2D hydraulic model results stored in the SHRC format.

- Convert SRH Results to Shapefile – Creates a polygons shapefile of the model hydraulics. This is useful for visualizing results on the model mesh (as opposed to interpolating results to a raster) and for sharing model results. The shapefile of model results is the basis for the rest of the tools in this toolbox.
- Compare Model Hydraulics to Observations – Compares observations of river hydraulics stored in ArcGIS point shapefiles, such as water surface elevation, depth, water’s edge, or velocity to hydraulic variables computed by the model. This tool in intended for use during model calibration and makes it easy to visualize how the model hydraulics differ spatially from the observed hydraulics.
- Compare Surface to Survey Points – Compares an elevation model in the form of a DEM, TIN, or ArcGIS Terrain to observations of elevation stored in a point shapefile. This tool is useful for estimating the amount error in a surface derived from LiDAR survey by comparing it to high precision field GPS measurements. The probability distribution of these elevation residuals can be used to evaluate model calibration.
- Compare SRH-Results – Differences two model runs that are based on the same model mesh. This tool can be used to analyze how a change of hydraulic roughness (such as during model calibration) changed the model water surface or to compare how the hydraulics vary with the model discharge.
- Compute Shields Stress – Computes a map of Shields Stress, a measure of incipient sediment motion, for a range of grain sizes from the hydraulic model results.
- Compute Gradient of SRH Result Field – Computes the gradient of a hydraulic variable such as water surface elevation or shear stress and stores the result in a polygon shapefile.

The Trinity Habitat Calculations Toolbox computes salmonid habitat metrics, either Weighted Useable Area (WUA) or Fish Capacity, that are specific to the Trinity River in Northern California. These tools could appropriate for use in nearby or similar watersheds and could be easily adapted to incorporate new metrics or different habitat suitability curves.

- Compute Distance to Cover – Calculates the distance between a model mesh element and the nearest fish escape cover polygon. Distance to cover is one of the variables in the Trinity River habitat metrics.
- Compute Habitat – Computes either the WUA or Fish Capacity metric from SRH-2D SRHC results for the fry and pre-smolt life stages of either Chinook or Coho salmon.
- Batch Compute Habitat – Computes either the WUA or Fish Capacity metric for a group of SRH-2D model results. Useful for computing habitat over a range of modeled flows.
- Compute Habitat vs. Flow – Creates a spreadsheet of habitat vs. flow curves for sites of interest defined in a polygon shapefile.

Python Scripting Tools

Daniel Dombroski
Sedimentation and River Hydraulics Group
Bureau of Reclamation Technical Service Center

Through support of multiple projects, Daniel Dombroski has developed a suite of Python modules with functions that can be invoked while building custom scripts for batch processing of model results.

Module convert – contains functions that convert one type of data to another and perform various analysis operations

- srh2dToPoint_convert(`srh_filepath`, `out_dir`, `sp_ref`)
  Produces a point shapefile with the srh2d results. Can also read other types of point text files. The current version of the routine does not use any ArcPy routines, but instead is dependent on Numpy and OGR. Will return a string with the output filepath. Return will be empty and no point shapefile will be created if input file is found empty.
- clipShape (`shape_filepath`,`clip_filepath`,`shape_filepath_out`)
  Uses ArcPy
  Clip shapefile
- pointToTin_convert (`shape_filepath`,`clip_filepath`,`shape_filepath_out`)
  Uses ArcPy
  If it is desirable to produce rasters from point srh2d data, necessary to first produce a continuous tin surface from the srh2d data, and then produce the raster from the tin. This is because of the variably spaced nature of the point data. Care should be taken in spacing max_edge_length appropriately (e.g., on the order of the spacing of the largest point spacing in the shapefile). Default value of max_edge_length is 100 if none given.
- tinToRaster_interp (`tin_filepath`,`out_rast_filepath`,`rast_cell_size`,`conversion`)
  uses ArcPy
  Recommend using "LINEAR" linear interpolation method. Seems to produce more accurate results at the edge of the wetted water surface. Using “NATURAL_NEIGHBORS" seemed to artificially stretch or exagerate the wetted area of the raster. If saving to system folder, extension ".tif" should be added to out_rast_filepath. If saving to geodatabase, no extension is needed.
- rasterize_grid (`grid_filepath`, `grid_attr`, `cell_size`)

uses ArcPy
This can work ok if input grid is uniform spacing such that raster cell_size is set to the grid spacing. Nonuniformly spaced data does not work well for conversion to raster - need to convert to continous surface (eg. TIN) first

