State of California
California Natural Resources Agency
DEPARTMENT OF WATER RESOURCES

Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh

43rd Annual Progress Report to the
State Water Resources Control Board in
Accordance with Water Right Decisions 1485 and 1641

June 2022

Gavin Newsom	Wade Crowfoot	Karla Nemeth
Governor	Secretary for Natural Resources	Director
State of California	California Natural Resources Agency	California Department of Water Resources

State of California
Gavin Newsom, Governor

California Natural Resources Agency
John Laird, Secretary for Natural Resources

California Department of Water Resources
Karla Nemeth, Director
Cindy Messer, Lead Deputy Director

General Counsel Tom Gibson	Public Affairs Margaret Mohr, Deputy Director	Legislative Affairs Kasey Schimke, Assistant Director

Assistant Lead Deputy Director Vacant	California Water Commission Joseph Yun, Executive Officer

Deputy Directors

John Paasch	Security and Emergency Management Program
Stephanie Varrelman	Business Operations
Gary Lippner	Flood Management and Dam Safety
Ted Craddock	State Water Project
John Andrew	Climate Resilience
Bianca Sievers	Special Initiatives
Kristopher A. Tjernell	Integrated Watershed Management
Paul Gosselin	Sustainable Groundwater Management
Joel Ledesma	Statewide Water and Energy

Bay-Delta Office
Vacant, Manager

Modeling Support Branch
Erik Reyes, Acting Manager

Delta Modeling Section
Nicky Sandhu, Supervising Engineer

Edited by
Min Yu, Bay-Delta Office
Robert Suits, Bay-Delta Office
(See individual chapters for authors.)

Editorial review
William O’Daly, Supervisor of Technical Publications
Frank Keeley, Research Writer

Foreword

This is the 43rd annual progress report of the California Department of Water Resources’ San Francisco Bay-Delta Evaluation Program, which is carried out by the Delta Modeling Section. This report is submitted annually to the State Water Resources Control Board pursuant to its Water Right Decision D-1485, Term 9, which is still active pursuant to its Water Right Decision D-1641, Term 8.

This report documents progress in the development and enhancement of computer models for the Delta Modeling Section of the Bay-Delta Office. It also reports the latest findings of studies conducted as part of the program. This report was compiled under the direction of Nicky Sandhu, Program Manager for the Bay-Delta Evaluation Program.

Online versions of previous annual progress reports are available at: https://www.water.ca.gov/Library/Modeling-and-Analysis.

For more information, contact:

Nicky Sandhu, Supervising Engineer
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Prabhjot.Sandhu@water.ca.gov
(916) 902-9945

Contents

Preface

Chapter 1. DSM2 V8.2.0 Calibration

This chapter documents the calibration effort of the latest version of Delta Simulation Model 2 (DSM2), V8.2.0. The previous release of the DSM2 model (V8.1.2) was modified to enable the use of Delta channel depletion estimated by the Delta Channel Depletion model rather than that by the Delta Island Consumptive Use model. The flow, stage, and salinity simulations from the calibrated DSM2 V8.2.0 were mostly similar to those from the previously calibrated DSM2 V8.1.2 but were in better agreement with observed data in most cases. In particular, DSM2 v8.2.0 better fit observed salinity data during the validation period at most key locations.

Chapter 2. DSM2 Georeferenced Grid Maps

This chapter describes the development of georeferenced grid maps for DSM2 (Tom et al. 2020). The georeferenced grid maps are stored as GIS shapefiles with symbology added to the various features to represent channels, nodes, gates, reservoirs, reservoir connections, and monitoring stations. The shapefiles are compatible with ArcGIS and QGIS and are available on the CNRA Open Data web site. The georeferenced grid maps are also available as Portable Document Format (PDF) files.

The workflow for creating the grid maps is automated as much as possible, streamlining updates and the development of new versions. DSM2 is a 1D model and uses input derived from georeferenced information rather than using georeferenced information directly. Georeferenced grid maps can help ensure that the DSM2 input derived from georeferenced information is sufficiently accurate.

The first versions of the DSM2 grid map were created using AutoCAD. Later versions created using AutoCAD were exported to PDF files, which were then printed on plotter paper. The most recent version created using this method dates back to 2002. The first ArcGIS version was created in 2009.

Chapter 3. DSM2 Water Temperature Modeling Input Extension: 1922–2015

The water quality module of the Delta Simulation Model II (DSM2 QUAL) was previously calibrated and validated from 1990–2008 to simulate water temperature in the Sacramento-San Joaquin Delta (Delta) in 2011

(Resources Management Associates 2011). In a follow-up study, the simulation period was extended to 2012 (Resources Management Associates 2015). Recently, the Delta Modeling Section (DMS) was tasked to extend the water temperature simulation period to water years 1922–2015 to align with the current simulation period used by DWR’s water resources planning model, CalSim3.

This chapter describes the input data requirements for modeling Delta water temperature via DSM2 QUAL and the methods applied to assemble or derive these data for the extended period.

Chapter 4. South Delta Salinity-Constituent Conversion via Machine Learning

Electrical Conductance (EC) is a water quality metric typically used to represent the salinity level. It can also be used as the predictor for other ion constituents, including Total Dissolved Solids (TDS), dissolved chloride (Cl-), dissolved sulfate (SO42-), dissolved sodium (Na+), dissolved calcium (Ca2+), dissolved magnesium (Mg2+), dissolved nitrate (NO3-), dissolved potassium (K+), dissolved bromide (Br-), dissolved boron (B), Alkalinity, and hardness in the Delta. These ion constituents are typically treated as water quality indicators and can be measured by standard laboratory methods. Regression models have also been developed and applied to simulate the concentrations of these ion constituents in the Delta (Jung 2000; Suits 2002; Hutton 2006; Denton 2015). Most recently, the North Central Region Office (NCRO) used parametric quadratic regression equations to estimate the concentrations of these 12 ion constituents, using EC as the predictor. This chapter provides an overview of the study, intended to identify and investigate sources of local salt loading in south Delta channels, which collected and used grab sample data from 2018–2020 at seven key locations in the south Delta (California Department of Water Resources North Central Region Office 2021). The goal of the current study is to develop machine learning models to emulate the regression equations in the NCRO study to simulate ion constituents. The results indicate that machine learning models can provide simulations comparable or superior to the regression equations.

Chapter 5. Hotstart and Nudging Preprocessors for Bay-Delta SCHISM

This chapter describes the methods and usages of preparing hotstart and nudging model input files for the Semi-implicit Cross-scale Hydroinformatics Simulation Model (SCHISM), a three-dimensional hydrodynamic and water quality model applied extensively to the Sacramento-San Joaquin River Delta.

The concept of hot-starting SCHISM is to start the model with accurate or realistic initial states of temperature, salinity, and other potential water quality constituents based on previous model states or observed data. Nudging is a process of relaxing the model toward local observations, creating final merged fields that reflect both the model dynamics and observations. Both developments provide important capability to improve SCHISM modeling results and increase flexibility in applying the model.

The scripts and examples described in this chapter are distributed publicly in the subdirectory of a Python preprocessing library on github called schimpy (https://github.com/CADWRDeltaModeling/schimpy).

43rd Annual Progress Report
June 2022

Chapter 1
DSM2 V8.2.0 Calibration

Authors: Minxue He, Yu Zhou, Bradley Tom, Parviz Nader-Tehrani, and Nicky Sandhu
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Contents

Figures

Tables

Chapter 1 DSM2 V8.2.0 Calibration

1.1 Introduction

1.1.1 Background

Delta Simulation Model II (DSM2) is a one-dimensional hydrodynamics and water quality model applied in historical simulation, real-time forecasting, and long-term planning practices in the Sacramento-San Joaquin Delta (Delta). DSM2 (Version 8.2) is a key analytical model being used to assess possible impacts to Delta conditions from the proposed Delta Conveyance Project (DCP), which includes several structural and operational changes to the Delta.

DSM2 relies on two modules, HYDRO and QUAL, to simulate Delta hydrodynamics and water quality conditions, respectively. Past DSM2 calibrations occurred in 1997, 2000, and 2009 (California Department of Water Resources 1997; Nader-Tehrani and Shrestha 2000; California Department of Water Resources 2009), along with some limited parameter fine-tuning when newer versions of DSM2 were released (Liu and Sandhu 2012; Liang and Suits 2018).

1.1.2 DSM2 enhancements since its previous release

The current version of DSM2 (V8.2.0) differs from the previous version, V8.1.2 (Liu and Sandhu 2012), mainly because of different Delta channel depletion estimates being used by DSM2. Specifically, V8.1.2 uses the Delta channel depletions estimated by the Delta Island Consumptive Use (DICU) model, while V8.2.0 employs those estimated by the Delta Channel Depletion (DCD) model. DICU is a monthly model that divides the Delta into 142 subareas. It simulates the water entering, leaving, or being stored in each of these subareas when estimating the monthly consumptive use of water for each subarea. Delta channel depletion for each subarea is then assumed to be the same as the consumptive use. In comparison, DCD has finer temporal and spatial resolutions that provide simulations on a daily scale for 168 subareas in the Delta. DCD also incorporates a number of enhancements, including updated parameterization and the addition of physical processes to distinguish Delta channel depletions from consumptive use (Liang and Suits 2017, 2018).

1.1.3 Purpose of DSM2 recalibration

The previous calibration (California Department of Water Resources 2009) used monitoring data up to 2008. Since then, additional flow, stage, and salinity measurements have become available, including those during the 2012–2015 drought. These new data collectively provide a better depiction of the current hydrodynamic and water quality conditions in the Delta and enable evaluation of model performance during an extreme event. In addition, DWR is considering operating the Suisun Marsh Salinity Control Gates (SMSCGs) during summer per ITP requirements. Modeling the possible impacts of this action on local hydrodynamics and water quality requires channel depletion estimates in the Suisun Marsh in addition to the legal Delta. The DCD model has been recently extended to cover the Suisun Marsh area (Liang 2020). The extended DCD, when coupled with DSM2, should enable improved hydrodynamic and water quality simulations in Suisun Marsh.

Considering those observations, it is necessary to recalibrate DSM2 (V8.2.0) to reflect revised channel depletion estimates which now include Suisun Marsh and additional observed data availability as well as the addition of the extended DCD model. For these reasons, DSM2 was recalibrated to ensure the adequacy of the model for the DCP Delta analyses. This effort is relatively limited when compared with a more comprehensive future recalibration of DSM2 planned for the next release (V8.3), which will incorporate further model enhancements.

1.2 Calibration Setup

1.2.1 Observations

Flow, stage, and salinity (represented by electrical conductivity [EC]) observations at 15-minute intervals were obtained from multiple sources, such as the California Data Exchange Center (CDEC), Division of Environmental Services (DES), and the Northern California Regional Office (NCRO), for a range of stations in the Delta, as presented in Table 1-A1 of Appendix 1-A. These stations were deemed important for DSM2 performance evaluation (Sections 3 and 4). The data were quality-controlled before being applied in calibration and validation.

1.2.2 Calibration and validation periods

The selection of the calibration period was based on:

1. Data availability. Observed data should faithfully (with minimal missing observations) reflect the more recent Delta structural and operational conditions. Previous calibrations used flow, stage, and salinity observations only up to 2008. The current calibration includes the use of more recent observations.
2. Data representativeness. Observed data should cover a wide range of variations in hydrodynamics and salinity, particularly for the latter, which typically needs a longer calibration period than the former does. Given this, the current calibration includes periods with both high and low ranges of flow and salinity.

With the above considerations, water years 2011–2012 (containing a wet year and a below-normal year; Table 1-1) were selected as the calibration period for flow and stage, while water years 2010–2017 (containing two wet years, three below normal water years, one dry year, and two critical years; Table 1-1) were chosen for salinity calibration. The calibrated model was then validated using all observations collected since water year 2001, as there were notable physical changes to the Delta in early 2000 (e.g., Liberty Island flooding). To be specific, the validation period for flow and stage were water years 2001–2017. The validation period for salinity were water years 2001–2009.

Table 1-1 Water year type classification in the Sacramento Valley

Water Year	Sacramento Valley*	Water Year	Sacramento Valley*	Water Year	Sacramento Valley*
2001	D	2007	D	2013	D
2002	D	2008	C	2014	C
2003	AN	2009	D	2015	C
2004	BN	2010	BN	2016	BN
2005	AN	2011	W	2017	W
2006	W	2012	BN	—	—

*Source: https://cdec.water.ca.gov/reportapp/javareports?name=WSIHIST.

1.2.3 Calibration metrics

During the calibration, DSM2 model performance is evaluated both quantitatively and qualitatively. A set of quantitative metrics was calculated to evaluate the goodness-of-fit between model simulations and the corresponding observations. Additionally, visual inspection was conducted to compare them qualitatively. Flow and stage simulations were examined on both a tidal scale and a net daily scale. For salinity, the evaluation was extended to a monthly scale to maintain consistency with previous calibration efforts (Nader-Tehrani and Shrestha 2000; California Department of Water Resources 2009).

For flow and stage simulated via DSM2 HYDRO, the metrics were:

- Visual inspection of instantaneous (15-minute) flow and stage simulations against observations. This time series plot provides an initial assessment on DSM2 HYDRO’s ability in simulating the amplitude, phase, and patterns of variation in flow and stage. For brevity, the results are only illustrated over one month from the calibration period given the fine timescale (i.e., 15-minute).
- Visual inspection of Godin-filtered flow and stage simulations against observations. This time series plot provides insights on how well the model simulates the flow field and water levels over the entire calibration period.
- Visual inspection of tidally averaged flow and stage simulations against observations. This scatter plot gives an indication on how modeled daily flows and stages compare with the corresponding observations (i.e., how close to the 1:1 line). It does not provide any assessment on the phase error in simulations.

- Error in flow/tidal amplitude. This metric measures the differences between modeled and observed flow/tidal amplitude. Probability density function (PDF) curves showing the occurrence frequency of discrepancies in percentage over the calibration period are presented. A mean amplitude error is also calculated by averaging all these discrepancies during the calibration period.
- Error in flow/tidal phase. This metric quantifies the differences between modeled and observed peak flow or tidal ebb/flood timing. A PDF curve showing the differences in minutes over the calibration period to indicate whether the model simulations are lagging or leading the observations. A mean phase error is also determined by averaging all these differences during the calibration period.
- Mean error in tidally averaged flow and stage. This metric serves as a measure of the difference between long-term (over the calibration period) average modeled and simulated tidally averaged flow and stage data. It shows whether the model has a dry (under-prediction) or wet (over-prediction) bias and by how much on average.
- Root Mean Squared Error (RMSE) of tidally averaged flow and stage. This metric allows assessing the average magnitude of the model error. Since it takes the square root of the discrepancy between modeled and observed data, it implicitly assigns relative higher weights to larger errors. This makes it particularly useful when large errors are especially undesirable, which is the case for HYDRO-simulated flow and stage in the current study. But, unlike the mean error, the RMSE does not reveal the direction (over or underprediction) of the error.

For salinity (EC) simulated via DSM2 QUAL, the metrics were:

- Visual inspection of Godin-filtered EC simulation simulations against observations. This time series plot provides an initial assessment on DSM2 QUAL’s ability to simulate the pattern of variation of EC.
- Visual inspection of tidally averaged EC simulations against observations. This indicates how well DSM2 QUAL simulates EC on a daily basis and whether there are seasonal biases over the calibration period.
- Mean error in tidally averaged EC and monthly average EC. This metric quantifies the difference between long-term (over the calibration

period) simulated and observed EC. It averages the negative (under prediction) and positive (over prediction) errors together and may result in a smaller error than via other error metrics.

- Root Mean Squared Error (RMSE) in tidally averaged EC and monthly average EC. This metric is complementary to the mean error in the sense that it provides a more realistic measure on mode errors (particularly for large errors) but does not discern between negative (under prediction) and positive (over prediction) errors.

These calibration metrics were also used in model validation.

1.3 Hydrodynamics calibration and validation

1.3.1 Calibration parameter

The channel roughness coefficient (i.e., Manning's n) was the main parameter for HYDRO calibration. Specifically, Manning's n values of six groups of channels were modified to improve the model's ability to simulate observed flow and stage conditions, including in Sutter Slough and Steamboat Slough, the Sacramento River upstream and downstream of Delta Cross Channel, Georgiana Slough, Suisun Marsh, and Old River at head. The modifications were made progressively (group-by-group), based on calibration metrics. After modifying Manning's n for the final group of channels, calibrated flow and stage well resembled corresponding observations. Channel ID number and Manning's n values before and after the calibration are tabulated in Table 1-2.

Table 1-2 Six groups of channels with modified channel roughness Coefficient (Manning's n) in the current calibration

Group Name	Channel Number	Before Calibration	After Calibration
Head of Old River	54–58	0.03	0.025
Georgiana Slough	366–374	0.028	0.027
Sutter and Steamboat Sloughs	379	0.025	0.034
Sutter and Steamboat Sloughs	380–381	0.025	0.032
Sutter and Steamboat Sloughs	382	0.031	0.038
Sutter and Steamboat Sloughs	383–384	0.029	0.031
Sutter and Steamboat Sloughs	388–390	0.036	0.034
Sacramento River above DCC	410–414	0.028	0.032
Sacramento River above DCC	415–418	0.028	0.034
Sacramento River below DCC	422	0.03	0.028
Sacramento River below DCC	423–429	0.031	0.028
Sacramento River below DCC	430–433	0.026	0.027
Sacramento River below DCC	434	0.025	0.027
Sacramento River below DCC	435	0.015	0.027
Suisun Marsh	461–462	0.035	0.015
Suisun Marsh	463–466	0.035	0.04
Suisun Marsh	470	0.025	0.04
Suisun Marsh	489–491	0.03	0.015
Suisun Marsh	516–517	0.021	0.015
Suisun Marsh	522–523	0.021	0.015
Suisun Marsh	528	0.03	0.04
Suisun Marsh	543	0.03	0.04

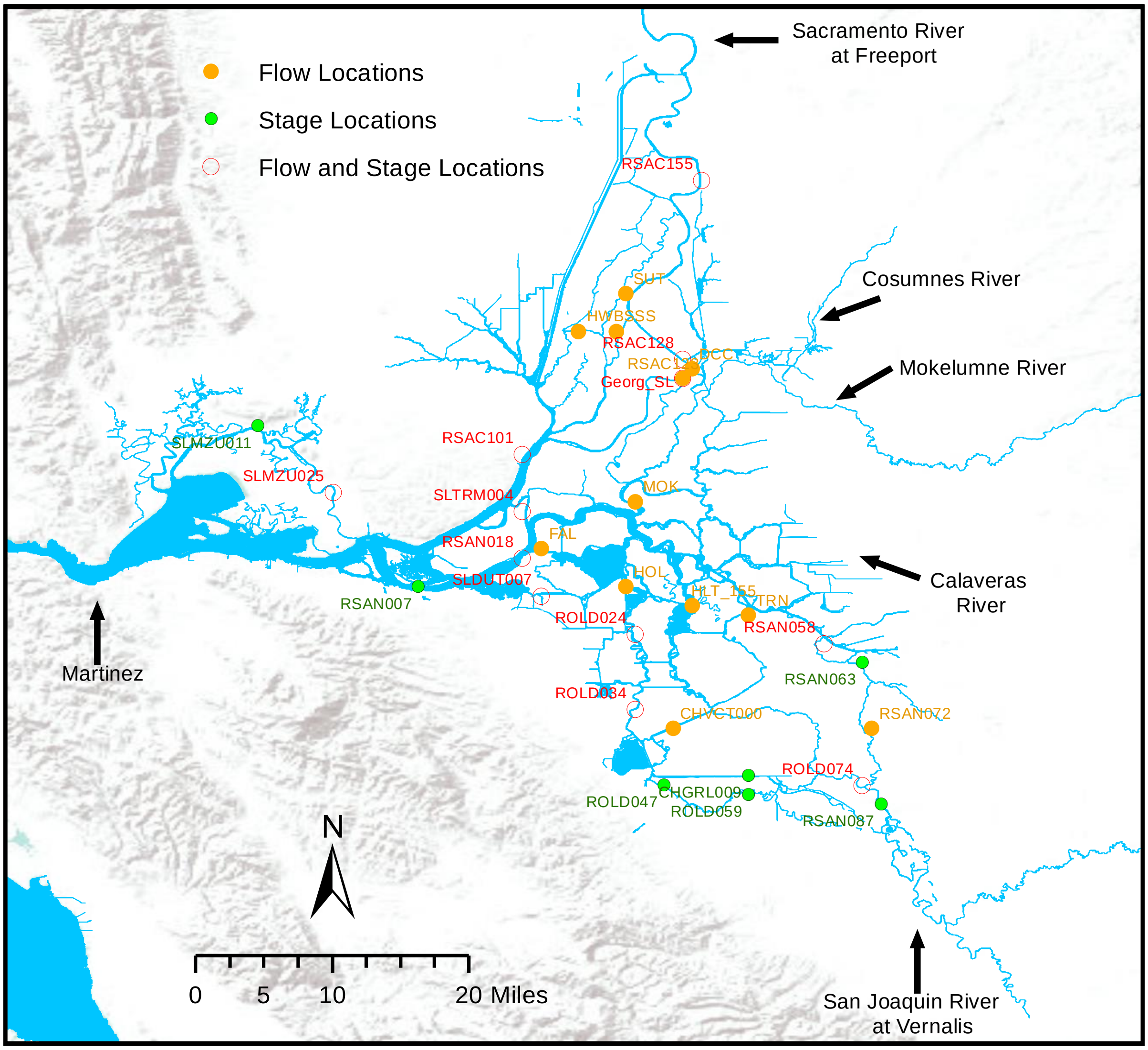
1.3.2 Calibration and validation stations

Table 1-3 lists the locations observed, and modeled data were used for the hydrodynamics calibration. A total number of 23 locations for flow and 19 locations for stage were selected to evaluate the performance of DSM2-HYDRO in simulating flow and stage. In addition, model-simulated cross-Delta flow (calculated as the flow difference between RSAC128 and RSAC123) is also compared with the corresponding observations. These locations are also illustrated in Figure 1-1.

