UC Merced
UC Merced Electronic Theses and Dissertations

Title

Field parameterization and evaluation of 2-D, finite-element modeling of the Merced River-San Joaquin River confluence

Permalink

https://escholarship.org/uc/item/17g6w929

Author

Pai, Henry

Publication Date

2010-12-14

Peer reviewed|Thesis/dissertation

University of California,

Merced

Field Parameterization and Evaluation of 2-D, Finite-Element Modeling
of the Merced River-San Joaquin River Confluence

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Environmental Systems

by

Henry Pai

2010

This thesis is of Henry Pai is approved.

Qinghua Guo

Carlos F.M. Coimbra

Thomas C. Harmon, Committee Chair

University of California, Merced

2010

For my parents and brother,

whose patience, support, and comedic relief

allowed me this opportunity.

Table of Contents

List of Figures

List of Tables

Acknowledgments

ABSTRACT OF THE THESIS

Field Parameterization and Evaluation of 2-D, Finite Element Modeling
of the Merced River-San Joaquin River Confluence

by

Henry Pai

Master of Science in Environmental Systems

University of California, Merced, 2010

Professor Thomas C. Harmon, Chair

River confluences present a challenging environment for both data collection and hydrodynamics and advection-dispersion modeling. The Merced River-San Joaquin River provides a site with distinct water salinity signatures where advection-dispersion modeling can be readily tested. This work describes the application of a robotic delivery system for water velocity and specific conductivity measurements whose infrastructure is sufficiently agile to enable analysis of several cross-sections along the confluence within a week-long study. The volumetric water flow estimates from the cross-sectional water velocity field were comparable (within 10%) to those recorded at a nearby gaging station. With the river bed elevation and water surface elevation determined by echo-sounding and surveying, a 2-D, finite element hydrodynamic model is parameterized with additional fitting parameters including bed roughness coefficients and eddy viscosities. Various parameter adjustment scenarios were undertaken to calibrate the RMA2 hydrodynamic model, and the resulting steady velocity field was then used as input for the advection dispersion model RMA4. The best results in RMA2 were automatically assigning the roughness coefficient by depth and a single, isotropic eddy viscosity value of 35 Pa-s for the entire model domain. This approach resulting in an average absolute percent difference (AAPD) between the modeled and observed lateral velocity profile for the furthest downstream cross-section of 31.87%, a -2.47% difference between modeled and observed water surface elevation, and a best qualitative shape-fit agreement between the modeled and observed lateral velocity profile. These results could be improved by quicker elevation data collection methods that are more closely synchronized with the times of the velocity/flow data collection. The estimated

eddy viscosity value is close to the theoretical value (34.69 Pa-s) estimated given the downstream cross-sectional dimensions and flow conditions. For the advection-dispersion modeling, the longitudinal dispersion coefficient of 0.1 m^2/s and lateral dispersion coefficient of 0.01 m^2/s assigned to the entire model domain generated the best fit, resulting in a 7.78% AAPD between the modeled and observed lateral concentration (salinity) profile for the furthest downstream cross-section, and a best qualitative fit shape-fit agreement between the modeled and observed lateral concentration profile. These fitted values are within the range of theoretically estimated longitudinal (0.06 to 0.13 m^2/s) and are similar lateral dispersion coefficients (0.014 to 0.015 m^2/s) based on the upstream San Joaquin River cross-sectional dimensions and flow conditions. This work demonstrates that high-resolution data collection coupled with appropriate model settings can provide reasonable 2-D hydrodynamic and advection-dispersion simulations in complex river environments, such as confluence zones. The models developed here are useful for informing reservoir operation and water quality management decisions in the SJR basin.

1 Introduction

River confluences are regions of complex flow and constituent mixing, often difficult to characterize and parameterize. Though two-dimensional (2-D), finite-element models do not describe secondary flows known to occur at confluences, the parameterization and computational requirements for the 2-D hydrodynamic models are significantly easier to satisfy. Still, previous 2-D confluence model studies are sparse, and the contribution of this work provides detailed application and analysis for 2-D hydrodynamic models and an additional advection-dispersion modeling component to also verify the accuracy of fitted model parameters. Additional contributions involve capturing the velocity and salinity concentration cross-sectional profiles in this complex system using a robotic delivery system known as the Rapidly Deployable Networked Infomechanical System (NIMSRD), and data-handling methods to define appropriate model inputs from the river bed topography and velocity and concentration field data.

Chapter 2 of the thesis provides a synopsis of the field site stretching along the Merced River-San Joaquin River confluence and field data collection methods. The subsequent chapter (Ch. 3) discusses data analysis aimed at characterizing the confluence. Integrated flow estimates from the field data are compared with a local gaging station near a transect downstream of the confluence for validation. Chapter 4 discusses the 2-D, finite-element modeling approach used to simulate the both the hydrodynamics can advection-dispersion. The chapter includes a discussion of the development and general motivation for river modeling, and concludes with a discussion covering previous confluence modeling studies. Chapter 5 and 6 describe the key 2-D, finite-element hydrodynamic and advection-dispersion model results. Several case scenarios for fitting parameters and situations described in previous modeling studies are analyzed. The compiled results provide recommendations for future data collection methods and appropriate estimation and fitting methods for modeling studies at river confluences.

2   Field Site and Methods

2.1   Field Site

From1991 through 2004, the San Joaquin-Tulare basins were included as study sites of the National Water-Quality Assessment (NAWQA) program. Studies of the water usage in the region tally agricultural use at over 90 percent whereas urban use remains below 10 percent [1], [2]. Such heavy agricultural land use of water and eventual discharge of irrigation return flows to the water supply caused known increases in concentrations of pesticides and nutrients that can have adverse effects on aquatic and sediment organisms as well as drinking water supplies [1], [3].

In this work, the area of interest is the tributary Merced River confluence with the San Joaquin River (SJR). The Merced River forms its own sub-basin which is located on the east part of the SJR Basin [4]. The river bed sediments near the site location were classified as medium-to-coarse sand and silty clay layers in the shallow subsurface [5]. By the lower parts of the Merced River, near our study site, agricultural land usage is roughly 55-56 percent [4], [5], and agricultural inputs and accumulation from upstream are expected. The SJR, in addition to normal agricultural diversions and inputs, intakes significant amounts of salts from the west part of the SJR Basin just upstream of the study site from the Mud and Salt Slough. In particular, half of the nitrate load on the SJR at this point is due to these sloughs as recent as a 1992-1995 USGS study [1]. Figure 2.1 shows a regional map including the sloughs and a waterway flow diagram with abbreviated names of USGS and California Department of Water Resources (DWR) gaging stations. As can be seen in Figure 2.2, the turbidity from the SJR is greater than that of the Merced River. Ultimately, these differences as well as individual constituents and temperatures serve as tracers in distinguishing one river from another, and mixing between the two.
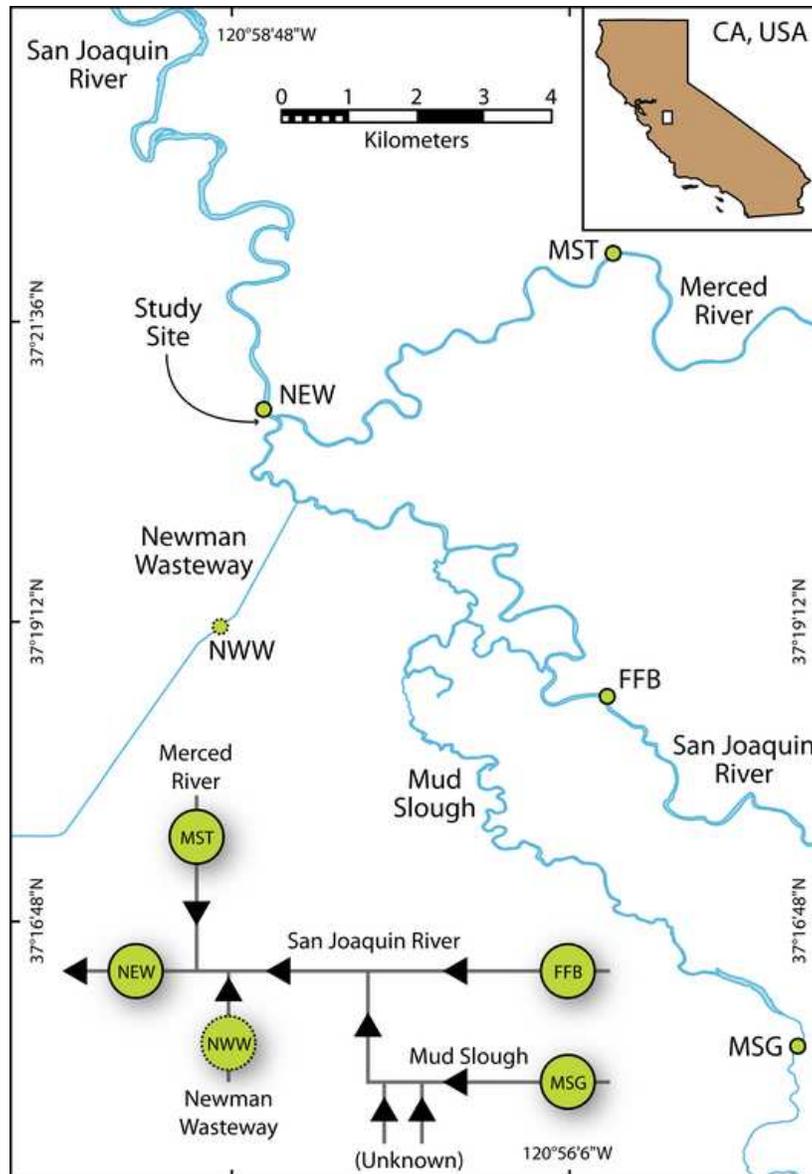
Figure 2.1: Regional map with flow diagram and abbreviated gauging station names are in light green circles (*courtesy of Jason Fisher*).

From August 7th to August 12th in 2007, extensive cross-sectional measurements characterizing water quality and flow were taken where the SJR meets the Merced River, the zone referred to henceforth as the: SJR-Merced confluence. Four cross-sectional data sets are used in this work: (1) transect T1 was located next to the USGS monitoring station at Newman, denoted as NEW and roughly 290 m downstream of the confluence, estimated by Google Earth [6]; (2) T2 was located roughly 115 m upstream of the confluence on the Merced River; (3) T3 was located roughly 85 m upstream of the confluence on the SJR, and T4 is located roughly 45 m

downstream of the river confluence point. Figure 2.2 below shows a satellite image generated from Google Maps to delineate transect locations and direction of flow. T2 and T3 served as the upstream boundaries of the model domain, while T1 served as the downstream boundary. Data collection was more intensive at T4 as it resided in an intermediately mixed zone with respect to the Merced and San Joaquin Rivers' salinity and other water quality parameters.
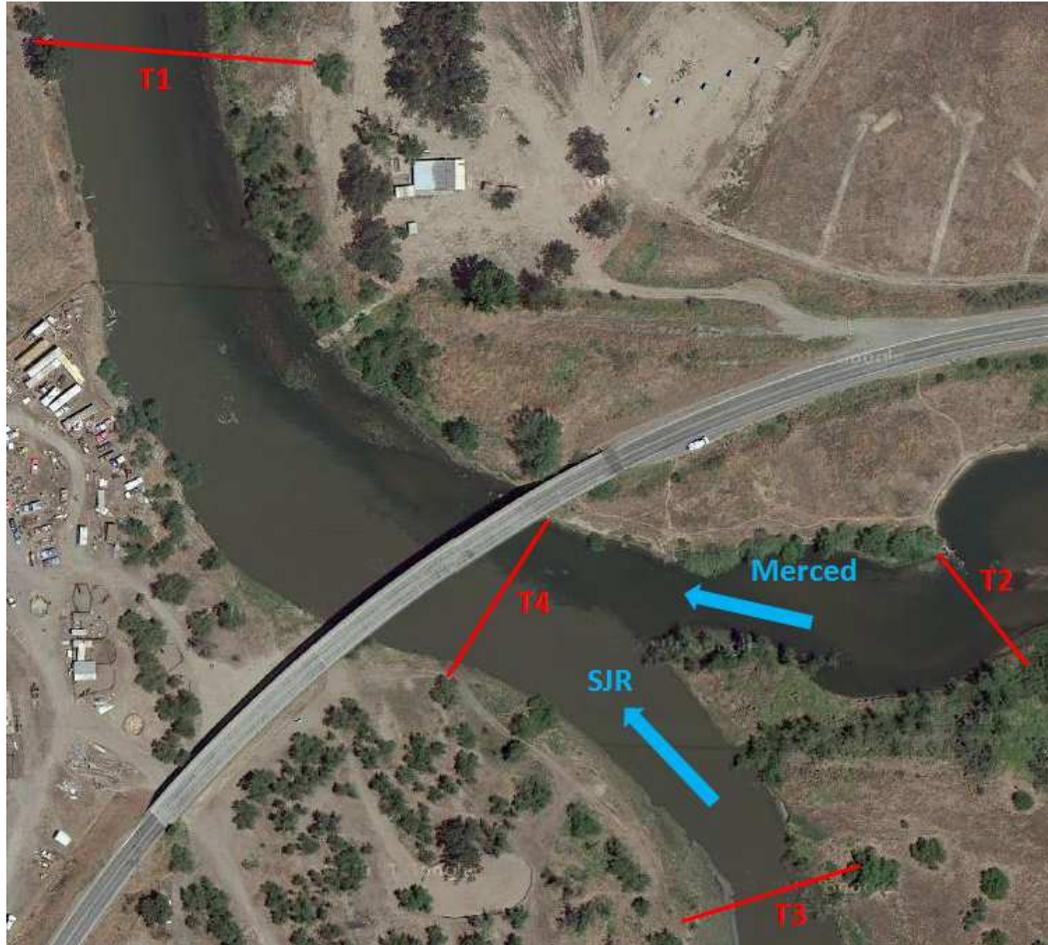
Figure 2.2: Zoomed in figure showing transect locations and denoting flow of the SJR and Merced River from Google Maps. Note that the satellite image was not taken at the same time of the study. Therefore, water levels will be different.

2.2 Methods Overview

Studies at the SJR-Merced confluence occurred over the summers of 2006 and 2007 and focused on:

1. Robustness of cable-based robotic sensor-delivery system known as Rapidly Deployable Networked Infomechanical System, NIMSRD [7], [8]

2. Adaptive sampling and path-planning techniques to maximize efficiency data collection [9]

3. Analysis of downstream length of the mixing layer generated by the confluence [6], and

4. Evaluating mass balance between upstream and downstream measurements

This work further examines data collected during the 2007 study using two different two-dimensional (2-D), depth-averaged flow and transport models for to simulate the confluence region. This chapter provides a brief summary of the overall method, each method for data collection, and probable sources of error from these methods.

As mentioned in previously, four transects were of particular interest for this work. The two transects upstream, T2 and T3, serve as the input flow boundary conditions. The downstream transect, T1, serves as a validation boundary for the model, since this transect is proximate to a USGS flow gauging station- (Newman Station). A fourth transect (T4) was located just upstream of the bridge. The stage at T4, however, was quite low, leading to higher probabilities of error in water velocity measurements. In total, 20 cross-sectional data sets were collected for T1, 2 for T2, 3 for T3, and 3 for T4.

Two NIMSRD systems were deployed simultaneously to minimize time between measurements and maximize the likelihood that stationarity was maintained within the confluence during the deployment. The limiting factor in this approach was that only one acoustic Doppler velocimeter (ADV) was available for velocity measurements. Thus, once a data set was collected at one transect, the ADV had to be removed and transported to another transect, a process requiring at least one hour. To measure the third transect, one NIMSRD system had to be transported and partially reassembled, requiring at least two hours. Water quality parameters measured by two multiparameter probes, (Hach Hydrolab DS5), were used to make simultaneous measurements across two transects.

After the deployment of NIMSRD systems and associated sensors, the topology of the river bed and surrounding floodplain were measured using surveying equipment from Leica and a kayak delivering a echo-sounding (sonar) device coupled with a GPS localization system (MIDAS Surveyor, Valeport Ltd., UK). The geometric file created by the MIDAS system is eventually fed into the flow models.

2.3   NIMSRD

The Rapidly Deployed Networked Infomechanical System (NIMSRD) provides reproducible localization of point-measurement sensors along a cable infrastructure. Typically, both anchoring towers (Figure 2.3) are georeferenced and the NIMSRD then localizes itself within that frame of reference. For this experiment, a handheld GPS device was used (Thales MobileMapper, Magellan Navigation, Inc., Santa Clara, CA). Figure 2.3 shows schematics and a photograph of the NIMSRD at T3.
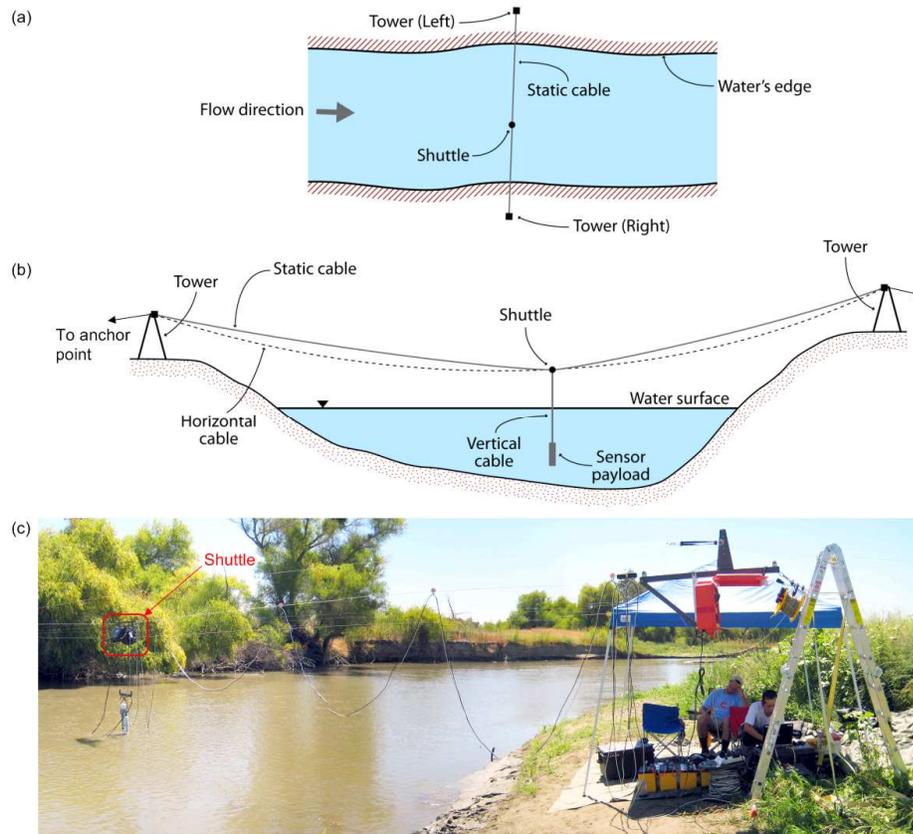
Figure 2.3: (a) Schematic bird's eye view shows the NIMSRD system perpendicular to river flow with labels; (b) Downstream, cross-section view [*diagrams by Jason Fisher*]; (c) Photograph of physical setup at T3 on the San Joaquin River.

The NIMSRD system requires several steps to install and major ones are listed below to aid in visualizing where errors associated with localization by the system are most prominent.

1. **Anchor system**. Once the transect section location is selected, anchors for a high-load static cable are sturdy trees, and ladders are used to provide control points on the static cable height on opposite sides of the transect (Figure 2.3b).

6

2. **Mount and connect shuttle motor box**. The motor box assembly is shown in the open (90°) case in Figure 2.3c, and shuttle on the static cable in all three images in Figure 2.3. The motor box consists of two servomotors (Parker model SM233, Rohnet Park, CA) and stays on-shore on the ladder on the operators' side of the river. Each motor has a small wheel attached to the axle. The shuttle provides attachment points for horizontal movements and pivot points for vertical movement.

3. **Attach horizontal and vertical cables**. The end points of the horizontal cable attach to the shuttle. This cable loops around a motor wheel which is modified with coarse sandpaper to improve cable traction. On the opposite shore, the horizontal cable loops through a pulley attached to a compressed spring that enables cable tension adjustments. The vertical cable is spooled on the second motor wheel and anchored on the shore opposite of the motor box.

4. **Attach sensor payload to the vertical cable**. The sensor payload includes a metal bracket providing attachment points for various sensors and the sensors.

5. **Attach tensioned calibration cable**. The calibration cable spans the system and facilitates calibrating the shuttle location horizontally with marked increments which is assumed to be parallel with the water surface.

6. **Connect system power and communications**. If power is required (e.g. externally powered sensors), then the system requires a festooned direct power cable. Data can be seen in real-time also through a separate data cable. Typically a laptop controls the NIMSRD system through a rewired RS232/serial interface. Power for the servomotors is provided by car batteries.

Although NIMSRD is designed to support precise delivery of the sensor payload to locations in the transect (x, z), localization errors are unavoidable. The key errors introduced in the setup are related to cable sag, horizontal/vertical calibration error, and horizontal/vertical sensor localization error caused by river currents. In spite of these factors, the system results in highly reproducible localization under stationary flow conditions, as verified by repetitive raster cross-sectional scans [7].

Cable sag occurs due to the weight of the sensor payload and static cable. Thus, actual motor-driven movements are along a sagged path and not a path parallel to the water surface, which is our local reference of study. To account for the sag, the NIMSRD calibration program calls a parabolic solver to relate axle rotations with fixed points along the water surface. The

calibration process involves moving the shuttle and sensor payload to a marked location on the calibration cable and visually confirming by a viewer often in the river. Visual confirmations are subject in nature, and can result in errors of 1 or 2 cm. For instance, if the viewer confirms the position at an angle, the horizontal position is likely inaccurate. Vertical confirmations are more consistent because a fixed point on the sensor payload is chosen, however, water surface fluctuations can lead to vertical calibration errors.

After calibration, raster scans are performed with predetermined locations, typically in a grid pattern. Despite calibration, local positions are subject to environmental errors primarily from water flow. For instance, as water currents increase, the sensor payload sways farther from the commanded position leading to inaccurate localization in both horizontal and vertical planes. Though currents were not particularly strong during the study, sway was observed in T3. While significant, these errors were typically systematic in nature under reasonable stationary flow conditions.

Additional errors are systematic. For example, in previous experiments involving the NIMSRD system, slippage occurred between the horizontal cable and the sandpaper mentioned in step 3. The experiments conducted at the confluence did not experience much slipping. Also, the system assumes the vertical spool diameter is constant when it actually changes when wrapping and releasing. This is an ongoing issue that is being addressed in the NIMSRD driver software, but not remedied in time for the studies reported on here.

2.4   Acoustic Doppler Velocimetry

Measurements of water flow velocity were made with an acoustic Doppler velocimeter (Sontek Argonaut model, San Diego, CA). In laboratory comparisons, the ADV agreed well with a laser Doppler velocimeter, LDV, (mean velocities within 1%), [10]. Also, the portability, ruggedness, ability for 3-D velocity measurements, and sensitivity to low flow conditions [11] make the ADV an ideal candidate for complex flow regimes such as the confluence [12].

The ADV probe consists of a single transmitter and three receivers whose arms are separated at 120 degrees around the transmitter and tips are oriented 30 degrees from the transmitter arm axis or z axis shown in Figure 2.4a. A schematic of the probe with labels is shown below in Figure 2.4a.
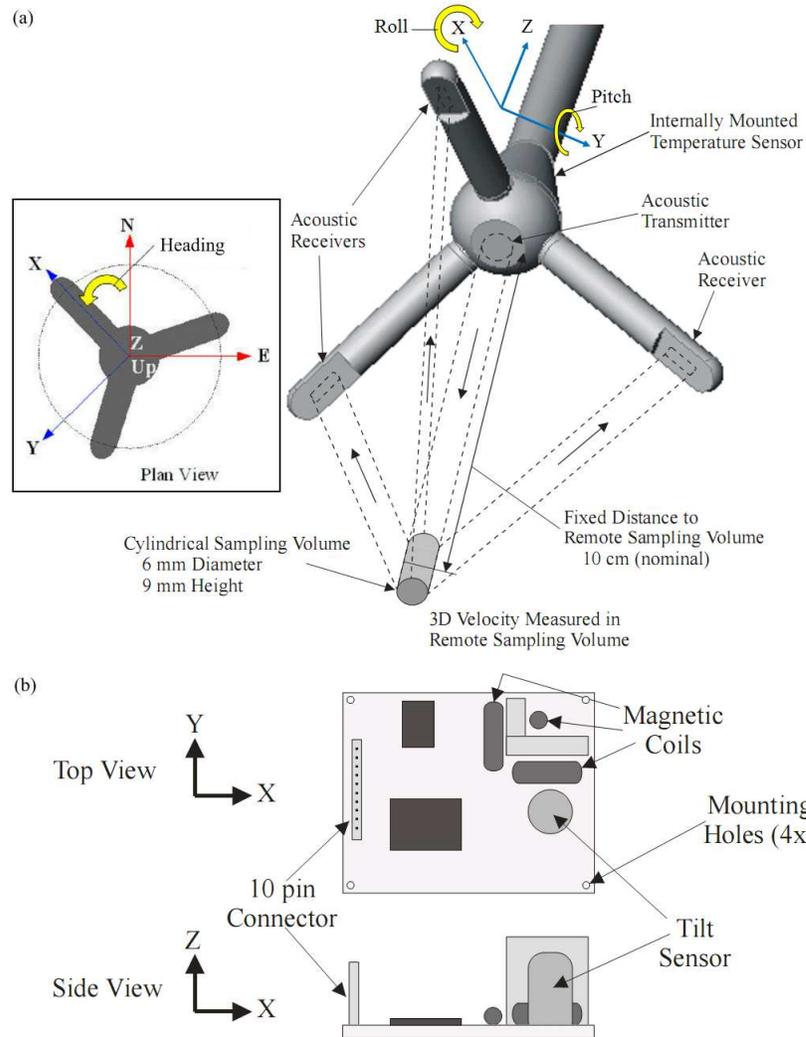
Figure 2.4: (a) Diagram of acoustic probe and related axes along with a plan view relating axes with magnetic axes [6], [13]; (b) diagram of internal sensor board measuring pitch, roll, and magnetic direction [13].

The acoustic transmitter emits pulses at 10 MHz for a specified duration. Energy from the pulsed waves will backscatter off particulates in the flowing water and is then measured by the receivers for a sample volume extending about 10 cm from the transmitter, as seen in Figure 2.4a. The actual sample volume is determined by the transmit beam pattern, receive beam pattern, length of the acoustic pulses, and the time allotted for the return signal to render a water velocity measurement [11]. Because of these factors, the location of the sample volume depends on the initial calibration of the equipment done by the manufacturer and can vary from ±0.5 cm

9

between different ADV probes. The sample volume shape is also simplified to represent a cylinder of diameter 0.6 cm and height 0.9 cm, where the volume edge accuracy is ±0.05 cm.