- mask_raster (`rast_filepath, clip_filepath, masked_rast_filepath`)
  uses ArcPy
  Intersect raster with raster or polygon that delineates valid data area
  rast_dir provides directory to folder where raster rast_file resides.
  clip_filepath provides directory to folder where mask resides
  masked_rast_filepath is output save directory
- depth_filter (`input_object,depth`)
  Filter Non-wetted Cell Data from Object.
  - Input 'input_object' can either be a pandas DataFrame (e.g., SRHhydraulics data) or a class object with attributes (e.g., class fio.SRHmesh object).
  - Returns 'filtered_object' which contains a copy of the 'input_object' with each attribute list filtered according to 'depth' values greater than zero.
  - Assumes that each object attribute list is indexed the same as the input depth list.
  - This seems a bit tedious; there's probably a more concise way to perform the filtering.
- spatial_classify (`shape_object,input_object`)
  Classify Object Spatially
  Input 'input_object' can either a pandas DataFrame, or a class object such as fio.Habitat, as long as it contains an 'xy' column or attribute.
  Returns 'classified_object' which contains either (a) pandas DataFrame with hierarchical indexing that groups according to classifications delineated in the 'shape_object' input OR (b) a list of type input objects, each of which is populated by the values in the type of input object that correspond to delineated classifications within the 'shape_object' object.
  The algorithm uses the 'points_in_polygons' function to determine what points within the type of input objects coincide within what polygon features in the 'shape object'.   The only requirement of the input object to be classified is that it contain a .xy attribute that gives spatial coordinates of each point.
  Inputs:
  shape_object is a fio.Shapedef object populated by fio.readshp
  input_object is any object (e.g., fio.SRHhydraulics) with {x,y} coordinates given in an object.xy attribute
- points_in_polygons (`attrib,vertex_xy,point_xy`)
  Finds points in polygons based on input polygon names 'attrib' and corresponding defining polygon vertices 'pg_vet'. Input 'XY' contains the {x,y} location of the points to be distributed within the polygons
  Returns dictionary 'pnt_id_attrib' that contains the unique polygon attrib names and the indices of the points XY that reside within the objects.
- wtd_stats (`object_list,attributes,weights_name=None`)

wtd_stats computes a weighted statistics of attributes in each entry in object_list using given weights. Useful for computing reach average statistics with the cell area being the weight for each cell.
object_list does not actually need to be a list; it can be a single object instead and the function can handle either case without specication
Returns a pandas data frame
Note to filter out any unwanted values first (e.g., zero depths)

- compute_cdf `(object,attribute_list,weights_in=None,filepath=None)`
  compute cumulative distribution function as cumulative sum of histogram.
  NOTE: using density = True in numpy.histogram doesn't seem to produce normalized results. Possibly because of weighting?

Module fio – contains functions for data input and output and perform various analysis operations

- lin_int `(pt1,pt2)`
  A = lin_int(pt1,pt2) calculates the line integral between points 1 & 2, where pt1 & pt2 are of the format {x,y} = [x y]. Formula derived by integrating y=mx+b
- SRHmesh.read2dm
  class object SRHmesh contains computational mesh information
  Read a sms .2dm file to load srh2d computational mesh information
  returns:

elements_df	(string list)	['E3T', 'E4Q', ...]
elements_id	(int list)	[1, 2, 3, ...]
elements_nd	(int list of lists)	[[1, 2, 3], [4, 5, 6, 7], ...]
nodes_id	(int list)	[1, 2, 3, ...]
nodes_xy	(float list of lists)	[[1802754.73, 526880.026], [1802742.16, 526895.553], ...]
area	(float list)	[283.87457275390625, 104.23651123046875, ...]