Table 1-3 **List of locations data was used in hydrodynamics (flow and stage) calibration**

DSM2 ID	CDEC ID	Location Name	Flow	Stage
RSAC155	FPT	Sacramento River at Freeport	X	X
RSAC128	SDC	Sacramento River above Delta Cross Channel	X	X
RSAC123	GES	Sacramento River below Georgiana Slough	X	—
RSAC101	SRV	Sacramento River at Rio Vista	X	X
RSAN087	MSD	San Joaquin River at Mossdale	—	X
RSAN072	BDT	San Joaquin River at Brant Bridge	X	—
RSAN063	SJG	San Joaquin River at Stockton	—	X
RSAN058	RRI	Rough and Ready Island	X	X
SLTRM004	TSL	Three Mile Slough	X	X
RSAN018	SJJ	San Joaquin River at Jersey Point	X	X
RSAN007	ANH	San Joaquin River at Antioch	—	X
ROLD074	OH1	Old River at Head	X	X
ROLD059	OLD	Old River at Tracy Road Bridge	—	X
ROLD047	OAD	Old river near Delta Mendota Canal	—	X
ROLD034	OH4	Old River at Highway 4	X	X
ROLD024	OBI	Old River at Bacon Island	X	X
HLT_159	HLT	Middle River near Holt	X	—
CHGRL009	GCT	Grant Line Canal at Tracy Boulevard Bridge	—	X
Georg_SL	GSS	Georgiana Slough at Sacramento River	X	X
SLMZU011	BDL	Montezuma Slough at Beldon Landing	—	X
SLDUT007	DSJ	Dutch Slough	X	X
SLMZU025	NSL	Montezuma Slough at National Steel	X	X
FAL	FAL	False River near Oakley	X	—
HOL	HOL	Holland Cut near Bethel Island	X	—
HWB	HWB	Miner Slough at HWY 84 Bridge	X	—
MOK	MOK	Mokelumne River at San Joaquin River	X	—
SSS	SSS	Steamboat Slough	X	—
SUT	SUT	Sutter Slough at Courtland	X	—
TRN	TRN	Turner Cut near Holt	X	—
CHVCT000	VCU	Victoria Canal near Byron	X	—
DCC	—	Cross Delta Flow (RSAC128 - RSAC123)	X	—

Figure 1-1 Schematic showing locations of modeled and observed data used in hydrodynamics calibration and validation



1.3.3 Calibration and validation results

1.3.3.1 Flow calibration

This section describes flow calibration results at five selected locations. These locations include three in the Sacramento River (at Rio Vista, Georgiana Slough, and Delta Cross Channel) and two in the San Joaquin River (at Three Mile Slough and Jersey Point). Figure 1-2 through Figure 1-6 show the calibration metrics at these locations, respectively. The calibration metrics of the previously calibrated DSM2 (V8.1.2) model are included for reference. The results for the remaining flow calibration stations are presented in a separate technical report (California Department of Water Resources 2021).

For Sacramento River locations (Figures 1-2 to 1-3), both the current calibration (V8.2.0) and the previous calibration (V8.1.2) well simulated the variation pattern, magnitude, and phase in observed flows. The mean errors and RMSE values of the tidally averaged flow simulations of the current calibration were consistently smaller than those in the previous calibration. The most significant difference in mean error was in cross-Delta flow (calculated as the difference in the Sacramento River flow above Delta Cross Channel and below Georgiana Slough) (Figure 1-4), where the error of the current calibration (17.7 cfs) was about half of that in the previous calibration (31.8 cfs). The most significant improvement (13 percent smaller) in RMSE was in flow at the Georgiana Slough at Sacramento River (Figure 1-3). The phase error distribution patterns of both calibration efforts were very similar, with the current calibration yielding slightly smaller mean phase errors. The distribution patterns of amplitude error were also similar from the two calibration efforts; however, the mean amplitude errors were noticeably different. For Sacramento River at Rio Vista, the previous calibration had smaller amplitude error on average. For Georgiana Slough, the current calibration yielded smaller mean errors from the observed.

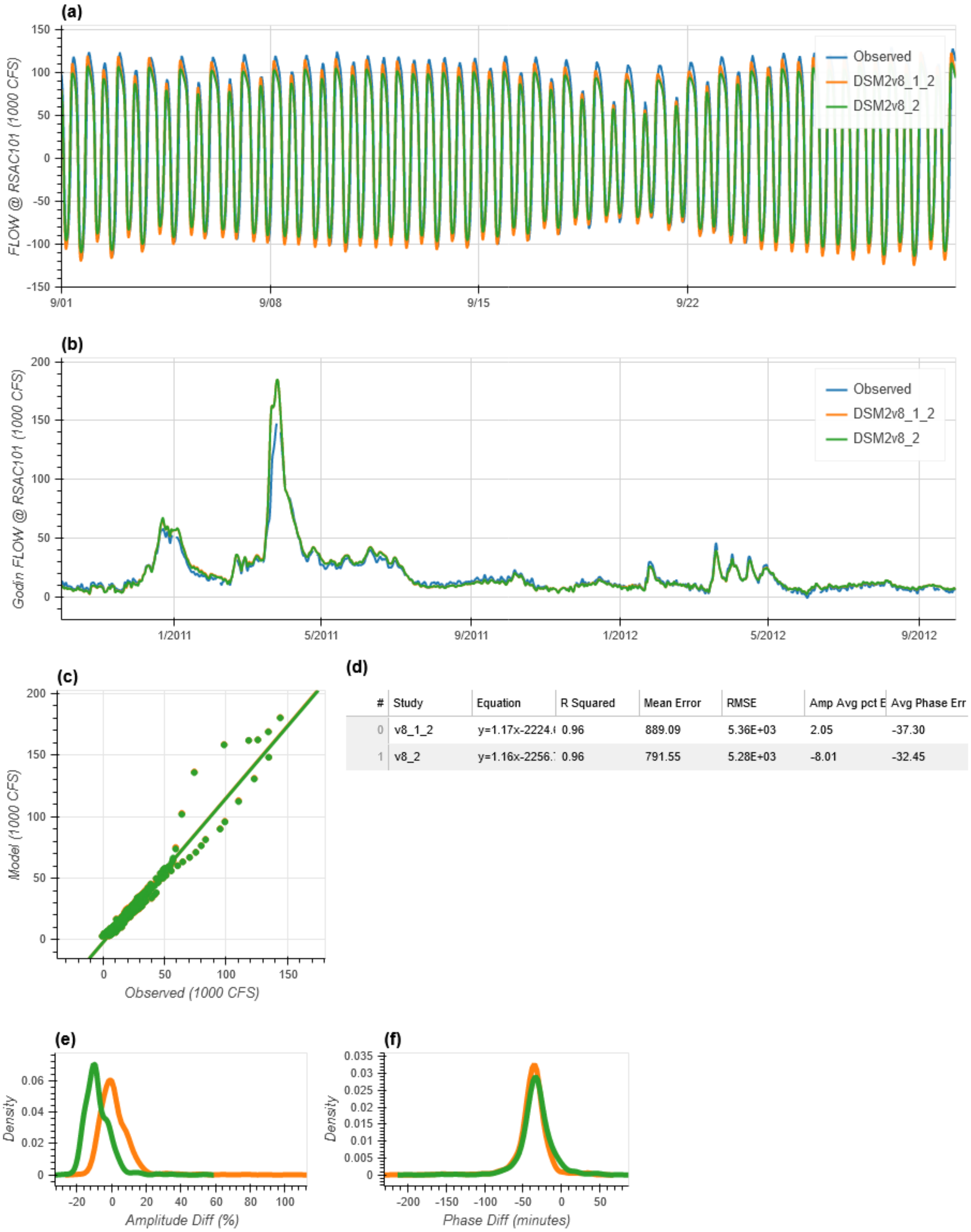
Similarly, the current calibration yielded flow simulations at two San Joaquin River locations very close to those of the previous calibration (Figures 1-5 and 1-6). These flows well mimicked the observations on both tidal scale and daily scale; however, calibrated V8.1.2 and V8.2.0 performed differently at these two locations in terms of statistical metrics derived from tidally averaged flows. For Three Mile Slough (Figure 1-5), flows under the current calibration had a smaller mean error, smaller average mean amplitude and phase errors, but a higher RMSE compared to the previous calibration. Flows in the San Joaquin River at Jersey Point showed opposite trends (Figure 1-6).

Among all five locations, Three Mile Slough was the only location where tidally averaged flow simulations deviated notably from the corresponding observations. The linear relationships (with R^2 less than 0.6) between observed and simulated flow from both calibration efforts at this location were considerably weaker than at other selected locations (with R^2 over 0.9). This is partly a result of net tidally averaged flows being much smaller than tidal flows at Three Mile Slough. Looking at the flow time-history, the largest discrepancy in flow apparently corresponded to the extreme high flow period during Spring 2011. Flow at Rio Vista showed a similar discrepancy during the same period. This could indicate an issue with the flow boundary conditions used and this issue needs to be revisited in future DSM2 model enhancement and calibration efforts.

Overall, the current calibration yielded satisfactory flow simulations at the selected locations. Compared with the previous calibration, the current calibration led to improvements in modeled flow at most of these locations; however, model performance at some locations (e.g., Three Mile Slough) needs to be improved in future model enhancement and calibration efforts.

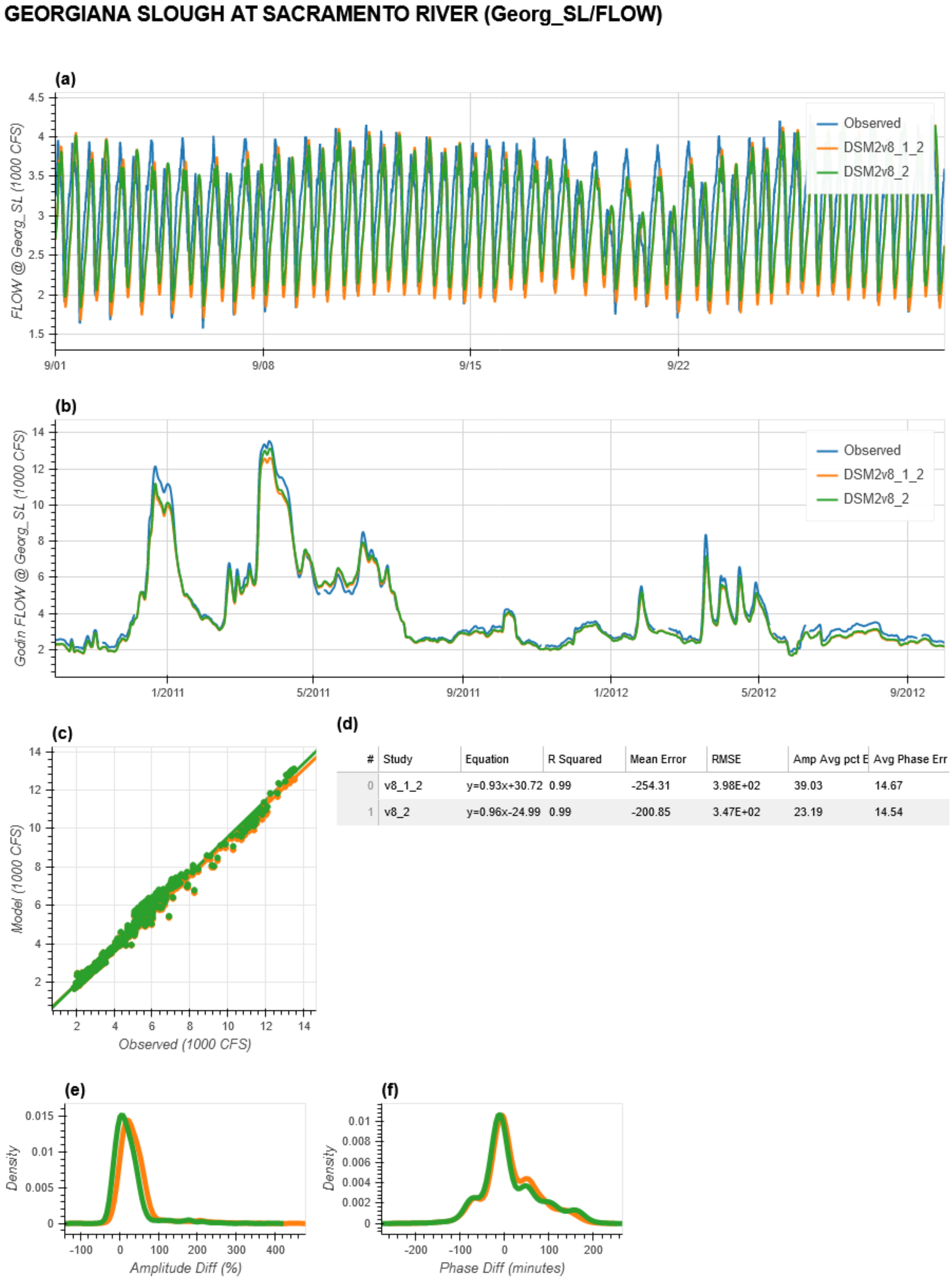
Figure 1-2 Flow calibration metrics for Sacramento River at Rio Vista

SACRAMENTO RIVER AT RIO VISTA (USGS) (RSAC101/FLOW)



#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=1.17x-2224.	0.96	889.09	5.36E+03	2.05	-37.30
1	v8_2	y=1.16x-2256.	0.96	791.55	5.28E+03	-8.01	-32.45

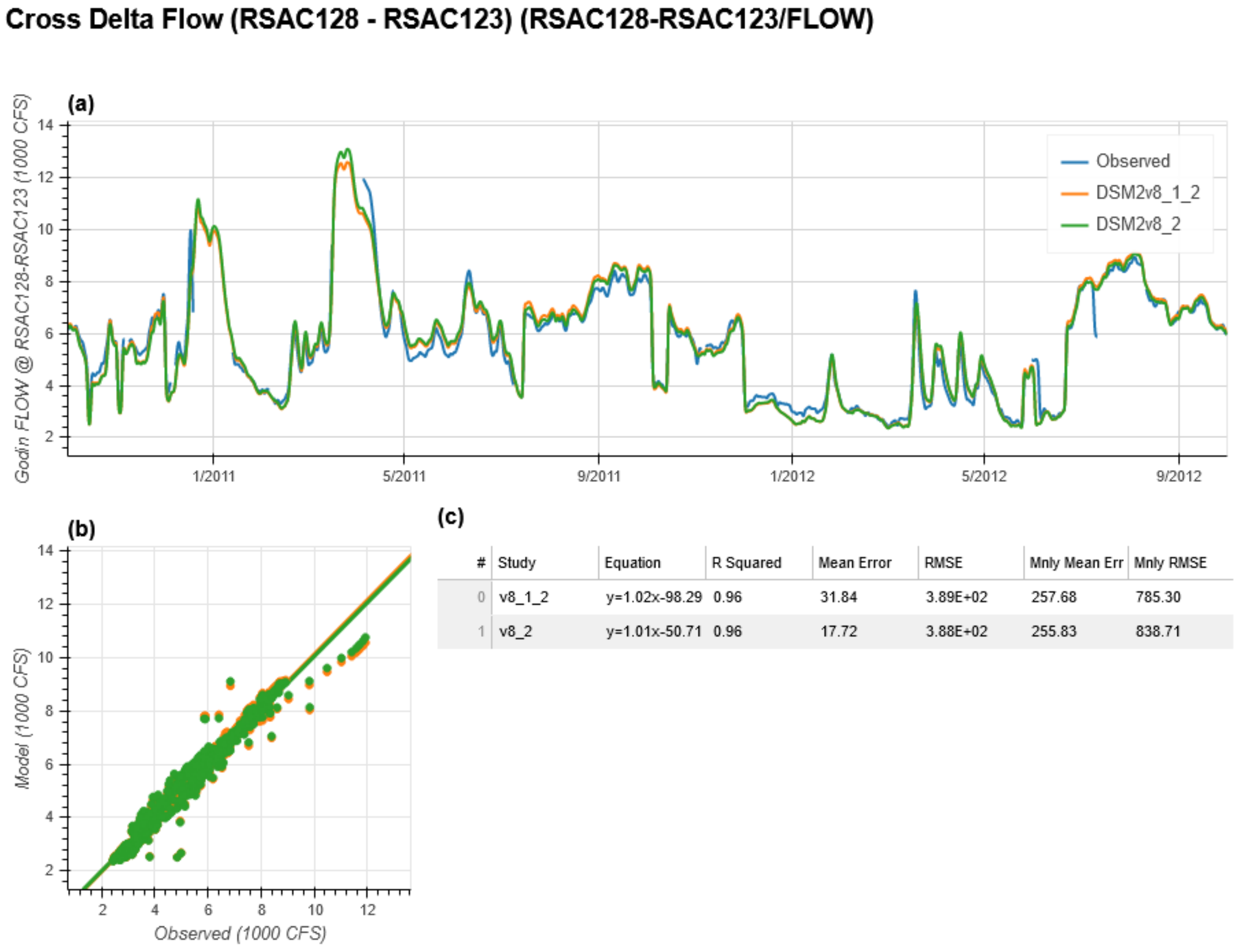
Figure 1-3 Flow calibration metrics for Georgiana Slough



GEORGIANA SLOUGH AT SACRAMENTO RIVER (Georg_SL/FLOW)