The ADV relates Doppler shift of the backscattered response to water velocities using a pulse-coherent processing technique. The technique measures the phase difference between successive pulses, where the time between pulses relate to different maximum velocities and respective velocity ranges. During the experiment, the ADV was operated in auto-range including all allowed velocity ranges from 0-600 cm/s corresponding to pulse-to-pulse time lags on the scale of milliseconds [10], [13].

Inside the ADV housing, an internal compass and roll/pitch sensors shown in Figure 2.4b separate water velocity components in terms of magnetic directions, given in northing, easting, and up vectors. Procedures from the USGS call for flows to be perpendicular to cross-section of the study requiring velocity component analysis [14].

The ADV accounts for water temperature and salinity affecting the speed of sound in water, which is needed in the Doppler relation to compute water velocity. A temperature sensor near the acoustic transmitter accounts for water temperature fluctuations. For salinity, a background measurement is set in the software by the user. At the confluence, the rivers have differing salinities across the cross-section, necessitating post-processing [11]. Sontek provides an approximate rule of thumb that a 1 percent increase in water sound speed corresponds to 12 ppt increase in salinity for ranges of 0 to 35 ppt. Additional empirical relations are available [15], [16].

During the confluence experiment, the ADV settings were chosen to maximize data available to post-process, although data integrity is also lowered by this approach. The data-averaging interval was set to the lower limit of the instrument (3 s), where the logger registers computed water velocities at a rate of 10 Hz. The low averaging interval is subject to 1% accuracy error [11]. Also, the low averaging interval prevented the ADV from detecting and filtering boundary interference. Sontek recommends that data within 20 cm from the acoustic transmitter can be influenced from boundary interference with averaging intervals less than 10 sec, and that 30 cm from the transmitter is safe from boundary interference [11].

2.5   Georeference

The Thales MobileMapper GPS marked locations for fixed points allowing a local plane of reference to be determined as well as tying data onto a global scale. Also, the device marked

the water's edge on August 3rd 2007 and will be used as a comparison for water surface elevation (WSE) versus USGS gaging station at Newman.

The GPS device uses up to 12 satellites to locate it position and maximizing accuracy to 2-3 m without post-processing. The software package, MobileMapper Office, enables post-processing to an accuracy of up to 0.7 m. The geodesic standard used by default by the system was WGS84. [17].

2.6   Surveying

From any two fixed points, the total station (Leica Builder R100M, Leica Geosystems, St. Gallen, CH) builds a reference line. Distances and elevations are saved and reconstructed from the reference line. After the initial deployment using NIMS-RD device, surveying commenced for several days to map the shallow river bottom and the nearby floodplain. Several fixed points were generated as reference planes became invisible along stretches due to changing elevations, vegetation, and limitations in distance.

The R100M transmits and receives a red laser that measures distance to objects that can reflect the laser. Leica provides a flat prism with reflective tape raised on a rod to reflect the laser. The general operation of the R100M involves:

1. Leveling the R100M base unit by adjusting the tripod leg lengths
2. Shooting (acquiring distance) fixed points and entering in their Northing, Easting, and elevation, and
3. Shooting visible points

After the second step in the operation, feedback about the accuracy of the reference line is generated and reference lines that were inaccurate by over 0.1 m were not used. The given manufacturer reported distance standard deviation for the reflective tape is 3 mm + 2 parts per million (ppm) with a suggested range of up to 150 m. External influences possibly affecting measurements involve objects within the beam path such as tree branches and leaves and heat shimmer [18].

2.7   Sonar Bathymetry

For deeper stretches of the river where surveying was not practical, a MIDAS Surveyor GPS Echosounder (Valeport Ltd.) mapped riverbed topography. The echosounder was mounted to a wooden frame attached to the center of a kayak. The kayak trolled or paddled back and forth

over the deeper channels. The device includes three major pieces: a data logger and power supply (integrated into the MIDAS Surveyor), a 12 channel GPS receiver, and a high frequency transducer (the echosounder). Data are post-processed using SurveyLog software also from Valeport Ltd.

The GPS computes the antenna location by an onboard Space Based Augmentation System (SBAS) differential correction the signal. The ability to perform the correction depends on the coverage with geostationary satellites common in North America. The manufacturer reports an accuracy of ±2 m with this correction and ±4 m without. The default projection for the GPS module is WGS84.

The echosounder transducer emits an acoustic pulse at 210 kHz and the data logger records data at a rate of 6 Hz. The manufacturer recommends an operational range of the transducer of 0.3-100m with an accuracy of ±0.1 m or ±0.02%, whichever is greater. Additional errors can be of depth measurements may be associated with the velocity of sound in the freshwater system, recommended as 1470 m/s, (1510 m/s for seawater) [19], [20].

The system needs horizontal and vertical correction between the GPS antenna and echosounder transducer and this was accounted for during post-processing. Additional error from tides and heaves from the kayak but were not considered important due to low flow conditions and the need for additional tide height and heave sensors.

2.8   Specific Conductivity

The Hydrolab DS5 (Hach Environmental, Loveland, CO) multi-parameter sonde measures several parameters including: temperature, pH, specific conductance (SC), depth up to 25m, chlorophyll, rhodamine, and dissolved oxygen (DO). For this study in particular, there was a prominent disparity between the two SC concentration inputs from the San Joaquin and Merced Rivers. Thus, SC and the 2-D velocity field became the focus for studying mixing effects downstream of the confluence. Also, SC is often used as a natural tracer albeit not exclusively due to spatio-temporal changes in lateral inflows and upstream boundary conditions [21], both of which were not resolved, unfortunately.

The SC sensor is composed four graphite plates acting as conductors of a cell with known dimensions measuring the resistivity or, inversely, the conductivity of the medium passing through cell [22], [23]. The manufacturer operational quoted range is 0 to 100 mS/cm with an

accuracy of ±1% of the reading or ±0.001 mS/cm, whichever is greater [24]. Sensor firmware corrects for local water temperature reports SC values at 25º C.

2.9   Summary

The instruments used in this study aim to resolve the two critical parameters necessary for a 2-D hydrodynamic model: river bed and bank topography and flow boundary conditions. Topography required the use of survey equipment whose initial reference is determined by the handheld GPS device and sonar bathymetry to map areas impractical for surveying. Flow boundary conditions required the use of multiple measurements from the ADV moved by the NIMSRD system to accurately quantify the flow across the cross-section. The handheld GPS device also helped map the WSE to compare versus the NEW gaging station. To study mixing length downstream of the confluence, the Hydrolab DS5 measured SC to provide semi-quantitative estimates assuming subsurface gains and losses are negligible and upstream flows are steady. Table 2.1 summarizes equipment, equipment purposes, operational ranges, errors, and feasibility to this work.

Table 2.1: Summary of instruments used to characterize river bed geometry, flow conditions and specific conductance (SC).

Equipment	Purpose	Range	Inherent accuracy	Operational error
NIMS-RD	Local reference	-	-	Subjective calibration
	Actuation			Flow displacing payload
				Slipping
				Non-constant vertical spool diameter
Sontek ADV	Fluid velocity measurement	0-600 cm/s	Spatial: 0.05 cm	Salinity content
			Flow velocity: 1%	Instrument faults
				Boundary interference
Magellan MobileMapper Pro	Fixed points on WGS84 projection	-	Without post-processing: 2-3 m	-
	WSE		With post-processing: 0.7 m	
Leica R100M	Shallow river bed topography	Up to 150 m	Standard deviation: 3 mm + 2 ppm	Reference line errors
	Floodplain topography			Reflective objects in beam path
				Heat shimmer
				Approximate level of rod with reflective tape
Midas Surveyor GPS Echosounder	Deep river bed topography	0.3-100 m	Spatial without SBAS correction: 4 m	Inaccurate correction between GPS receiver and echosounder transducer
			Spatial with SBAS correction: 2 m	
			Depth: greater of 0.1 m or 0.02%	
Hydrolab DS5	Mixing length with SC as proxy	0-100 mS/cm	Greater of 1% or 0.001 mS/cm	Unaccounted inputs from subsurface flow and unsteady upstream inflow

3   Field Data and Analysis

3.1   Overview

The data collected from the various sensors discussed previously supplied layers of information for geometry, flow boundary conditions, and water surface elevations needed parameterize and to initialize river models accurately. Figure 3.1 is a flow diagram showing major data analysis steps to obtain model boundary conditions. First, this chapter presents the merging of raw geometry data. Challenges involve resolving the accuracy of Easting and Northing coordinates between the echosounder GPS and surveying equipment and the depths (assuming a static river bed) for differing measurement times. Merely combining the data leads to sharp local elevation discontinuities in elevation, which can lead to model instabilities. To remedy this, a multilevel B-spline approximation technique was used to smooth the combined data. Once the model geometry is defined, the flow boundary conditions need to be estimated correctly. For accurate flow estimates, the automated motion and velocity and salinity data need to be localized. Due to interference from the river bed and need for direction correction, the ADV data needed post-processing prior to its application to flow calculations, as will be discussed. After data correction and post-processing, the point velocity data are integrated to estimate flow using the mid-section method and compared with a US Geological Survey (USGS) gauging station for accuracy.
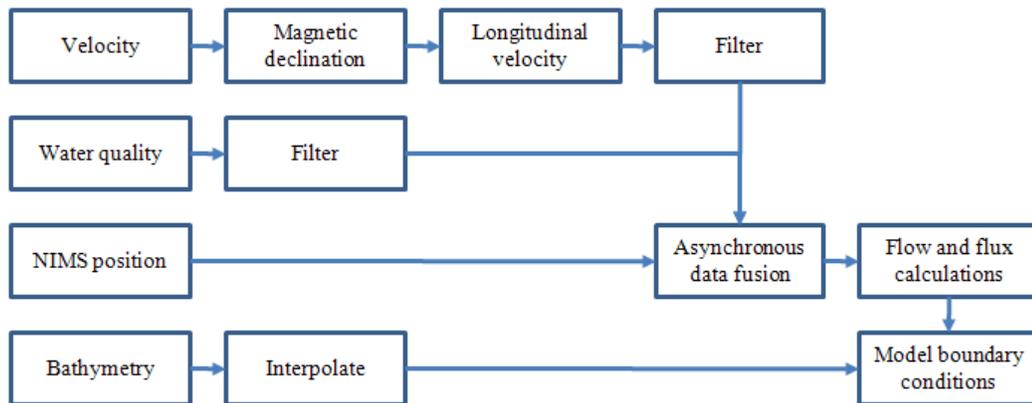
Figure 3.1: Flow diagram showing major data analysis steps.

3.2   Topography

Geometries of the river bed and floodplain are typically required for establishing a 2-dimensional river model domain. The data are collectively referred to as the digital elevation

model (DEM) for the area of interest. The DEM data collected for the Merced River-San Joaquin River confluence is shown in Figure 3.2. The solid blue lines represent data taken with the handheld Magellan GPS unit to map the water edge and base flow conditions. The solid red line represents the echosounder track showing deeper river channels that could not be surveyed. The rest of the individual points were collected using the surveying equipment. The depths for the echosounder and surveyed points are plotted by color; the handheld GPS elevation data was not included because those data were found to be of insufficient accuracy compared to the other data.
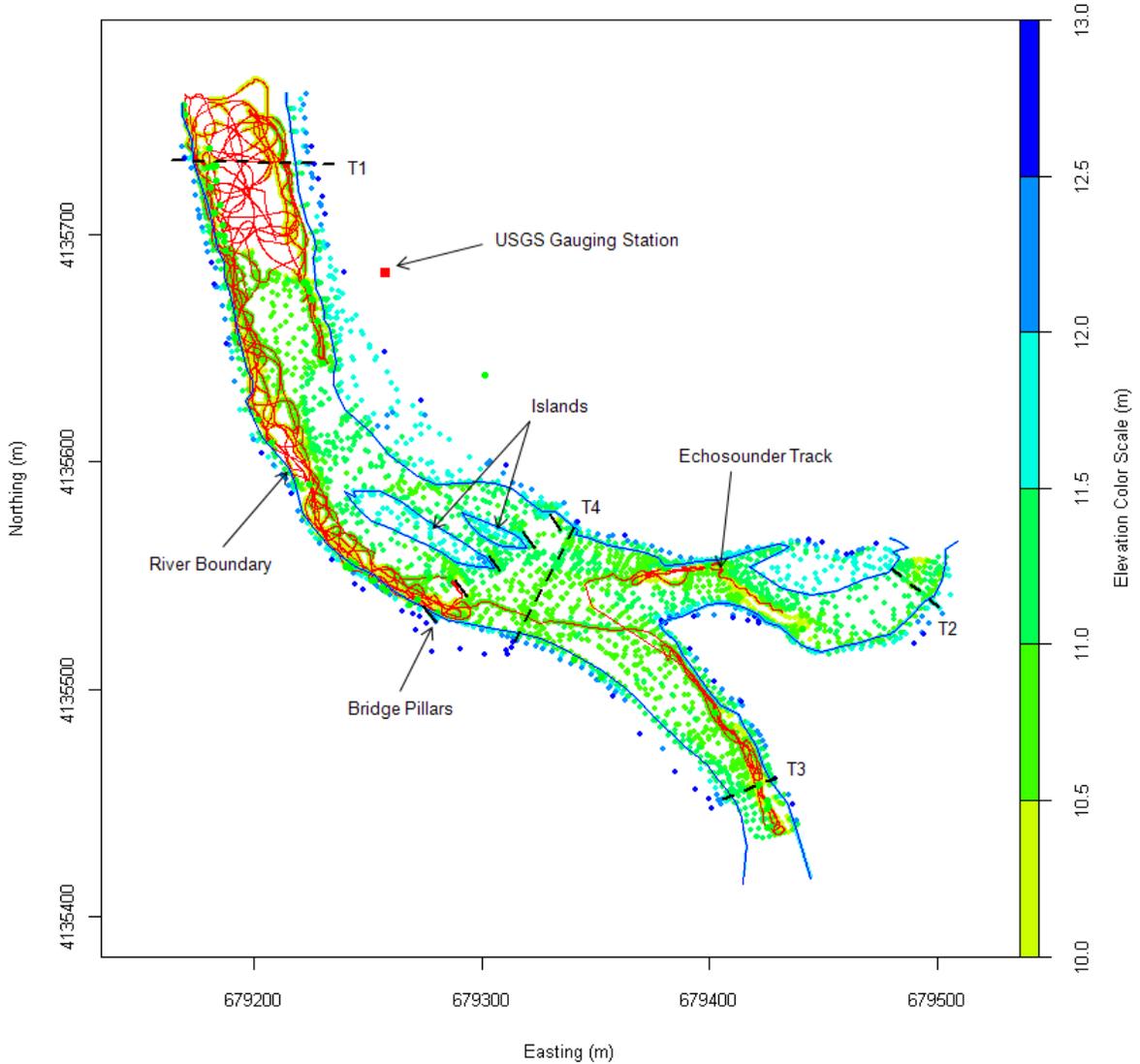
Figure 3.2: The DEM showing observed elevation using both the echosounder (tracked in red) and Leica surveying equipment. Additional information is added for reference such as transects (dashed lines), bridge pillars (solid black lines), and labels for physical features.

The Surface-water Modeling Systems (SMS 9.2) is a commercial program providing a GUI and computational mesh generator front-end and compiler for separate hydrodynamic model codes. SMS also provides three interpolation methods (linear, inverse distance, and natural neighbor) for mapping the DEM to mesh elements. The linear interpolation method has been reported to poorly represent the actual topography leading to inaccurate model results [25]. To account for the weak DEM interpolation, a code called RSurvey in the R-programming language was used to interpolate and generate a dense DEM using an interpolation method known as multilevel B-spline approximation [26], [27]. The algorithm is described in detail by Lee et al [28], and applies uniform cubic B-spline basis functions as a weighting function around observed elevations and interpolates the surrounding, unknown grid points, known as the control lattice, by multiplying the weighting function with the observed elevations. Using the B-spline approximation alone results in a tradeoff between surface smoothness and accuracy depending on the density of the control lattice. To resolve this, the authors suggest a multilevel approach involving hierarchies, h, of control lattices where the B-spline approximation is applied h times. The density of the grid increases with each hierarchal level and the levels are blended together to create a smoother shape. It is worth noting that this method is inefficient computationally in accurately representing local features [29], [30]. Figure 3.3 shows the interpolated DEM when the interpolated points on a grid with a spacing of 0.5 m and 11 hierarchal levels which is assumed to be sufficiently dense to preserve local features and smooth data.

To evaluate the accuracy of the interpolated surface, the elevation difference is calculated between every observed and interpolated point. Overall, the agreement was reasonably close for the majority of the domain (Figure 3.4). The figure designates locations where the elevation difference is greater than 0.25 m as green points. This occurs often along the echosounder path near the river edge. Differences are likely due to steep drop offs at the river edge where the interpolation scheme is reducing accuracy in favor of smoothness or due to measurement disagreements between the echosounder and surveyed data, in which case the interpolation may serve as a compromise between values. An example of measurement disagreement can be seen at transect 3 (T3), as shown in Figure 3.5. The variable portion of the red line depicts accuracy differences between the echosounder and surveying equipment. The interpolated blue line provides a noticeably smoother compromise for the transect shape and was chosen to best represent the bathymetric data.

17
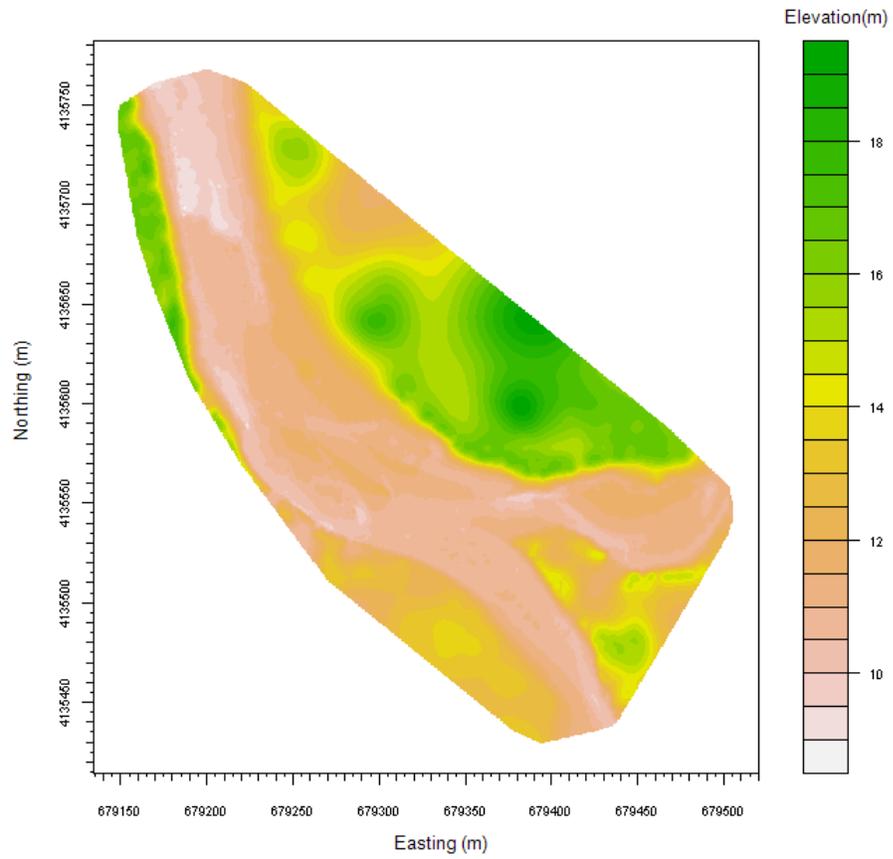
Figure 3.3: Interpolated DEM generated using RSurvey showing smoothed surfaces.
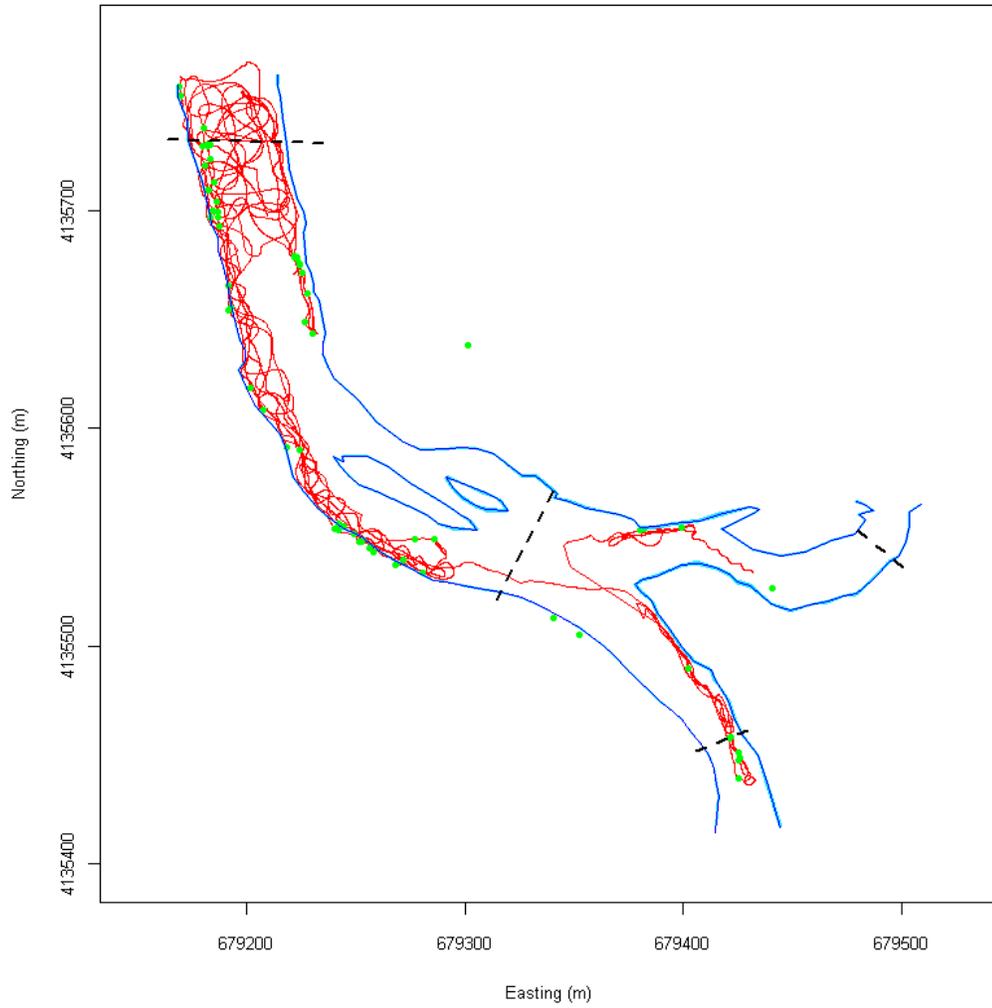
Figure 3.4: Map showing where the elevation difference between the observed elevation value and closed interpolated point differ by more than 0.25 m notated by the green dots. The red line represents the echosounder path, blue line represents the river boundary, and dashed lines represent the NIMSRD transects.
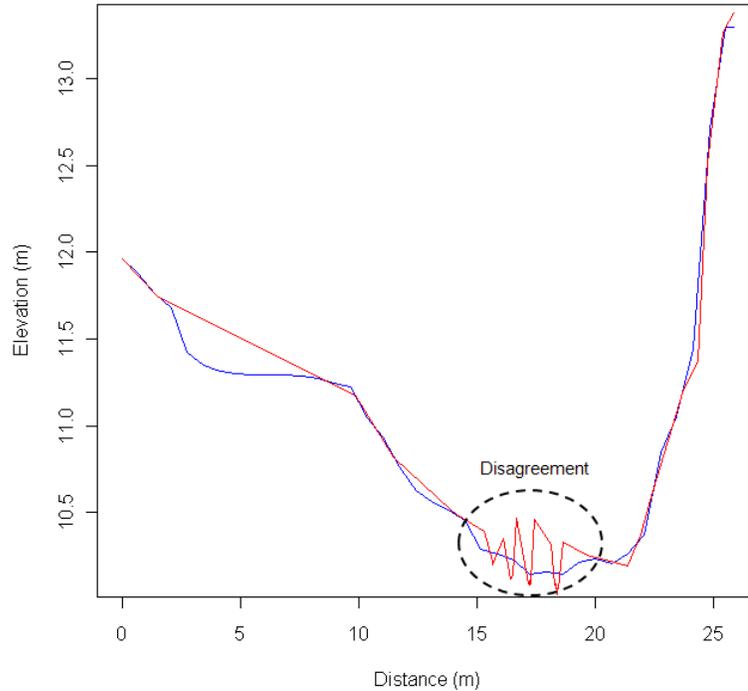
Figure 3.5: Distance (x-axis) refers to the distance from left anchor point of T3. The disagreement between measurement devices (echosounder vs. surveyed) is shown in red within the dashed-ellipse.

3.3   Flow data

Volumetric flow data from transect 2 (T2) and transect 3 (T3) are required as input upstream boundaries for the 2-dimensional hydrodynamic models used. NIMS-RD provided automated motion for ADV point measurements in a grid-like fashion. The major steps to calculate flow were to:

1. Combine NIMS-RD position with the water velocity and quality parameters
2. Post-process ADV data
3. Calculate flow using the USGS midsection method, and
4. Validate flow methods and results with the local gauging station near transect 1 (T1).

Each step will be discussed further. R-code developed to support these steps and applied in this section is presented in Appendix A.

3.3.1 Asynchronous Data Fusion

Combining the position involved defining when the motor was stopped and the sensor payload was dwelling in a desired location. Dwell states were necessary to obtain reliable ADV

point measurements. Timestamps of commands for dwell and move states were initiated by the laptop but were subject to transmission lag. To further refine dwell start and end times, the NIMS-RD motors relayed encoder information back to the laptop. A detailed flow diagram and pseudocode are shown in Figure 3.6 representing the algorithm programmed in an R-script to consistently define dwell times for all the raster runs. (One raster run is defined as one cross-sectional dataset).
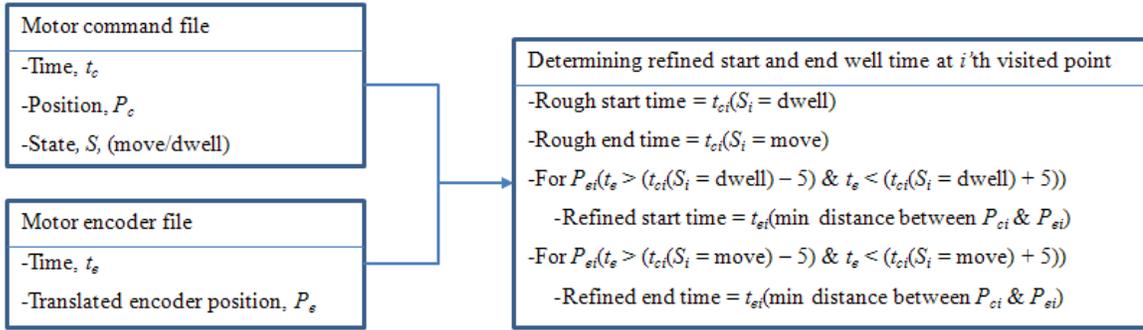
Figure 3.6: Data from files shown on left entering algorithm on right to define dwell start and end times.