- readsrh `( srh_filepath )`
  Reach SRH Results Text File, return contents in SRHydraulics pandas Dataframe
  Returns:

attrib	(list of strings)	string descriptors from SRH file header columns (NOT USED)
id	(list of integers)	point id number
x	(list of floats)	point x coordinates
y	(list of floats)	point y coordinates
z	(list of floats)	bed elevations
wse	(list of floats)	water surface elevation
wd	(list of floats)	water depth
vx	(list of floats)	x-direc velocity
vy	(list of floats)	y-direc velocity
v	(list of floats)	velocity magnitude, vx vy resultant
frd	(list of floats)	froude number
strs	(list of floats)	bed shear stress estimate

  If sediment results included:

d50	(list of floats)	D50 mm

erodep (list of floats) Erosion or deposition depth
conc (lsit of floats) Sediment concentration ppm

- Shapedef.readshp (self,shp_filepath,shp_attrib_name)
  Class Shapedef contains geometry and attributes defining shapes (e.g., polygons)
  Read shapefile
  inputs:
  shp_filepath: full path to shapefile for reading
  shp_attrib_name: the column heading name in the attribute table that contains the attributes to be read.
  Returns:
  attrib: attribute table entry for each element in column shp_attrib_name
  xy: xy coords at point vertices defining each shape element
  shptpe: flag indicating the type of each shape element
- loadHSI (direc,sheet_name)
  Load HSI (habitat suitability index) data from excel spreadsheet
  Uses fio.readxls function to return data from each worksheet organized into a dictionary. Then the data from the dictionary is parsed according to the desired species classification (i.e., sheet name) into depth-based and velocity-based HSI data lists. Assumptions are that the species classification corresponds to the worksheet names in the file and that the data is organized into column format within each worksheet with headers denoted by 'Velocity (ft/s)', 'VHSI', 'Depth (ft)', 'DHSI', etc.
- readxls (direc)
  Read an excel xls or xlsx workbook and return the contents as a data dictionary organized according to keys corresponding to worksheet names. Each worksheet data is contained in a nested dictionary with column headings.
  - Data is assumed to be in column format
  - If no text-based heading is detected, the dictionary key is set based on the column index
  - If a text-based heading is detected, the key is set according to the first row of each column. If more than one row of headers exist, error will result.

  Example data dictionary returned would be:
  data:
  {'Sheet1': {'Col1': [1, 2, 3, 4], 'Col2': [0.1,0.2,0.3,0.4]}, ...
  'Sheet2': {'Col1': [1, 2, 3, 4], 'Col2': [0.1,0.2,0.3,0.4]}}
  where data contains two worksheets named 'Sheet1' and 'Sheet2' which each contain columns of data with the headers 'Col1' and 'Col2'

Module georiv – contains functions that perform geomorphic and riverine-specific analyses

- Lc2dm (filepath_2dm)
  [Lc4Q,Lc3T] = Lc2dm(filepath) reads in a *.2dm file identified by the string FILEPATH, and calculates the characteristic mesh resolution Lc for the quadrilateral (E4Q) and triangular (E3T) elements, respectively. The characteristic mesh resolution is defined as the following:

Lc4Q = sqrt(sum(A_E4Q)/N_E4Q)
Lc3T = sqrt(2*sum(A_E3T)/N_E3T)

Thus, the metric Lc4Q (Lc3T) is essentially an idealized lengthscale of a square (triangular) element of average area in the mesh. The idea is to gain a quantitative (albeit, loose) metric for the grid resolution of a given mesh.

- morphology (dep_filepath,vel_filepath,str_filepath,frd_filepath,z_filepath)
  compute metrics of river morphology based on hydraulic variables
- compute_centerline_geom(IN_polyline,pnt_filepath,tolerance=None,filepath_2dm=None
  Uses ArcPy
  compute_centerline_geom computes physical attributes of point SRH hydraulics results based on a centerline delineated polyline.
  - SRH results file is specified by input pnt_filepath
  - Delineated centerline is specified by input IN_polyline

  The function returns a 2d numpy array meas_z_wse_array that contains the incremental length along the centerline (column 1) and corresponding bed slope (column 2) and wse slope (column 3). Also returned are the total reach length reach_length, the reach averaged bed slope z_slope_reach, wse slope wse_slope_reach length, bed slope, and wse slope over the entire polyline.