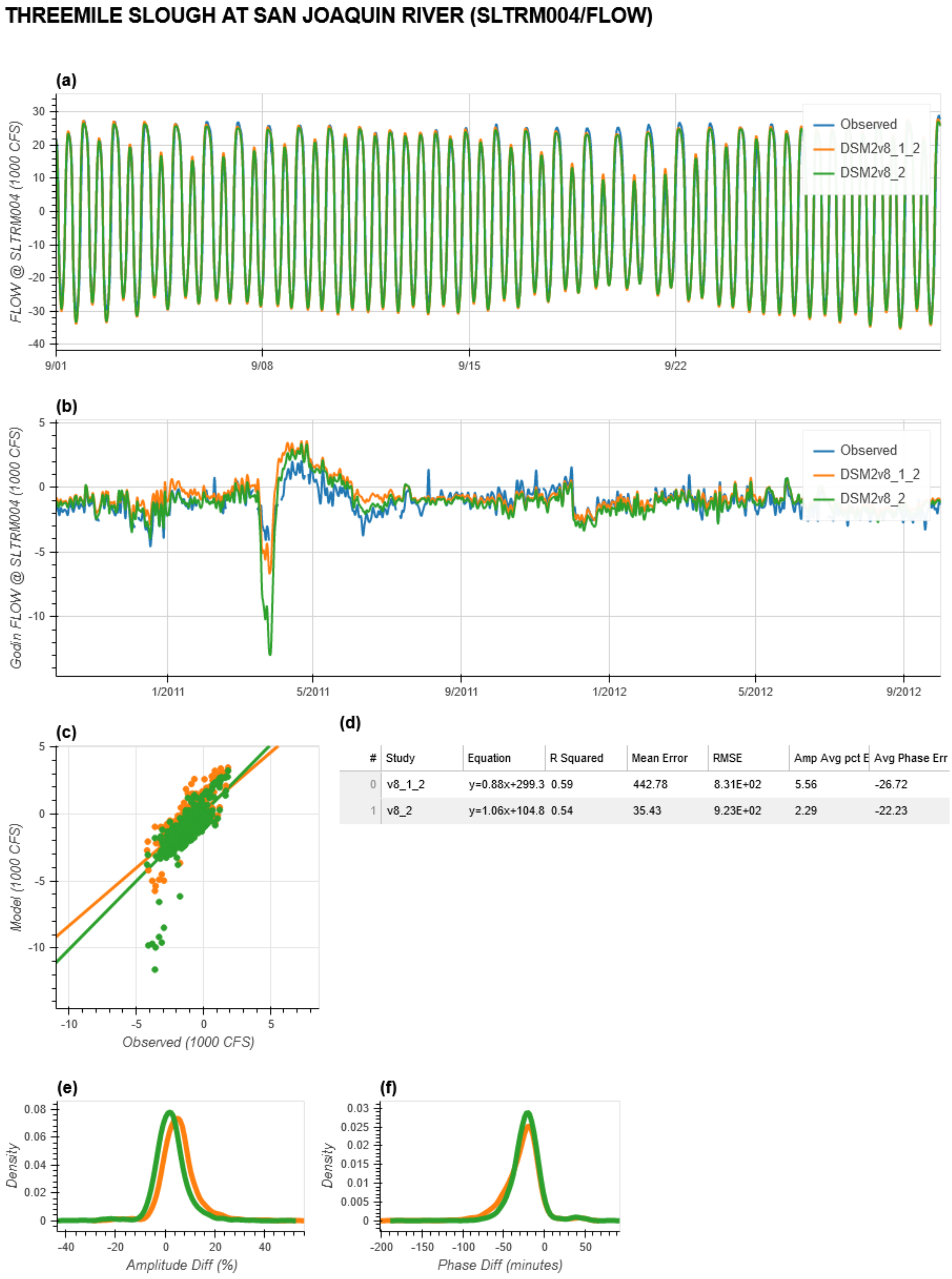
#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.93x+30.72	0.99	-254.31	3.98E+02	39.03	14.67
1	v8_2	y=0.96x-24.99	0.99	-200.85	3.47E+02	23.19	14.54

Figure 1-4 Flow calibration metrics for Cross Delta Flow


Cross Delta Flow (RSAC128 - RSAC123) (RSAC128-RSAC123/FLOW)

#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=1.02x-98.29	0.96	31.84	3.89E+02	257.68	785.30
1	v8_2	y=1.01x-50.71	0.96	17.72	3.88E+02	255.83	838.71

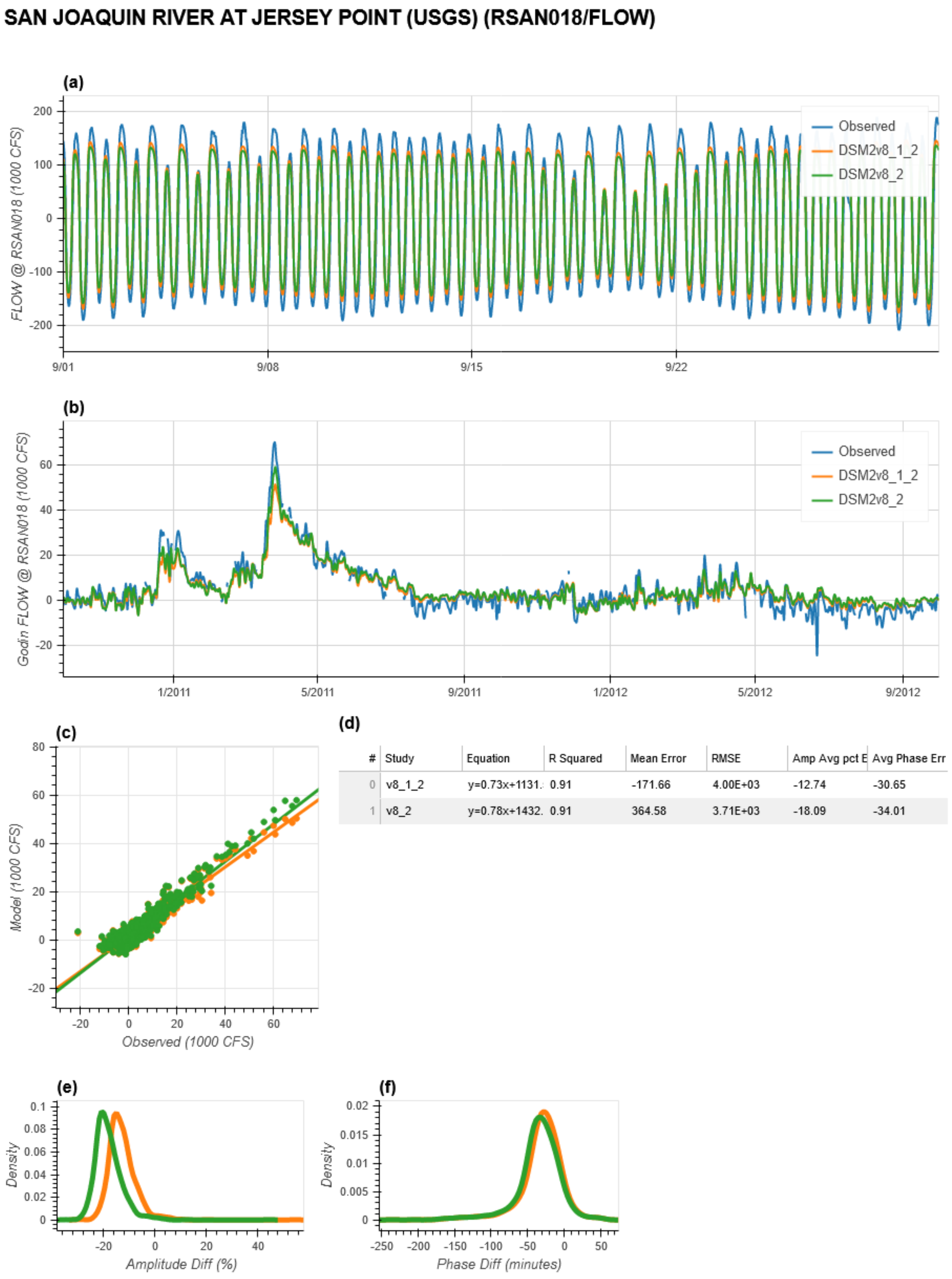
Figure 1-5 Flow calibration metrics for Three Mile Slough



THREEMILE SLOUGH AT SAN JOAQUIN RIVER (SLTRM004/FLOW)

#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.88x+299.3	0.59	442.78	8.31E+02	5.56	-26.72
1	v8_2	y=1.06x+104.8	0.54	35.43	9.23E+02	2.29	-22.23

Figure 1-6 Flow calibration metrics for San Joaquin River at Jersey Point



#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.73x+1131.:	0.91	-171.66	4.00E+03	-12.74	-30.65
1	v8_2	y=0.78x+1432.	0.91	364.58	3.71E+03	-18.09	-34.01

1.3.3.2 Stage calibration

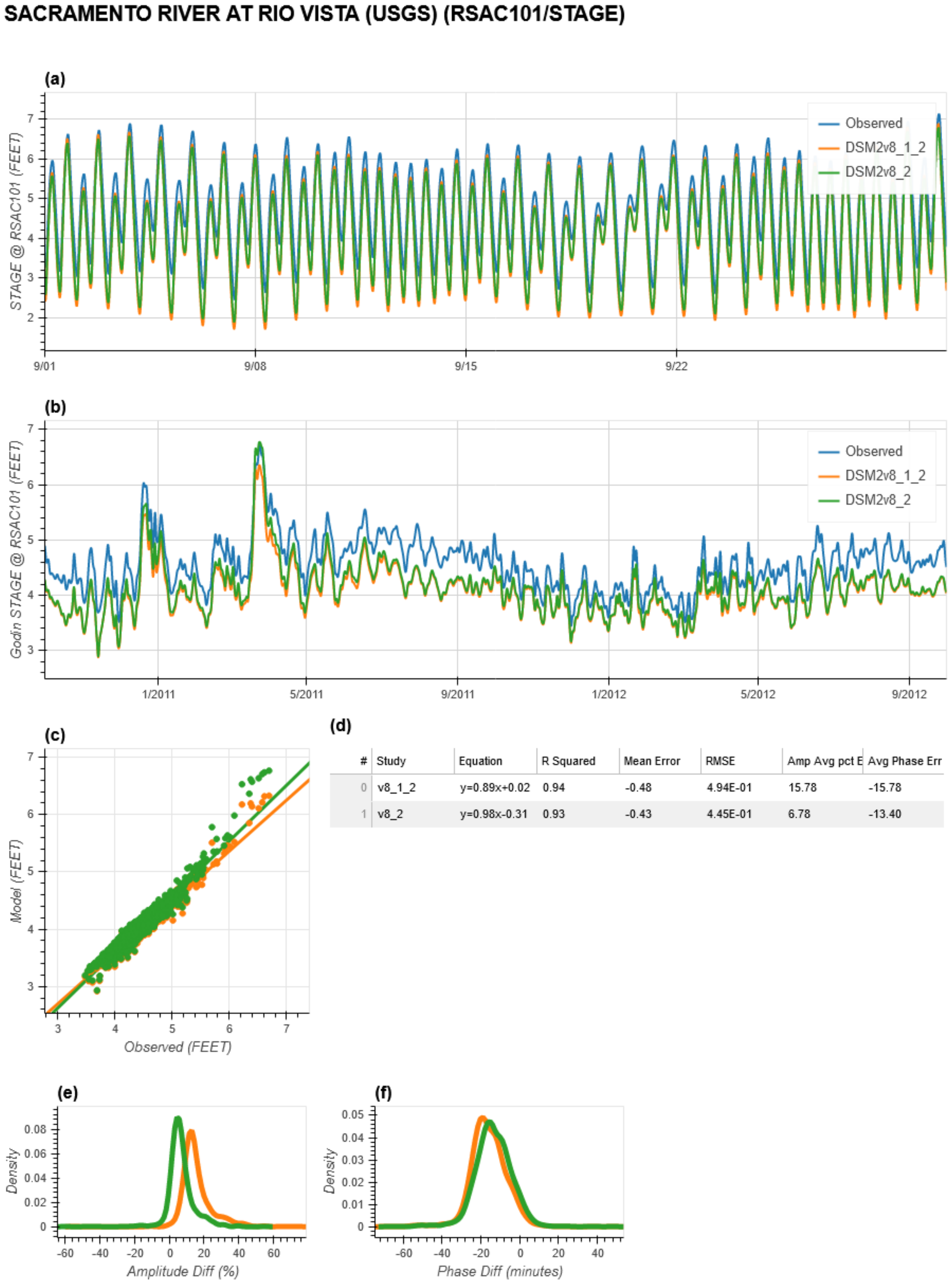
Stage calibration results at four locations are presented. One flow calibration location (Sacramento River at the Delta Cross Channel) is not included here as its corresponding flow (cross-Delta flow) is derived from flows at two different locations. Results for other stage calibration locations are provided in a separate technical report (California Department of Water Resources 2021).

Figures 1-7 and 1-8 show stage calibration metrics for Sacramento River at Rio Vista and Georgiana Slough, respectively. Both the current calibration (V8.2.0) and the previous calibration (V8.1.2) simulated the tidal variation pattern well (panel (a) of Figure 1-7) but under-estimated the observed stage (panel (b)) at both locations. In comparison, the current calibration better matched peak stage at Rio Vista (panel (b)). On a daily scale (panel (c)), both calibration efforts yielded stage simulations highly correlated with observed stage, with R^2 over 0.92 at both locations. In terms of statistical metrics, the mean error and RMSE for Rio Vista were smaller in the current calibration than in the previous one. The average amplitude error and phase error were also smaller in the current calibration. For Georgiana Slough, though, the mean error and RMSE for the current calibration were slightly higher than for the previous calibration.

Similar to the performance results at Sacramento River locations, stage simulations from both calibration efforts at both San Joaquin River locations faithfully mimicked but under-estimated the observed stage (panels (a) and (b) in Figures 1-9 and 1-10). On a daily scale (panel (c)), the tidally averaged simulated stage aligned very well with the observed stage. The R^2 values between the observed stage and the corresponding stage simulations from both calibration efforts at both locations were around 0.95. Both calibration efforts had nearly identical mean bias and RMSE values at two locations, but the current calibration had smaller amplitude error on average. In terms of phase error, though, the previous calibration outperformed the current calibration at both locations.

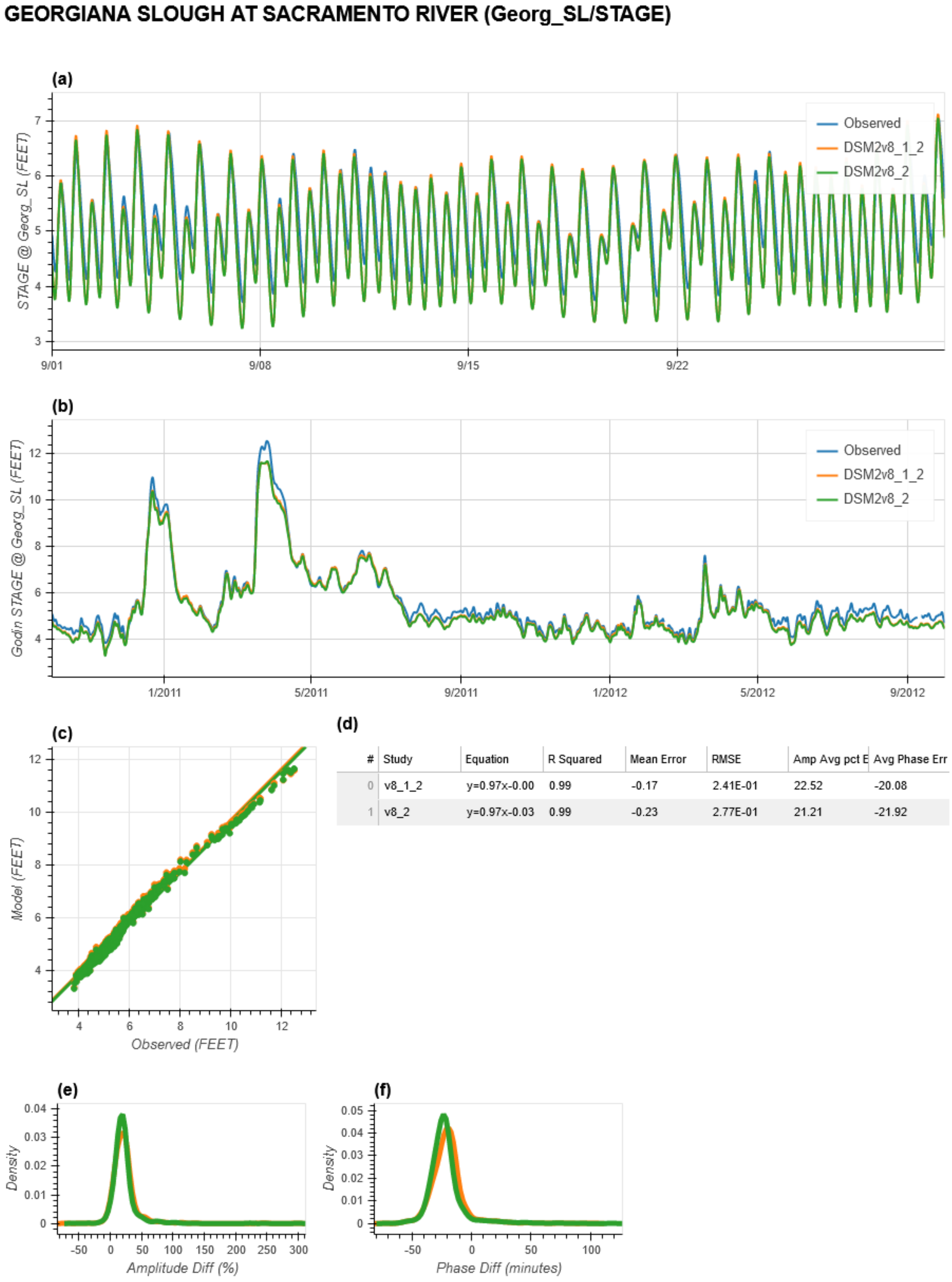
In general, the calibrated model (V8.2.0) was able to simulate the measured stage very well. Its performance was fairly similar or superior to that of the previously calibrated model (V8.1.2). There was a consistent dry bias (under estimation) in the calibrated model, which will need to be addressed in future model development and calibration efforts.