Likewise, ADV and Hydrolab sensor data were time stamped as the data were received on the laptop. With dwell state start and end times well defined, data could then be separated with slight buffers to account for sensor data transmission lags to the laptop. For the Sontek ADV, data were not sent to the laptop until the averaging period (set to 3 seconds) for the measurements was completed. Hence, an additional buffer was added to the dwell start time to account for combining the ADV and positional data.

3.3.2 Data Preparation

The next major step is to prepare the velocity data for flow calculations using the USGS midsection method [31]. The longitudinal velocity vector, the vector perpendicular to the transect or parallel with the direction of the river flow, needed to be retrieved from individual point velocity measurements, reported in three vectors (northing, easting, and upwards) based on the ADV's built-in compass. First, corrections were applied for the magnetic declination (magnetic north to true north). The magnetic north was found to be 14.25 degrees east of true north for the study area at the particular time based on NOAA's National Geophysical Data Center (NGDC) [32]. Velocity vectors were adjusted using a rotation matrix where the displacement is 14.25 degree counterclockwise rotation using the following expression

$$\begin{bmatrix} v_E{}' \\ v_N{}' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_E \\ v_N \end{bmatrix}, \qquad (3.1)$$

where θ is the counter clockwise rotation of 14.25 degrees, v_E and v_N are the velocities from the ADV whose direction is dictated by magnetic easting and northing respectively, and v_E' and v_N' are the rotated velocities to account for true easting and northing respectively. Next, the velocity is transformed so the northing component is rotated to the longitudinal direction of flow. For transect 1, a clockwise rotation of 1.59 degrees is made; for transect 2, a counterclockwise rotation of 139.66 degrees; for transect 3, a counterclockwise rotation of 21.79 degrees. A similar rotation expression as Equation (3.1) is applied for each transect (clockwise rotation needed for transect 1 uses a negative angle).

The final preparation step is to filter for operational errors listed in the previous chapter. One noticeable operational error is evident when the compass reads zeroes for heading, pitch, and roll indicating circuit board failure, and these data were filtered out. The other filter was applied to provide a safe buffer distance between the river bed and ADV point velocity measurements. As mentioned before, this distance was 30 cm to avoid boundary effects, or 20 cm from the sampling volume, for averaging periods less than 10 seconds. For this work, filters for 10, 20, and 30 cm from the transmitter were analyzed. Figure 3.7 shows an example of filtering for the buffer distance of 10 cm for the transects one, two, and three. The plots of the point measurement locations versus the boundaries show that: (1) in transect 1, the extra spacing on the left side of the channel may indicate inaccurate measurement positions along the stream width, and (2) in transect 2, the bed movement between when the velocity measurements and surveyed data were taken was significant, causing greater sources of error in discharge calculations because the point velocity measurements were taken before bed depth measurements were made. Ultimately, the 10 cm boundary was selected as a reasonable filter due to deletion of several points seen for the 20 and 30 cm buffers.
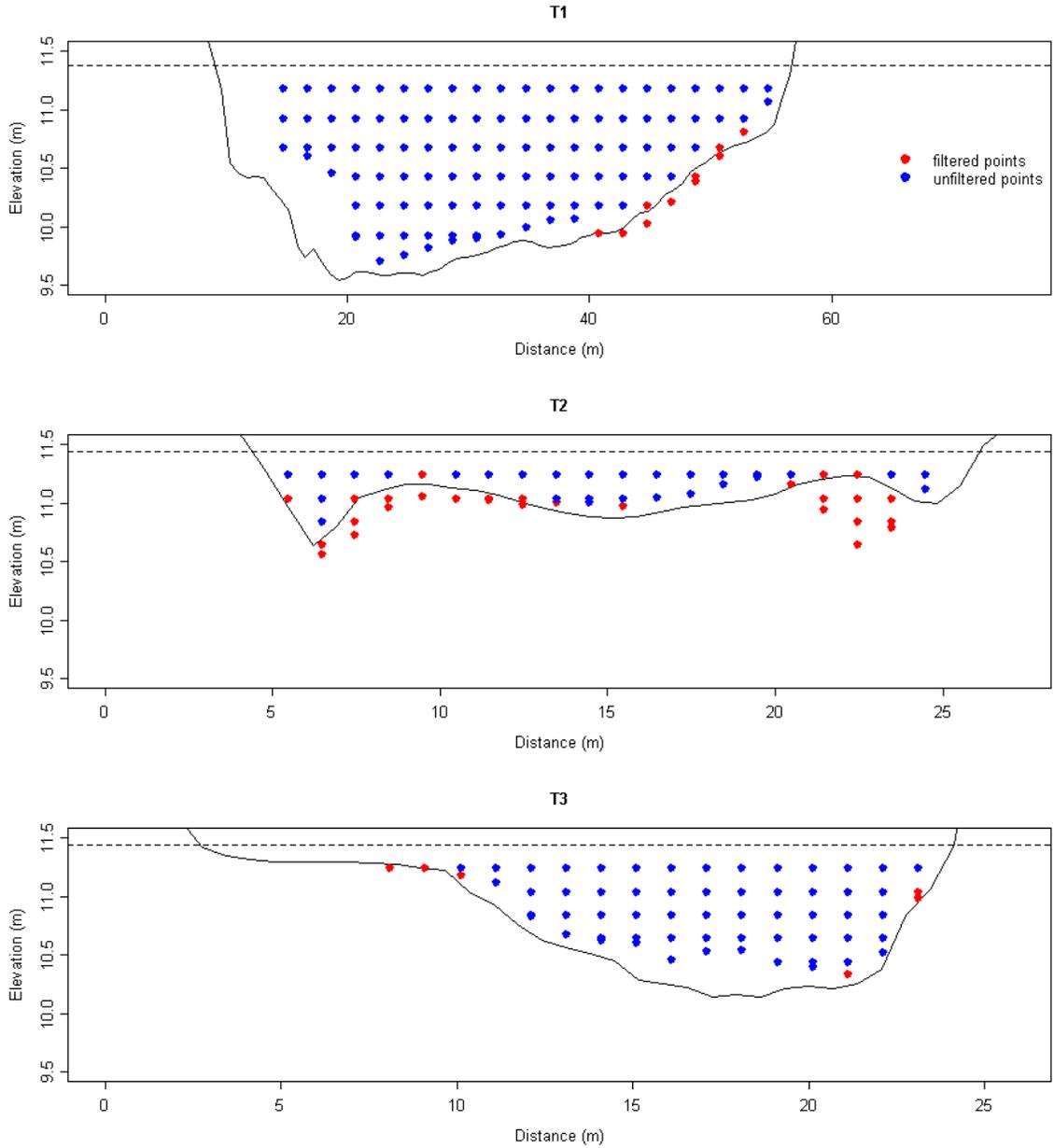
Figure 3.7: Showing river bed boundary for each transect (solid line), water surface elevation (dashed line), filtered points for a buffer of 10 cm (red points), and unfiltered points (blue points).

3.3.3 Calculating Flow

After the velocity data preparation, the midsection method was applied. This method sums vertical rectangular subsections of flow to calculate total flow, Q, through a cross section or transect and expressed as

$$Q = \sum av_{ave}, \tag{3.2}$$

where a is the area of a vertical rectangle and v_{ave} is the average velocity for the vertical profile. Due to the grid measurements provided by the NIMS-RD and ADV, several point velocity measurements were available for most vertical velocity profiles in each transect. The vertical velocity profiles in this work were assumed to be the shape of theoretical vertical velocity profiles shown in Figure 3.8 and the vertical-velocity curve method [31] was chosen to estimate average velocity values for flow calculations.
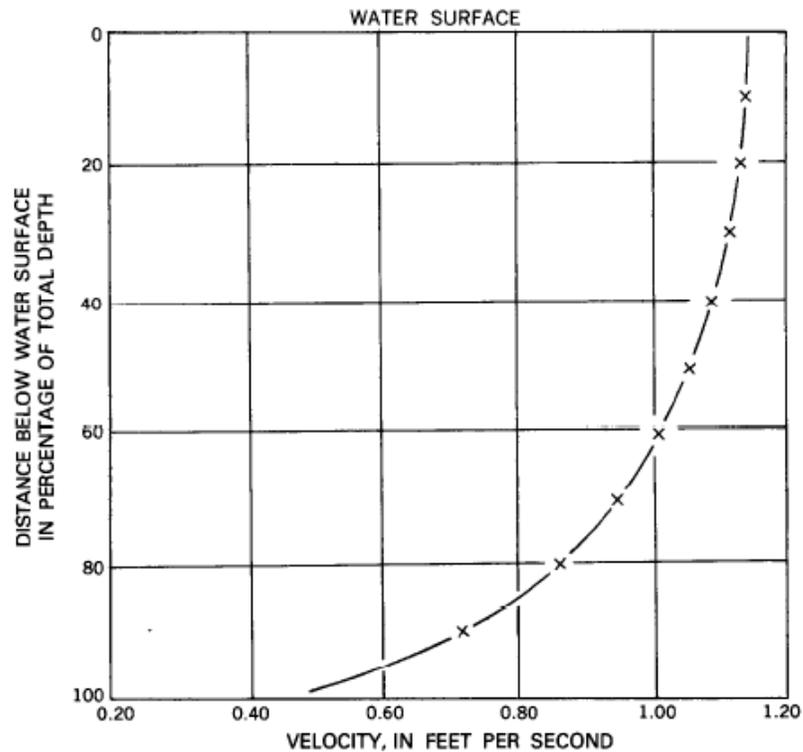
Figure 3.8: Velocity profile is for a typical vertical velocity profile whose average velocity is 1ft/s, shown to be at a depth 60% of the total depth (Plot taken from pg. 133 in [31]).

This method was appropriate because the ADV measurements were not taken at specific sampling depth ratios required by other methods. Instead, depth ratios were related to actual ratios associated with point water velocities to mean velocity for the theoretical vertical profiles given in Table 3.1 [33].

Table 3.1: Coefficients are for standard vertical-velocity curve, ([31], pg. 133)

Ratio of observation depth to depth of water, R_d	Ratio of point velocity to mean velocity in the vertical, R_v
0.05	1.160
0.10	1.160
0.20	1.149
0.30	1.130
0.40	1.108
0.50	1.067
0.60	1.020
0.70	0.953
0.80	0.871
0.90	0.746
0.95	0.648

With these coefficients, the average velocity for the vertical profile can be calculated using the following expression

$$v_{ave} = \frac{\sum_i^n v_i R_{vi}^{-1}}{n},$$
(3.3)

where n is the number of velocity measurements in the vertical profile, i is the particular measurement at a location, v_i is the longitudinal velocity measurement for i, and R_{vi} is the specific ratio for that particular of point velocity to mean velocity chosen from the second column in Table 3.1 as determined by the ratio of the i'th measurement depth to depth of water for that vertical profile.

Once the average velocities are calculated, the midsection method can be applied. The method calculates the flow through each vertical subsection, q, as follows

$$q_j = v_j \left[\frac{b_{(j+1)} - b_{(j-1)}}{2} \right] h_j,$$

$$q_1 = v_1 \left[\frac{b_2 - b_1}{2} \right] h_1,$$
(3.4)

$$q_n = v_n \left[\frac{b_n - b_{(n-1)}}{2} \right] h_n,$$

where:

      1, 2, 3… n = subscript number denoting particular vertical subsection,

      $q_1, q_2, q_3… q_n$ = flow through particular vertical subsection,

$v_1, v_2, v_3 \ldots v_n$ = average longitudinal velocity calculated from Equation (3.3) for the particular subsection,

$b_1, b_2, b_3 \ldots b_n$ = distance from an arbitrary initial point to the observation vertical, and

$h_1, h_2, h_3 \ldots h_n$ = depth at the observation vertical.

The middle multiplicative term indicates a width and is then multiplied by the depth for the area of the cross section. Special cases are used for the first and last vertical subsections in the transect denoted by the last two expressions in Equation (3.4) respectively. Otherwise, the first expression is used for all other subsections. All the subsection flows are then summed and the expression becomes equivalent to Equation (3.2). Figure 3.9 visually describes Equation (3.4).
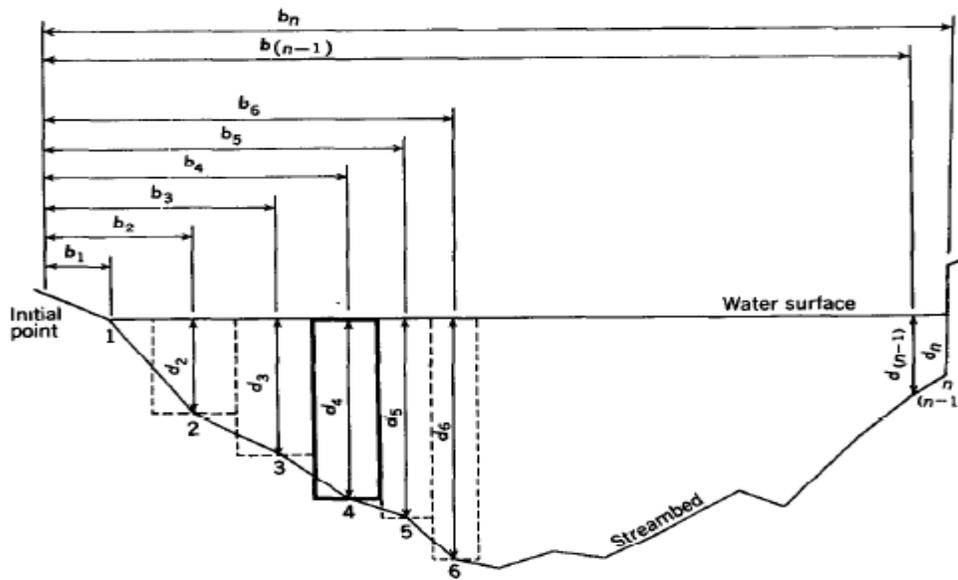
Figure 3.9: Plot from pg. 81 in [31] to help explain the method in calculating total flow found in Equations (3.4). The dashed lines are the boundaries between the subsections and the bolded was used an example in [31].

The final flow values were computed using the above-described method in conjunction with the interpolated bathymetry values and buffered (10 cm) velocity measurements. The resulting flow estimates are shown in the table below.

Table 3.2: Flow estimates based on integrated ADV measurements from all four transects.

Date	Transect	Run	Computed Flow (m^3/s)
Aug 7	1	1	9.31
Aug 7	1	2	9.78
Aug 7	1	3	10.23
Aug 8	1	6	11.14
Aug 9	1	17	9.24
Aug 10	1	26	9.32
Aug 11	1	27	8.10
Aug 11	1	33	7.56
Aug 8	2	4	2.68
Aug 8	2	5	2.74
Aug 9	3	1	7.80
Aug 10	3	2	6.92
Aug 10	3	3	6.92
Aug 11	4	2	10.48
Aug 12	4	3	9.66
Aug 12	4	4	9.53

3.3.4 Validation

The measured and computed flows at transect 1 are compared to the USGS gauging station near Newman, CA (NEW) which is close to the location of (37.351º N, 120.976º W) shown in Figure 3.2 as a red square. The station is maintained by the USGS and the data are available online at both the California Data Exchange Center (CDEC) and the USGS Water Resources site [34], [35]. Both CDEC and the USGS provide real-time data for the NEW station at 15-minute intervals and the USGS also publishes daily flows that have been corrected for rating curve changes. Figure 3.10 below is a time-series plot comparing our measured and computed flows with that of the NEW station. The USGS real-time (RT) data appears to apply a simple shift to the CDEC RT data over this relatively short period. As a note, the USGS field streamflow measurements conducted during the experiment study period were reported as being in "poor" (>8% error) to "fair" (5-8% error) agreement with the rating curve.
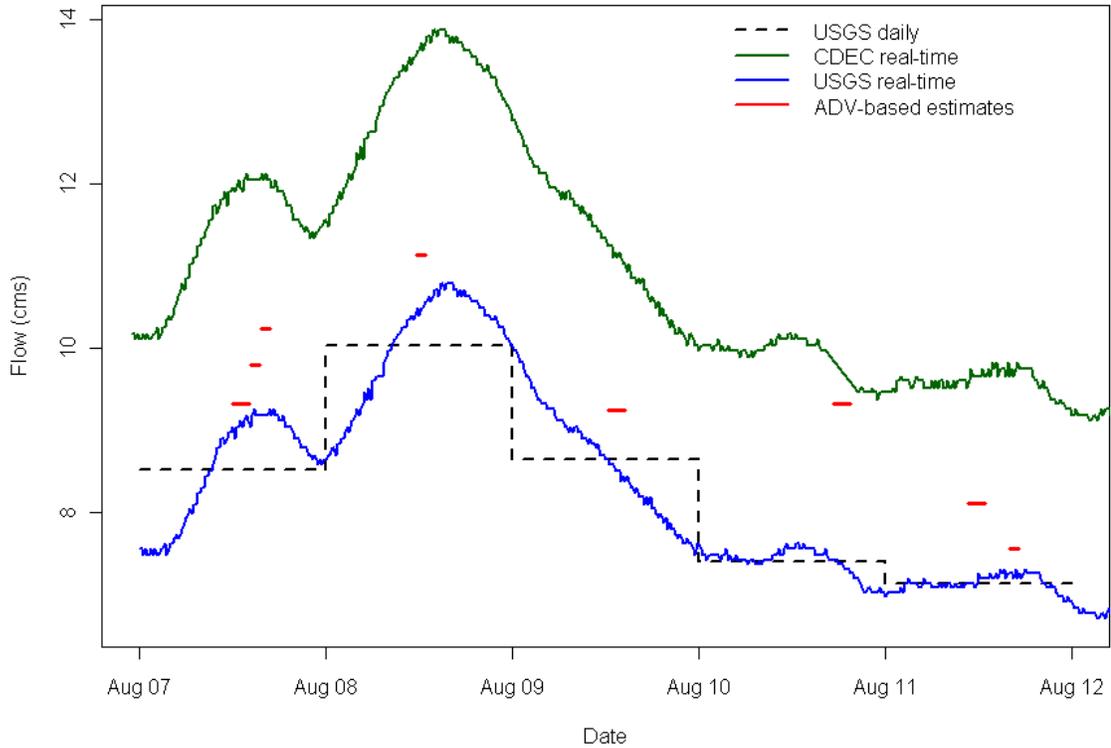
Figure 3.10: Plot comparing the measured and computed data to the CDEC real-time, USGS real-time corrected, and the USGS daily flow data. The variation of line lengths for the measured data encompasses the varied time each transect run took.

The resulting flow estimates compare relatively well with the USGS data. Table 3.3 summarizes the percent difference between the measured and computed flows with the data provided from CDEC and USGS. For daily USGS flow calculations, the measured and computed flows were first averaged daily and then the percent difference computed. There was a period when the CDEC data reported large negative values, which were excluded, leading to the NA value on August 10th. On average, our flow estimates agreed well with the USGS RT and the USGS daily-averaged flow data with exception to the August 10th data.

Table 3.3: Table shows % differences between our measured and computed data with that of CDEC and USGS (RT and daily).

	Date	Run	CDEC RT	USGS RT	Date	USGS daily
			% difference between measured and observed			
	Aug 7	1	-22.33	2.72	Aug 7	14.67
	Aug 7	2	-18.70	6.39	Aug 8	11.09
	Aug 7	3	-15.27	11.26	Aug 9	6.96
	Aug 8	6	-18.44	6.29	Aug 10	26.13
	Aug 9	17	-16.97	8.78	Aug 11	9.76
	Aug 10	26	NA	26.73		
	Aug 11	27	-16.20	13.15		
	Aug 11	33	-22.22	4.49		
Average % difference			-18.59	9.98		13.72

4   River Modeling Background

4.1   Overview

Before incorporating the field data into river models, this chapter will discuss the motivation for and background on the river models used in the study. In general, river models rely upon the computational power and the resolution of the field data to adequately parameterize the models. As model dimensions increase from one to three dimensions, data requirements likewise increase dramatically, with river bed topography being the most important parameter. Model applications primarily aim to estimate flow with varying degrees of detail. For instance, 1-D models estimate volumetric flows, 2-D models estimate depth-averaged velocities, and 3-D models estimate point velocities. Common applications from flow alone often include flood prediction and structural design, as for bridge piers. Additional applications such as geomorphology (bed transport) and constituent transport are typically driven from flow and velocity information. This chapter will cover the different river models and their applications with a more detailed discussion on two-dimensional, depth-averaged river models and their utility in studying flow and constituent transport and mixing.

4.2   Governing Principles

Hydrodynamics in most river models are governed by the mass and momentum conservation equations. Since 2-dimensional models are the focus of this work, the 2-dimensional forms of these equations will be discussed. Depth components are averaged in the representation and the fluid is assumed to be incompressible. The resulting mass conservation or continuity equation is expressed as

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(vh) = 0, \tag{4.1}$$

where h is the water surface elevation, t is time, u is the depth-averaged velocity in the x or longitudinal direction, and v is the depth-averaged velocity in the y or transverse direction. For momentum conservation equations, 2-D models include terms to account for water surface elevation changes, river bed slope, and shear stress terms to account for turbulent effects. The resulting x and y momentum conservation equations are

$$h\left[\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g\frac{\partial h}{\partial x} + g\frac{\partial}{\partial x}(z_b) - \frac{\varepsilon_t}{\rho}\frac{\partial^2 u}{\partial x^2} - \frac{\varepsilon_t}{\rho}\frac{\partial^2 u}{\partial y^2}\right] + \frac{1}{\rho}(\tau_{bx}) = 0,$$

$$h\left[\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g\frac{\partial h}{\partial y} + g\frac{\partial}{\partial y}(z_b) - \frac{\varepsilon_t}{\rho}\frac{\partial^2 v}{\partial x^2} - \frac{\varepsilon_t}{\rho}\frac{\partial^2 v}{\partial y^2}\right] + \frac{1}{\rho}(\tau_{by}) = 0,$$

(4.2)

where g is gravity, z_b is the river bed elevation, ρ is the fluid density, ε_t is eddy viscosity, and τ_{bx} and τ_{by} are the bed shear stresses in the x and y directions respectively. Similarly, the transport equation for a conservative constituent and without source and sinks takes a similar form to Equation (4.2):

$$h\left[\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} - D_x\frac{\partial^2 c}{\partial x^2} - D_y\frac{\partial^2 c}{\partial y^2}\right] = 0,$$

(4.3)

where c is the concentration of the constituent, and D_x and D_y are the dispersion coefficients in the x and y directions respectively.

Model calibration for the hydrodynamics described in Equation (4.2) requires parameters associated with eddy viscosity and bed shear stresses. Eddy viscosity is described more a property of the flow rather than the fluid and calibrated by comparing the velocity distribution results from the model with the measured [36]. The parameter is commonly calculated using

$$\varepsilon_t = \varepsilon_c \rho u_* h,$$

(4.4)

where ε_c is a fitted parameter and u_* is the shear velocity calculated with Equation (4.6). The fitted parameter, ε_c, for natural channels often estimated as 0.6 with an error of ±50% [37] or with various empirical formulas with similar forms [38]. One such recent formulation for ε_c is

$$\varepsilon_c = 0.145 + \frac{1}{3,520}\left(\frac{u}{u_*}\right)\left(\frac{b}{h}\right)^{1.38},$$

(4.5)

where b is the river channel width [39]. Shear velocity is commonly expressed as

$$u_* = \sqrt{gRS},$$

(4.6)

where R is the hydraulic radius, which is the area of the cross section divided by the wetted perimeter, and S is also the slope that is estimated as the slope of the water surface elevation in this work.

The other fitted parameter is a bed roughness factor included in the formulation of bed shear stress. The chosen model bed friction factor is calibrated by comparing the water surface elevations between the model results and the measured [36]. The formulation of bed shear stress is known as

$$\tau_{bx} = \rho c_f u \sqrt{u^2 + v^2},$$
$$\tau_{by} = \rho c_f v \sqrt{u^2 + v^2},$$

(4.7)

where c_f is a dimensionless friction coefficient similar to the friction factor found in the Darcy-Weisbach equation [40]. This coefficient relates with better-known Manning coefficients with the relations as follows:

$$c_f = \frac{gn^2}{h^{1/3}},$$

(4.8)

where n is the Manning coefficient. The Manning formula relates the coefficient to flow as follows:

$$\bar{u} = \frac{R^{2/3}S^{1/2}}{n},$$

(4.9)

where $\bar{u}$ is the mean cross-sectional velocity. The Chezy formula is expressed as

$$\bar{u} = C\sqrt{RS}.$$

(4.10)

These formulas provide methods for estimating flow for a single channel and therefore collapsing roughness to one dimension. For 2-D modeling, roughness coefficients for the bed material alone, such as sandy or gravelly beds, are adequate for the model [36]. With these fitting parameters, eddy viscosity (ε_t) and roughness coefficient (n or C), the 2-D model solves for water surface elevations (h) and velocities (u and v) at each nodal element.

For transport modeling, the dispersion coefficient is similarly fitted. The dispersion coefficients can be assumed to isotropic ($D_x = D_y$). For fitting purposes, however, Suh (2009) found anisotropic assignments for the dispersion coefficients to work the best [41]. The lateral dispersion coefficient, D_y, is equal to the eddy viscosity calculated in Equation (4.4) divided by density. Fischer (1979) provides two approximate expressions for the longitudinal dispersion coefficient, D_x, and shown below as

$$D_x = 0.593 h u_*,$$

(4.11)

$$D_x = \frac{0.011 \bar{u}^2 b^2 u_*}{h}.$$

(4.12)

2-D river models are generally solved numerically using finite difference or finite element approaches. To deal with complex geometries, the 2-D models often employ the finite-element method to solve the mass and momentum conservation equations [36], [42], [43]. The method divides the model spatial domain into smaller elements, often in triangular or quadrilateral shapes. Then, the finite element method integrates weighted residuals for Equations

32

(3.1) and (3.4) at the corners of each element. The estimates of weighted residuals involves testing approximations for the dependent variables, *h*, *u*, and *v*, expressed as

$$\tilde{h} = \sum_{i=1}^{m} N_i h_i,$$

$$\tilde{u} = \sum_{i=1}^{m} N_i u_i, \qquad (4.13)$$

$$\tilde{v} = \sum_{i=1}^{m} N_i v_i,$$

where the accented variables, $\tilde{h}, \tilde{u},$ and $\tilde{v}$ are approximated values, *m* is the number of nodes used in the finite element method, e.g. *m* = 3 for triangular elements, and *N* is the set of weight or test functions. Then, the approximated values are substituted into Equations (4.1) and (4.2) leaving residuals rather than values of zero. The residuals are then integrated over the element domain,

$$\int_R N_i f(\tilde{h}, \tilde{u}, \tilde{v}) dR = 0, \qquad (4.14)$$

where *f* are the continuity or momentum conservation expressions which equal the residuals and *R* is the element domain. The final step solves this expression for the actual dependent variables, *h*, *u*, and, *v*, for the element nodes [36], [42].