  Optional input tolerance specifies search radius to look for points along the delineated centerline.
  IF no tolerance is given, then a filepath to a 2dm file MUST be given, and the tolerance is then calculated as:
  tolerance = Lc4Q/2, where Lc4Q is a characteristic mesh lengthscale returned from the function georiv.Lc2dm
- compute_segment_geom (IN_polyline,pnt_filepath,tolerance)
  Uses ArcPy
  computes physical attributes of segments within delineated polyline.
  Returns the length, bed slope, and wse slope of each segment within the polygon.
- compute_seg_azi (IN_polyline)
  Computes azimuth of line segments using clockwise-from-due-North convention
  Returns dataframe with array of azimuth anges, {X,Y} coordinates of segment midpoint and length of each segment.
  IN_polyline must be a valid .shp shapefile. It can contain multiple disconnected line segments - for example, delineating both channel centerline and side channels
- compute_bed_slope (centerline,pnt_data,max_xsec_l,tolerance)
  compute_bed_slope computes approximate bed slope of segments delineated by centerline through pnt_data.
  Minimum bed elevation is found at vertices of centerline by cutting cross-sections
  Inputs
  pnt_data must contain an xy coordinate and z elevation attribute
  max_xsec_l is the extent of cross-section length to create

tolerance is how far to search for points along the cross-sections
Outputs
slope_df is a pandas dataframe containing x,y,slope data
NOTE: Best to err on side of keeping max_xsec_l and tolerance small here
to prevent accidentally picking up unwanted points from differently
classified regions. Recommend maximum values are:
max_xsec_l = 10 x Lc2dm[0]
tolerance = Lc2dm[0]

- compute_pnt_hydraulics `(pnt_data,xy_data,tolerance)`
  compute_pnt_hydraulics extracts the nearest data points to a set
  of given xy data points.
  pnt_data can either be a complete filepath to data file OR a pandas
  dataframe such as loaded from fio.readSRH
  xy_loc should be a list of x,y coordinates or a panas dataframe
  containing columns labeled x and y
- compute_xsec_hydraulics `(tables)`
  compute_xsec_hydraulics computes hydraulic attributes from table
  data defining cross-sections cut through SRH results output from
  compute_xsec function. Input "tables" can either be a list or
  single object consisting of strings containing the directory
  paths to saved dbf tables OR it can be a Pandas dataframe object
  or list of pandas Dataframe objects.
  Each table must have a 'MEAS' column the contains the length
  along the path.
  Calculated arrays are:
  Wlist = wetted width at each station
  Vlist = cross-sectional average velocity at each station
  WDlist = cross-sectional average depth at each station
  Qlist = discharge at each station
  Llist = cumulative length along channel at each station
  WSElist = water surface elevation along approx thalweg
  Zlist = bed elevation along approx thalweg
  Xlist,Ylist = {X,Y} coordinates of approx thalweg
  Output is returned as a pandas Dataframe object with attribute
  columns corresponding to above calculated arrays.interp_xsec_srhc
- interp_xsec_srhc `(xs_filepath,srhc_filepath,tolerance,resolution)`
  interp_xsec_srhc interpolates point data to cross-sections defined by
  polylines within a shapefile (such as exported from HEC-RAS). The point
  data comes from an srh2d results file (SRHC or SRHN file)
  - input xs_filepath is
  - input srhc_filepath is full path to srh2d results file
- interp_xsec
  uses ArcPy
  "interp_xsec" interpolates point data to cross-sections defined by
  polylines within a shapefile (such as exported from HEC-RAS).

- Input "xs_filepath" is the complete filepath to a shapefile with an attribute table containing polylines that define cross-sections. It is assumed that each row in the attribute table defines a distinct cross-section
- Input "pt_filepath" is a filepath to .shp point shapefile
- Input "tolerance" is the search radius to use in associating cross-section route with points from the shapefile.

The routine attempts to extract Stationing from the cross-section input file and include the cross-section station in the export table. If that doesn't work, the routine tries to conduct the cross-sectioning without including the stationing

- compute_xsec `(centerline,pnt_data,tolerance,max_xsec_l,resolution,clip=None)` uses ArcPy

  "compute_xsec" calculates cross sections at mid-points of line segments within channel centerline that is delineated within SRH results data.