Figure 1-7 Stage calibration metrics for Sacramento River at Rio Vista



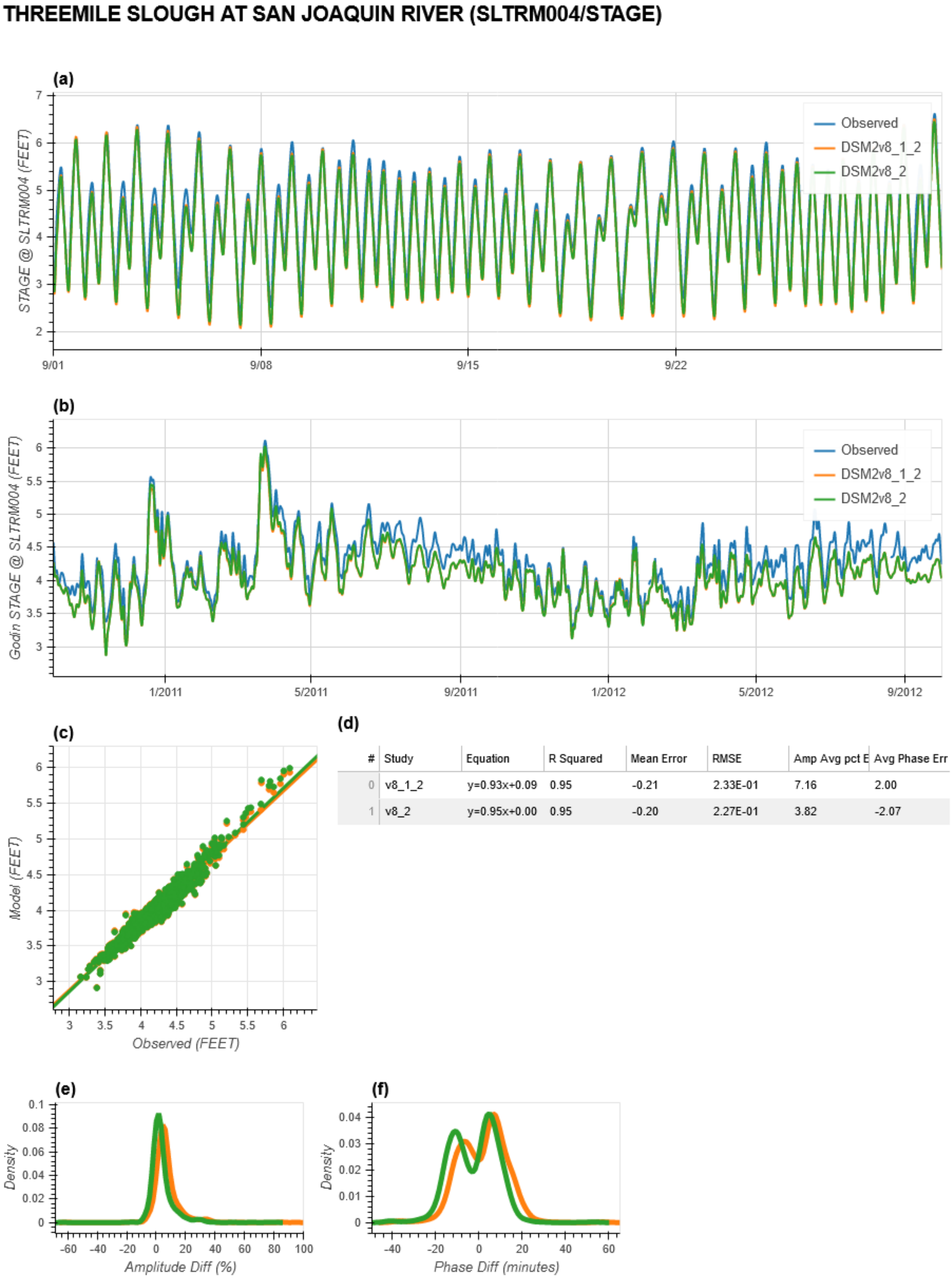
#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.89x+0.02	0.94	-0.48	4.94E-01	15.78	-15.78
1	v8_2	y=0.98x-0.31	0.93	-0.43	4.45E-01	6.78	-13.40

Figure 1-8 Stage calibration metrics for Georgiana Slough


GEORGIANA SLOUGH AT SACRAMENTO RIVER (Georg_SL/STAGE)

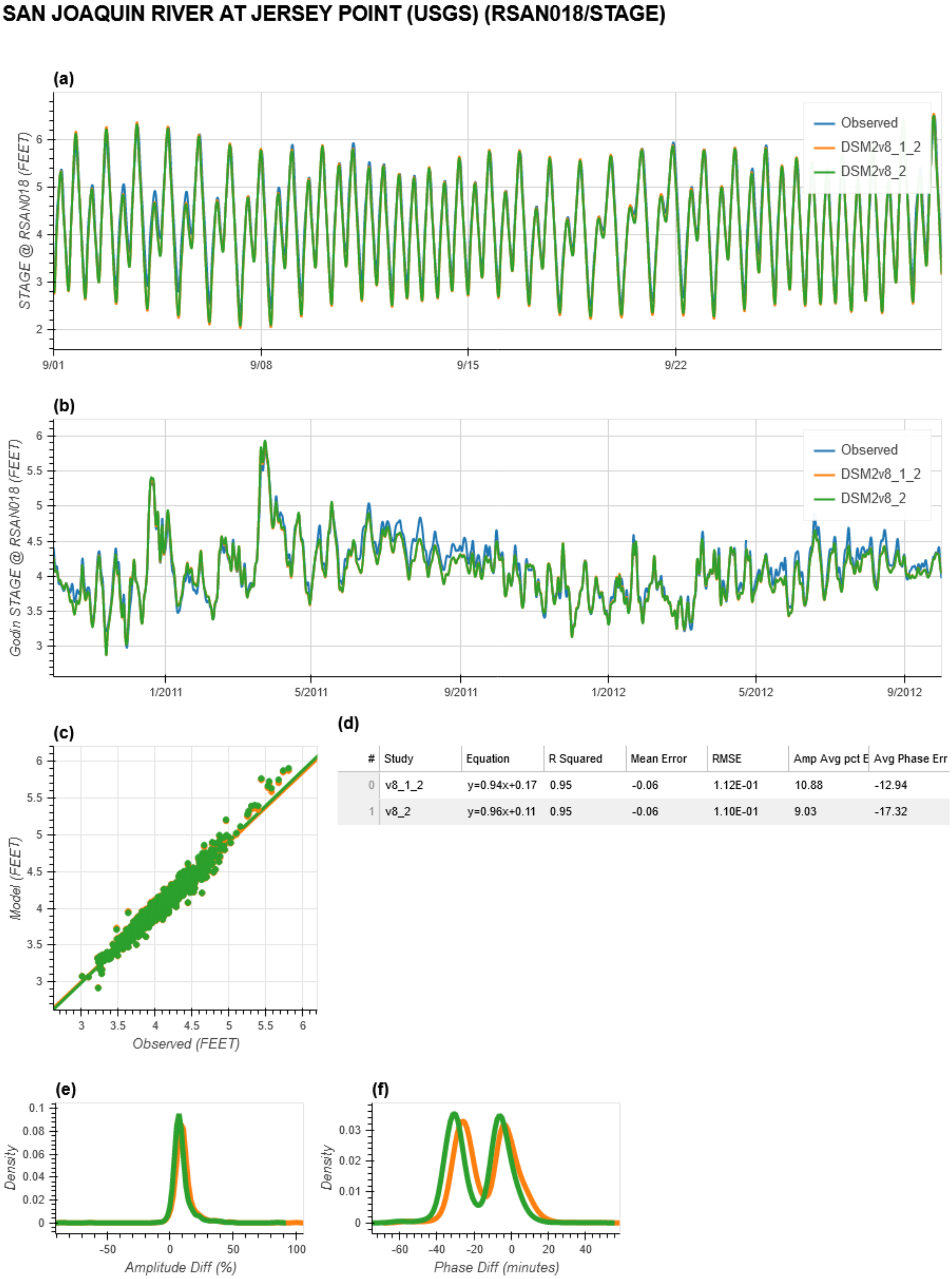
#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.97x-0.00	0.99	-0.17	2.41E-01	22.52	-20.08
1	v8_2	y=0.97x-0.03	0.99	-0.23	2.77E-01	21.21	-21.92

Figure 1-9 Stage calibration metrics for Three Mile Slough



#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.93x+0.09	0.95	-0.21	2.33E-01	7.16	2.00
1	v8_2	y=0.95x+0.00	0.95	-0.20	2.27E-01	3.82	-2.07

Figure 1-10 Stage calibration metrics for San Joaquin River at Jersey Point



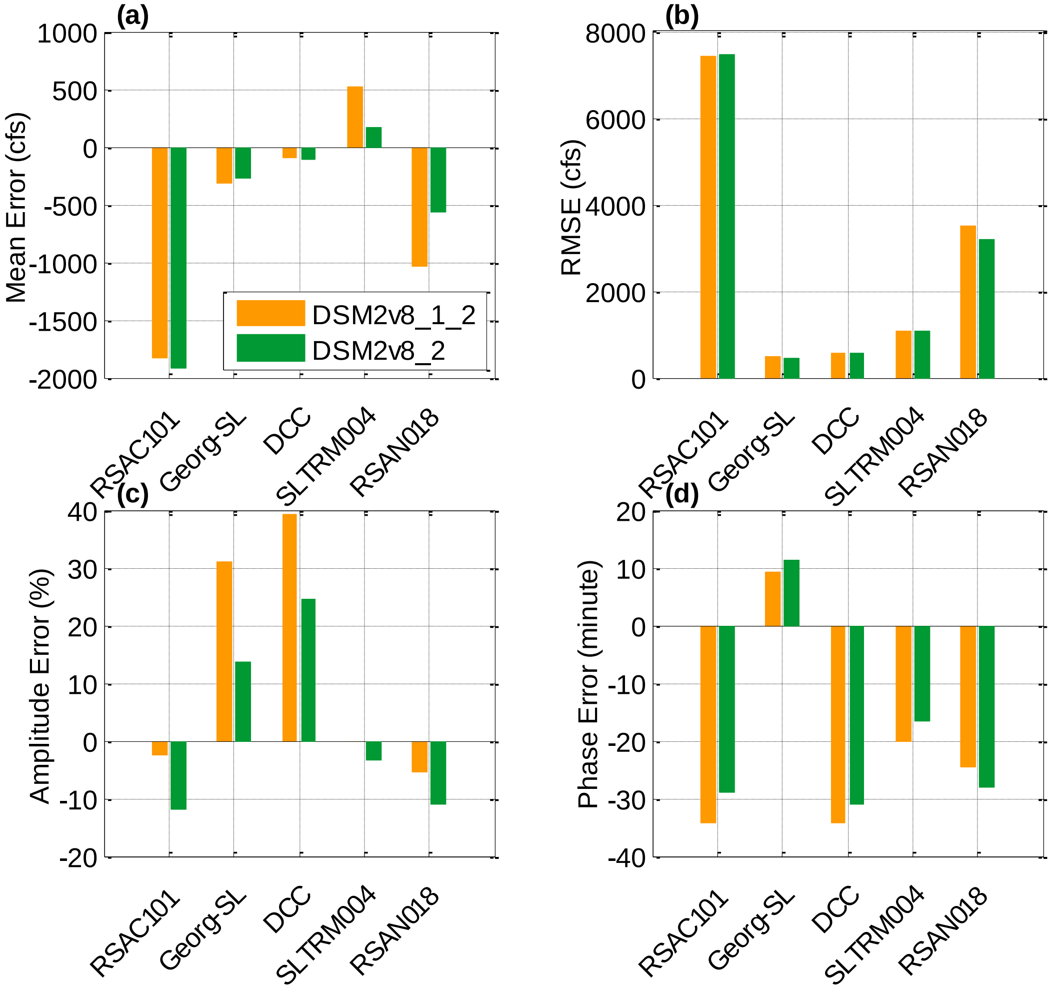
#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg pct E	Avg Phase Err
0	v8_1_2	y=0.94x+0.17	0.95	-0.06	1.12E-01	10.88	-12.94
1	v8_2	y=0.96x+0.11	0.95	-0.06	1.10E-01	9.03	-17.32

1.3.3.3 Flow and stage validation

This section describes statistical metrics of flow and stage simulations during the validation periods at the key locations illustrated in the previous two sections. Other metrics, including time series plot, scatter plot, distribution curves of amplitude, and phase difference are provided in a separate technical report (California Department of Water Resources 2021), along with validation results at other locations.

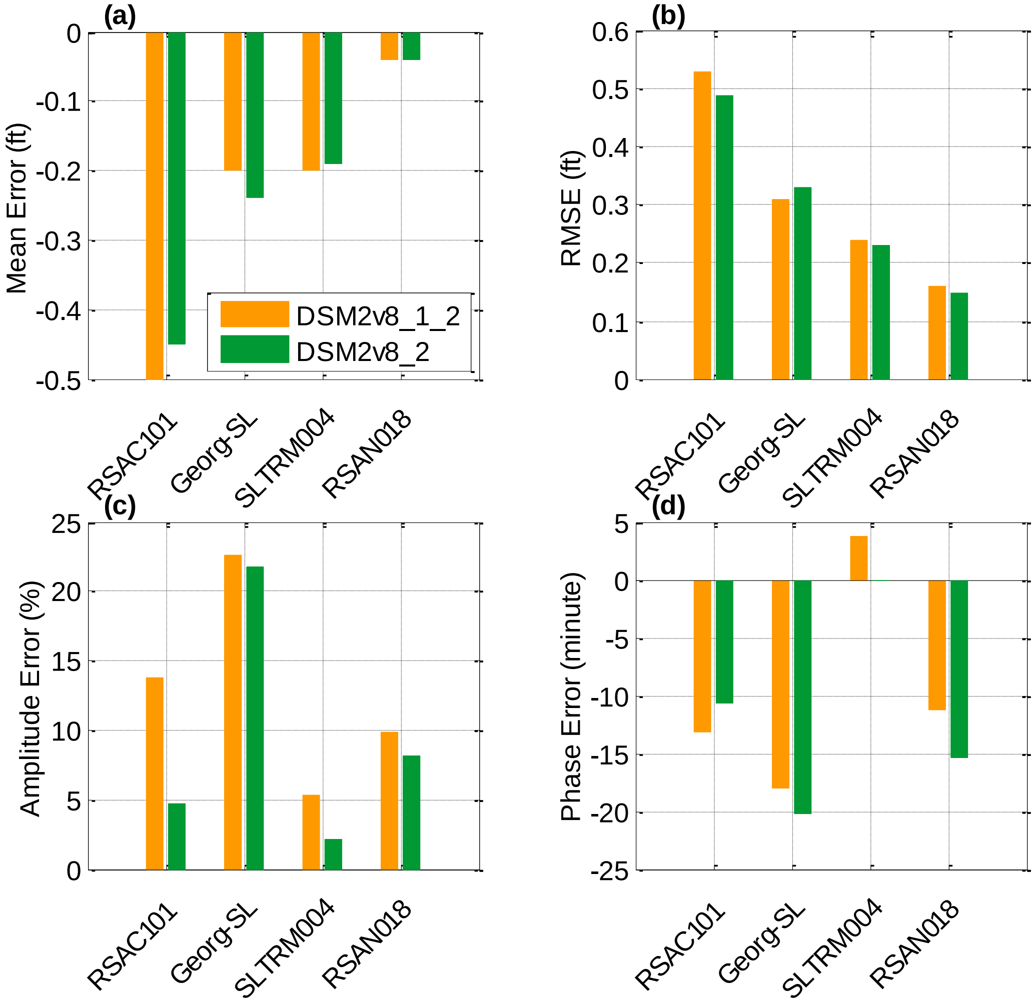
Figure 1-11 presents the mean error (a), RMSE (b), average amplitude error (c), and average phase error (d) between simulated and observed flows at five selected locations during the validation period. The current calibration clearly yielded generally more desirable mean error and RMSE. The only exception was that the mean error and RMSE of the current calibration at Rio Vista were slightly larger than those from the previous calibration. At locations where the observed amplitude was over-estimated (Georgiana Slough and cross Delta Cross Channel), the current calibration had considerably smaller amplitude error; however, at the other three locations where the amplitude was under-estimated in both calibration efforts, the current calibration had a notably larger error. In terms of phase error, the current calibration had smaller errors at three out of five locations.

Figure 1-11 Summary of statistical metrics of flow validation



Stage validation metrics at selected locations are illustrated in Figure 1-12. Similar to what was noted during the calibration period, both calibration efforts (V8.1.2 and V8.2.0) underestimated the observed stage during the validation period as well. In comparison, the current calibration had a smaller mean error and RMSE at all locations except for Georgiana Slough. Additionally, the current calibration had smaller amplitude errors at all four locations; however, the results were mixed for phase error. The differences in average phase error in both calibration efforts were generally small. The largest difference was about four minutes (at Three Mile Slough), while the smallest difference was about two minutes (at Georgiana Slough).

Figure 1-12 Summary of statistical metrics of stage validation



Overall, flow and stage simulations from the calibrated V8.2.0 during the validation period were generally in line with the corresponding observations at selected stations. These simulations were generally comparable or superior to those from the calibrated V8.1.2.

1.4 Water quality calibration and validation

1.4.1 Calibration parameter

The channel dispersion factor was the main parameter of QUAL that was calibrated. The factor is defined as the dispersion-to-advection ratio in a channel. A higher value of the dispersion factor indicates higher mixing and thus faster salinity transport. Based on the flow field simulated by the calibrated HYDRO, modifications of QUAL dispersion factors were required in only a limited number of channels to yield desirable salinity simulations in the current calibration. Table 1-4 tabulates the IDs of these channels as well as the corresponding modifications in dispersion factors.

Table 1-4 List of channels with dispersion factors modified in the current calibration

Group Name	Channel Number	Before Calibration	After Calibration
South Delta	276–279	720	900
North Delta	47	360	1000
North Delta	48	260	1000
North Delta	309–310	360	720
North Delta	430–432	260	120
North Delta	433	700	540
North Delta	434–435	1000	900

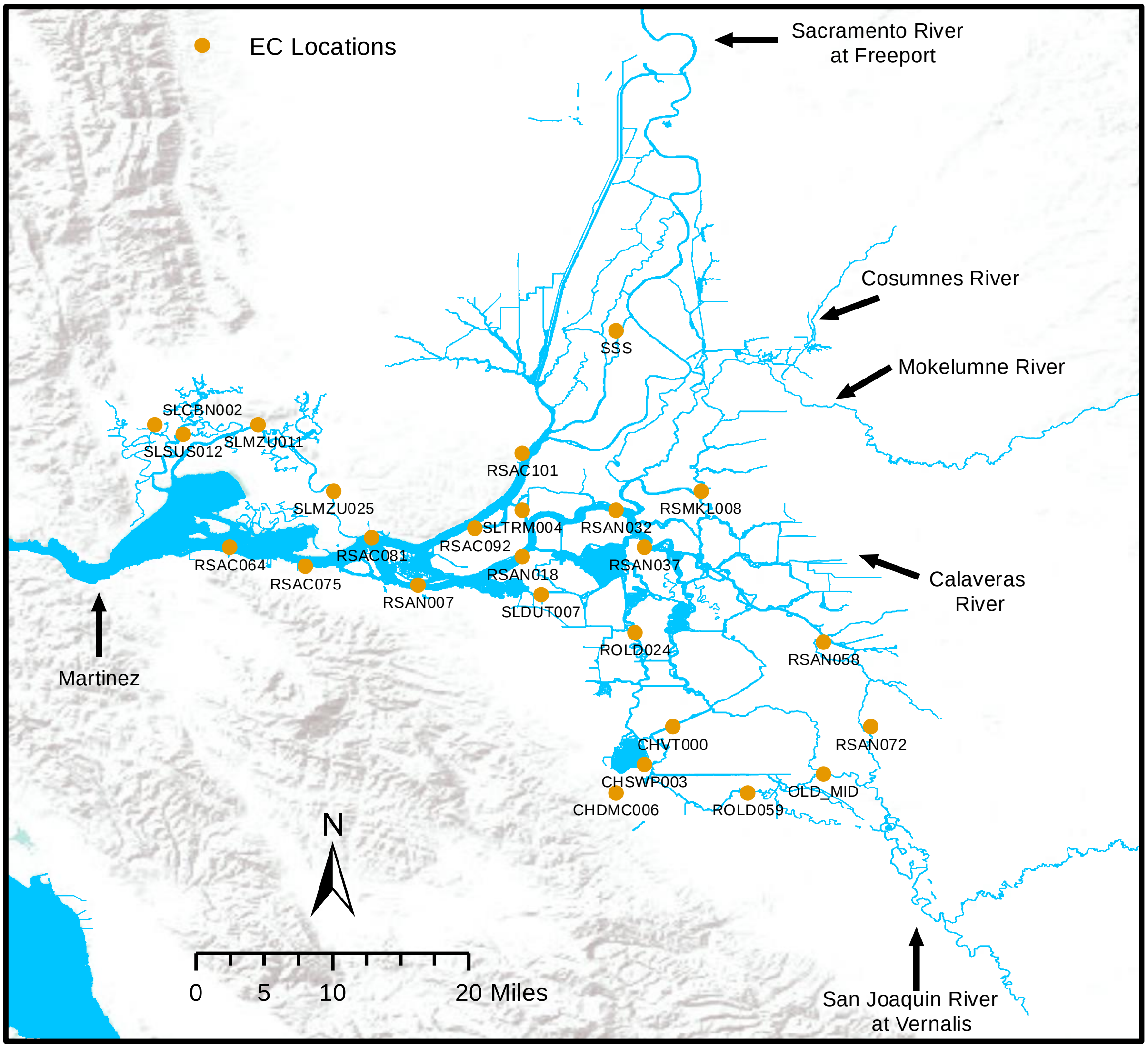
1.4.2 Calibration and validation stations

A total number of 25 locations were selected in assessing DSM2-QUAL's performance in simulating EC. These locations are listed in Table 1-5 and illustrated in Figure 1-13.