4.3   River Modeling Background

River modeling initially took form in 1-D modeling often called 'streamflow routing'. This method primarily estimates river stage, flow discharge, and discharge related parameters helpful for flood inundation prediction and structural systems. Early versions of this model initially solved for the conservation of mass equation only for practical purposes without computers. With computers, the models discretized and applied finite difference methods to solve equations of conservation of mass, momentum, and energy [44-46]. Currently watershed-scale hydrologic models used in predicting water large-scale water supplies include streamflow routing models along with other water reservoirs and fluxes such as precipitation, groundwater, and evapotranspiration. These models are typically easy to parameterize requiring average channel areas, widths, slopes, and flows given by gauging stations. Roughness, typically given in the Manning roughness coefficient, is often fitted to reasonable values but can be estimated to account for other factors affecting roughness not included in 1-D channel representations such as channel meandering and vegetation [36], [47]. Figure 4.1(a) shows an example of stream routing

nodes for a USGS model called Diffusion Analogy Surface-Water Flow Model (DAFLOW) on a gridded layer representing the discretization of the groundwater model. Additional 1-D models, such as the Hydrologic Engineering Center's River Analysis System (HEC-RAS), allow input of varying cross-sectional geometries to account for differing friction parameters between the main channel, river banks, and the floodplain when such conditions arise as seen in Figure 4.1(b). With focus on rivers alone, HEC-RAS model estimates for flows are often applied to sediment transport capacity and bridge scouring [48].
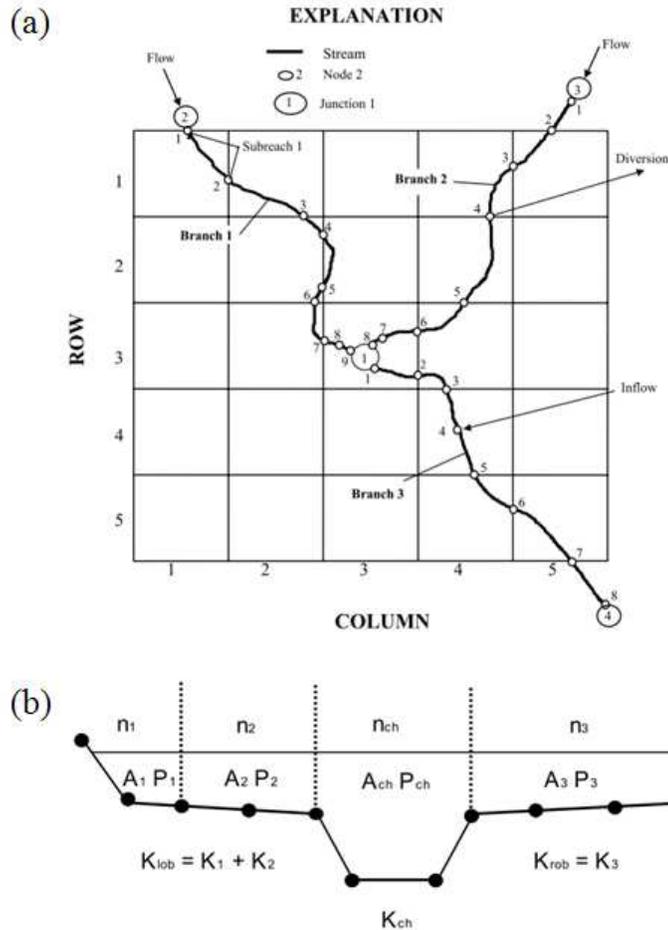
Figure 4.1: (a) Example streams showing the flexibility in node placement for DAFLOW [46]. (b) Example cross-section in HEC-RAS assigning different roughness and conveyance values to subsections [48].

Two-dimensional river models provide further detail for similar estimates reported from one-dimensional models. Two-dimensional estimates for water velocity magnitudes and water surface elevations generally performed better than one-dimensional models around structures

such as bridges [49], and the detailed spatial velocities help identify advantageous flow regimes for sustainable fish habitats [50], [51]. Commonly, 2-D models solve the mass conservation and two horizontal components of the momentum conservation equations in grids that satisfy finite difference, volume, or element methods. The models assume uniform vertical velocity distributions and hydrostatic pressure distributions [36]. These models can execute in reasonable times on personal computers depending on the number of cells from the grid. Model accuracy, however, requires more river bed topography data than that of 1-D models and similar to the extensive data collected in Chapter 3. Unlike roughness estimates for 1-D models, roughness for 2-D models only account direct bed shear where observed bed material, (e.g. sand, gravel, vegetation), and bed form geometries not included in the bed topography are adequate to estimate a roughness coefficient [36], [51]. The model requires the lateral eddy viscosity to be calibrated and fitted, often to "unrealistically" high values to maintain model stability [36]. With the advances in data collection for the bed topography and flow and stage boundary conditions, 2-D models have become more widely available freely and commercially.

Finally, three-dimensional models are still sparsely used and mainly for research purposes for the past 15 to 20 years [52]. Potential applications of the model help describe secondary flows that non-turbulent 2-D models do not include and treatment of bed roughness with porosity [52]. The prevalent limitation is the bed topography resolution at common reach scales [52], [53].

4.4   River Confluence Studies

River confluences exhibit complex flow environment where limited studies have been performed. Confluences occur naturally at braided rivers but are often channelized for navigation purposes. Focus on flows involved lab-scale flumes and intensive data collection to accurately describe flow behavior for models. Mixing downstream of the confluence, geomorphology, and the subsequent effects on ecology have been studied in varying degrees [54]. This section focuses on flow characteristics, two-dimensional modeling, and downstream constituent mixing.

Characterization of flow structure at confluences remains difficult to measure. Early studies focused on laboratory-scale models [55], [56]. With advancements to velocimeters or current meters, flow structure at small [57], [58] and large [59] rivers has been studied in further detail. Rice et al. summarize six particular flow regimes with some seen in Figure 4.2:

1. stagnated flow at upstream corner,

2. deflected flow as flows converge,

3. flow separation at downstream corners or bends,

4. accelerated flow as flows converge,

5. shear layer possibly leading to helical flow cells seen in Figure 4.2(b),

6. and flow recovery downstream.
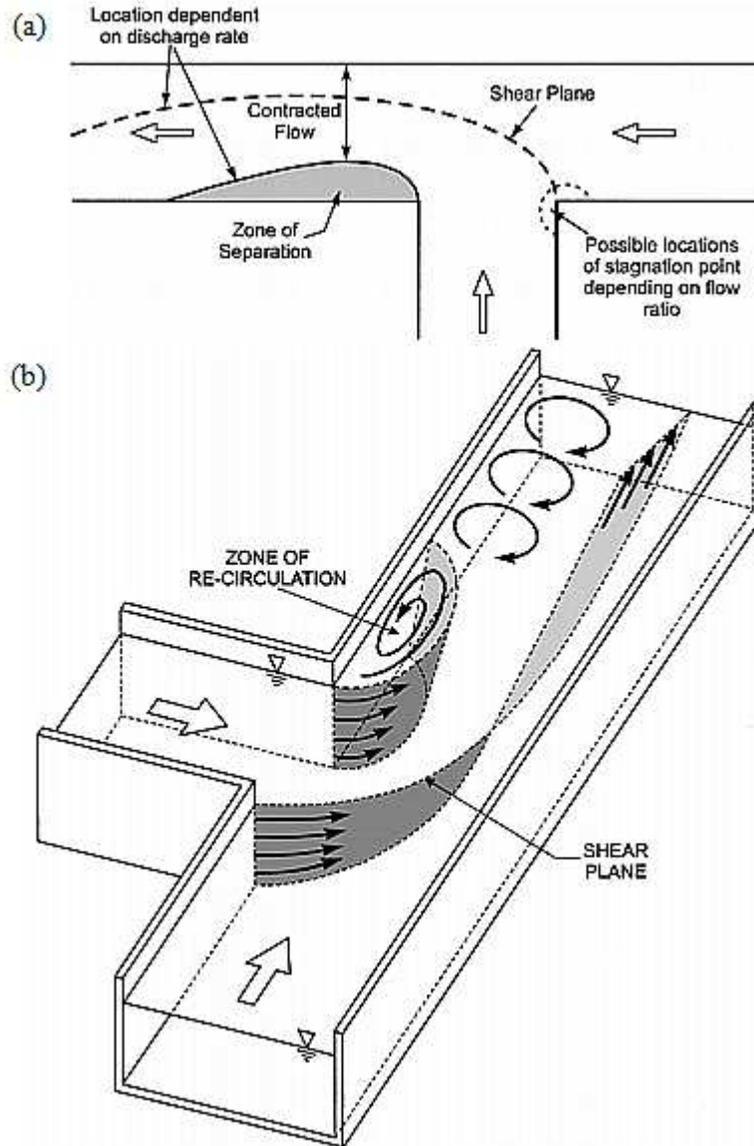
Figure 4.2: (a) Top down view of 90 degree confluence with identified flow regimes. (b) Three-dimensional view of confluence showing helical flow cells. Both from [56].

Major influencing factors to the flow regimes include the angle of the upstream junction corner, and discharge ratio between the two channels. Other influencing factors include upstream

channel bend shapes (convex or concave), bed roughness for shallow streams, and bed geometries of discordant beds and extreme scour zones [54]. This work will focus on the hydrodynamics flow recovery zone downstream of the confluence and mixing processes.

Applications of two-dimensional models to river confluences remain sparse. The models does not characterize complex flows such mixing and flow separation. Still, the 2-D models require less computational demands, less channel topography data, and have less numerical instabilities making them more advantageous for particular cases [60]. Weerakoon et al. (2003) applied a depth-averaged model to a confluence and performed well given appropriately estimated bed roughness and eddy viscosity values. Likewise, Roca et al. (2009) achieved good results river confluence under flood conditions but had to adjust bed roughness parameters from the original bed material for the best fit [61]. In both cases, model performance was judged by comparing observed and modeled water surface elevations. This work, with the available field data, aims to apply 2-D models to a confluence and judge performance using the velocity profile averaged at cross-sections in the recovery zone.

The modeling of constituent mixing processes at river confluences has also received only limited attention. Mixing models often focus on end-member analysis analyzing both appropriate background and injected tracer concentrations [62-64]. These studies effectively define the discharge ratios based on sources. Advective-dispersive models are often limited to simplified solutions, often to find mixing lengths or distance downstream where complete mixing occurs. Complete mixing is commonly said to be achieved when all measured concentrations in the cross section are within 5% of the mean. Fischer (1979) presented an analytic solution for an ideal case assuming equal discharge between two rectangular channels and uniform width downstream of the transect [37]. His solution is expressed as follows:

$$c = \frac{c_0}{2} \sum_{i=-\infty}^{\infty} \left\{ erf\left(\frac{y' + \frac{1}{2} + 2i}{\sqrt{4x'}} \right) - erf\left(\frac{y' - \frac{1}{2} + 2i}{\sqrt{4x'}} \right) \right\} \tag{4.15}$$

where y' is

$$y' = \frac{y}{b}, \tag{4.16}$$

x' is

$$x' = \frac{xD_y}{\bar{u}b^2}, \tag{4.17}$$

37

c_0 is the initial injected concentration, and *erf* is the error function. In this work, this model will be compared with the finite-element, two-dimensional advection-dispersion model described in Equation (4.3).

In summary, 2-D modeling efforts aimed as characterizing the hydrodynamics and mixing processes at confluences have not been extensive. Though complex flow structure may overwhelm the capabilities for 2-D models, this work focuses far enough downstream where flow begins to recover and where 2-D models may be adequate. And whereas previous works validated 2-D models of confluences by comparing water surface elevations, this work aims to compare the model and measured cross-sectional, depth-averaged velocity profiles. For further validation of the estimated eddy viscosity, the results for advection-dispersion model are compared to the measured cross-sectional salinity profile. Finally, from a more practical user's perspective, the 2-D model results are compared to pre-existing ideal solution to determine if there are appreciable performance differences for lateral mixing estimates.

5   Hydrodynamic Model Results and Analysis

5.1   Overview

Field data collected for flow, salinity, river bed topography, and water surface elevations parameterize depth-averaged, two-dimensional hydrodynamic and advection dispersion models. Two finite-element hydrodynamic codes are the focus of this work. The Federal Highway Administration authors the Depth-averaged Flow and Sediment Transport Model (FST2DH) which is part of the Finite Element Surface-water Modeling System (FESWMS). Many developers including Resource Management Associates (RMA) and the US Army Waterways Experiment Station contributed to the second hydrodynamic code, RMA2. Using the depth-averaged hydrodynamics results from RMA2, RMA4 models the two-dimensional advection-dispersion. This chapter first discusses model parameterization and then analyzes the model results with the measured data for:

1. Relationship between node spacing and the model accuracy
2. Better performance between the two hydrodynamic codes, FESWMS and RMA2
3. Determine the best RMA2 model parameters for the advection-dispersion model, RMA4, and
4. Accuracy of estimated dispersion coefficients.

Marginally better hydrodynamic results came from using the FESWMS code, closer node spacing in all models, and automatic assignment of the roughness coefficient. Closer node spacing, however, leads to increased instabilities and greater computational time. Direct and automatic eddy viscosity assignment produced similar model results, but direct assignment has the advantage for parameter estimation using measured data.

5.2   Model Parameterization

Aside from flow (Table 3.2) and river channel topography (Figure 3.3), both models require channel friction, expressed as Manning's coefficient, and eddy viscosity estimates. The Manning coefficient is solved for by rearranging Equation (4.9) to

$$n = \frac{R^{2/3}S^{1/2}}{\bar{u}}, \tag{5.1}$$

where the mean velocity is calculated by dividing the flow by the channel cross-sectional area. The slopes are estimated from water surface elevations (WSE's), and because they differ slightly for each channel, two Manning's roughness coefficients are calculated for the Merced and San

Joaquin Rivers. Note that the cross-sectional areas and wetted perimeters were calculated using the MBA interpolated river bed topography. The distance was estimated using the Google Maps line tool. Due the steady-state assumption for the both of the models codes, FESWMS and RMA2, the volumetric flows, Q, were averaged from August 8 onward. Table 5.1 below compiles the data for the Manning's roughness coefficient estimation.

Table 5.1: Summary of calculations for Manning's roughness coefficient.

	Downstream		Upstream
Transect ID	T1	T2	T3
Cross-sectional area (m^2)	61.89	8.57	14.42
Wetted perimeter (m)	48.12	20.95	20.99
R (m)	1.29	0.41	0.69
Estimated channel length to T1		423.68	383.07
WSE (m)	11.379	11.442	11.442
Slope, S, to T1		0.000149	0.000164
Q (m^3/s)	9.07	2.71	7.21
$\bar{u}$ (m/s)	0.15	0.32	0.50
n		0.0212	0.0200

The roughness coefficients for both rivers fall within the value range for sand, $n = 0.012$ to 0.026, for diameters 0.2 to 1 mm [47]. Table 5.1 also includes the flow boundary conditions that include the upstream flow conditions at T2 and T3 and the downstream WSE at T1. For RMA2, roughness can also be set automatically for each mesh node by depth, and model runs in this work will compare this setting to manual assignment of Manning roughness coefficients.

Similar eddy viscosity estimates are calculated using Equation (4.4) and (4.5). Values for ε_c estimated using Equation (4.5) used the mean cross-sectional velocity and depth, h_{xs}. Eddy viscosity values for both the constant and calculated ε_c are summarized below in Table 5.2. As mentioned previously, low eddy viscosity values ($\varepsilon_t <$ approximately 30 Pa-s) lead to unstable solutions [36]. The, the ratio between the eddy viscosities is scaled accordingly. For sensitivity analysis, the ratios for the Merced River to San Joaquin River eddy viscosities, R_{ms}, were varied to assess the effects of the ±50% error when ε_c equals the constant value, 0.6 [37].

Table 5.2: Summary of calculations for eddy viscosity values.

	Variables	Merced River	SJR
	ε_c	0.6	0.6
	u_* (m/s)	0.024	0.033
	b (m)	21.68	21.42
	h_{xs} (m)	0.40	0.67
	ε_t (Pa-s)	5.99	13.72
Equation (4.5)	ε_c	1.07	0.65
	ε_t (Pa-s)	10.33	14.59

Table 5.3: Summary of calculations for ratios between the Merced and San Joaquin River eddy viscosities.

	% errors in ε_t	
Merced	SJR	R_{ms}
50	-50	1.31
15	-15	0.59
5	-5	0.48
0	0	0.44
-5	5	0.39
-15	15	0.32
-50	50	0.15
Equation (4.5) 0	0	0.71

Eddy viscosities computed above are assumed to be isotropic. However, given the equivalence between the dispersion coefficients and eddy viscosities discussed previously and the good agreement Suh (2007) found for anisotropic dispersion coefficients, anisotropic eddy viscosity and dispersion coefficient terms were also examined [41]. Suh (2007) found two ratios, R_{xy}, between longitudinal, D_x, and transverse, D_y, dispersion with low normalized mean squared error, NMSE, ranged from $R_{xy} = 6.43$ (NMSE = 1.5) and 0.643 (NMSE = 2.7) [41]. The average for each channel ratios for Equations (4.11) and (4.12) are 9.62 and 4.95 respectively. A few ratios are examined in this work for further model evaluation. As this discussion suggests, the calibration of eddy viscosities and dispersion coefficients become increasingly complex. In RMA2, Osting (2007) suggests automatic eddy viscosity assignment using the Peclet number, Pe, and expressed as

$$\mathrm{Pe} = \frac{\rho u l}{\varepsilon_t}, \tag{5.2}$$

where l is the spacing between nodes [65]. Here, the model takes the locally computed velocity and estimate the eddy viscosity given a prescribed Peclet number. From the values of eddy viscosity from Table 5.2, the estimated Peclet number for each cross section becomes

unrealistically large (Pe = 73 for T3 and Pe = 105 for T2) compared to the recommended range for Pe (15 to 40) stated in the RMA2 manual. Hence, Peclet numbers were chosen arbitrarily within the manual-recommended range and compared with the directly assigned eddy viscosities.

Given these estimated fitting parameters and boundary conditions, the software package known as Surface-water Modeling System (SMS) that includes hydrodynamic and advection-dispersion codes discussed solves the models if the models converge. SMS provides a graphical user interface and generates a finite element mesh for the model domain. Figure 5.1 describes the main data inputs and processes of the SMS hydrodynamic and advection-dispersion models. The red outlined boxes in the figure summarize the adjustable parameters for output accuracy and model stabilities. The last adjustable model parameter is node spacing. Osting (2007) discusses that node spacing should always be greater than the river depth. A few different node spacing's are tested in the following sections to see how to optimize accuracy of model results and computation time.

The remaining parameters that affect model stability and runtime performance include defining the river bank boundaries, wetting and drying of mesh elements, and the depth convergence limit. SMS can define the river bank by defining a particular contour elevation as the boundary arcs. The range of WSE's from Table 5.1 from the downstream to upstream transect was 11.38 to 11.44 m respectively. The lower WSE led to less element sides to cross steep contour gradients found in the river banks. Element wetting and drying settings include drying depth for RMA2 or storativity depth for FESWMS. Convergence limits sets the maximum changes between model iterations that are considered acceptable for converged solutions. FESWMS has convergence limits for depth and flow, and RMA2 only has depth convergence limit. Common settings for these parameters along with the upstream concentrations needed for the advection-dispersion modeling are described below in Table 5.4.

Table 5.4: Remaining input variables in the models.

Remaining parameters	Value
Contour boundary (m)	11.38
Dry depth (m)	-0.12
Storativity (m)	0.5
Depth convergence limit (m)	0.05
Unit flow convergence limit (m)	0.05
SJR concentration (μS/cm)	1531.6
Merced concentration (μS/cm)	286.8
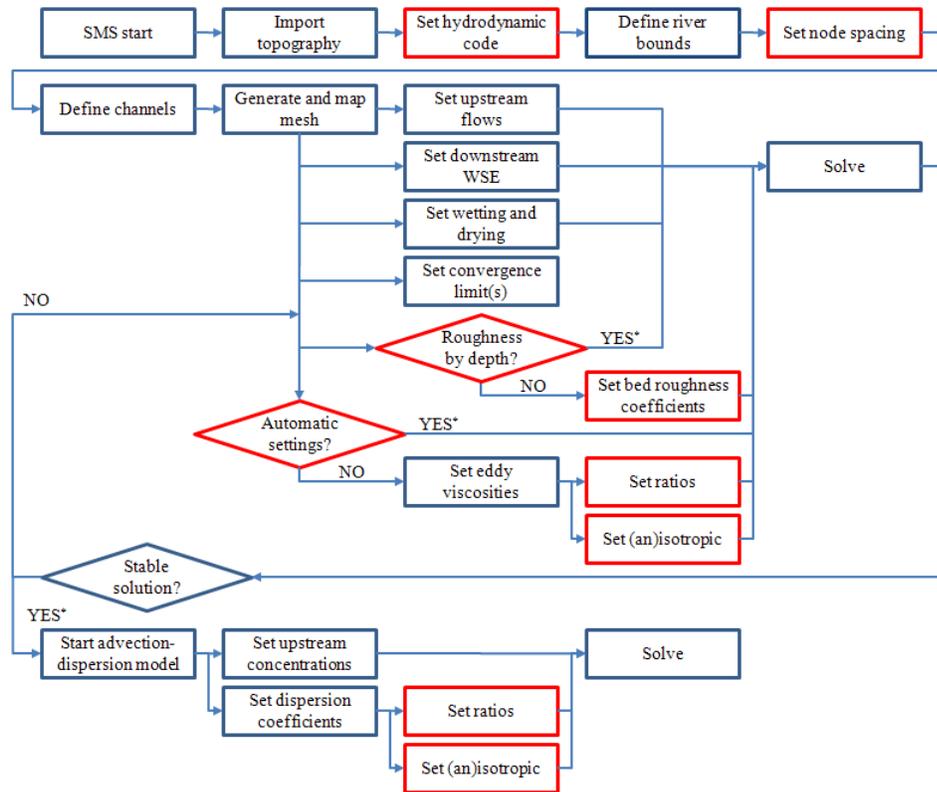
Figure 5.1: Flow diagram of the processes for both hydrodynamic and advection-dispersion modeling using SMS. Red boxes denote variables in this work that are adjusted. [*]Specific features for RMA2 hydrodynamic model.

For the following sections, judgment for best hydrodynamic settings involves examining the model velocity profiles and the difference between upstream and downstream WSE's, Δh. Quantitative examinations involve taking the average absolute percent difference, AAPD, between the modeled and measured lateral velocity profile and the percent difference between the modeled and measured Δh. One qualitative consideration includes the shape of the lateral velocity profiles.

5.3   Node Spacing

To examine the effect of node spacing, four FESWMS models computed solutions for spacing values of 1, 1.5, 1.8, and 2 m. Roughness values from Table 5.1 were used for the Merced and San Joaquin channels. Eddy viscosity values were increased for stability to 50 and 115 Pa-s for the Merced and San Joaquin channels respectively, maintaining the theoretical ratio from Table 5.2. Figure 5.2 plots the model outputs to the observed depth averaged velocity profiles for transects downstream of the confluence joint with error bars for observational

uncertainty. Calculated uncertainty was propagated from standard deviation of the velocity at each raster point to the depth-averaged velocity. Large error bars were observed and were likely due to low number of measured point velocities and turbulent eddies. At T4, model outputs differ slightly from each other. At T1, however, the model output for 1 m node spacing begins to drift away from other model outputs. The average absolute percent differences (AAPD's) between the observed depth-averaged velocity magnitudes and the linearly interpolated depth averaged model results are compared in Table 5.5.
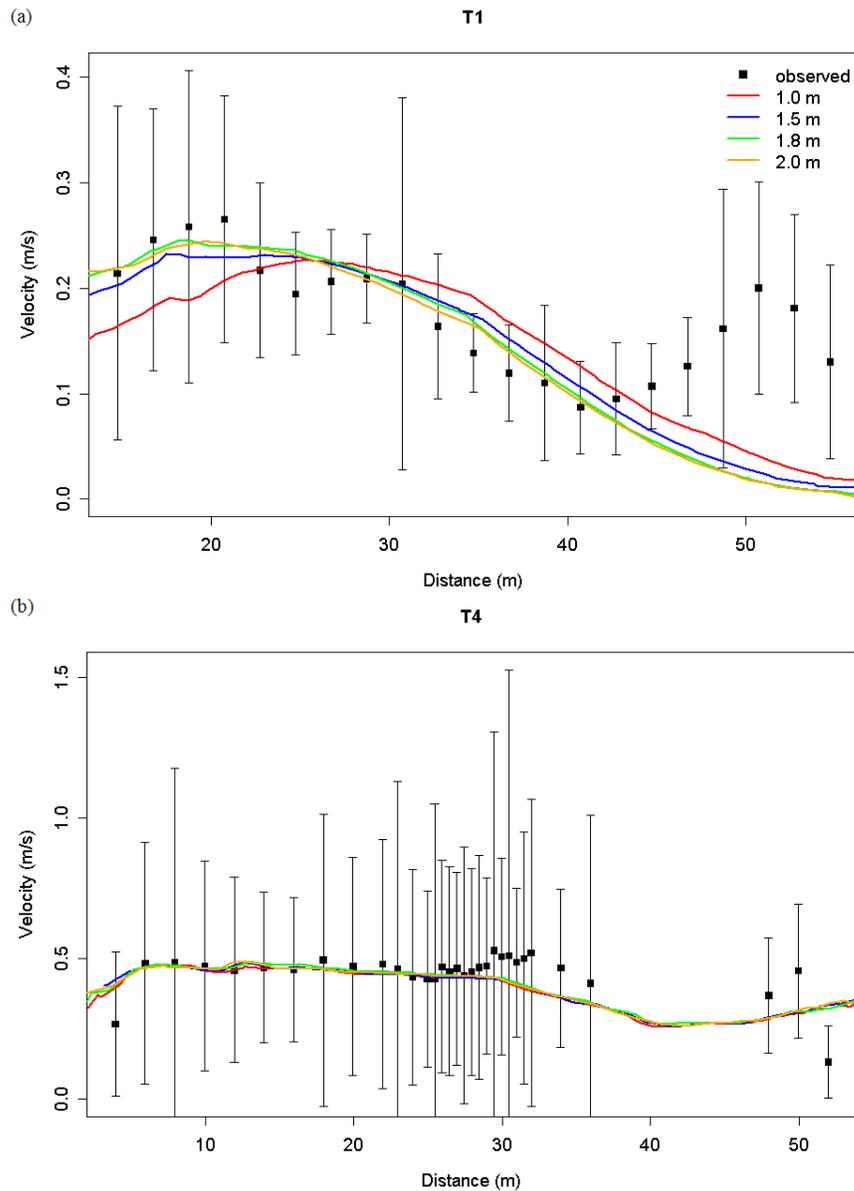
Figure 5.2: (a) Modeled and measured velocity results at T1. (b) Modeled and measured velocity results at T4. Distances are from the left bank facing downstream.

Table 5.5: Average absolute percent differences (AAPD) between modeled and measured velocities for different node spacing values.