  - Input "centerline" is either a dataframe with columns containing x,y,length,azimuth (as returned from compute_seg_azimuth) OR is a filepath to a polyline that delineates the route of the cross-sections.
    Cross sections are created perpendicular to the centerline at the midpoint of each segment within the centerline.
    Thus, the number and location of cross-sections can be controlled by the placement of vertices within the centerline that define segments.
    "centerline" must point to a .shp shapefile.
  - Input "pnt_data" is either (a) filepath to .shp point shapefile or (b) a SRHhydraulics class with attributes returned from module fio.

  Logic is built in to automatically detect what format of "pnt_data" is given and how to handle the processing. If "pnt_data" is a string pointing to a .shp shapefile, then arcpy routines are used to locate features along the cross-sections. If "pnt_data" is a SRHhydraulics class object, scipy interpolation routines are used to collect nearest neighbor data values along each cross section. If "pnt_data" is a pandas DataFrame, the data access is handled such that scipy interpolation routines can be used.

  - Input "tolerance" is the search radius to use in associating cross-section route with points from the shapefile.
  - Input "max_xsec_l" is the maximum allowable extent of a calculated cross-section, most useful for sinuous systems in order to avoid multiple intersections with channel.
  - Input "resolution" is the approximate spacing between points in the calculated cross-section from which to search within tolerance.
    This is not used if arcpy is used (if centerline is a polyline)
  - Input "clip" (OPTIONAL) provides a path to a .shp shapefile polygon

used to spatially filter out unwanted data. Only computed cross-section points and points from "pnt_data" that reside inside the "clip" polygon will be included in the returned cross-sections, regardless of the specified "tolerance" and "max_xlec_l". NOT YET FULLY IMPLIMENTED!!!

- Output includes the {"X","Y"} coords of the centerpoint, length "L", and channel orientation "chan_orient" of each segment within the channel centerline. Output "xsec_tables" is either (a) a list of the folder locations where the cross-section .dbf data tables are to be written OR (b) a panel data object containing data frames with cross-section data.

- nearest_neighbor `(XY_coords_data,XY_coords_loc,tolerance=None)`

  "nearest_neighbor" finds the index of each point in "XY_coords_data" that is nearest to each point in "XY_coords_loc". If the distance to the nearest point is greater than tolerance (OPTIONAL), then the point is disregarded.

  Returned are:

  - "i_data" which contains the indices of each point in "XY_coords_data" that are the nearest neighbor of each point returned in "XY_coords_loc"
  - "distance", which is the distance between each point in "XY_coords_loc" and the the nearest neighbor point in "XY_coords_data"
  - "XY_coords_loc", which is the same as the input "XY_coords_loc" but with points removed that did not meet the tolerance criteria.
  - "i_success" are the indices of XY_coords_loc that resulted in a successful point in "XY_coords_data"

  Note on optional "tolerance" input: If a tolerance is given as input and no data points from "XY_coords_data" are within the tolerance distance of the points in "XY_coords_loc", then the returned arrays "i_data", "distance", and "XY_coords_loc" will all be empty!

- filter_feature `(Infeature)`

  uses ArcPy

- get_polygon_attr `(XYcoords,feature,attrStr)`

  uses ArcPy

  get_polygon_attr associates the attribute "attrStr" within the feature class "feature" at the coordinate locations "XYcoords".

- write_dat `(data,title_header,data_header,filepath)`

  output data to file
  "data" is expected to be a concatenation of 1D arrays and/or lists
  example: data = [X,Y,W]
  "title_header" is a string header that appears at top of file
  "filpath" is output filepath where to the data is to be written, appears as string in line 2
  "data_header" is string header that appears above the actual data