Table 1-5 List of EC calibration locations

DSM2 ID	CDEC ID	Location Name
RSAC101	RVB	Sacramento River at Rio Vista
RSAC092	EMM	Sacramento River at Emmaton
RSAC081	CLL	Sacramento River at Collinsville
SLTRM004	TSL	Three-Mile Slough
RSAC075	MAL	Sacramento River at Mallard Island
RSAC064	PCT	Sacramento River at Port Chicago
RSAN072	BDT	San Joaquin River at Brandt Bridge
RSMKL008	STI	South Fork Mokelumne River at Terminous
RSAN037	PRI	San Joaquin River at Prisoners Point
RSAN032	SAL	San Joaquin River at San Andreas Landing
RSAN018	SJJ	San Joaquin River at Jersey Point
RSAN007	ANC	San Joaquin River at Antioch
OLD_MID	UNI	Old River near Middle River
ROLD059	OLD	Old River at Tracy Road Bridge
SLDUT007	DSJ	Dutch Slough
CHVT000	VCU	Victoria Canal near Byron
CHSWP003	CLC	Banks Pumping Plant /Clifton Court Forebay
CHDMC006	DMC	Jones Pumping Plant
ROLD024	OBI	Old River at Bacon Island
SLMZU025	NSL	Montezuma Slough at National Steel
SLMZU011	BDL	Montezuma Slough at Beldon Landing
SLCBN002	SNC	Chadbourne Slough near Sunrise Duck Club
SLSUS012	VOL	Suisun Slough 300 ft south of Volanti Slough
RSAN058	RRI	Rough and Ready Island (SJR)
SSS	SSS	Steamboat Slough

Figure 1-13 Schematic showing EC calibration and validation locations



1.4.3 Calibration and validation results

1.4.3.1 Salinity calibration

The calibration of QUAL in terms of simulated EC focused on several key locations in the Delta, including Emmaton, Jersey Point, Rio Vista, Antioch, Old River at Bacon Island, Banks Pumping Plant, and Jones Pumping Plant. Calibration metrics, including time series plot of Godin-filtered EC, scatter plot of tidally averaged EC, and mean error and RMSE of tidally averaged and monthly EC at these locations are illustrated in Figures 1-14 to 1-20. Calibration metrics at other EC calibration locations are documented in a separate technical report (California Department of Water resources 2021).

At Emmaton, both V8.1.2 and V8.2.0 under-estimated EC observations (Figure 1-14). The under estimation was particularly evident during the period of 2013–2015, which contains two critical years (2014–2015) and one dry year (2013). During this period, V8.1.2 largely under-simulated the high range salinity while V8.2.0 produced significantly closer simulations. Over the entire calibration period, the mean error of V8.1.2 on a daily scale was about -200 microsiemens per centimeter (μs/cm) (versus -37 μs/cm of V8.2.0). The RMSE (about 646 μs/cm) of V8.1.2 was also notably larger than that of V8.2.0 (428 μs/cm). This was also the case on the monthly scale. The tidally averaged EC simulations of V8.2.0 also aligned better with observations. The R^2 of V8.2.0 was 0.92 (versus 0.85 of V8.1.2). Overall, the current calibration led to improved EC simulations at Emmaton when compared with the previous calibration. Similarly, the current calibration improved EC simulations at Rio Vista when compared with the previous calibration (Figure 1-15). The mean error and RMSE of the current calibration were smaller than their counterparts in the previous calibration at both daily and monthly scales. In addition, the current calibration better simulated EC during the dry period from 2013–2015. Furthermore, tidally averaged EC from the current calibration also aligned better with the observations and had a higher R^2.

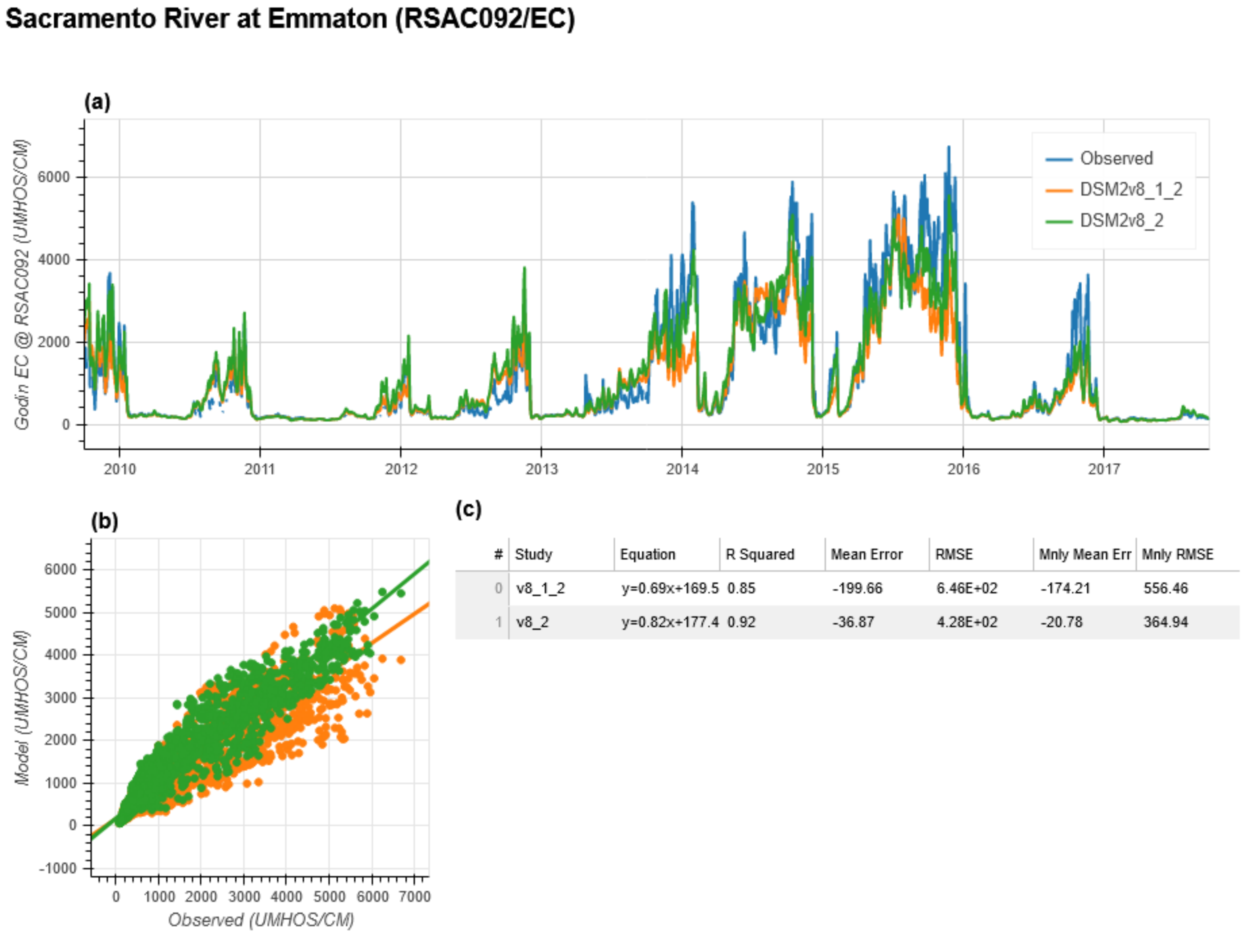
Similar improvements over the previous calibration were also observed at Antioch (Figure 1-16) and Jersey Point (Figure 1-17). The improvements were most notable during the period from 2013–2015, when V8.1.2 largely over or under-estimated the salinity observed in the field. The statistical metrices, including mean error, RMSE, and R^2 of V8.2.0 were also more desirable. Nevertheless, there were periods when both the current calibration and previous calibration performed relatively poorly. For instance, in late 2016, both calibration efforts under simulated the observed EC at both locations. This could also be the result of inaccuracies in the assumed flow boundary conditions and warrants further investigation in future calibration efforts.

For Old River at Bacon Island, both calibration efforts tended to under simulate the measured EC field (Figure 1-18). This under prediction was particularly pronounced in 2012 and 2016. Likely there were local salinity sources (during these two periods at this location) not considered in DSM2. The under prediction was also evident in the scatter plot showing tidally averaged EC simulations and observations (Panel (b)). When comparing the two calibration efforts, though, the current calibration had more satisfactory statistical metrics.

At both pumping plants, EC was also under-predicted in both calibration efforts (Figures 1-19 and 1-20). Compared to the previous calibration, the current calibration yielded consistently better statistical metrics at both locations, with smaller mean error and RMSE as well as higher R^2.

Overall, the current calibration outperformed the previous calibration in terms of providing EC simulations that better matched observed EC at all selected locations; however, there was a dry bias (underestimation) in the calibrated model. This issue will need to be addressed in future model enhancement and calibration efforts.

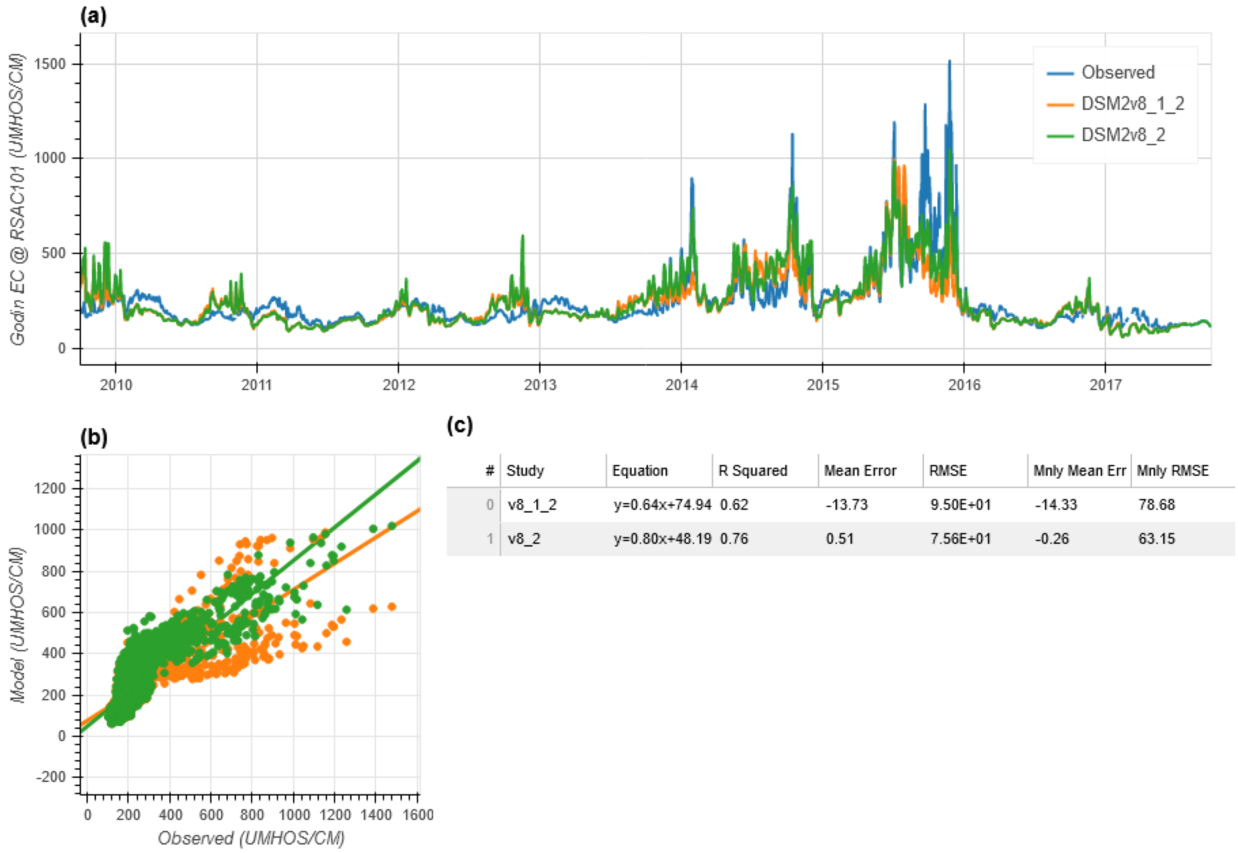
Figure 1-14 EC calibration metrics for Sacramento River at Emmaton



#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.69x+169.5	0.85	-199.66	6.46E+02	-174.21	556.46
1	v8_2	y=0.82x+177.4	0.92	-36.87	4.28E+02	-20.78	364.94

Figure 1-15 EC calibration metrics for Sacramento River at Rio Vista

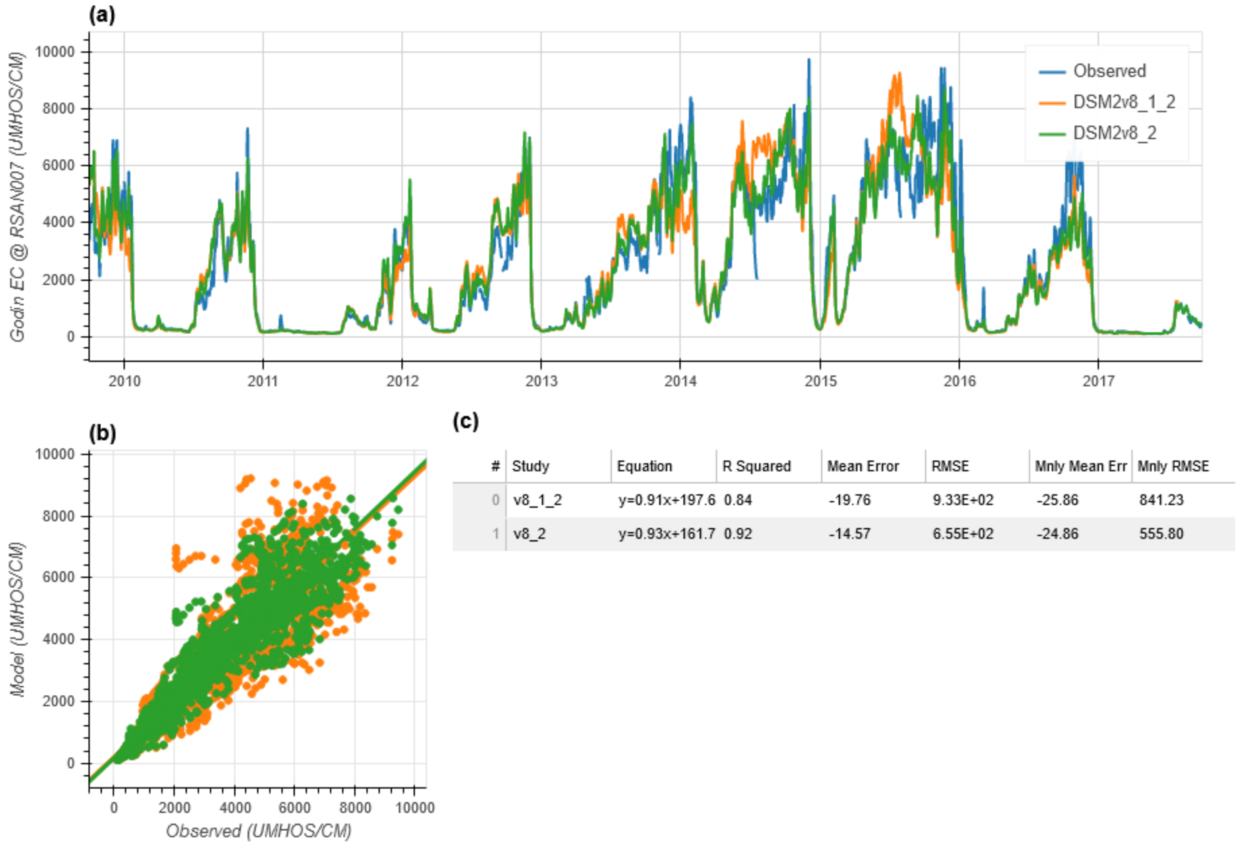
Rio Vista (RSAC101/EC)



#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.64x+74.94	0.62	-13.73	9.50E+01	-14.33	78.68
1	v8_2	y=0.80x+48.19	0.76	0.51	7.56E+01	-0.26	63.15

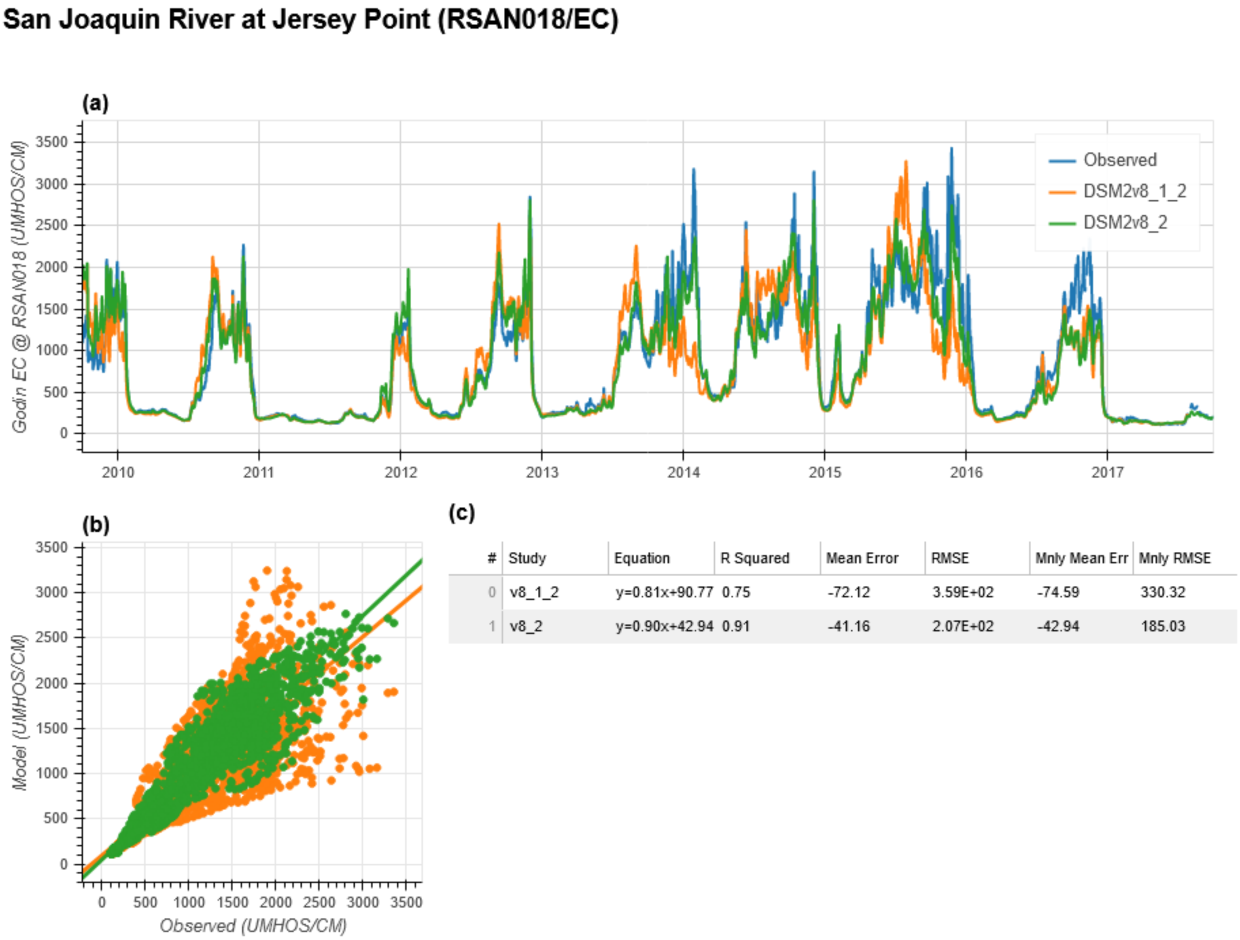
Figure 1-16 EC calibration metrics for San Joaquin River at Antioch

San Joaquin River at Antioch (RSAN007/EC)