Node spacing (m)	Absolute % difference	
	T1	T4
1.0	33.61	15.39
1.5	30.49	15.83
1.8	30.49	14.72
2.0	29.97	15.46

AAPD is used as a metric to avoid averaging out error that could be apparent between the modeled and measured velocity results. Most obvious from Table 5.5 and the accompanying figures are the comparatively large errors associated with T1. These seemed to have been caused by a systemic error in the model structure. In short, the velocity profile suggests a secondary channel in that cross section which was not captured well by the model regime. Quantitatively, 1 m node spacing showed greater AAPD for T1. Osting (2007) notes model instability and less accurate results as node spacing approaches the depth of the water [65]. For analyzing WSE, the average upstream WSE at T3 estimated by the model was 11.438 m which is marginally different from the measured T3 measurement of 11.442 m indicating relatively good estimates for the Manning roughness coefficients. For model compilation times, the spacing of 1.8 m is chosen for subsequent modeling efforts.

5.4   Comparing FESWMS and RMA2 Codes

RMA4 uses the RMA2 hydrodynamic solution flow field to drive the advection-dispersion process. To validate RMA2 model performance, RMA2 model results were compared with FESWMS under similar conditions. For the RMA2 solution to converge, the downstream WSE boundary conditions needed to be relaxed to 11.48 m rather than the independently estimated 11.38 m value. Table 5.6 shows model performances using similar eddy viscosity values, Manning's roughness, downstream WSE, and node spacing parameters for both FESWMS and RMA2 models. The AAPD values indicate better FESWMS performance at T1 in the later runs. RMA2 were marginally better for T4, although the previously limitations with this cross-section may negate any significance in this finding.

Table 5.6: AAPD values between modeled and measured velocities comparing RMA2 and FESWMS codes.

			RMA2		FESWMS	
			Absolute % difference
Run	Merced ε_t (Pa-s)	SJR ε_t (Pa-s)	T1	T4	T1	T4
1	30.00	68.73	33.31	24.60	35.45	25.32
2	40.00	91.64	33.16	24.94	31.93	25.72
3	50.00	114.55	32.75	25.08	30.46	26.04
4	60.00	137.46	32.85	25.22	29.94	26.32
5	70.00	160.36	33.41	25.33	29.43	26.59

As a result of the WSE downstream boundary condition, the model results for upstream were also shifted upward. However, assuming that the modeled difference between upstream and downstream WSE, Δh, the modeled and the measured ($\Delta h = 0.063$m) differences should be similar, and the percent difference between the modeled and measured Δh is chosen as an appropriate additional metric to verify model accuracy. Table 5.7 below summarizes the comparisons.

Table 5.7: Percent differences between modeled and measured Δh (0.063 m) comparing RMA2 and FESWMS codes.

	RMA2			FESWMS		
Run	T3 WSE (m)	Δh (m)	% difference	T3 WSE (m)	Δh (m)	% difference
1	11.512	0.033	-47.72	11.513	0.033	-48.07
2	11.514	0.034	-46.40	11.514	0.034	-45.48
3	11.515	0.035	-45.18	11.516	0.036	-43.39
4	11.515	0.035	-43.80	11.518	0.038	-40.41
5	11.516	0.036	-42.39	11.519	0.039	-38.53

Both models underestimated the changes in WSE from upstream to downstream. Figure 5.3 below compares the measured and simulated velocity profiles at T1 and T4. Again, the modeled T4 velocity profile results do not differ greatly from one another, and this is a continuing trend for remaining model results. Notice, however, that the modeled T4 velocities shown in Figure 5.3(b) are generally lower than the velocities in Figure 5.3(a). For T1, the lower eddy viscosity values for Run 1 denoted by the green and red lines captures the maximum velocity better for the San Joaquin River side. In addition, this first set of RMA2 runs seem to begin to capture the secondary channel observed on the Merced River side of T1, albeit poorly. In summary, as would be expected altering the downstream WSE boundary conditions significantly affects the

difference between upstream and downstream WSE. Unfortunately, this alteration was necessary to achieving stable execution of the RMA2 model.
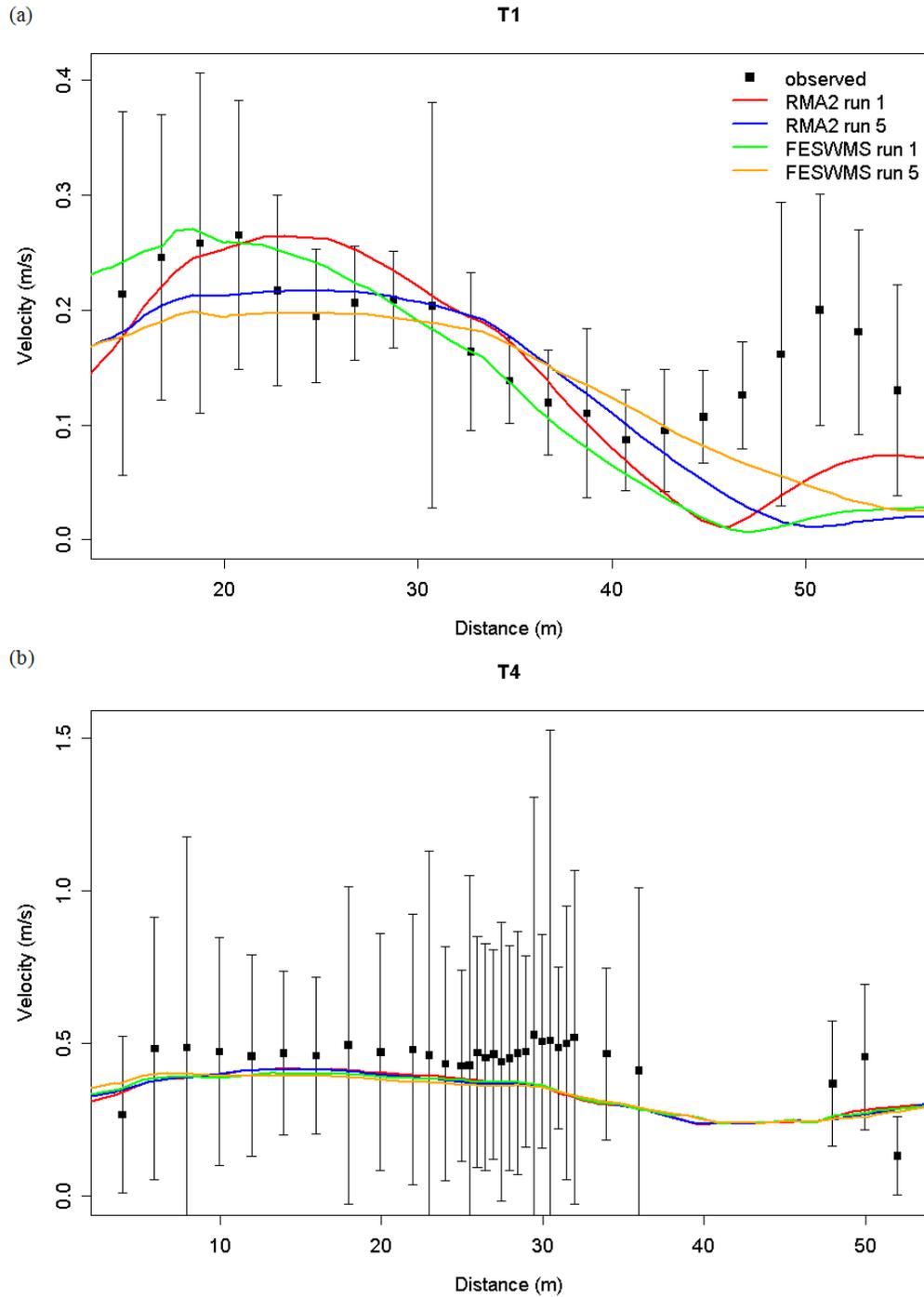
Figure 5.3: (a) Modeled and measured velocity results at T1. (b) Modeled and measured velocity results at T4. The run numbering refers to Table 5.6 comparing RMA2 and FESWMS codes.

Although significant differences between RMA2 and FESWMS simulations were observed in the water velocity profiles at T1, RMA2 results remained reasonable with lower eddy viscosities, and exhibited a qualitatively better fit for the SJR channel of T1. Hence, from this point forward, results are limited to those based on the RMA2 model.

5.5   Comparing Manually and Automatically Assigned Roughness Coefficients

The automatic roughness by depth feature provided by RMA2 was compared with the approach of manually assigning Manning roughness values. Again, model runs were performed with varying the eddy viscosity values. The assigned Manning roughness values are summarized in Table 5.1. The AAPD values between modeled and measured water velocities for T1 and T4 are summarized below in Table 5.8.

Table 5.8: AAPD values between RMA2-modeled and measured velocities comparing manually and automatically assigned roughness coefficients.

			Manual assignment		Assignment by depth	
			Absolute % difference			
Run	Merced ε_t (Pa-s)	SJR ε_t (Pa-s)	T1	T4	T1	T4
1	30.00	68.73	33.31	24.60	45.39	24.72
2	40.00	91.64	33.16	24.94	31.24	24.98
3	50.00	114.55	32.75	25.08	31.30	25.16
4	60.00	137.46	32.85	25.22	31.58	25.34
5	70.00	160.36	33.41	25.33	31.87	25.45

Velocity results for run 1 and model applying roughness by depth resulted in relatively large AAPD values for T1, but otherwise the AAPD values do not differ greatly between the two approaches. Table 5.9 also summarizes the percent differences between the upstream (T3) and downstream (T1) WSE. Parameterizing using roughness by depth greatly increases the model accuracy, with lower eddy viscosity values performing better. The simulated and measured velocity profiles for T1 and T4 are shown in Figure 5.4. For T1, large error occurs for model run 1 using roughness by depth (green line), but otherwise the assigned roughness model runs (red and blue lines) seem to characterize the shape of the SJR channel better than the roughness by depth model runs. For T4, the roughness-by-depth models may capture some of the Merced channel velocity profile.
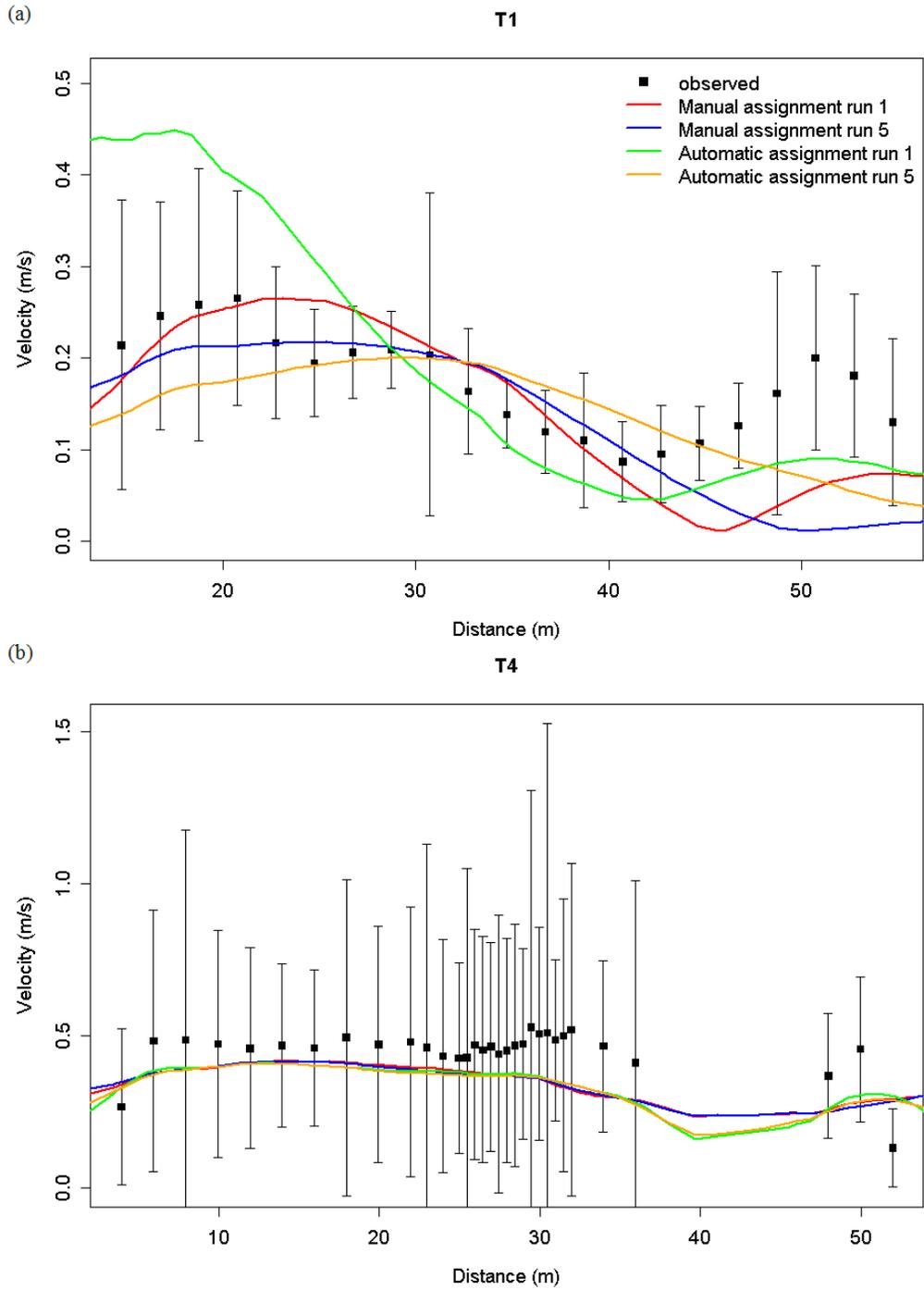
Figure 5.4: (a) Modeled velocity and measured results at T1. (b) Modeled and measured velocity results at T4. The run numbering refers to Table 5.8 comparing methods of assigning roughness coefficients.

Table 5.9: Percent differences between modeled and measured Δh comparing manually and automatically assigned roughness coefficients.

	Manual assignment			Assignment by depth		
Run	T3 WSE (m)	Δh (m)	% difference	T3 WSE (m)	Δh (m)	% difference
1	11.512	0.033	-47.72	11.542	0.065	2.65
2	11.514	0.034	-46.40	11.545	0.065	2.76
3	11.515	0.035	-45.18	11.546	0.066	5.35
4	11.515	0.035	-43.80	11.548	0.068	7.88
5	11.516	0.036	-42.39	11.549	0.069	10.05

In summary, the models using roughness by depth greatly increases accuracy for Δh and qualitatively shows better velocity results at T4. These advantages outweigh the qualitatively worse T1 velocity results, and therefore subsequent RMA2 modeling efforts employed the roughness by depth feature.

5.6   Automatically Assigned Eddy Viscosity

RMA2 also includes the feature that automatically assigns the eddy viscosity by assigning the Peclet number defined by Equation (5.2). The RMA2 manual recommends Peclet number values of approximately 20. For the model to converge, Peclet number values that ranged from 10 to 30 were used, and results from these simulations are shown in Figure 5.5. The resulting AAPD values for velocity profiles and percent differences for differences in WSE's are summarized in Table 5.10 and Table 5.11. For velocity, Table 5.10 shows that better performance was obtained at low Peclet number values at T1. For WSE, the percent differences for Δh are comparable to those from Table 5.9, for which the model use assigned eddy viscosity values and automatically assigned roughness coefficients. Qualitatively, the Peclet numbers of 15 and 20 show a good shape fit for the velocity profiles at T1 seen in Figure 5.5. At T4, there is no noticeable difference between the model runs.

Table 5.10: AAPD values between RMA2-modeled and measured velocities for varying Peclet numbers.

	Absolute % difference	
P	T1	T4
10	29.73	24.04
15	30.05	23.85
20	33.56	23.79
25	36.41	23.73
30	38.06	23.73
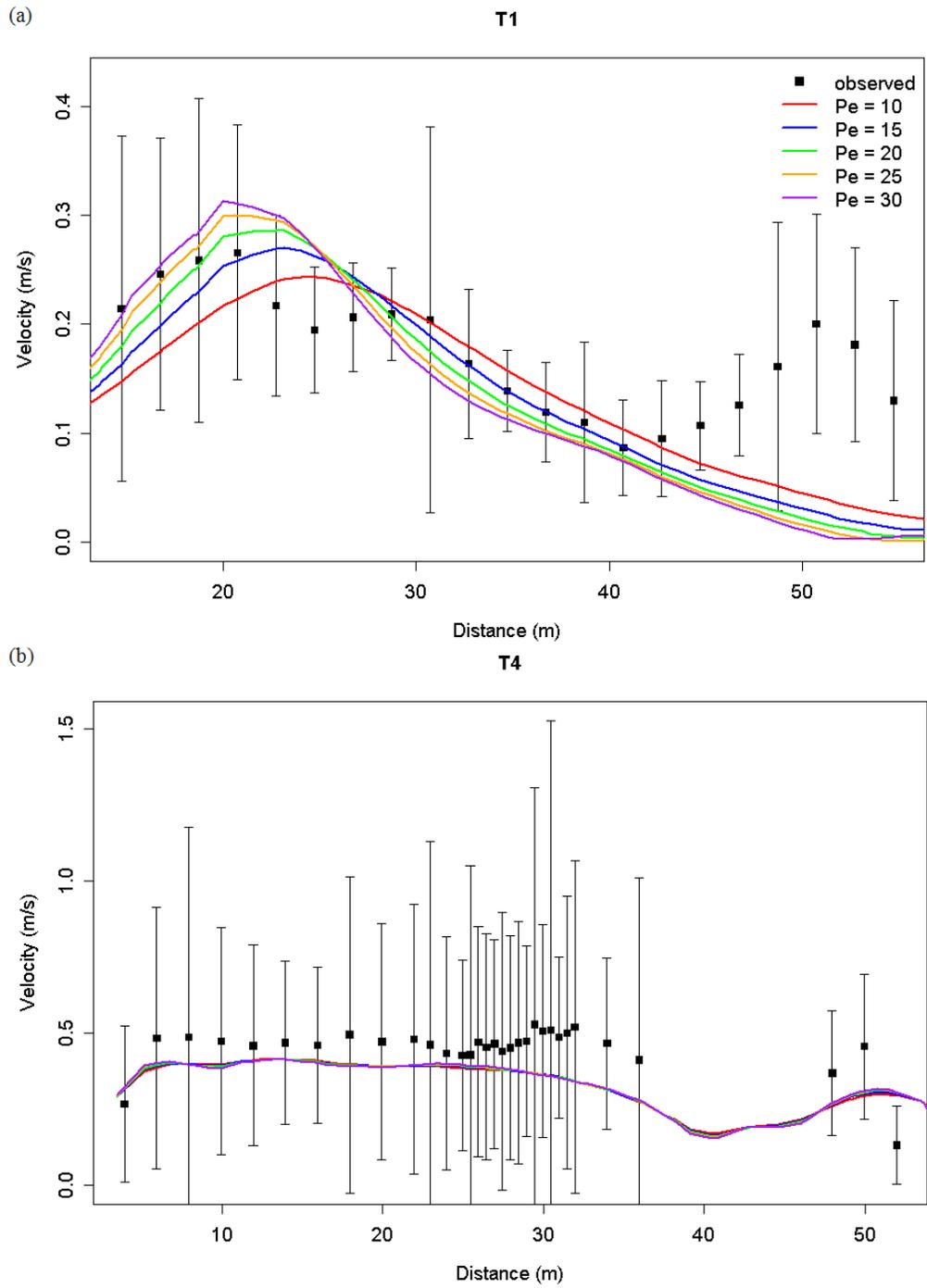
Figure 5.5: (a) Modeled and measured velocity results at T1. (b) Modeled and measured velocity results at T4. The results are for different Peclet numbers.

Table 5.11: Percent differences between modeled and measured Δh for varying Peclet numbers.

P	T3 WSE (m)	Δh (m)	% difference
10	11.544	0.064	1.59
15	11.542	0.062	-1.59
20	11.541	0.061	-3.41
25	11.540	0.060	-4.64
30	11.540	0.060	-5.40

In summary, the performance of models automatically assigning eddy viscosity values by choosing Peclet numbers is reasonable for velocity comparisons especially around the values of 15 and 20 (as recommended). In terms of the WSE differences (Δh), lower Peclet valued perform better and overall compared well with the observed. Therefore, automatic assignment of the eddy viscosity for Peclet numbers of 15 and 20 are compared with subsequent model results.

5.7   Ratios Between Channel Eddy Viscosities

Until now in this work, the assumption has been that the calculated ratio between the Merced and SJR eddy viscosities, R_{ms}, needs to be maintained. To account for ±50 percent errors associated with Equation (4.4), however, wide ranges of values can be assigned for either the SJR or Merced River channels, as shown in Table 5.3. These values are further explored here by maintaining the Merced $\varepsilon_t = 35$ Pa-s and varying the SJR eddy viscosity. Table 5.12 shows the AAPD values for different R_{ms} values employed. The AAPD does not vary substantially from case to case except for $R_{ms} = 0.15$. Table 5.13 shows the percent differences of Δh for different R_{ms} values and shows an inverse relation between R_{ms} and the percent difference of Δh values.

Table 5.12: AAPD values between RMA2-modeled and measured velocities for varying Merced and San Joaquin River eddy viscosity ratio (R_{ms}) values.

			Absolute % difference	
R_{ms}	Merced ε_t (Pa-s)	SJR ε_t (Pa-s)	T1	T4
1.31	35	26.73	34.87	23.71
1.00	35	35.00	31.87	23.82
0.71	35	49.40	31.59	24.03
0.59	35	59.27	32.46	24.13
0.48	35	72.55	32.95	24.29
0.44	35	80.18	33.16	24.38
0.39	35	88.62	33.74	24.50
0.32	35	108.48	34.48	24.74
0.15	35	240.55	37.92	25.89
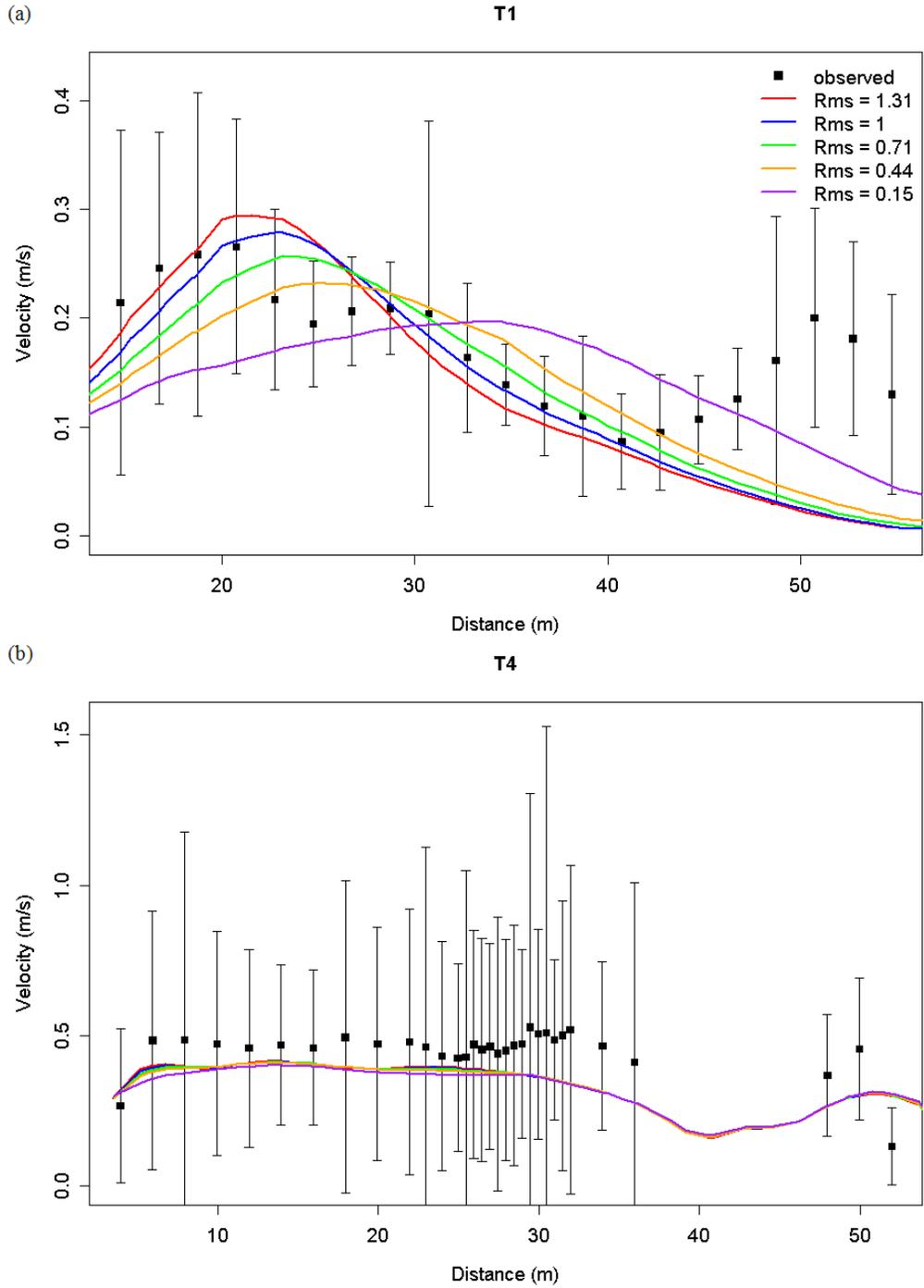
Figure 5.6: (a) Modeled and measured velocity results at T1. (b) Modeled and measured velocity results at T4. The results are for varying R_{ms} values.

Table 5.13: Percent differences between modeled and measured Δh for varying R_{ms} values.

R_{ms}	T3 WSE (m)	Δh (m)	% difference
1.31	11.541	0.061	-3.53
1.00	11.541	0.061	-2.47
0.71	11.543	0.063	-0.29
0.59	11.544	0.064	1.00
0.48	11.545	0.065	2.47
0.44	11.545	0.065	3.23
0.39	11.546	0.066	4.64
0.32	11.547	0.067	6.41
0.15	11.554	0.074	17.77

From the shape of the modeled velocity profiles, the assumption of a single eddy viscosity value for the whole model domain or $R_{ms} = 1$ appears to satisfy the velocity profile for the SJR channel as well or better than the other ratios. The specific scenario with $R_{ms} = 1$ and eddy viscosity value of 35 Pa-s compares well with the automatically assigned eddy viscosity values for Pe values of 15 or 20. The eddy viscosity value of 35 Pa-s is near the eddy viscosity value estimated at T1 (34.69 Pa-s) using Equation (4.4) with the geometry of T1 ($b = 47.58$ m, $h_{xs} = 1.29$ m), the averaged slopes between the two channels ($S = 0.000157$), and $\varepsilon_c = 0.6$. This scenario provides a distinct advantage given that this fitting parameter can be estimated rather than arbitrarily choosing from a range of Peclet number values.

5.8   Anisotropic Eddy Viscosities

Suh (2007) studied fitting different ratios between longitudinal and lateral dispersion coefficients for advection-dispersion modeling. Given the relation of dispersion coefficients to eddy viscosity values (divide by fluid density to equal dispersion coefficients), similar anisotropic ratios were explored here. Suh found low values of normalized mean square errors for R_{xy} values of 0.643 and 6.43, defining an approximate range for R_{xy} values. Simulations in this section maintains the lateral eddy viscosity, ε_y (note $\varepsilon_y = \varepsilon_t$), constant at 35 Pa-s while varying the longitudinal eddy viscosity. R_{ms} is set to 1 (based on section 6.7), meaning a single value of ε_y is applied to the whole model domain.