Module plotting – contains functions that perform data plotting and display

- plot_hist_stack `(object_list,attribute_list,log=0,savefile=0,filename=None)`

plot_hist_stack creates a figure with length of attribute_list subplots
 where each subplot contains a stacked histogram of the data attribute data
 in each object in object_list. Object_list is a list of objects that each
 contain data with attributes named in attribute_list. 'log' (optional) is
 set to 0 or 1 to control whether the y-axis of the bar plot is plotted on
 log scale or linear scale. Optional arguments savefile (0 or 1) and
 filename (string) control output of figure file to disk.
  Example:
   object_list = SRHhydraulics_classified
   each entry in object_list is a fio.SRHhydraulics object with attributes
   in attribute_list
   attribute_list = ['v','wd','frd']

Matlab Scripting Tools

Daniel Dombroski
Sedimentation and River Hydraulics Group
Bureau of Reclamation Technical Service Center

Through support of multiple projects, Daniel Dombroski has developed a suite of Matlab functions that can be invoked while building custom scripts for batch processing of model results.

- write2dm.m
  write2dm(filepath,filename,elements,element_nds,nodes_xy,material,elevation)
  writes the 2dm file named filename.2dm to disk in location filepath using
  the input cell array ELEMENTS containing the element definitions (e.g., E3T,
  E4Q, etc.), the cell array ELEMENT_NDS containing the node id's
  comprising each element, the numeric array NDS_XY containing the
  {x,y} locations of each node in the mesh, numeric arrays 'MATERIAL'
  defining the material index of each element, and numerical array
  'ELEVATION' defining the elevation at each node.
  Note that write2dm cannot handle nodestrings or the parameter definition
  portion of the 2dm file format that appears below the node definitions.
  Therefore this portion will have to manually entered or copied and pasted
  from an exesint 2dm file.
- readSRH.m
  [varargout] = readSRH(filepath,varargin) reads in a results data file
  output from SRH-2D, where the string FILEPATH contains the full path to
  the file to be loaded, and VARARGIN specifies what variables are to be
  read from file. Options for VARARGIN include any combination of 'XY',
  'ELEV','DEPTH','WSE','VMAG','FROUDE', or 'STRESS'. If the SRH-2D mobile bed
  sediment model was implemented, the output options may also include 'D50',
  'EROSION', and 'CONCENTRATION'. If 'XY' is input, the % corresponding
  output array will have 2 columns containing the x & y data points. All
  other inputs correspond to a single column output.
  Format: readSRH can be used to load results files in either SRH or SMS

formats. From the SRH-2D manual: SMS format stores all variables at the mesh (element) centers but SRH format stores all variables at the mesh nodal points. So its important to know how the information being loaded by readSRH was recorded.

- readSRH_MonitorLineInlet
  [time, Qw, Qs, Qi] = readSRH_MonitorLineInlet(monitoringLineFile)

  Read inlet monitoring line file. Algorithm checks every data point individually from the file due to an issue where SRH2D omits the 'E' in scientific notation format for very small numbers.
- readSRH_MonitorLine
  [time, Qw, Qs, ssC, meanWSE, meanZbed, minZbed, Qi] = readSRH_MonitorLine(monitoringLineFile)
  Read monitoring line file. Algorithm checks every data point individually from the file due to an issue where SRH2D omits the 'E' in scientific notation format for very small numbers.
- plot_z_points.m
  plot_z_points(x,y,z) plots arbitrary point values z at locations x,y using marker colors that are mapped to the range of values in set z. The algorithm checks if the data contains both positive >0 and negative <0 values. If the data contains both positive and negative values, the bins are setup such that zero is a bound between two bins - that is, to prevent any one bin from straddling zero.
  plot_z_points(x,y,z,exclusion) excludes values listed in array exclusion, e.g., find(z~=exclusion(i)) where exclusion = [0,-999,NaN]
  plot_z_points(x,y,z,exclusion,'log') plots point values z with color mapped to logarithmic scaling. 'log' flag will only work if all points in z are positive > 0. Exclusion = 0 flag (above) can be used to remove zero values.
  plot_z_points(x,y,z,exclusion,'log',nbins) discretizes the data into nbins number of bins for plotting purposes. Default is nbins = 5

Bibliography

Melcher, A., Hauer, C., & Zeiringer, B. (2018). Aquatic Habitat Modeling in Running Waters. In S. Schmutz, & J. Sendzimir, *Riverine Ecosystem Management* (Vol. 8). Springer.