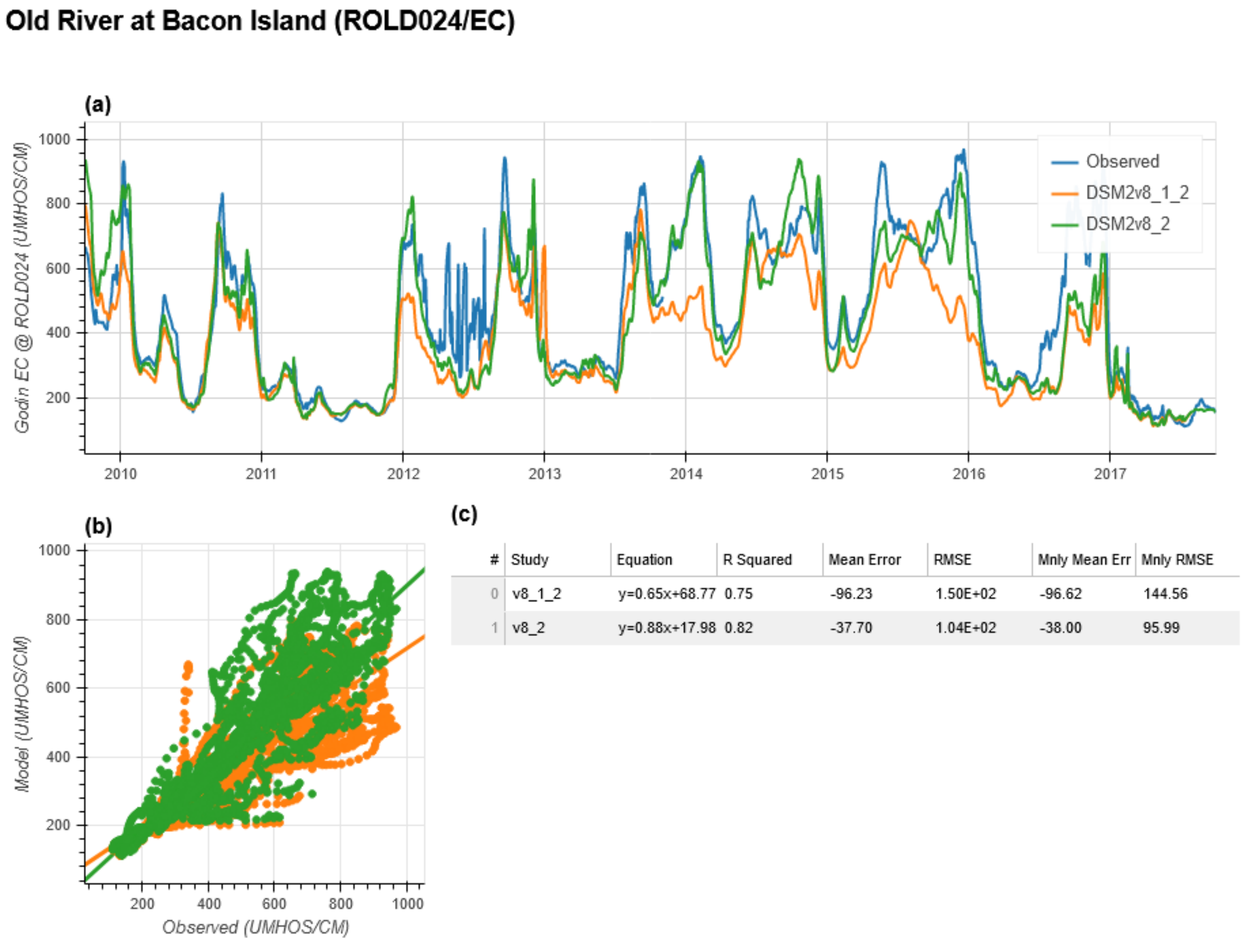
#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.91x+197.6	0.84	-19.76	9.33E+02	-25.86	841.23
1	v8_2	y=0.93x+161.7	0.92	-14.57	6.55E+02	-24.86	555.80

Figure 1-17 EC calibration metrics for San Joaquin River at Jersey Point



#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.81x+90.77	0.75	-72.12	3.59E+02	-74.59	330.32
1	v8_2	y=0.90x+42.94	0.91	-41.16	2.07E+02	-42.94	185.03

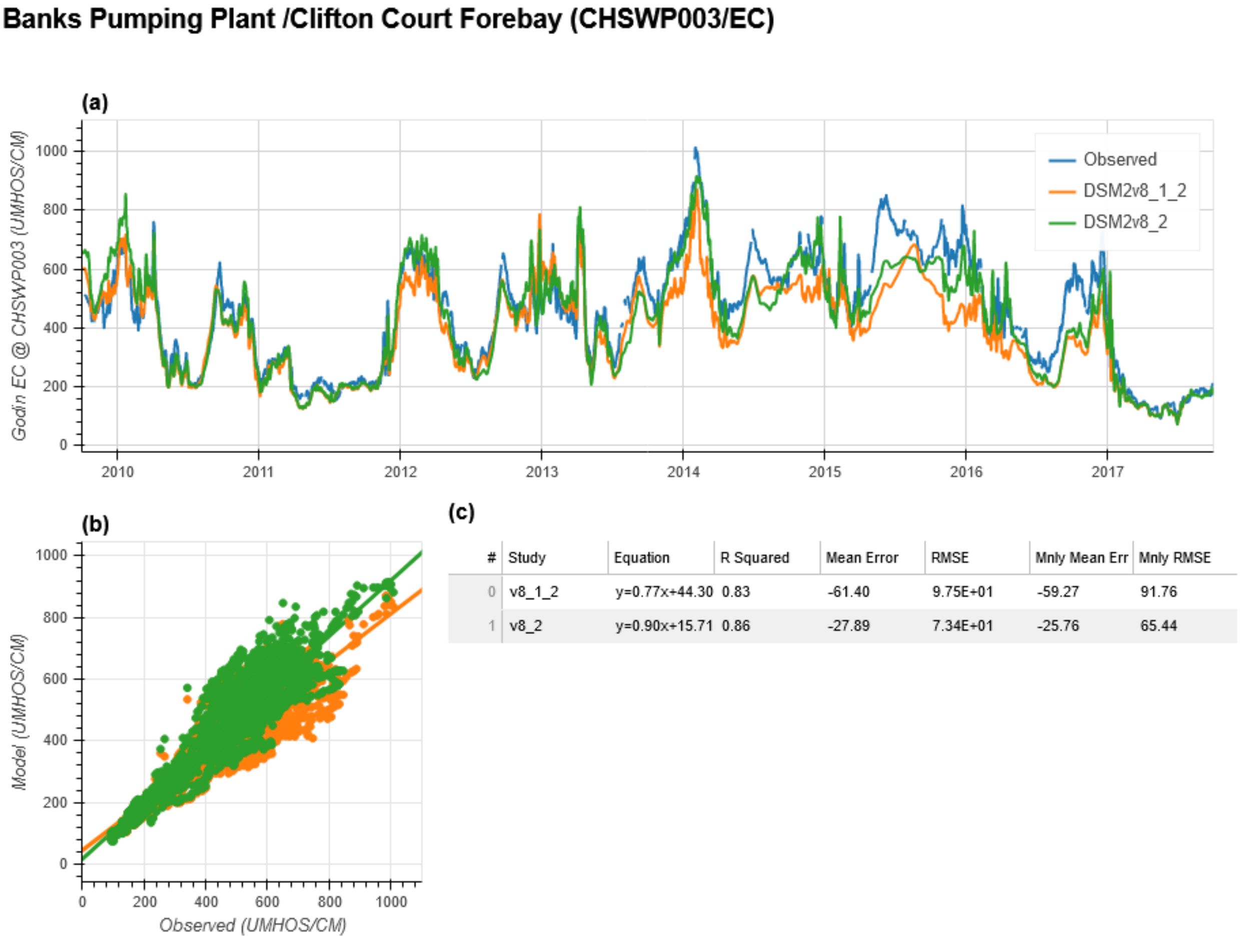
Figure 1-18 EC calibration metrics for Old River at Bacon Island



Old River at Bacon Island (ROLD024/EC)

#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.65x+68.77	0.75	-96.23	1.50E+02	-96.62	144.56
1	v8_2	y=0.88x+17.98	0.82	-37.70	1.04E+02	-38.00	95.99

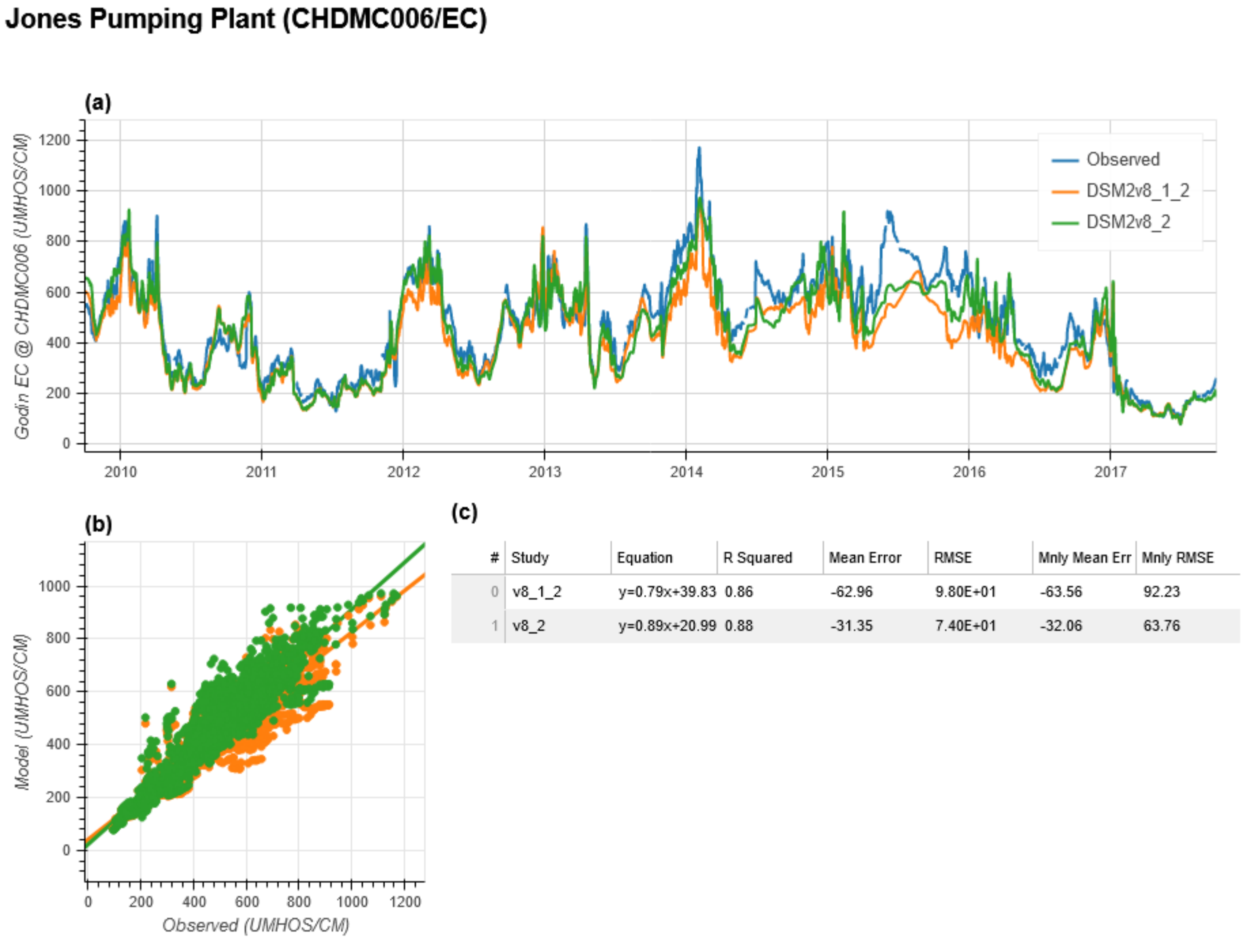
Figure 1-19 EC calibration metrics for Banks Pumping Plant



Banks Pumping Plant /Clifton Court Forebay (CHSWP003/EC)

#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.77x+44.30	0.83	-61.40	9.75E+01	-59.27	91.76
1	v8_2	y=0.90x+15.71	0.86	-27.89	7.34E+01	-25.76	65.44

Figure 1-20 EC calibration metrics for Jones Pumping Plant


Jones Pumping Plant (CHDMC006/EC)

#	Study	Equation	R Squared	Mean Error	RMSE	Mnly Mean Err	Mnly RMSE
0	v8_1_2	y=0.79x+39.83	0.86	-62.96	9.80E+01	-63.56	92.23
1	v8_2	y=0.89x+20.99	0.88	-31.35	7.40E+01	-32.06	63.76

1.4.3.2 Salinity validation

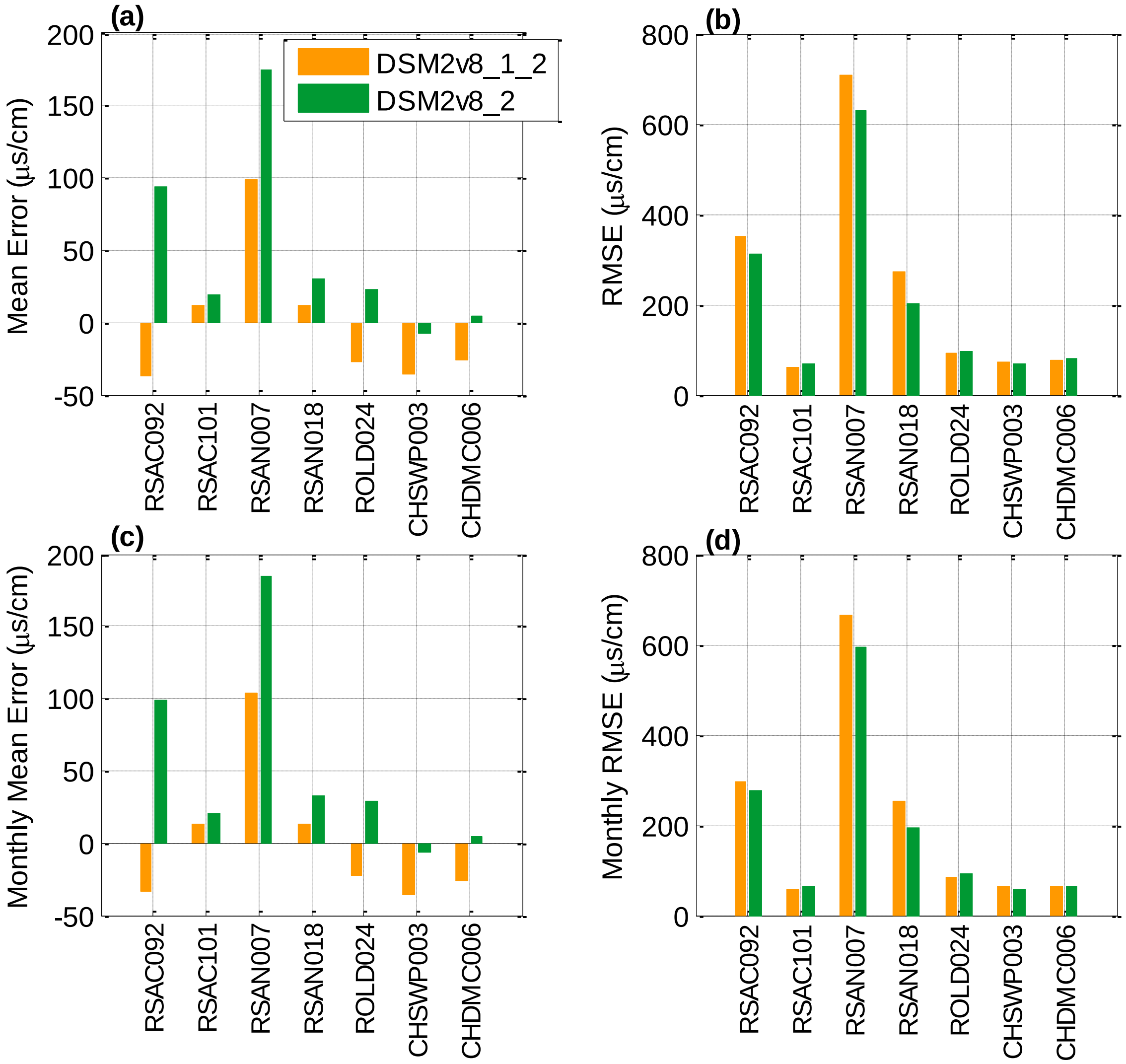
As in the flow and stage validation presented in Section 3.3.3, salinity validation in this section only showcases statistical metrics at selected locations. Detailed time series plots and scatter plots during the validation period, along with detailed validation results at other salinity calibration locations, are provided in a separate technical report (California Department of Water Resources 2021).

Figure 1-21 shows the mean error and RMSE at the daily and monthly scale between observed EC and simulated EC during the validation period. At both temporal scales, the magnitudes of mean error associated with both calibration efforts were comparable. Compared with V8.1.2, the calibrated V8.2.0 had larger mean errors at all locations except for the Banks and Jones pumping plants. At those locations, V8.2.0 produced saltier conditions (positive mean error) than the field measurements. At two pumping plants, the errors of V8.2.0 simulations were insignificant (with absolute values less than 10 µs/cm) and much smaller than their counterparts of V8.1.2.

The RMSE values also varied largely across different locations. The largest values occurred at Emmaton (RSAC092) and Antioch (RSAN007). This was expected, as these two locations are relatively saltier than other locations. In terms of magnitude, both calibration efforts yield similar RMSE values at those locations. In comparison, though, V8.2.0 had relatively smaller RMSE at most locations.

Overall, the calibrated V8.2.0 yielded similar statistical metrics as V8.1.2 at the selected locations during the validation period. While V8.2.0 consistently outperformed V8.1.2 during the calibration period (Section 4.3.1), this was not the case during the validation period. V8.1.2 had smaller mean error for most locations, while V8.2.0 produced smaller RMSE values at most locations.

Figure 1-21 Summary of statistical metrics of EC validation



1.5 Conclusions

The current calibration effort is to prepare a calibrated DSM2 model for hydrodynamic and water quality modeling and analysis in the Delta Conveyance Project. The previous release of the DSM2 model (V8.1.2) was modified to enable use of Delta channel depletion estimated by the Delta Channel Depletion model rather than that by the Delta Island Consumptive Use model. The hydrodynamics module of the updated model (DSM2 V8.2.0) was calibrated using observed flow and stage from water year 2011–2012 and by varying Manning’s n values in six groups of channels. The module was then validated using flow and stage data from water years 2001–2017. The water quality module of DSM2 V8.2.0 was calibrated using observed EC from water years 2010–2017, and the flow field was simulated via the calibrated hydrodynamics module, by modifying the dispersion factors of a number of channels. The calibrated water quality module was validated using observed and simulated EC from water years 2001–2009.

The flow, stage, and EC simulations from the calibrated DSM2 V8.2.0 were mostly similar to those of the previously calibrated DSM2 V8.1.2 but were in better agreement with observed data in most cases. In particular, DSM2 v8.2.0 better fit observed EC data during the validation period at most key locations; however, the calibrated model tended to under-predict the observed stage and flow at most key locations examined. These biases need to be addressed in future model enhancement and calibration efforts.

1.6 Acknowledgements

The authors like to thank Chandra Chilmakuri (State Water Contractors) for sharing his experience on calibrating DSM2. The authors would also like to thank their colleagues Lan Liang and Bob Suits for incorporating DCD into DSM2 V8.2.0. Hari Rajbhandari and Hans Kim reviewed an early version of the chapter and provided insightful comments that improved the quality of it.