Table 5.14 lists the resulting AAPD values for the downstream velocity profiles and do not differ considerably for different cases. Likewise, differences between modeled upstream and downstream WSE's are all close to the measured as seen in Table 5.15. Figure 5.7 shows the modeled and observed velocity profiles for the downstream transects. From Figure 5.7 and from

the low variations between runs, it appears that employing an isotropic eddy viscosity is adequate for this system.

(a)

(b)
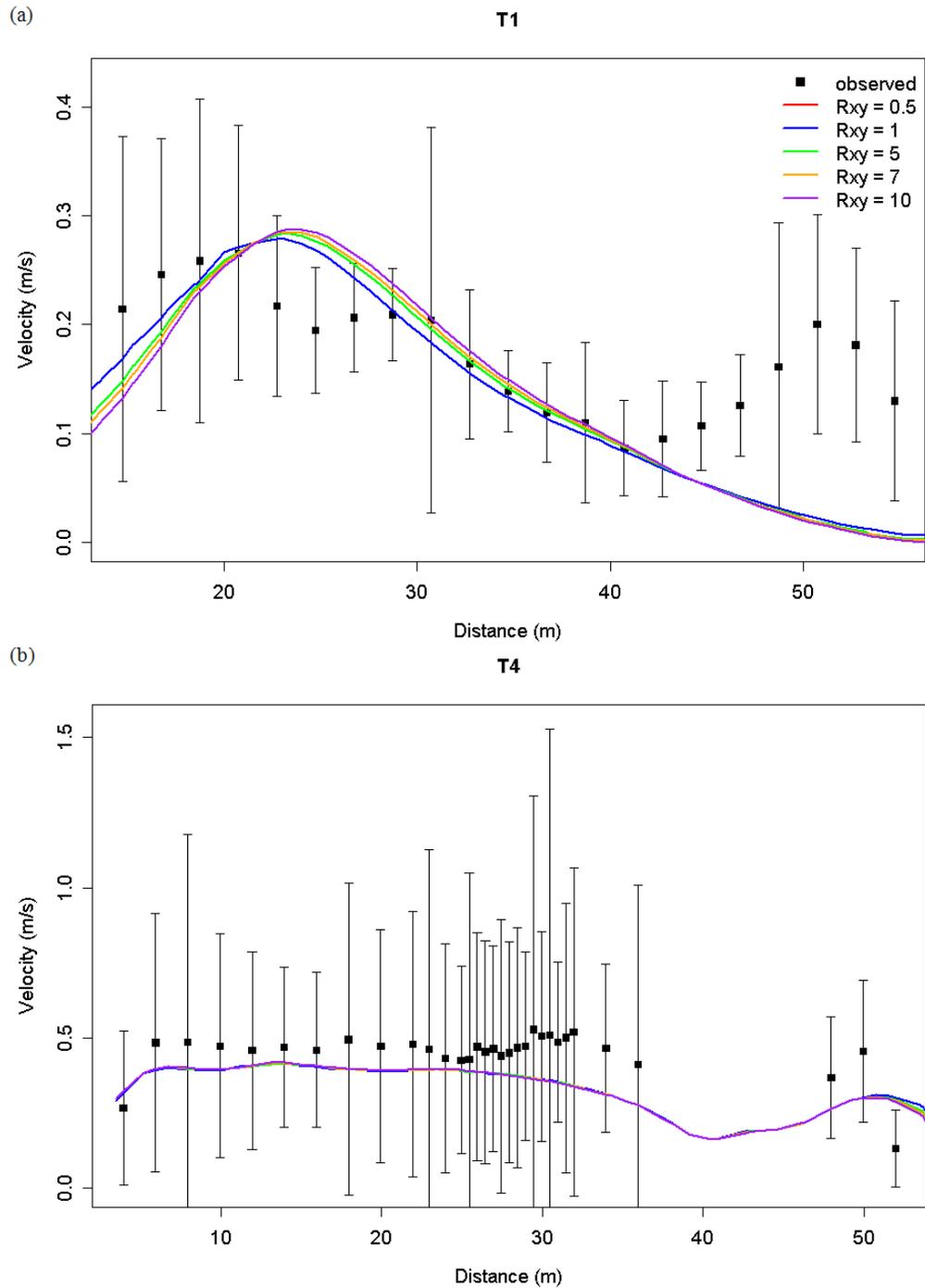
Figure 5.7: (a) Modeled and measured velocity results at T1. (b) Modeled and measured velocity results at T4. The results are for varying R_{xy} values.

Table 5.14: AAPD values between RMA2-modeled and measured velocities for varying R_{xy} values.

R_{xy}	ε_x	ε_y	Absolute % difference	
			T1	T4
0.5	17.5	35	31.87	23.82
1	35	35	31.87	23.82
3	105	35	32.01	23.78
5	175	35	32.68	23.72
7	245	35	33.60	23.66
10	350	35	34.95	23.59

Table 5.15: Percent differences between modeled and measured Δh for varying R_{xy} values.

R_{xy}	T3 WSE (m)	Δh (m)	% difference
0.5	11.541	0.061	-3.53
1	11.541	0.061	-2.47
3	11.543	0.063	-0.29
5	11.544	0.064	1.00
7	11.545	0.065	2.47
10	11.545	0.065	3.21

5.9   Summary and Conclusions

Although it was necessary to increase downstream WSE above observed values to achieve stable simulations, the resulting 2-D simulations provide a reasonable approximation of the complex dynamics of this real river confluence system. General observations that yielded a semi-quantitative calibration for the RMA2 hydrodynamic model of the SJR-Merced confluence are:

- Lower eddy viscosity values yields better velocity profile results (30 to 40 Pa-s),

- Automatically assigned roughness coefficient performs better than manual assignment exhibited the percent difference between the modeled and observed Δh (~2.7% for automatic roughness assignment compared to ~ -47% for manual assignment at low eddy viscosities),

- Manually assigned and fitted eddy viscosity value, ε_t = 35 Pa-s, performance is comparable to automatic assignment for Peclet numbers of 15 and 20. There is close agreement between the fitted value and the estimated value using the downstream geometry and flow conditions (34.69 Pa-s).

- Single eddy viscosity value for the model domain performs adequately shown qualitatively in velocity profiles in Figure 5.6 ($R_{ms} = 0.71$ also performs well but a single eddy viscosity value for the model domain is recommended for simplicity),

- Isotropic eddy viscosity performs adequately given that the AAPD values between the modeled and observed velocity profiles, percent differences between modeled and observed Δh, and qualitative velocity profile shapes did not differ greatly from the R_{xy} range from 0.5 to 10.

6 Advection-dispersion Model Results and Analysis

6.1 Overview

Given the RMA2 hydrodynamic solution in the previous model iterations, RMA4 code generates advection-dispersion constituent modeling results. As noted before, only the longitudinal and lateral dispersion coefficients, D_x and D_y, are adjusted for model fitting. Similar fitting questions to fitting the eddy viscosity for the hydrodynamic solution arise. Do the channels require distinct dispersion coefficients? Is the dispersion coefficient isotropic ($D_x = D_y$)? The last part of this work explores these questions. The first set of RMA4 model runs assumes an isotropic, global dispersion coefficient for the entire domain to gain a general range of values to be used. The last two sets of model runs apply the ratios and anisotropic conditions similar to the hydrodynamic eddy viscosity cases.

In addition to these dispersion parameter questions, one final question is asked: is it possible to obtain a reasonable approximation of constituent mixing using a one-dimensional transport model? This question is a significant one given the level of effort required to parameterize and calibrate the 2-D hydrodynamic model (Ch. 6). Hence, the last analysis compares one-dimensional analytic solutions of dispersion for blending streams seen in Equation (4.15). Like analysis of the hydrodynamic models, use of AAPD for salinity concentration profiles at T1 and T4 and qualitative judgment of the shape of the concentration profile decides model performance.

6.2 Defining Dispersion Coefficient Range

In RMA4, the dispersion coefficients are expressed in m^2/s. The lateral eddy viscosity, shown in Equations (4.4) and (4.5) and expressed in units of Pa-s, can be divided by density, as a first estimate of the lateral dispersion coefficient, D_y. For the longitudinal dispersion coefficient, D_x, Equations (4.11), (4.12) and ratios, $R_{xy} = 6.43$ and 0.643, given by Suh, provide a range of feasible values. To determine a range of dispersion coefficient test values, isotropic conditions ($D_x = D_y$) and a globally assigned dispersion coefficient (D) were applied to generate the range in Table 6.1.

Table 6.1: Calculating D_x and D_y using several methods and ratios discussed by Suh to generate the range of values for D.

Descriptor	Dispersion coefficients ID	Merced (m^2/s)	SJR (m^2/s)
Equation (4.4), $\varepsilon_c = 0.6$	D_y	0.006	0.014
$R_{xy} = 6.43$	D_x	0.039	0.088
$R_{xy} = 0.643$	D_x	0.004	0.057
Equation (4.4), ε_c from Equation (4.5)	D_y	0.010	0.015
$R_{xy} = 6.43$	D_x	0.066	0.094
$R_{xy} = 0.643$	D_x	0.007	0.009
Equation (4.11)	D_x	0.057	0.133
$R_{xy} = 6.43$	D_y	0.009	0.021
$R_{xy} = 0.643$	D_y	0.089	0.207
Equation (4.12)	D_x	0.032	0.062
$R_{xy} = 6.43$	D_y	0.005	0.010
$R_{xy} = 0.643$	D_y	0.050	0.097
	Minimum D (m^2/s)		0.004
	Maximum D (m^2/s)		0.207

Model runs with D values ranging from 0.003 to 0.25 m^2/s were performed. Figure 6.1 shows the resulting concentration profiles at transects T1 and T4. Qualitatively, values of D from 0.02 to 0.1 m^2/s result in promising degree of mixing for T1. For T4, lower values of D, 0.003 to 0.02 m^2/s, capture the mixing gradient better. However, the sharp mixing gradient is shifted along the lateral axis. Table 6.2 shows the AAPD for the concentration profiles at the downstream transects.

Table 6.2: AAPD values between RMA4-modeled and measured concentrations for varying D values (m^2/s).

	Absolute % difference	
D (m^2/s)	T1	T4
0.003	40.85	76.37
0.02	31.08	67.20
0.05	16.52	62.78
0.1	6.25	60.37
0.15	11.25	59.62
0.25	19.06	59.34

T1

T4
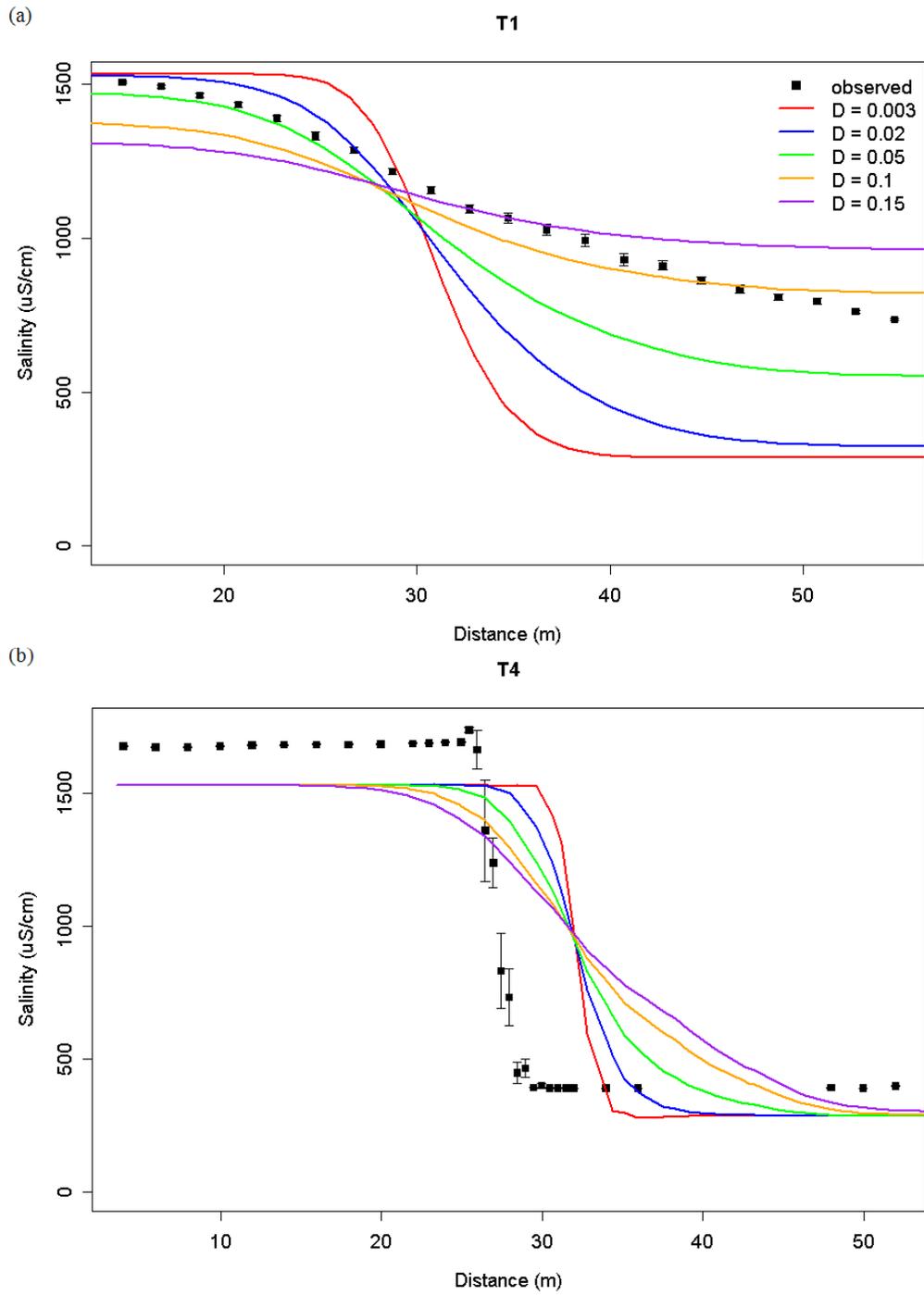
Figure 6.1: (a) Modeled and measured concentration results at T1. (b) Modeled and measured concentration results at T4. The results are for varying D values (m^2/s).

For T1, $D = 0.1$ m^2/s clearly result in the lowest AAPD between the modeled and measured concentration profiles. For T4, $D = 0.25$ m^2/s has the lowest AAPD, but Figure 6.1 shows the higher dispersion values increase mixing not seen in the measured concentration profile. In summary, the dispersion value domain greater than 0.003 m^2/s and less than 0.15 m^2/s performed adequately and this range is used for the subsequent sections.

6.3   Ratios Between Channel Dispersion Coefficients

Similar to analyzing the ratios between channel eddy viscosities, the same ratios, R_{ms}, from Table 5.12 are used to examine the effects of varying R_{ms} values. The dispersion coefficient for the Merced River channel is held at 0.02 m^2/s while the dispersion values of the SJR channel changes. Also, the following model runs assume isotropic conditions for the dispersion coefficients. Table 6.3 shows the AAPD's for the downstream concentration profiles. As R_{ms} decreases, the AAPD decreases for T1. For T4, the AAPD's remain relatively constant. Figure 6.2 shows the modeled concentration profiles.

Table 6.3: AAPD values between RMA4-modeled and measured concentration profiles for varying R_{ms} values.

			Absolute % difference	
R_{ms}	Merced D (m^2/s)	SJR D (m^2/s)	T1	T4
1.31	0.02	0.015	32.88	67.12
1.00	0.02	0.020	31.08	67.20
0.71	0.02	0.028	28.39	67.32
0.59	0.02	0.034	26.75	67.40
0.48	0.02	0.041	24.86	67.49
0.44	0.02	0.046	23.94	67.54
0.39	0.02	0.051	23.15	67.59
0.32	0.02	0.062	21.46	67.69
0.15	0.02	0.137	15.02	68.15

For T4, there is no noticeable difference between model runs. For T1, $R_{ms} = 0.15$ showed a large departure from the other model runs, where the sharp drop-off appears to occur along the shear boundary between the two channels. As R_{ms} decreases, or as the dispersion coefficient of the SJR increases, mixing increases with noticeable effects occurring when $R_{ms} > 0.44$ which is the ratio estimated from Equation (4.4). In summary, R_{ms} variation does not appear to affect the model $R_{ms} > 0.44$ which may be due to increase the in SJR dispersion coefficient. Though the lower R_{ms} values do yield better AAPD's, the discontinuity between channels seems unnatural. Therefore, global value of dispersion coefficients ($R_{ms} = 1$) is reasonable.
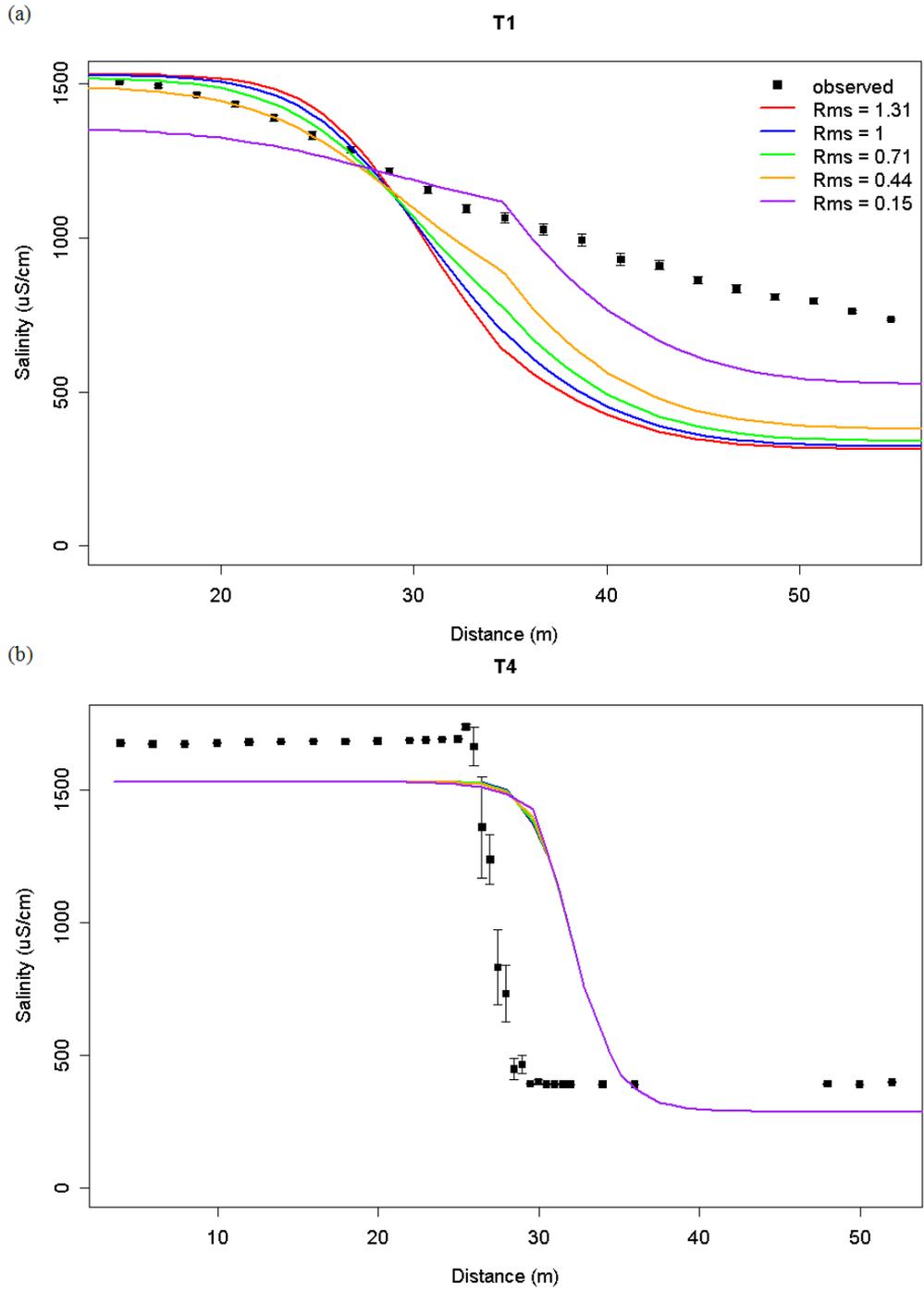
Figure 6.2: (a) Modeled and measured concentration results at T1. (b) Modeled and measured concentration results at T4. The results are for varying R_{ms} values.

6.4   Anisotropic Dispersion Coefficients

Given the previous model fittings, two R_{ms} cases, $R_{ms} = 1$ and $R_{ms} = 0.44$, are examined further in a study of potential merits of parameterizing dispersion as anisotropic. Similar ratios, R_{xy}, between eddy viscosities shown in Table 5.14 are used for setting longitudinal and lateral dispersion coefficients. For the first set of model runs, the dispersion coefficients are the same for both channels. To ensure the anisotropic dispersion coefficients fall within the range described in Section 6.2 ($0.15 \text{ m}^2/\text{s} > D > 0.003 \text{ m}^2/\text{s}$), the lateral dispersion coefficient, D_y, is set to $0.01 \text{ m}^2/\text{s}$ while changing the longitudinal dispersion coefficient, D_x. Table 6.4 shows the AAPD statistics for the model runs and clearly demonstrates that as R_{xy} increases, the AAPD decreases substantially at T1. When $R_{xy} = 10$, the model has the lowest AAPD. Consistent with other results, there is not much variation in model performance at T4. Similar conclusions are drawn from qualitatively looking at the model runs in Figure 6.3. R_{xy} values greater than 7 produced the best fits.

Table 6.4: AAPD values between modeled and measured concentration profiles for varying R_{xy} values and $R_{ms} = 1$.

			Absolute % difference	
R_{xy}	D_x (m^2/s)	D_y (m^2/s)	T1	T4
0.5	0.005	0.01	38.59	71.40
1	0.01	0.01	36.55	71.33
3	0.03	0.01	28.06	71.05
5	0.05	0.01	19.94	70.78
7	0.07	0.01	14.15	70.51
10	0.1	0.01	7.78	70.12

For $R_{ms} = 0.44$, another set of model runs further tested the anisotropic conditions. To stay within the appropriate range of dispersion coefficient values, the Merced and SJR channel lateral dispersion coefficients are set to $0.006 \text{ m}^2/\text{s}$ and $0.014 \text{ m}^2/\text{s}$ respectively. The same ratios as before are examined and the AAPD values for the concentration profiles are summarized in Table 6.5. Again, $R_{xy} = 10$ performed the best at T1, but not as well as the similar case for $R_{ms} = 1$. Figure 6.4 shows the concentration profiles.
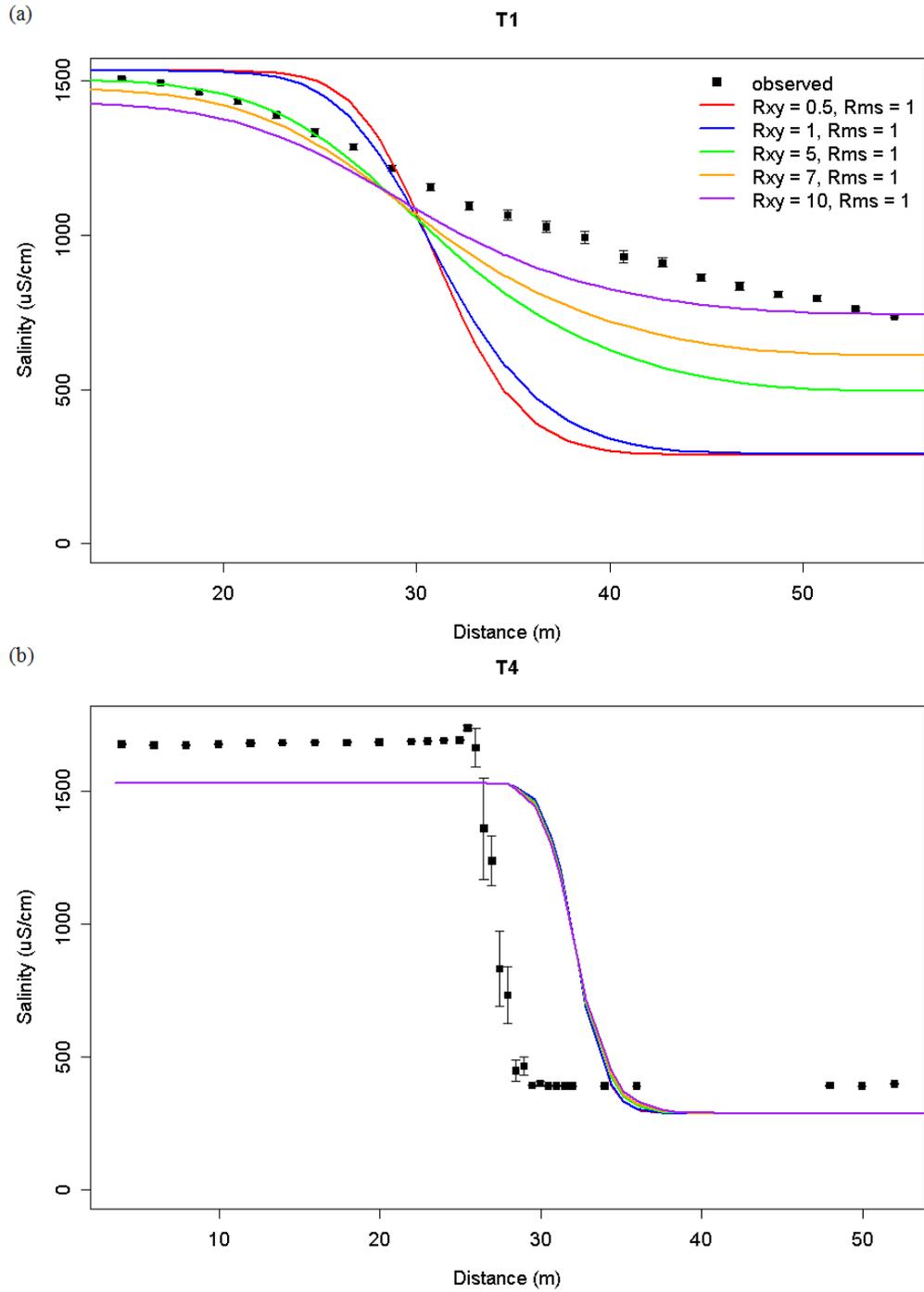
Figure 6.3: (a) Modeled and measured concentration results at T1. (b) Modeled and measured concentration results at T4. The results are for varying R_{xy} values and $R_{ms} = 1$.
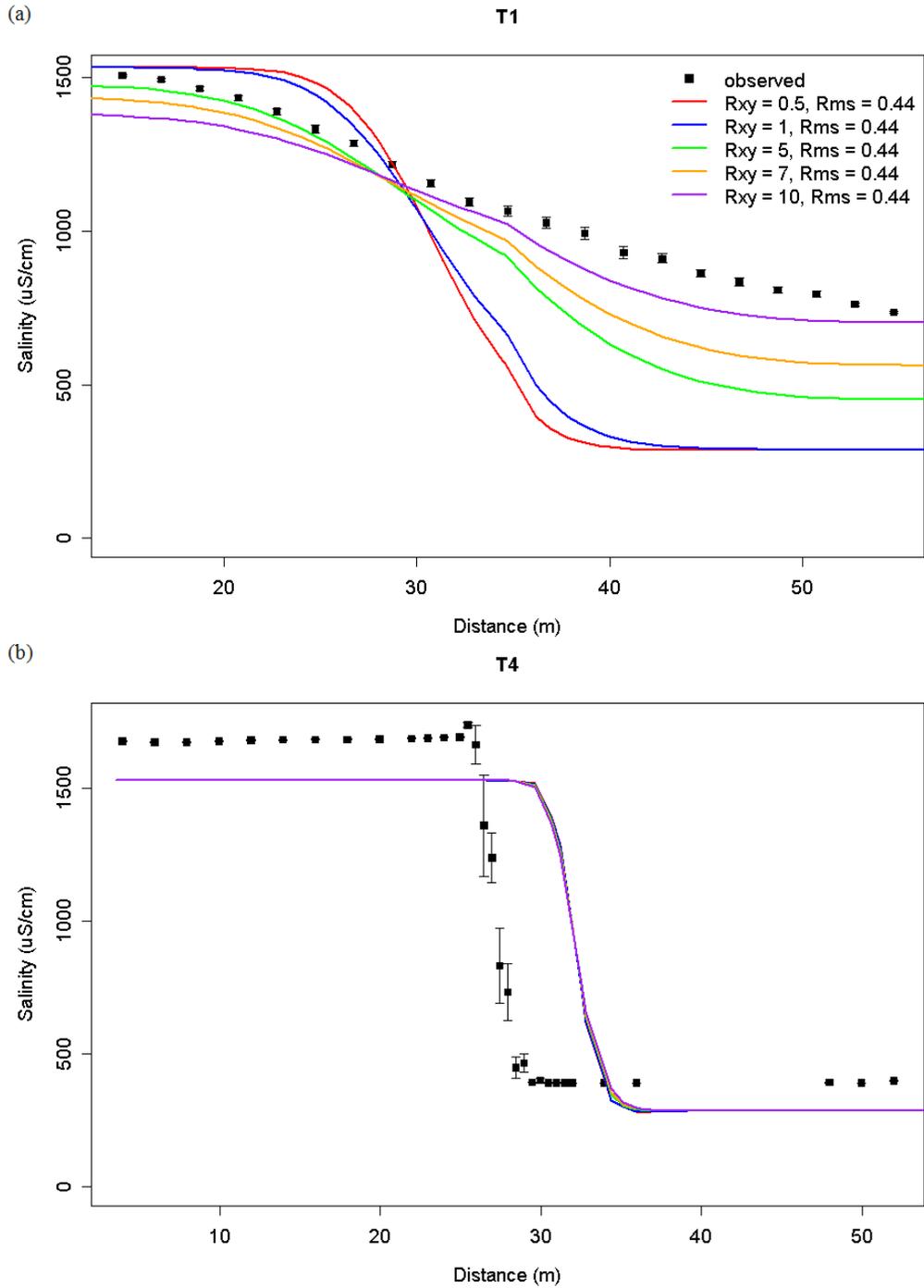
Figure 6.4: (a) Modeled and measured concentration results at T1. (b) Modeled and measured concentration results at T4. The results are for varying R_{xy} values and $R_{ms} = 0.44$.