Appendix 1-A: Inventory of calibration and validation stations

Table 1-A1 List of flow, stage, and salinity stations and the data record periods

Location ID	CDEC ID	Location Name	Flow	Stage	EC
RSAC155	FPT	Sacramento River at Freeport	1/2/2000–1/1/2020	1/1/2000–3/26/2020	—
RSAC128	SDC	Sacramento River above Delta Cross Channel	12/22/1992–1/1/2020	10/1/2005–3/26/2020	—
RSAC123	GES	Sacramento River below Georgiana Slough	1/8/1993–1/1/2020	—	—
RSAC101	SRV/RVB	Sacramento River at Rio Vista	4/20/1995–1/1/2020	10/1/2005–3/26/2020	10/17/2003–1/1/2020
RSAC092	EMM	Sacramento River at Emmaton	—	—	3/27/1999–1/1/2020
RSAC081	CLL	Sacramento River at Collinsville	—	—	5/3/1999–1/1/2020
RSAC075	MAL	Sacramento River at Mallard Island	—	—	1/4/1989–1/1/2020
RSAC064	PCT	Sacramento River at Port Chicago	—	—	1/1/2000–1/1/2020
RSAN087	MSD	San Joaquin River at Mossdale	—	3/1/1983–11/19/2020	
RSAN072	BDT	San Joaquin River at Brant Bridge	1/23/2008–1/1/2020	—	4/6/2005–1/1/2020
RSMKL008	STI	South Fork Mokelumne River at Terminous	—	—	1/1/2000–1/1/2020
RSAN063	SJG	San Joaquin River at Stockton		8/1/2006–5/7/2020	—
RSAN058	RRI	Rough and Ready Island	1/10/2007–8/10/2018	9/30/2005–8/4/2020	8/30/2000–1/1/2020
SLTRM004	TSL	Three Mile Slough	2/14/1997–1/1/2020	3/1/2005–3/26/2020	6/16/2008–1/1/2020
RSAN037	PRI	San Joaquin River at Prisoners Point	—	—	6/27/2006–1/1/2019
RSAN032	SAL	San Joaquin River at San Andreas Landing	—	—	3/27/1999–1/1/2020
RSAN018	SJJ	San Joaquin River at Jersey Point	5/11/1994–1/1/2019	10/1/2005–3/26/2020	3/26/1999–1/1/2020

Location ID	CDEC ID	Location Name	Flow	Stage	EC
RSAN007	ANC	San Joaquin River at Antioch	—	10/1/1982–12/2/2020	1/1/2000–8/1/2018
OLD_MID	UNI	Old River near Middle River	—	—	3/27/1999–1/1/2020
ROLD074	OH1	Old River at Head	2/7/2000–1/1/2000	10/1/1982–5/4/2020	—
ROLD059	OLD	Old River at Tracy Road Bridge	—	10/1/1982–10/26/2020	4/6/2005–1/1/2020
ROLD047	OAD	Old river near Delta Mendota Canal	—	10/1/1991–11/6/2020	—
ROLD034	OH4	Old River at Highway 4	1/1/2000–1/1/2020	10/1/2005–5/7/2020	—
ROLD024	OBI	Old River at Bacon Island	1/6/1987–1/1/2020	9/27/2005–5/7/2020	3/8/2000–1/1/2020
HLT_159	HLT	Middle River near Holt	1/9/1987–1/1/2020	—	—
CHGRL009	GCT	Grant Line Canal at Tracy Boulevard Bridge	—	10/1/1982–9/4/2020	—
Georg_SL	GSS	Georgiana Slough at Sacramento River	9/17/2001–1/1/2020	10/1/2005–3/26/2020	—
SLMZU011	BDL	Montezuma Slough at Beldon Landing		10/1/1998–1/1/2020	1/1/1989–1/1/2020
SLDUT007	DSJ	Dutch Slough	2/9/1996–1/1/2020	10/1/2005–5/7/2020	12/9/2009–1/1/2020
SLMZU025	NSL	National Steel	1/16/2008–1/1/2020	5/21/2009–1/1/2020	9/16/2001–1/1/2020
CHVCT000	VCU	Victoria Canal near Byron	2/15/2006–1/1/2020	—	6/26/2007–1/1/2020
CHSWP003	CLC	Banks Pumping Plant /Clifton Court Forebay	—	—	12/30/2000–12/31/2019
CHDMC006	DMC	Jones Pumping Plant	—	—	3/27/1999–1/1/2020
SLCBN002	SNC	Chadbourne Slough near Sunrise Duck Club	—	—	9/16/2001–1/1/2020
SLSUS012	VOL	Suisun Slough 300 ft south of Volanti Slough	—	—	2/28/2008–1/1/2020
SLBAR002	BKS	Barker Slough at North Bay Aqueduct	—	—	1/1/2001–1/1/2020
FAL	FAL	False River near Oakley	8/9/2007–1/1/2020	—	—
FCT_280	FCT	Fisherman's Cut	6/24/2015–1/1/2020	—	—
HOL	HOL	Holland Cut near Bethel Island	10/23/2007–1/1/2020	—	—
HWB	HWB	Miner Slough at HWY 84 Bridge	5/17/2006–1/1/2019	—	—

Location ID	CDEC ID	Location Name	Flow	Stage	EC
MOK	MOK	Mokelumne River at San Joaquin River	2/9/2007–1/1/2020	—	—
PDC	PDC	Paradise Cut	4/16/2014–1/1/2019	—	—
SSS	SSS	Steamboat Slough	9/26/2003–1/1/2019	—	5/22/2014–5/5/2016
SUT	SUT	Sutter Slough at Courtland	12/11/2006–1/1/2020	—	—
TRN	TRN	Turner Cut near Holt	2/15/2006–1/1/2020	—	—

1.7 References

California Department of Water resources 1997. DSM2 Model Development, in 18th Annual Progress Report on Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh.

California Department of Water resources 2009. DSM2 Recalibration. Technical Report.

California Department of Water resources 2021. DSM2 V8.2.0 Calibration. Technical Report, https://data.cnra.ca.gov/dataset/dsm2-v8-2-0.

Liang and Suits 2017. Implementing DETAW in Modeling Hydrodynamics and Water Quality in the Sacramento-San Joaquin Delta, in 38th Annual Progress Report on Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh.

Liang and Suits 2018. Calibrating and Validating Delta Channel Depletion Estimates., in 39th Annual Progress Report on Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh.

Liu and Sandhu 2012. DSM2 Version 8.1 Recalibration, in 33rd Annual Progress Report on Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh.

Nader-Tehrani and Shrestha 2000. DSM2 Calibration, in 21st Annual Progress Report on Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh.

43rd Annual Progress Report
June 2022

Chapter 2
DSM2 Georeferenced Grid Maps

Authors: Brad Tom, Minxue He, Nicky Sandhu
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Contents

Figures

Chapter 2 DSM2 Georeferenced Grid Maps

2.1 Introduction

This chapter describes the development of georeferenced grid maps for the Delta Simulation Model 2 (DSM2) (Tom et al. 2020). The georeferenced grid maps are stored as GIS shapefiles with symbology added to the various features to represent channels, nodes, gates, reservoirs, reservoir connections, and monitoring stations. The shapefiles are compatible with ArcGIS and QGIS and are available on the CNRA Open Data web site. The georeferenced grid maps are also available as Portable Document Format (PDF) files.

The workflow for creating the grid maps is automated as much as possible, streamlining updates and the development of new versions. DSM2 is a 1D model and uses input derived from georeferenced information rather than using georeferenced information directly. Georeferenced grid maps can help ensure that the DSM2 input derived from georeferenced information is sufficiently accurate.

The first versions of the DSM2 grid map were created using AutoCAD. Later versions created using AutoCAD were exported to PDF files, which were printed on plotter paper. The most recent version created using this method dates back to 2002. The first ArcGIS version was created in 2009.

2.2 GIS Layers

This section describes how DSM2 model features are represented in the GIS layers included in the georeferenced grid maps, including symbology. In each of the georeferenced grid maps, the display of each layer can be toggled on or off. Each subsection below identifies the default display setting of the layer in the georeferenced grid maps.

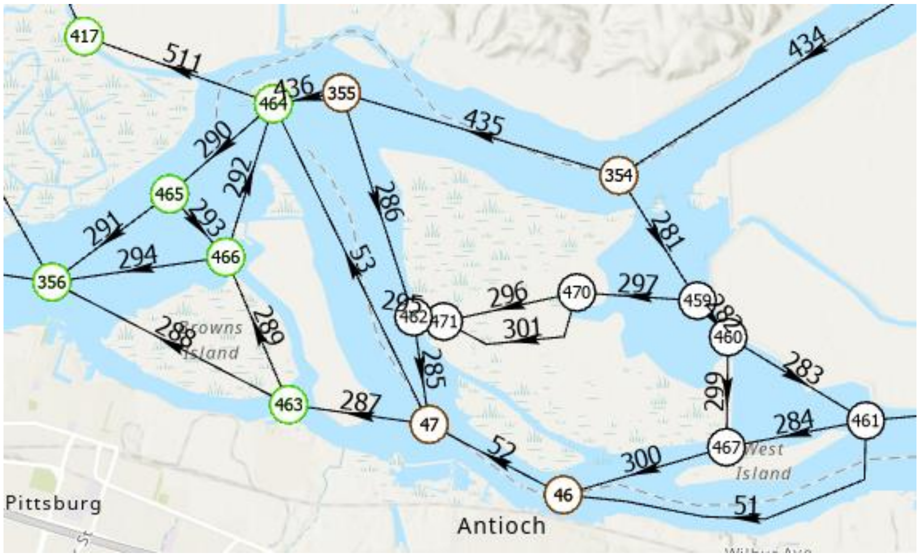
2.2.1 Channel Layers

The grid maps include three layers to represent DSM2 channels. Each layer represents different channel attributes, as described in the following subsections.

2.2.1.1 Network Channels Layer

Most users will probably want to use the *Network Channels* layer, which is designed to show connectivity to nodes and is most similar to the channel representations used in most previous grid map versions. This layer uses mostly straight lines to represent channels, with additional line segments added as needed to prevent overlapping with other channels or other features such as node symbols (see Figure 2-1, channels 296 and 301). The lines are colored black, with inline arrows indicating positive flow direction, and offset numbers indicating the DSM2 channel number. This layer is displayed by default in DSM2 georeferenced grid maps.

Figure 2-1 The Network Channels Layer uses mostly straight lines to represent channels

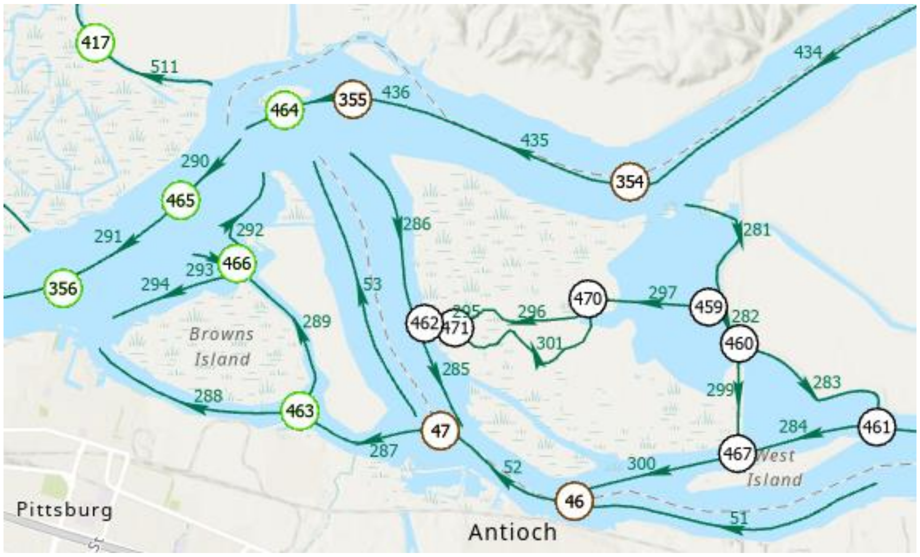


Note: Some channels are edited to prevent overlap or to improve appearance.

2.2.1.2 Centerline Channels Layer

The *Centerline Channels* layer is created directly from the Cross-Section Development Program (CSDP) centerlines, which sometimes have endpoints that are not located at the nodes. Endpoints for some centerlines are placed away from the nodes because CSDP centerlines are intended to represent a portion of the volume of a physical channel, and placing all centerline endpoints at nodes would result in overlapping volumes in some areas (Tom et al. 2020). Figure 2-2 shows a number of centerline channels with endpoints located away from nodes for this reason. The lines are colored dark green, with inline arrows indicating positive flow direction and offset numbers indicating the DSM2 channel number. This layer is not displayed by default in DSM2 georeferenced grid maps.

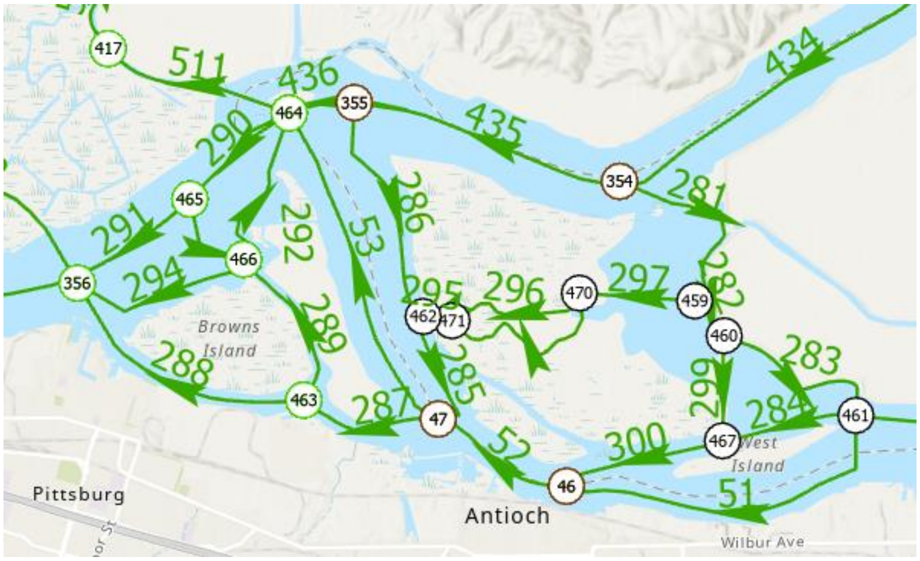
Figure 2-2 The centerline channels layer is created from the CSDP centerlines



2.2.1.3 Centerline Channels Connected to Nodes Layer

The *Centerline Channels Connected to Nodes* Layer (Figure 2-3) is the same as the Centerline Channels Layer, but with line segments added to connect the endpoints to the nodes. This layer is only intended for use as a reference when using the Particle Tracking Model (PTM) Animator to display model results. The lines are colored light green, with inline arrows indicating positive flow direction and offset numbers indicating the DSM2 channel number. This layer is not displayed by default in DSM2 georeferenced grid maps.

Figure 2-3 The Centerline Channels Connected to Nodes layer is created by adding extra line segments to connect CSDP centerlines to nodes



2.3 Node Layers

The DSM2 georeferenced grid maps include four different node layers, each colored differently to identify the model or models that use the nodes. The models identified include DSM2, the Delta Channel Depletion Model (DCD), and the Suisun Marsh Channel Depletion Model (SMCD). The colors that are used to differentiate the node layers identified in each subsection below are applied to the symbol outline and text only. All node symbols have white backgrounds, making the node numbers easier to read, but these sometimes obscure the features beneath. Users who are unable to distinguish the colors used by these layers will need to display one layer at a time to determine which model(s) use the node(s) in the layer. All nodes are represented by circles with the node number in the middle. All node layers are displayed by default in DSM2 georeferenced grid maps. Example node layers are displayed in Figure 2-4.

2.3.1 DSM2 Nodes

The *DSM2 Nodes* layer includes all nodes used by the DSM2 model, and its color is black. All other layers are displayed on top of this layer, so that the colors displayed will always correctly identify which model(s) use the node(s).

2.3.2 DSM2 and SMCD Nodes

This layer includes all nodes that are used by the Suisun Marsh Channel Depletion Model (SMCD), and its color is green. All SMCD nodes are also DSM2 nodes. This layer is called *DSM2 and SMCD Nodes* so that it can be displayed on top of the DSM2 nodes layer, correctly identifying the nodes as belonging to both models.

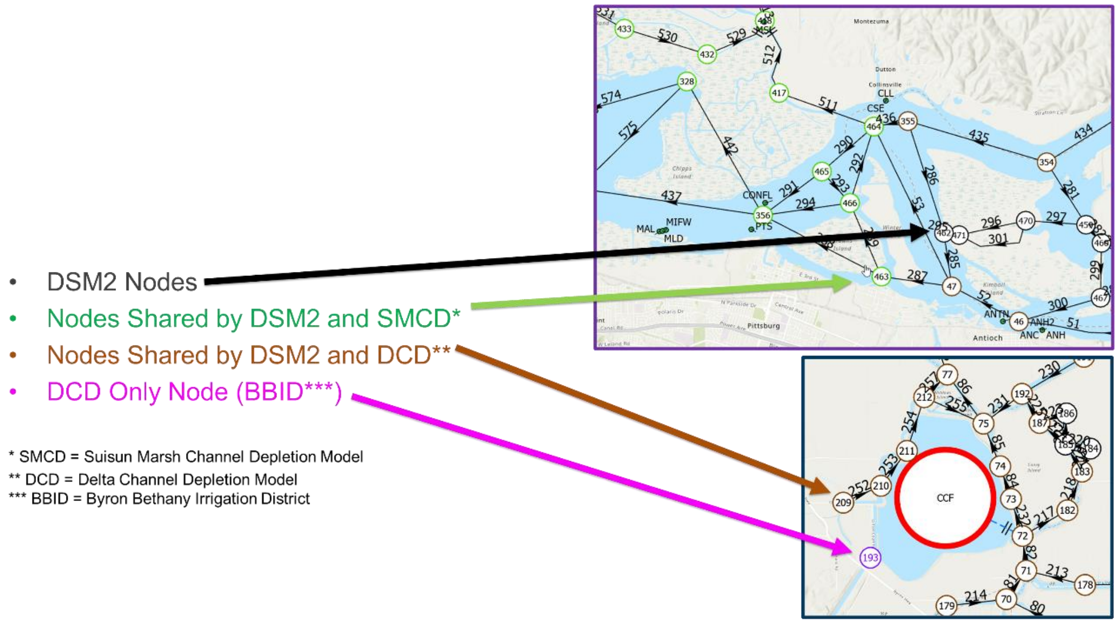
2.3.3 DSM2 and DCD Nodes

This layer includes all nodes that are used by the Delta Channel Depletion Model (DCD), and its color is brown. All DCD nodes are also DSM2 nodes. This layer is called *DSM2 and DCD Nodes* so that it can be displayed on top of the DSM2 nodes layer, correctly identifying the nodes as belonging to both models.

2.3.4 DCD Only Node

The *DCD Only Node* layer contains the node that is used only by DCD, and not by DSM2, and its color is pink.

Figure 2-4 Node layers are colored to identify which model(s) use the nodes



2.4 Gate Layers

DSM2 requires gates to be located at the ends of channels. This often results in differences between the locations of gates in the model grid and their actual locations. This section describes the two layers used to represent the approximate actual location and approximate grid location of each gate (Figure 2-5).

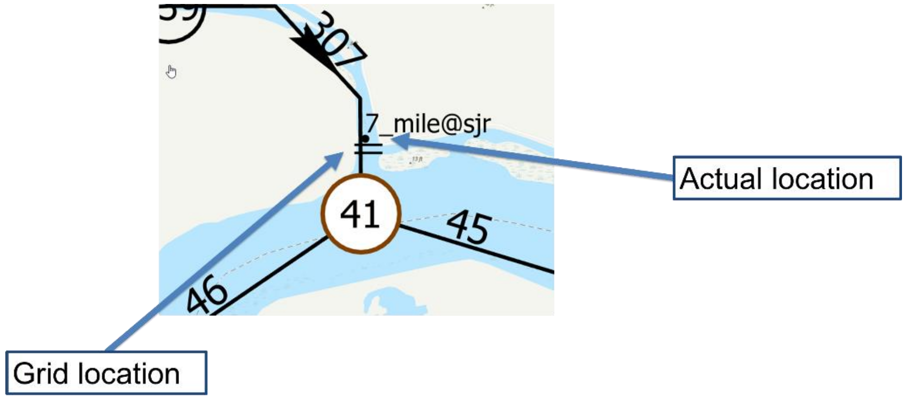
2.4.1 Actual Gate Location Layer

The Actual Location layer uses a dot symbol (see Figure 2-5 for an example representing station “7_mile@sjr”) to represent the approximate location of the physical structure that each DSM2 gate represents. This layer is not displayed by default in DSM2 georeferenced grid maps.