Table 6.5: AAPD values between RMA4-modeled and measured concentration profiles for varying R_{xy} values and $R_{ms} = 0.44$.

	Merced		SJR		Absolute % difference	
R_{xy}	D_x (m^2/s)	D_y (m^2/s)	D_x (m^2/s)	D_y (m^2/s)	T1	T4
0.5	0.003	0.006	0.007	0.014	37.78	73.58
1	0.006	0.006	0.014	0.014	35.64	73.54
3	0.018	0.006	0.041	0.014	27.38	73.35
5	0.03	0.006	0.069	0.014	20.32	73.16
7	0.042	0.006	0.096	0.014	14.72	72.97
10	0.06	0.006	0.137	0.014	8.03	72.71

In summary, the results for defining two separate sets of dispersion coefficients for the two channels did not improve results. The best estimate lateral dispersion coefficient, $D_y = 0.01$ m^2/s, did fall within the range of values estimated from Equation (4.4) for the Merced (0.006 m^2/s) and SJR channels (0.0146 m^2/s). To approach the appropriate ratio and longitudinal dispersion coefficient value of $D_x = 0.1$ m^2/s, Equation (4.11) for the SJR side had the best estimate of $D_x = 0.133$ m^2/s. Therefore, in general it appears that dispersion characteristics estimated based on the SJR channel could be applied to the whole model domain rather than attempting to separate the model domain into two channels.

6.5   Comparisons with Analytic Solutions

A lingering question asks if the remaining finite-element solution performs better than the analytic solution for the ideal case expressed in Equation (4.15), referred to here as the Fischer model. Since both channels have salinity concentrations, c_0 is adjusted as difference between the more saline SJR and less saline Merced River. This adjusted difference is then added after the Equation (4.15) is solved. Additionally, although the equation's assumptions of constant width, rectangular channel shape, and equal discharge between channels are unrealistic, the lateral dispersion coefficient, D_y, can similarly be adjusted to obtain a model fit. Only data from the furthest downstream transect, T1, are included in this analysis due to the assumption of constant channel width. Given that T1 has different cross-sectional area and wetted perimeter than the upstream transects, a separate shear velocity is calculated from Equation (4.6), which creates a separate estimate for the lateral dispersion coefficient. The slope needed in the shear velocity estimate was taken as the averaged slopes of the Merced and SJR channels found in Table 5.1 resulting in a lateral dispersion coefficient value of 0.034 m^2/s. Three mixing cases with differing lateral dispersion coefficients are analyzed for the Fischer model. The three lateral dispersion

values are the estimates for the upstream Merced River at T2 ($D_y = 0.006$ m^2/s), upstream SJR at T3 ($D_y = 0.014$ m^2/s), and the downstream SJR at T1 estimated above as 0.034 m^2/s. The different mixing cases are compared with the best-fitting RMA4 model results employing anisotropic mixing using the dispersion coefficients estimated from the SJR channel ($D_x = 0.133$ m^2/s and $D_y = 0.014$ m^2/s). A plot of the concentration profiles for these cases is shown in Figure 6.5.
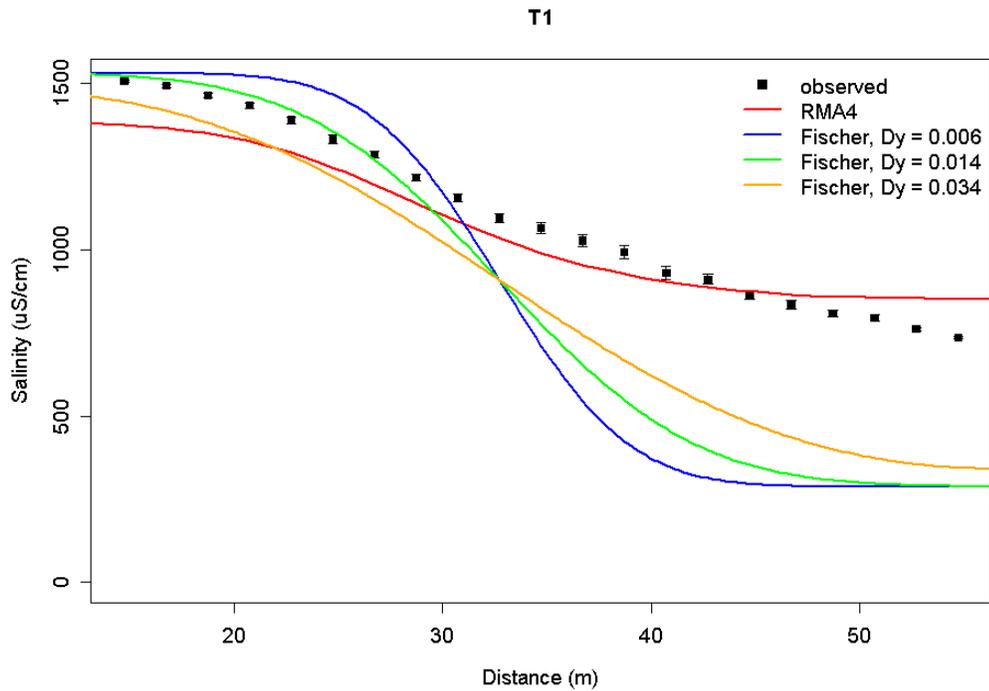
Figure 6.5: Modeled and measured concentration results at T1. The RMA4 results are for when $D_x = 0.133$ m^2/s and $D_y = 0.014$ m^2/s.

Generally, the analytic solution estimates did not reproduce the mixing behavior nearly as well as the 2-D model, a result which is reasonable given the relative simplicity of the Fischer model and the gross assumptions required to apply it to a real river channel. Quantitative results for the AAPD's are shown in Table 6.6 verifying that RMA4 is more accurate.

Table 6.6: AAPD values between RMA4-modeled and measured concentration profiles for the analytic solution and RMA4 whose anisotropic values are from theory.

Model type	D_x (m^2/s)	D_y (m^2/s)	Absolute % difference at T1
Fischer	-	0.006	34.06
Fischer	-	0.014	29.79
Fischer	-	0.034	26.35
RMA4	0.133	0.014	6.87

The RMA4 results using the theoretically estimated dispersion coefficients have a lower AAPD than the fitted values shown in Table 6.5. The Fischer model AAPD values indicate better performance as D_y increases. The slope from $D_y = 0.034$ m^2/s qualitatively showed the best slope fit from Figure 6.5. Curiously, the associated eddy viscosity, calculated by multiplying D_y by the density of water ($\varepsilon_t = 34$ Pa-s), is close to the fitted case of 35 m^2/s. Ultimately, the finite-element model showed better performance qualitatively and quantitatively relative to the analytic solution.

7 Conclusions

This work presented two-dimensional hydrodynamic and advection-dispersion modeling methods and the data collection necessary to parameterize the models at the Merced River-San Joaquin River confluence. The repetitive delivery of an acoustic velocimeter and Hydrolab multiprobe by the NIMSRD generated many velocity and concentration fields for transects. The estimated flows from the velocity fields compared well with a nearby gaging station downstream of the confluence. Given the good performance of the system for mobile data collection and the mobility of the infrastructure, the system was found to be capable of providing good upstream flow field estimates necessary for driving depth-averaged, finite-element hydrodynamic models. However, to reduce error in depth-averaged velocities seen in the results in Chapter 5, the Sontek ADV should be set to longer averaging periods.

Standard methods applying surveying and echosounder equipment defined the river bed topography. This work applied a multi-level B-spline approximation method to resolve discontinuities between the two data sets and smooth the data. Smoother bathymetry data help assure numerical model convergence. To help satisfy model boundary conditions for both defining the river model boundary and downstream water surface elevations, one outcome of this work is a recommendation that river topography data and water surface along all banks be recorded be collected as close to the time when flow data is collected.

For hydrodynamic and advection-dispersion modeling, this work focused on 2-dimensional, finite-element models rather than 3-dimensional models due to the data requirements and computational power needed to support the latter, and to examine whether the 2-D models presented reasonable were capable of providing reasonable approximations of the complex flow regime in a confluence zone. Overall, the models performed well with the following summarized observations from model optimization:

1. Closer node-spacing did not drastically improve model results.
2. FESWMS code performed better than RMA2 and allowed for hydrodynamic model convergence at downstream WSE close to the observed. The RMA2 code failed to converge consistently unless the downstream WSE boundary condition was increased beyond the measured levels.
3. In RMA2, automatically assigning the roughness coefficient by depth performed better than choosing a global Manning roughness coefficient.

4. In RMA2, a low, isotropic eddy viscosity value performed the best in terms of matching the downstream (T1) cross-sectional velocity field. Fitted eddy viscosity values were frequently found to be greater than values estimated from upstream transect parameters but close to the downstream transect parameters.

5. For advection-dispersion modeling computed with the RMA4 code, the results showed good agreement with measured specific conductance (SC) profiles using anisotropic dispersion coefficients, which were best defined by dispersion coefficients attributed with the SJR flow and channel geometry (the river with substantially greater flow).

6. A simplified analytic solution mixing was capable of only roughly approximating the observed behavior, suggesting that 2-D modeling is necessary in cases where accurate knowledge of chemical gradients is necessary.

7. For both RMA2 and RMA4, a single domain with similar eddy viscosity and dispersion coefficients resulted in better agreement with observations than defining two separate channels.

Appendix A   Code for Calculating Flow

This appendix section includes major code sections written in R for handling and processing data needed for flow estimates. Not included are some of the globally-defined variables, input, and output code unless embedded in functions that also contain processing code. Simple functions not pertinent to process of flow calculations, such as identifier functions, are also no included.

A.1   Main Function

The main function calls all other functions needed for data input, classification, processing, and periodic output. Files associated with this code involve motor location files (2 files), sensor data (2 files), and filter with cross-sectional data files (4 files). Additional embedded operations not included in separate independent functions are the transform of velocity vectors from ENU coordinates to longitudinal and transverse dimensions and shifting of transverse horizontal distances to begin at the left bank looking downstream and zero defined at the WSE at the left bank.

```r
# ===== main function, goes thru all the steps and calls other functions =====
main_redo = function(index){
  # ===== indexing which file is which =====
  file_indices = (index-1) * 4 + (1:4)
  files_full = all_files[file_indices]

  # === file check ===
  file_check = mapply(fileCheck, 1:length(files_full), MoreArgs = list(files_full),
   SIMPLIFY = F)
  filecheck_dat = data.frame(do.call(rbind, file_check))
  filecheck_labels = c("trans_id", "run_id")
  colnames(filecheck_dat) = filecheck_labels

  # if the folder and immediate folder above are the same, then the files are ok
  if(length(unique(filecheck_dat$trans_id) == 1) &
   length(unique(filecheck_dat$run_id) == 1)){
    tran_id = as.character(unique(filecheck_dat$trans_id))
    run_id = as.character(unique(filecheck_dat$run_id))
  } else{
    stop("are the files in the same folder?")
  }

  # back to id'ing files
  motorPos_id = grep("allposition", files_full)
  hydrolab_id = grep("hydrolab", files_full)
  sontek_id = grep("sontek", files_full)

  if(length(motorPos_id) == 0 | length(motorPos_id) > 1)
    stop("allposition.txt error")
  if(length(hydrolab_id) == 0 | length(hydrolab_id) > 1)
    stop("hydrolab file error")
  if(length(sontek_id) == 0 | length(sontek_id) > 1 )
    stop("sontek file error")
```

```r
file_positions = c(motorPos_id, hydrolab_id, sontek_id)

commandPos_id = which((1:4 %in% file_positions) == F)

if(length(commandPos_id) == 0 | length(commandPos_id) > 1)
  print("command position file error")

# ===== reading in files, classifying data types, small filtering =====

# === reading in motor position log file
motorPos_dat = read.table(files_full[motorPos_id], header = F, as.is = T,
 comment.char = "", fill = T)

#filtering out lines that say System_Start_UP
string_MotId = grep("System_Start_UP", motorPos_dat[, 2])
motorPos_dat = motorPos_dat[!(1:nrow(motorPos_dat) %in% string_MotId), ]

#changing column data types
motorPos_timeStr = gsub(":", ".", motorPos_dat[, 1])
options(digits.secs = 2)
motorPos_time = strptime(motorPos_timeStr, "%F_%H.%M.%OS")
motorPos_dat = cbind(motorPos_time, motorPos_dat[, 2:ncol(motorPos_dat)])
motorPos_dat[, 2] = as.numeric(motorPos_dat[, 2])
motorPos_dat[, 3] = as.numeric(motorPos_dat[, 3])

# === reading in command log file
commandPos_dat = read.table(files_full[commandPos_id], header = F, as.is = T,
 comment.char = "", fill = T)

#filtering lines denoting beginning and end of run
string_commandId1 = grep("BEGIN", commandPos_dat[, 2])
string_commandId2 = grep("END", commandPos_dat[, 2])
string_commandId3 = c(string_commandId1, string_commandId2)
commandPos_dat = commandPos_dat[!(1:nrow(commandPos_dat) %in% string_commandId3),
 ]

#changing column data types
commandPos_timeStr = gsub(":", ".", commandPos_dat[, 1])
commandPos_time = strptime(commandPos_timeStr, "%F_%H.%M.%OS")
commandPos_dat = cbind(commandPos_time, commandPos_dat[, 2:ncol(commandPos_dat)])
commandPos_dat[, 2] = as.numeric(commandPos_dat[, 2])
commandPos_dat[, 3] = as.numeric(commandPos_dat[, 3])

# === defining dwell state ===
dwell_id = which(commandPos_dat[, 4] == "DWELL")
move_id = dwell_id + 1

#making sure dwell is longer than 5 seconds
fullDwell_id = which(commandPos_dat[move_id, 1] - commandPos_dat[dwell_id, 1] > 5)
dwell_id = dwell_id[fullDwell_id]
move_id = dwell_id + 1

#checking to make sure last dwell has an end
if(max(move_id) > nrow(commandPos_dat))
  move_id[length(move_id)] = dwell_id[length(dwell_id)]

#taking smaller slice of allposition.txt or motor position file
motor_sliceDat = motorPos_dat[motorPos_dat[, 1] > (commandPos_dat[dwell_id[1], 1]
 - 5) & motorPos_dat[, 1] < (commandPos_dat[move_id[length(move_id)], 1] + 5), ]

#makes two data frames with start, end times, and position
comm_dat = data.frame(start_t = commandPos_dat[dwell_id, 1],
```

```r
                          end_t = commandPos_dat[move_id, 1],
                          x = commandPos_dat[dwell_id, 2],
                          y = commandPos_dat[dwell_id, 3])

if(nrow(motor_sliceDat) != 0){
  dwell_dat = mapply(find_times, 1:length(dwell_id), MoreArgs = list(comm_dat,
  motor_sliceDat),SIMPLIFY = F)

  #collapsing list into data frame
  dwell_dat = data.frame(do.call(rbind, dwell_dat))
  dwell_labels = c("mot_startTime", "mot_endTime", "com_X", "com_Y")
} else{
  #if no smaller slice of allposition.txt found
  dwell_dat = comm_dat
  dwell_labels = c("com_startTime", "com_endTime", "com_X", "com_Y")
}

colnames(dwell_dat) = dwell_labels
dwell_units = c("%Y-%m-%d_%H:%M:%OS", "%Y-%m-%d_%H:%M:%OS", "m", "m")
dwell_headers = rbind(dwell_labels, dwell_units)

# ===== read in hydrolab file =====
hydro_table = read.table(files_full[hydrolab_id], header = F, as.is = T,
 comment.char = "", fill = T)
colnames(hydro_table) = HYDROLAB_NEW[, 1]
hydro_timeStr = gsub(":", ".", hydro_table[, 1])
hydro_time = strptime(hydro_timeStr, "%Y-%m-%d_%H.%M.%OS")
hydro_EC = hydro_table[, "SpCond"]
hydro_dat = data.frame(hydro_time, hydro_EC)

hydro_aveDat = mapply(hydro_ave, 1:nrow(dwell_dat), MoreArgs = list(dwell_dat,
 hydro_dat), SIMPLIFY = F)
hydro_aveDat = data.frame(do.call(rbind, hydro_aveDat))

hydro_aveDatLabels = c("hyd_startTime", "hyd_endTime", "mean_SC")

colnames(hydro_aveDat) = hydro_aveDatLabels
hydro_aveDatUnits = c("%Y-%m-%d_%H:%M:%OS", "%Y-%m-%d_%H:%M:%OS", "uS/cm")
hydro_aveDatHeaders = rbind(hydro_aveDatLabels, hydro_aveDatUnits)

# ===== read in sontek file =====
sontek_table = read.table(files_full[sontek_id], header = F, as.is = T,
 comment.char = "", fill = T)
colnames(sontek_table) = SONTEK_STREAMED[, 1]
sontek_timeStr = gsub(":", ".", sontek_table[, 1])
sontek_time = strptime(sontek_timeStr, "%Y-%m-%d_%H.%M.%OS")
sontek_table = cbind(sontek_time, sontek_table[, 2:ncol(sontek_table)])

#filter out when heading, pitch, and roll = 0
bad_id = which(sontek_table[, "Pitch"] == 0 &
               sontek_table[, "Roll"] == 0 &
               sontek_table[, "Heading"] == 0)
sontek_filt = sontek_table[!(1:nrow(sontek_table) %in% bad_id), ]
son_intTimeStr = paste(sontek_filt[, "Year"],
                       sontek_filt[, "Month"],
                       sontek_filt[, "Day"],
                       sontek_filt[, "Hour"],
                       sontek_filt[, "Minute"],
                       sontek_filt[, "Second"],
                       sep = "_")
son_intTime = strptime(son_intTimeStr, "%Y_%m_%d_%H_%M_%S")

#there is a large delay between internal sontek and laptop
```

```r
#but the internal sontek clock is sync'ed before deployment
#hence a large processing time can be added to transmission time
sontek_dat = data.frame(sontek_filt[, 1], son_intTime, sontek_filt[,
 8:ncol(sontek_filt)])
colnames(sontek_dat) = c("son_time", "son_intTime",
 SONTEK_STREAMED[8:nrow(SONTEK_STREAMED), 1])

sontek_aveDat = mapply(sontek_ave, 1:nrow(dwell_dat), MoreArgs = list(dwell_dat,
 sontek_dat), SIMPLIFY = F)
sontek_aveDat = data.frame(do.call(rbind, sontek_aveDat))
sontek_aveDatLabels = c("son_startTime", "son_endTime", "son_intStartTime",
 "son_intEndTime", "numPts", "velE", "velN", "velU", "rangeToBound", "velMag")
sontek_aveDatUnits = c("%Y-%m-%d_%H:%M:%OS", "%Y-%m-%d_%H:%M:%OS", "%Y-%m-
 %d_%H:%M:%S", "%Y-%m-%d_%H:%M:%S", "units", "cm/s", "cm/s", "cm/s", "m", "cm/s")
colnames(sontek_aveDat) = sontek_aveDatLabels
sontek_aveDatHeaders = rbind(sontek_aveDatLabels, sontek_aveDatUnits)

# === accounting for declination ===
# radians, degrees from noaa
shift_rad = 14.25 * pi/180

# essentially counter-clockwise shift
velE_decl = sontek_aveDat$velN * sin(shift_rad) + sontek_aveDat$velE *
 cos(shift_rad)
velN_decl = sontek_aveDat$velN * cos(shift_rad) - sontek_aveDat$velE *
 sin(shift_rad)

# === making velocities longitudinal
T1_angle = -1.594128997
T2_angle = 139.6636998
T3_angle = 21.78905762
T4_angle = 62.01801994

T1_rad = abs(T1_angle*pi/180)
T2_rad = abs(T2_angle*pi/180)
T3_rad = abs(T3_angle*pi/180)
T4_rad = abs(T4_angle*pi/180)

# if its transect 1 make a clockwise move
if(tran_id == "tran1"){
  vel_long = velN_decl * cos(T1_rad) + velE_decl * sin(T1_rad)
  vel_tran = velE_decl * cos(T1_rad) - velN_decl * sin(T1_rad)
} else if(tran_id == "tran2"){
# counterclockwise for transect 2
  vel_long = velN_decl * cos(T2_rad) - velE_decl * sin(T2_rad)
  vel_tran = velE_decl * cos(T2_rad) + velN_decl * sin(T2_rad)
} else if(tran_id == "tran3"){
#counterclockwise for transect 3
  vel_long = velN_decl * cos(T3_rad) - velE_decl * sin(T3_rad)
  vel_tran = velE_decl * cos(T3_rad) + velN_decl * sin(T3_rad)
} else if(tran_id == "tran4"){
#counterclockwise for transect 3
  vel_long = velN_decl * cos(T4_rad) - velE_decl * sin(T4_rad)
  vel_tran = velE_decl * cos(T4_rad) + velN_decl * sin(T4_rad)
}

newVel_dat = cbind(velE_decl, velN_decl, vel_long, vel_tran)
newVel_names = c("velE_decl", "velN_decl", "vel_long", "vel_tran")
colnames(newVel_dat) = newVel_names
newVel_units = c("cm/s", "cm/s", "cm/s", "cm/s")

newVel_headers = rbind(newVel_names, newVel_units)
```

```r
# === combine all data ===
newDat01 = cbind(dwell_dat, hydro_aveDat, sontek_aveDat, newVel_dat)
newDat01_headers = cbind(dwell_headers,
                         hydro_aveDatHeaders,
                         sontek_aveDatHeaders,
                         newVel_headers)
time_id = grep("Time", colnames(newDat01))
timeStr = mapply(timeToStr, 1:length(time_id), MoreArgs = list(time_id, newDat01))
newDat01[, time_id] = timeStr

# ===== filtering boundary =====

# adds distance from left endpt and elevation
T1_WSE = 11.379
# subjective b/c it wasn't measured, the T3 WSE was chosen
T2_WSE = 11.442
T3_WSE = 11.442
T4_WSE = 11.393

T1_HOFFSET = 20.335
T2_HOFFSET = 7.46
T3_HOFFSET = 10.1
T4_HOFFSET = 4.948

# cm the sample volume is under the surface
sontek_depth = 0.20

transect_folder = "C:/Documents and Settings/hank/Desktop/201006 data thesis
 organized/xs calcs/transect_bath/"

# provide proper adjustment and assign bottom file
if(tran_id == "tran1"){
  h_dist = 75.05604888- dwell_dat[, 3] - T1_HOFFSET
  v_elev = T1_WSE + dwell_dat[, 4] - sontek_depth
  bound_file = paste(transect_folder, "T1_bottomInterp.csv", sep = "")
  wse = T1_WSE
} else if(tran_id == "tran2"){
  h_dist = dwell_dat[, 3] + T2_HOFFSET
  v_elev = T2_WSE + dwell_dat[, 4] - sontek_depth
  bound_file = paste(transect_folder, "T2_bottomInterp.csv", sep = "")
  wse = T2_WSE
} else if(tran_id == "tran3"){
  h_dist = dwell_dat[, 3] + T3_HOFFSET
  v_elev = T3_WSE + dwell_dat[, 4] - sontek_depth
  bound_file = paste(transect_folder, "T3_bottomInterp.csv", sep = "")
  wse = T3_WSE
} else if(tran_id == "tran4"){
  h_dist = dwell_dat[, 3] + T4_HOFFSET
  v_elev = T4_WSE + dwell_dat[, 4] - sontek_depth
  bound_file = paste(transect_folder, "T4_bottomInterp.csv", sep = "")
  wse = T4_WSE
}

newDat02 = cbind(newDat01, h_dist, v_elev)
newDat02_labels = c("dist_LeftPt", "elevation")
newDat02_units = c("m", "m")

colnames(newDat02) = c(newDat01_headers[1, ], newDat02_labels)
newDat02_headers = rbind(newDat02_labels, newDat02_units)
newDat02_headers = cbind(newDat01_headers, newDat02_headers)

bound_dat = read.table(bound_file, header = T, sep = ",", as.is = T)
```

```r
    visited_xPts = unique(newDat02[, "dist_LeftPt"])

    # returns distance, elevation, index of points closest to the x-position passed to
     it
    bound_datPts = mapply(findBound, 1:length(visited_xPts), MoreArgs =
     list(bound_dat, visited_xPts),SIMPLIFY = F)
    bound_datPts = data.frame(do.call(rbind, bound_datPts))

    # finally to the filtering
    newDat03 = mapply(boundDat, 1:nrow(bound_datPts),
                      MoreArgs = list(bound_datPts, newDat02, visited_xPts, wse),
                      SIMPLIFY = F)
    newDat03 = data.frame(do.call(rbind, newDat03))
    newDat03 = newDat03[newDat03[, 1] != -1, ]

    newDat03_labels = c("velLongWeighted", "pts_deleted")
    newDat03_units = c("cm/s", "units")
    newDat03_headers = rbind(newDat03_labels, newDat03_units)
    newDat03_headers = cbind(newDat02_headers, newDat03_headers)
    colnames(newDat03) = newDat03_headers[1, ]

    print(paste(tran_id, run_id))

    # === depth averaging ===
    depAveNames = c("dist_LeftPt", "mean_SC", "velMag", "vel_long", "velLongWeighted")
    depAve = aggregate(newDat03[, depAveNames], list(newDat03[, "dist_LeftPt"]), mean)

    newDat04 = depAve[, 2:ncol(depAve)]
    newDat04_units = c("m", "uS/cm", "cm/s", "cm/s", "cm/s")
    newDat04_headers = rbind(depAveNames, newDat04_units)


    # === outputting this before boundary filter ===
    # no boundary output
#   output_fn1 = paste(tran_id, run_id, "noBound.csv", sep = "_")
#   output_full1 = paste(getwd(), "processing", output_fn1, sep = "/")

#   write.table(newDat02_headers, output_full1, quote = F, sep = ",", col.names = F,
#   row.names = F)
#   write.table(newDat02, output_full1, quote = F, sep = ",", col.names = F,
#   row.names = F, append = T)

    # boundary output
    boundName = paste("bound", boundary, "cm.csv", sep = "")
    output_fn2 = paste(tran_id, run_id, boundName, sep = "_")
    output_full2 = paste(getwd(), "processing", output_fn2, sep = "/")

    write.table(newDat03_headers, output_full2, quote = F, sep = ",", col.names = F,
    row.names = F)
    write.table(newDat03, output_full2, quote = F, sep = ",", col.names = F, row.names
     = F, append = T)

    # depth average output
    depName = paste("depAve", boundary, "cm.csv", sep = "")
    output_fn3 = paste(tran_id, run_id, depName, sep = "_")
    output_full3 = paste(getwd(), "processing", output_fn3, sep = "/")

    write.table(newDat04_headers, output_full3, quote = F, sep = ",", col.names = F,
     row.names = F)
    write.table(newDat04, output_full3, quote = F, sep = ",", col.names = F, row.names
     = F,append = T)

    # finds midpoints
```

```
  sortedPts = sort(newDat04[, "dist_LeftPt"])

  filtered_bound = bound_dat[bound_dat[, 2] < wse, ]
  leftPtID = which(filtered_bound[, 1] == min(filtered_bound[, 1]))
  leftX = filtered_bound[leftPtID, 1]
  rightPtID = which(filtered_bound[, 1] == max(filtered_bound[, 1]))
  rightX = filtered_bound[rightPtID, 1]

  #left_mid = (leftX + midpts[1])/2
  #right_mid = (rightX + midpts[length(midpts)])/2

  # variable names corresponding to midsection method
  b_n = c(leftX, sortedPts, rightX)

  #=== calculating flow via midsection method, note q1 and qn = 0 ===
  flow_rectangles = mapply(findFlow, 2:(length(b_n)-1), MoreArgs = list(b_n,
   filtered_bound, newDat04, wse))

  # computed flux
  Q_transect = sum(flow_rectangles)

  start_time = min(dwell_dat[, 1])
  start_timeStr = strftime(start_time, "%Y-%m-%d_%H:%M:%OS")
  end_time = max(dwell_dat[, 2])
  end_timeStr = strftime(end_time, "%Y-%m-%d_%H:%M:%OS")

  dat = cbind(start_timeStr, end_timeStr, tran_id, run_id, Q_transect)

  return(dat)
}
```

A.2   Defining Dwell States

This code defines the time-domain of every dwell state by comparing two different motor log files. One file is the command file with the timestamp of when the dwell begins and ends. The other file is the motor log file periodically updated with timestamps, more often when the motors are moving. The code will by default favor the motor log file timestamps, but if the time entry is missing, it will choose the command file timestamps. An additional buffer of 5 seconds at the beginning and end of the dwell states are added to avoid collecting sensor data when the apparatus is actually in motion.

```
# ===== function to define dwell state =====
find_times = function(index, comm_dat, motor_sliceDat){
  #takes closest time between motor log and start and end of dwells from command log
  motorStart_t = which.min(abs(as.numeric(comm_dat[index, "start_t"] -
   motor_sliceDat[, 1])))
  motorEnd_t = which.min(abs(as.numeric(comm_dat[index, "end_t"] - motor_sliceDat[,
   1])))

  #creates 10 second buffer for the start and end of dwell times in motor logs
  motor_sDatRange = motor_sliceDat[motor_sliceDat[, 1] >
   motor_sliceDat[motorStart_t, 1] - 5 & motor_sliceDat[, 1] <
   motor_sliceDat[motorStart_t, 1] + 5, ]
  motor_eDatRange = motor_sliceDat[motor_sliceDat[, 1] > motor_sliceDat[motorEnd_t,
```

```
   1] - 5 & motor_sliceDat[, 1] < motor_sliceDat[motorEnd_t, 1] + 5, ]

   #takes closest distance between motor log and start and end of dwells from command
    log
   start_dist = sqrt((comm_dat[index, "x"] - motor_sDatRange[, 2])^2 +
                     (comm_dat[index, "y"] - motor_sDatRange[, 3])^2)
   end_dist = sqrt((comm_dat[index, "x"] - motor_eDatRange[, 2])^2 +
                   (comm_dat[index, "y"] - motor_eDatRange[, 3])^2)

   #checking distances
   if(min(start_dist) > 0.3 | min(end_dist) > 0.3)
     stop("something fishy between motor and command position logs")

   #id start/end minimum distances
   startDist_minId = min(which(start_dist == min(start_dist)))
   endDist_minId = max(which(end_dist == min(end_dist)))

   #checks the indices to see if they're pointing at the same thing
   timeDiff_dist = as.numeric(motor_eDatRange[endDist_minId, 1] -
                              motor_sDatRange[startDist_minId, 1])

   #if the time difference is 0, use command log time
   if(timeDiff_dist == 0){
     dat = comm_dat[index, 1:4]
   } else{

   #otherwise, use motor log time
     dat = data.frame(start_t = motor_sDatRange[startDist_minId, 1],
                      end_t = motor_eDatRange[endDist_minId, 1],
                      x = comm_dat[index, 3],
                      y = comm_dat[index, 4])
   }
   return(dat)
}
```

A.3   Averaging Sensor Data in Dwell States

This code averages the raw data the sensor provides according to the sensor timestamps. Special care needs taken for the Sontek ADV whose timestamp is the beginning of a 3 second averaging period. Therefore, an additional time buffer is added for averaging.

```
# ===== hydrolab averaging by dwell state =====
hydro_ave = function(index, dwell_dat, hydro_dat){

  # slices data from the dwell start and end time with 0.5 second buffer at the
   beginning and end
  hydro_slice = hydro_dat[hydro_dat[, 1] > dwell_dat[index, 1] + 0.5 &
                          hydro_dat[, 1] < dwell_dat[index, 2] - 0.5, ]
  hydro_startT = hydro_slice[1, 1]
  hydro_endT = hydro_slice[nrow(hydro_slice), 1]
  hydro_mean = mean(hydro_slice[, 2])
  dat = data.frame(h_startT = hydro_startT, h_endT = hydro_endT, SC_mean =
       hydro_mean)

  return(dat)
}

# ===== averages sontek data and calculates magnitude velocity =====
sontek_ave = function(index, dwell_dat, sontek_dat){
```

```
#sontek stamps the beginning of each 3 second averaging interval
#internal clock considered to be reliable probably due to internal processing time
sontek_slice = sontek_dat[sontek_dat[, 2] > dwell_dat[index, 1] + 5 &
                          sontek_dat[, 2] < dwell_dat[index, 2] - 5, ]
sontek_startT = sontek_slice[1, 1]
sontek_endT = sontek_slice[nrow(sontek_slice), 1]

#takes the internal sontek time, the clocks are drastically different
intSontek_startT = sontek_slice[1, 2]
intSontek_endT = sontek_slice[nrow(sontek_slice), 2]

#takes only useful columns
col_names = c("WaterVel1_X_E", "WaterVel2_Y_N", "WaterVel3_Z_U", "EndLoc")
sontek_slice = sontek_slice[, col_names]

#calculates velocity magnitude
sontek_velMag = sqrt(sontek_slice[, "WaterVel1_X_E"]^2 +
                     sontek_slice[, "WaterVel2_Y_N"]^2 +
                     sontek_slice[, "WaterVel3_Z_U"]^2)
sontek_slice = cbind(sontek_slice, sontek_velMag)

#conditional for exorbitant velocity values
badVel_id = which(sontek_slice$sontek_velMag > 180)
if(length(badVel_id) > 0){
  sontek_slice = sontek_slice[!(1:nrow(sontek_slice) %in% badVel_id), ]

}

#takes average over columns
sontek_mean = colMeans(sontek_slice)

#matrix helps fill data.frame by column
sontek_mean = matrix(sontek_mean, nrow = 1)
numPts = nrow(sontek_slice)

dat = data.frame(s_startT = sontek_startT,
                 s_endT = sontek_endT,
                 s_intStartT = intSontek_startT,
                 s_intEndT = intSontek_endT,
                 numPts = numPts,
                 sontek_mean)

return(dat)
}
```

A.4   Mapping Sensor Data River Bed Topography

The interpolated cross-sectional river bed topography is chosen for each sensor data by choosing the closest corresponding lateral location.

```
# ===== finds the corresponding elevation to the visited points ====
findBound = function(index, bound_dat, visited_xPts){
  min_id = which.min(abs(bound_dat[, 1] - visited_xPts[index]))
  dat = cbind(bound_dat[min_id, ], min_id)
  return(dat)
}
```

A.5   Filtering, Calculating Depth-averaged Velocity, and Velocity Integration

The remaining code first filters ADV measurements close to or beyond the river bed. The code then applies the vertical velocity method to estimate the depth-averaged velocity, assigning weights to each velocity depending on the depth ratio of the velocity point measurement. Finally the code integrates the velocity field by multiplying the depth-averaged velocity to the associated cross-sectional slice area estimated with a rectangle, and finally summing the pieces.

```
# ===== mean velocity weighted by ratio =====
velWeight = function(index, dat_slice02, elevBound, wse){
  # ratio between depths
  ratio = (wse - dat_slice02[index, "elevation"])/(wse - elevBound)

  # vertical-velocity curves determined by Hulsing, Smith, & Cobb
  if(ratio > 0 & ratio < 0.2){
    ratio_toAve = 1.160
  } else if(ratio >= 0.2 & ratio < 0.3){
    ratio_toAve = 1.149
  } else if(ratio >= 0.3 & ratio < 0.4){
    ratio_toAve = 1.130
  } else if(ratio >= 0.4 & ratio < 0.5){
    ratio_toAve = 1.108
  } else if(ratio >= 0.5 & ratio < 0.6){
    ratio_toAve = 1.067
  } else if(ratio >= 0.6 & ratio < 0.7){
    ratio_toAve = 1.020
  } else if(ratio >= 0.7 & ratio < 0.8){
    ratio_toAve = 0.953
  } else if(ratio >= 0.8 & ratio < 0.9){
    ratio_toAve = 0.871
  } else if(ratio >= 0.9 & ratio < 0.95){
    ratio_toAve = 0.746
  } else if(ratio >= 0.95 & ratio <= 1){
    ratio_toAve = 0.648
  }

  vel_longWeight = dat_slice02[index, "vel_long"] * (1/ratio_toAve)
  dat = cbind(dat_slice02[index, ], vel_longWeight)
  return(dat)
}

# ===== goes the the id'd boundary points and filters out data ====
boundDat = function(index, bound_datPts, newDat02, visited_xPts, wse){
  dat_slice = newDat02[newDat02[, "dist_LeftPt"] == visited_xPts[index], ]

  # safe distance is 20 cm from transducer
  # note that the 10 cm sample volume is already accounted for
  safe_distID = which(dat_slice[, "elevation"] - bound_datPts[index, "elevation_m"]
   > (boundary/100))
  #safe_distID = which(dat_slice[, "elevation"] - bound_datPts[index, "elevation_m"]
   > 0)

  # record of pts deleted
  pts_deleted = nrow(dat_slice) - length(safe_distID)

  if(length(safe_distID) == 0){
```

```r
    dat = -1
  } else {
  # applies typical velocity curve ratios to longitudinal velocities
    vel_weight = mapply(velWeight, 1:length(safe_distID),
                        MoreArgs = list(dat_slice02 = dat_slice[safe_distID, ],
                        elevBound = bound_datPts[index, "elevation_m"], wse),
                        SIMPLIFY = F)
    vel_weight = data.frame(do.call(rbind, vel_weight))
    dat = cbind(vel_weight, pts_deleted)
  }
  return(dat)
}

# ===== calculates flow according to midsection method =====
findFlow = function(index, b_n, filtered_bound, newDat04, wse){
  # finds closest interpolated point
  closest_bnID = which.min(abs(filtered_bound[, 1] - b_n[index]))

  # accounts for wse
  depth = wse - filtered_bound[closest_bnID, 2]
  vel = newDat04[newDat04[, "dist_LeftPt"] == b_n[index], "velLongWeighted"]

  # convert to meters
  vel_m = vel/100

  # computes flow at rectangle
  dat = vel_m * (b_n[index + 1] - b_n[index - 1])/2 * depth
  return(dat)

}
```

References

[1] N. M. Dubrovsky, C. R. Kratzer, L. R. Brown, J. A. M. Gronberg, and K. R. Burow, *Water Quality in the San Joaquin-Tulare Basins, California, 1992-1995.* 1998.

[2] K. R. Burow, J. L. Shelton, J. A. Hevesi, and G. S. Weissmann, *Hydrogeologic Characterization of the Modesto Area, San Joaquin Valley, California.* 2004.

[3] J. A. M. Gronberg, N. M. Dubrovsky, C. R. Kratzer, J. L. Domagalski, L. R. Brown, and K. R. Burow, *Environmental Setting of the San Joaquin-Tulare Basins, California.* 1998.

[4] J. A. M. Gronberg, C. R. Kratzer, K. R. Burow, J. L. Domagalski, and S. P. Phillips, *Water-Quality Assessment of the San Joaquin-Tulare Basins- Entering a New Decade.* 2004.

[5] C. Zamora, *Estimating Water Fluxes Across the Sediment-Water Interface in the Lower Merced River, California.* 2007.

[6] S. Amaya, *Descriptive Analysis of the Turbulent Velocity Profiles for Two Cross Sections in a River Confluence Zone*. University of California, Merced, 2007.

[7] T. Harmon, R. F. Ambrose, R. M. Gilbert, J. Fisher, M. Stealey, and W. Kaiser, "High-Resolution River Hydraulic and Water Quality Characterization Using Rapidly Deployable Networked Infomechanical Systems (NIMS RD)," *Environmental Engineering Science*, vol. 24, no. 2, pp. 151-159, Feb. 2007.

[8] B. Jordan, M. Batalin, and W. Kaiser, "NIMS RD: A Rapidly Deployable Cable Based Robot," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 144-150, 2007.

[9] A. Singh et al., "Autonomous Robotic Sensing Experiments at San Joaquin River," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 4987-4993, 2007.

[10] G. Voulgaris and J. H. Trowbridge, "Evaluation of the Acoustic Doppler Velocimeter (ADV) for Turbulence Measurements," *Journal of Atmospheric and Oceanic Technology*, vol. 15, no. 1, pp. 272-289, Feb. 1998.

[11] Sontek Inc., *Argonaut-ADV Principles of Operation.* 2000.

[12] S. N. Lane et al., "Three-dimensional measurement of river channel flow processes using acoustic doppler velocimetry," *Earth Surface Processes and Landforms*, vol. 23, no. 13, pp. 1247-1267, 1998.

[13] Sontek Inc., *Argonaut-ADV Operation Manual Firmware Version 11.2.* 2000.

[14] T. Buchanan and W. Somers, *Discharge measurements at gaging stations*. USGS, 1969.

[15] C. Chen and F. J. Millero, "Speed of sound in seawater at high pressures," *The Journal of the Acoustical Society of America*, vol. 62, no. 5, pp. 1129-1135, Nov. 1977.

[16] V. A. Del Grosso, "New equation for the speed of sound in natural waters (with comparisons to other equations)," *The Journal of the Acoustical Society of America*, vol. 56, no. 4, pp. 1084-1091, Oct. 1974.

[17] Magellan Navigation, Inc., *MobileMapper Pro Datasheet.* 2006.

[18] Leica Geosystems, *Leica Builder User Manual, Version 2.0.* 2006.

[19] Valeport Ltd, *MIDAS Surveyor Operation Manual.* 2003.

[20] Valeport Ltd, *Datasheet Reference Number: Surveyor v1A.* .

[21] M. N. Gooseff and B. L. McGlynn, "A stream tracer technique employing ionic tracers and specific conductance data applied to the Maimai catchment, New Zealand," *Hydrological Processes*, vol. 19, no. 13, pp. 2491-2506, 2005.

[22] Hach Environmental, *Conductivity Calibration Video Transcript.* .

[23] R. L. Miller, W. L. Bradford, and N. E. Peters, *Specific Conductance: Theoretical Considerations and Application to Analytical Quality Control.* 1998.

[24] Hach Environmental, *Hydrolab DS5X, DS5, and MS5 Water Quality Multiprobes: User*

Manual. 2006.

[25] G. B. Pasternack, C. L. Wang, and J. E. Merz, "Application of a 2D hydrodynamic model to design of reach-scale spawning gravel replenishment on the Mokelumne River, California," *River Research and Applications*, vol. 20, no. 2, pp. 205-225, 2004.

[26] J. C. Fisher, "CRAN - Package RSurvey," 04-Apr-2009. [Online]. Available: http://cran.r-project.org/web/packages/RSurvey/index.html. [Accessed: 25-Aug-2009].

[27] A. O. Finley and S. Banerjee, "CRAN - Package MBA," 17-Jun-2009. [Online]. Available: http://cran.r-project.org/web/packages/MBA/index.html. [Accessed: 25-Aug-2009].

[28] S. Lee, G. Wolberg, and S. Shin, "Scattered data interpolation with multilevel B-splines," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 3, no. 3, pp. 228-244, 1997.

[29] J. Pouderoux, J. Gonzato, I. Tobor, and P. Guitton, "Adaptive hierarchical RBF interpolation for creating smooth digital elevation models," in *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pp. 232-240, 2004.

[30] M. Bertram, X. Tricoche, and H. Hagen, "Adaptive smooth scattered-data approximation for large-scale terrain visualization," in *Proceedings of the symposium on Data visualisation 2003*, pp. 177-184, 2003.

[31] S. Rantz, "Measurement of Stage and Discharge," in *Measurement and Computation of Streamflow*, vol. 1, 2 vols., Washington DC, USA: US Govt Printing Office, 1982.

[32] "NOAA's Geophysical Data Center - Geomagnetic Data." [Online]. Available: http://www.ngdc.noaa.gov/geomagmodels/struts/calcDeclination. [Accessed: 24-Nov-2009].

[33] H. Hulsing, W. Smith, and E. Cobb, "Velocity-head coefficients in open channels," .

[34] "California Data Exchange Center." [Online]. Available: http://cdec.water.ca.gov/. [Accessed: 01-Dec-2009].

[35] "USGS Water Resources of the United States." [Online]. Available: http://water.usgs.gov/. [Accessed: 01-Dec-2009].

[36] P. Steffler and J. Blackburn, "River 2D: Two-Dimensional Depth Averaged Model of River Hydrodynamics and Fish Habitat," 2002.

[37] H. B. Fischer, *Mixing in inland and coastal waters*. Academic Press, 1979.

[38] T. M. Jeon, K. O. Baek, and I. W. Seo, "Development of an empirical equation for the transverse dispersion coefficient in natural streams," *Environmental Fluid Mechanics*, vol. 7, no. 4, pp. 317-329, 2007.

[39] Z. Deng, V. P. Singh, and L. Bengtsson, "Longitudinal Dispersion Coefficient in Straight Rivers," *Journal of Hydraulic Engineering*, vol. 127, no. 11, pp. 919-927, Nov. 2001.

[40] R. P. Schwarzenbach, P. M. Gschwend, and D. M. Imboden, *Environmental organic chemistry*. John Wiley and Sons, 2003.

[41] K. Suh, K. Kim, S. Jung, and J. Lee, "Determination of dispersion coefficients using radioisotope data in river environment," *Applied Radiation and Isotopes*, vol. 67, no. 7, pp. 1499-1502, 2009.

[42] D. Froelich, *User's Manual for FESWMS FST2DH: Two-dimensional Depth-averaged Flow and Sediment Transport Model*. US DOT, FHWA, 2002.

[43] B. P. Donnell, J. V. Letter, Jr., W. H. McAnally, Jr., and L. C. Roig, "Users Guide to RMA2 WES Version 4.5," US Army, Engineer Research and Development Center, Waterways Experiment Station, Coastal Hydraulics Lab, 2008.

[44] M. Amein, "Streamflow routing on computer by characteristics," *Water Resources Research*, vol. 2, no. 1, pp. PP. 123-130, 1966.

[45] A. Laenen and J. C. Risley, *Precipitation-Runoff and Streamflow-Routing Models for the Willamette River Basin, Oregon*. USGS, 1995.

[46] H. E. Jobson and A. W. Harbaugh, *Modifications to the Diffusion Analogy Surface-Water Flow Model (DAFLOW) for Coupling to the Modular Finite-Difference Ground-Water Flow Model (MODFLOW)*. USGS, 1999.

[47] G. J. Arcement, Jr. and V. R. Schneider, *Guide for Selecting Manning's Roughness Coefficients for Natural Channels and Flood Plains*. Federal Highway Administration, 1984.

[48] G. W. Brunner, *HEC-RAS, River Analysis System: Hydraulic Reference Manual*. U.S. Army Corps of Engineers, Institute for Water Resources, 2010.

[49] C. R. Wagner, *Simulation of Water-Surface Elevations and Velocity Distributions at the U.S. Highway 13 Bridge over the Tar River at Greenville, North Carolina, Using One- and Two-Dimensional Steady-State Hydraulic Models*. USGS, 2007.

[50] G. B. Pasternack, C. L. Wang, and J. E. Merz, "Application of a 2D hydrodynamic model to design of reach-scale spawning gravel replenishment on the Mokelumne River, California," *River Research and Applications*, vol. 20, no. 2, pp. 205-225, 2004.

[51] G. J. Barton, R. R. McDonald, and J. M. Nelson, *Simulation of Streamflow Using a Multidimensional Flow Model for White Sturgeon, Kootenai River near Bonners Ferry, Idaho- Suplement to the Scientific Investigations Report 2005-5230*. USGS, 2009.

[52] S. N. Lane, R. J. Hardy, L. Elliott, and D. B. Ingham, "High-resolution numerical modelling of three-dimensional flows over complex river bed topography," *Hydrological Processes*, vol. 16, no. 11, pp. 2261-2272, 2002.

[53] S. N. Lane, K. F. Bradbrook, K. S. Richards, P. A. Biron, and A. G. Roy, "The application of computational fluid dynamics to natural river channels: three-dimensional versus two-dimensional approaches," *Geomorphology*, vol. 29, no. 1, pp. 1-20, Aug. 1999.

[54] S. P. Rice, A. G. Roy, and B. L. Rhoads, *River Confluences, Tributaries and the Fluvial Network*. John Wiley and Sons, 2008.

[55] M. P. Mosley, "An Experimental Study of Channel Confluences," *The Journal of Geology*, vol. 84, no. 5, pp. 535-562, Sep. 1976.

[56] L. J. Weber, E. D. Schumate, and N. Mawer, "Experiments on Flow at a 90-degree Open-Channel Junction," *Journal of Hydraulic Engineering*, vol. 127, no. 5, pp. 340-350, May. 2001.

[57] B. De Serres, A. G. Roy, P. M. Biron, and J. L. Best, "Three-dimensional structure of flow at a confluence of river channels with discordant beds," *Geomorphology*, vol. 26, no. 4, pp. 313-335, Jan. 1999.

[58] B. L. Rhoads and S. T. Kenworthy, "Time-averaged flow structure in the central region of a stream confluence," *Earth Surface Processes and Landforms*, vol. 23, no. 2, pp. 171-191, 1998.

[59] S. N. Lane, D. R. Parsons, J. L. Best, O. Orfeo, R. A. Kostaschuk, and R. J. Hardy, "Causes of rapid mixing at a junction of two large rivers: Río Paraná and Río Paraguay, Argentina," *Journal of Geophysical Research*, vol. 113, p. 16 PP., Jun. 2008.

[60] S. B. Weerakoon, N. Tamai, and Y. Kawahara, "Depth-averaged flow computation at a river confluence," *Proceedings of the ICE - Water and Maritime Engineering*, vol. 156, no. 1, pp. 73-83, 2003.

[61] M. Roca, J. Martín-Vide, and P. Moreta, "Modelling a torrential event in a river confluence," *Journal of Hydrology*, vol. 364, no. 3, pp. 207-215, Jan. 2009.

[62] N. Kabeya et al., "Isotopic investigation of river water mixing around the confluence of the Tonle Sap and Mekong rivers," *Hydrological Processes*, vol. 22, no. 9, pp. 1351-1358,

2008.

[63] L. E. Schemel, M. H. Cox, R. L. Runkel, and B. A. Kimball, "Multiple injected and natural conservative tracers quantify mixing in a stream confluence affected by acid mine drainage near Silverton, Colorado," *Hydrological Processes*, vol. 20, no. 13, pp. 2727-2743, 2006.

[64] L. Maurice-Bourgoin, B. Quemerais, P. Moreira-Turcq, and P. Seyler, "Transport, distribution and speciation of mercury in the Amazon River at the confluence of black and white waters of the Negro and Solimoes Rivers," *Hydrological Processes*, vol. 17, no. 7, pp. 1405-1417, 2003.

[65] T. D. Osting and B. R. Hodges, *Estimating uncertainty of 2D hydraulic models used for aquatic habitat modeling studies*. Bureau of Engineering Research: The University of Texas at Austin, 2007.