2.4.2 Grid Gate Location Layer

The Grid Location layer uses a double-line gate symbol (See Figure 2-5 for an example). In DSM2, the gates are located at the end of a channel. In GIS, we place the symbol near the end of the channel, to avoid interference with the display of the node symbol. This layer is displayed by default in DSM2 georeferenced grid maps.

Figure 2-5 Actual versus Grid gate location



2.5 Monitoring Station Layer

The monitoring station layer is labeled by default with station identifiers that are based on the station codes used by the California Data Exchange Center (CDEC) (Figure 2-6). For stations not on CDEC, the identifier may be an identifier used by the operating agency or a mnemonic.

Sometimes you will see more than one station at the same location. This means there are different agencies operating near to one another, each with its own station identifiers and data distribution system. They may or may not be sharing facilities and telemetry.

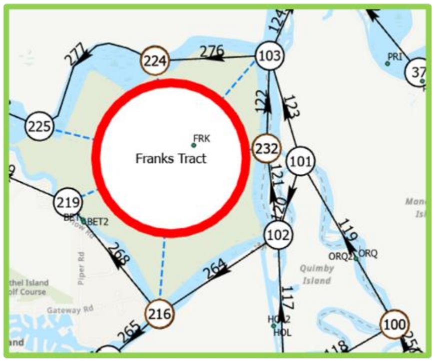
The layer is created from a list that is maintained by the Delta Modeling Section of the California Department of Water Resources' Bay-Delta Office. We try to keep this list accurate and up to date, but we are not always informed when new stations begin operating. Consequently, we cannot promise that our list will always be up to date, and we cannot promise that the station layer on our grid maps will always be up to date.

2.6 Reservoir Layer

The Reservoir layer displays DSM2 reservoirs as red circles with white backgrounds, with the reservoir name in the middle (Figure 2-6). Depending on the size of the circle, the reservoir name may be abbreviated.

Figure 2-6 The Monitoring Station, Reservoir, and Reservoir Connections layers

- Monitoring stations identified by station codes
  - Multiple equipment types and/or agencies at a location
- Reservoirs & Connections
  - Red circle, blue dashed connecting lines
  - Names sometimes abbreviated



* CDEC=California Data Exchange Center

2.7 Reservoir Connections Layer

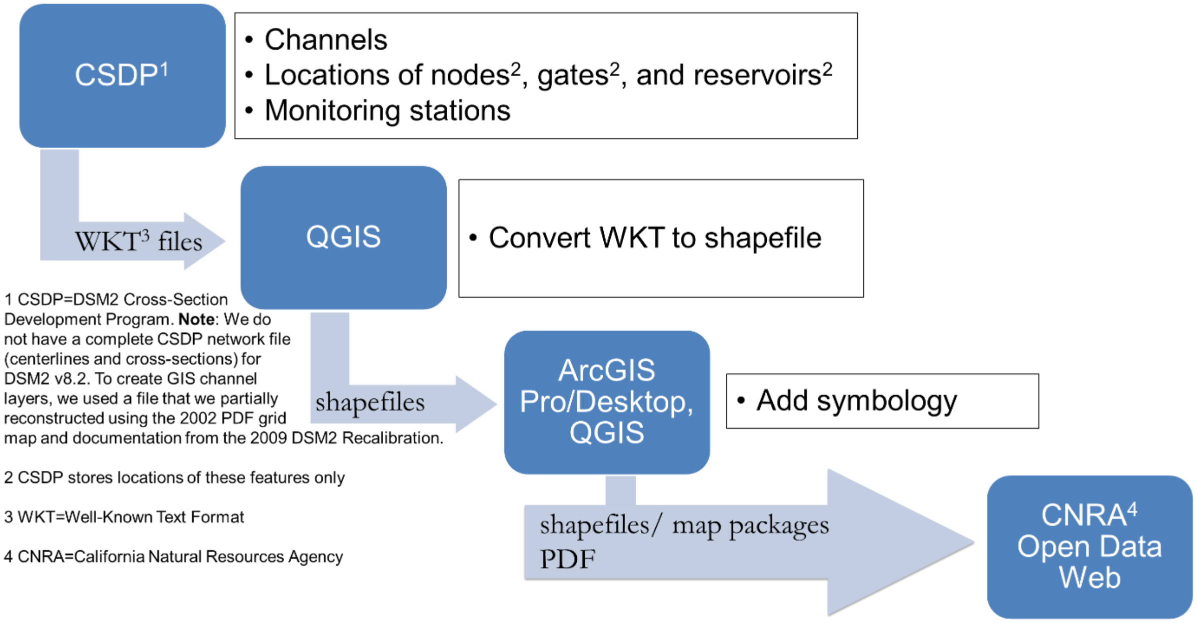
The reservoir connections layer consists of straight dashed blue lines that show connectivity of DSM2 reservoirs to nodes (Figure 2-6).

2.8 Workflow

For DSM2 v8.2, the process of creating georeferenced grid maps has mostly been automated. The workflow (Figure 2-7) for creating the grid maps begins with the CSDP, which is the tool used to create DSM2 geometry (Tom 1998). All editing of features, including channels, nodes, gates, and reservoirs, occurs in the CSDP. This process helps ensure that the shapefiles match the geometry input used by DSM2[1]. The process also makes creating updates and new versions easier and less error prone.

To create DSM2 grid maps from CSDP data, the following process is used: (1) Use the CSDP to export CSDP data to Well-Known Text (WKT) files, (2) Use QGIS to convert the WKT files to GIS shapefiles, (3) Use GIS to add symbology and create the georeferenced grid maps, which are in the form of GIS map packages and PDF documents. The DSM2 georeferenced grid maps are available on the CNRA open data web site at https://data.cnra.ca.gov/dataset/dsm2-georeferenced-model-grid.

Figure 2-7 Workflow for creating DSM2 georeferenced grid maps



1 CSDP=DSM2 Cross-Section Development Program. **Note**: We do not have a complete CSDP network file (centerlines and cross-sections) for DSM2 v8.2. To create GIS channel layers, we used a file that we partially reconstructed using the 2002 PDF grid map and documentation from the 2009 DSM2 Recalibration.

2 CSDP stores locations of these features only

3 WKT=Well-Known Text Format

4 CNRA=California Natural Resources Agency

[1] For DSM2 v8.2 and prior versions, channel lengths do not always match centerline lengths. This discrepancy will be corrected in later versions.

2.8.1 Creating DSM2 Channel Shapefiles

This section describes the process exporting CSDP data to GIS shapefiles. First, bathymetry, channel network, and node landmark files are loaded in to the CSDP (Tom 1998).

2.8.1.1 Creating the Centerline Channels WKT file

After loading a CSDP network file, in the CSDP menu bar, select *Network-Export-Export to WKT format for GIS* (Figure 2-8a). In the dialog that appears (Figure 2-8b), click *Select File* to specify the name and location of the WKT file to be created. Make sure the two checkboxes in the dialog are unchecked, then click *ok*.

Figure 2-8a Using the CSDP to export to a network file to a WKT file

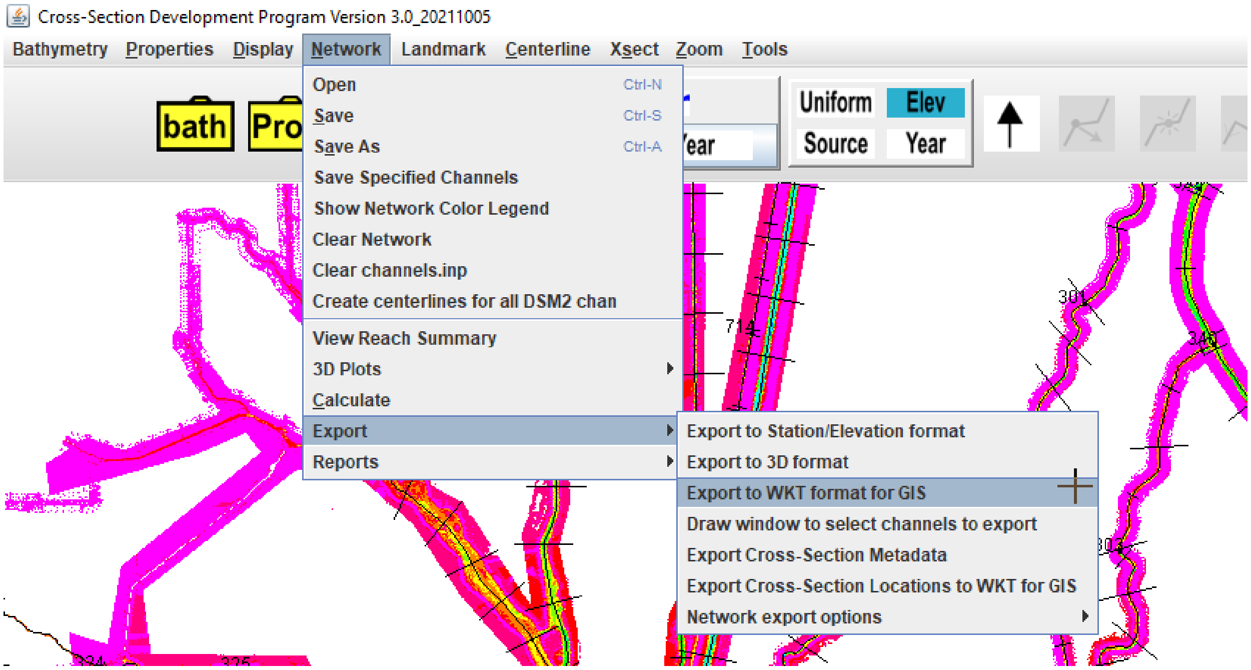
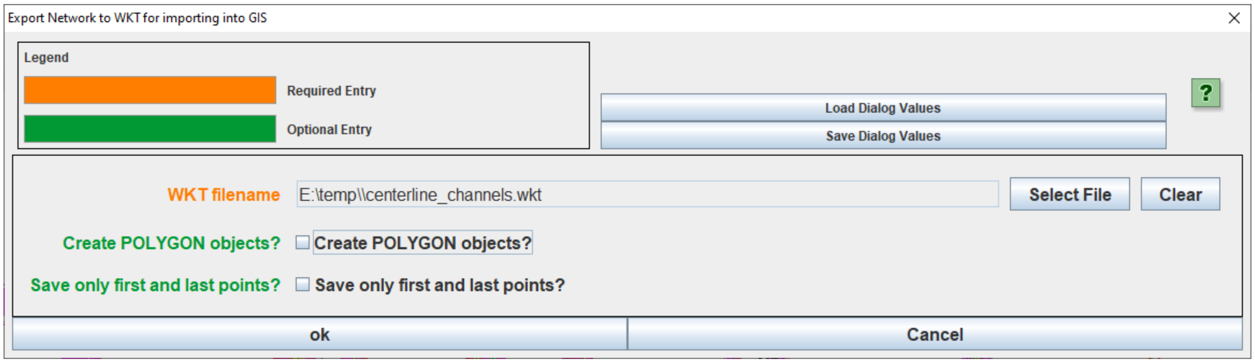

Cross-Section Development Program Version 3.0_20211005
Bathymetry
Properties
Display
Network
Landmark
Centerline
Xsect
Zoom
Tools
bath
Pro
Uniform
Elev
Source
Year
Open Ctrl-N
Save Ctrl-S
Save As Ctrl-A
Save Specified Channels
Show Network Color Legend
Clear Network
Clear channels.inp
Create centerlines for all DSM2 chan
View Reach Summary
3D Plots
Calculate
Export
Reports
Export to Station/Elevation format
Export to 3D format
Export to WKT format for GIS
Draw window to select channels to export
Export Cross-Section Metadata
Export Cross-Section Locations to WKT for GIS
Network export options
714
301
340
303
324
325

Figure 2-8b CSDP Export Network to WKT for importing into GIS dialog


Export Network to WKT for importing into GIS

Legend

🟧 Required Entry

🟩 Optional Entry

Load Dialog Values

Save Dialog Values

WKT filename: E:\temp\\centerline_channels.wkt Select File Clear

Create POLYGON objects? □ Create POLYGON objects?

Save only first and last points? □ Save only first and last points?

ok Cancel

2.8.1.2 Creating the Network Channels WKT file

To create a *Network Channels* WKT file, you do not need to have a CSDP network file loaded. Instead, you will use the CSDP to create new straight line (consisting of two endpoints and no other points) CSDP centerlines, using the information from a CSDP landmark file containing node coordinates, and a DSM2 channel connectivity input file (such as channel_std_grid_delta.inp).

In the CSDP menu bar, select *Create centerlines for all DSM2 chan* (Figure 2-9a). If you have not already loaded a DSM2 channel connectivity file, a dialog will appear asking you to specify a file (Figure 2-9b). Select a file that matches the channel network in the currently loaded network file and click *Open*. Channels consisting of straight lines connected to the nodes will be created (Figure 2-9c). Edit channel centerlines as needed to prevent overlapping features and to create the desired appearance.

Create the WKT file using the procedure described at the end of Section 2.8.1.1.

Figure 2-9a Using the CSDP to create the *Network Channels* network file

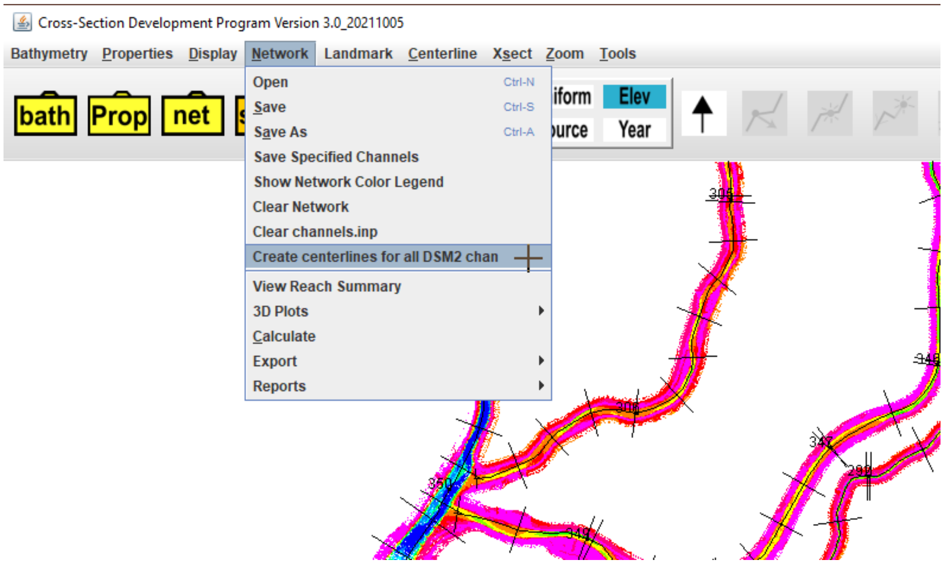
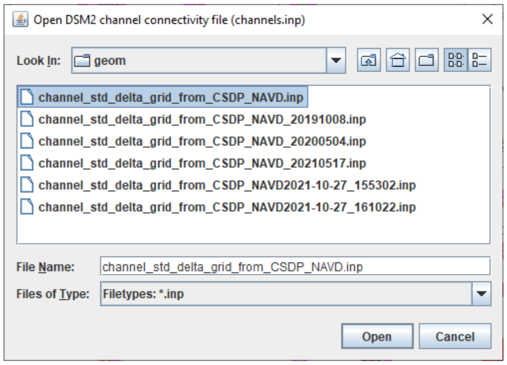


Figure 2-9b Using the file selector dialog to load a DSM2 channel connectivity file

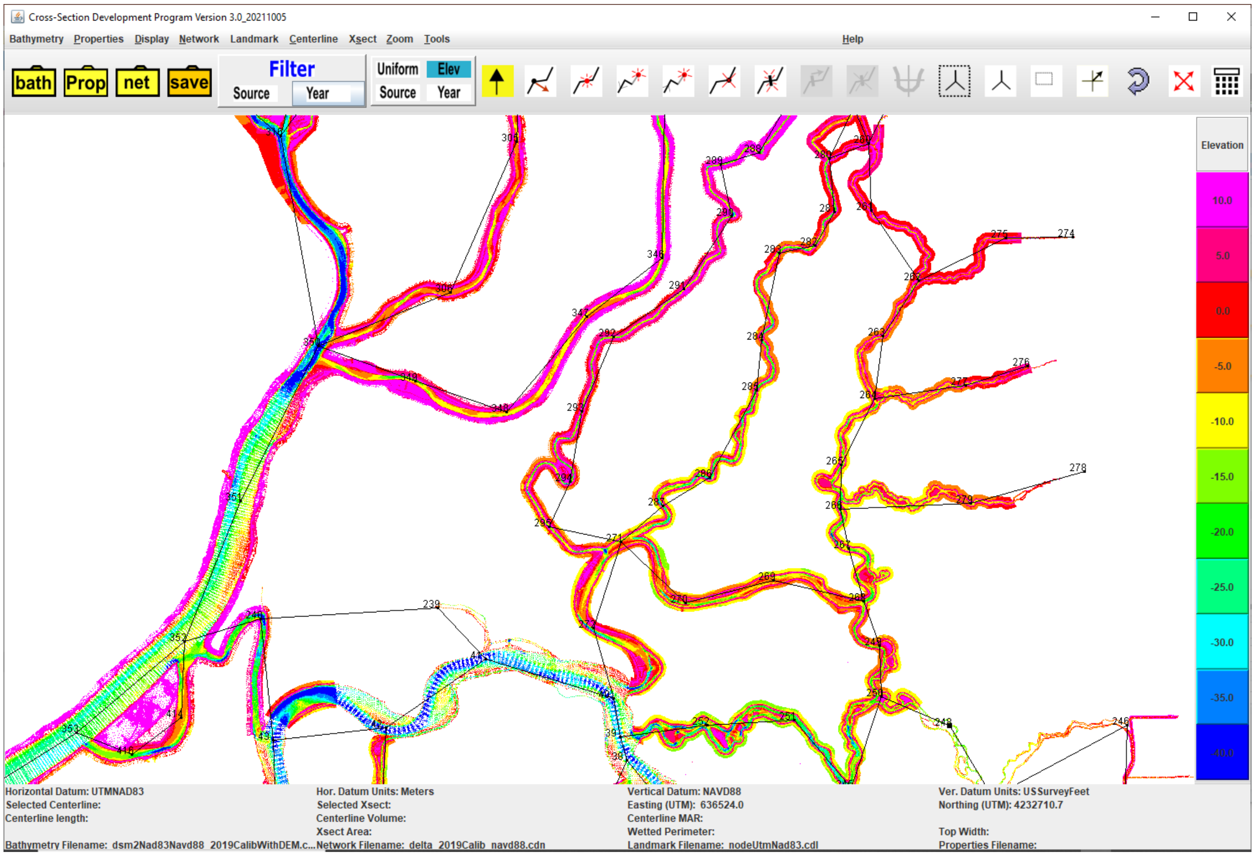

Open DSM2 channel connectivity file (channels.inp)

Look In: geom

channel_std_delta_grid_from_CSDP_NAVD.inp
channel_std_delta_grid_from_CSDP_NAVD_20191008.inp
channel_std_delta_grid_from_CSDP_NAVD_20200504.inp
channel_std_delta_grid_from_CSDP_NAVD_20210517.inp
channel_std_delta_grid_from_CSDP_NAVD2021-10-27_155302.inp
channel_std_delta_grid_from_CSDP_NAVD2021-10-27_161022.inp

File Name: channel_std_delta_grid_from_CSDP_NAVD.inp
Files of Type: Filetypes: *.inp

Figure 2-9c Network channels automatically created by CSDP



Horizontal Datum: UTMNAD83
Selected Centerline:
Centerline length:
Bathymetry Filename: dsm2Nad83Navd88_2019CalibWithDEM.c...

Hor. Datum Units: Meters
Selected Xsect:
Centerline Volume:
Xsect Area:
Network Filename: delta_2019Calib_navd88.cdn

Vertical Datum: NAVD88
Easting (UTM): 636524.0
Centerline MAR:
Wetted Perimeter:
Landmark Filename: nodeUtmNad83.cdl

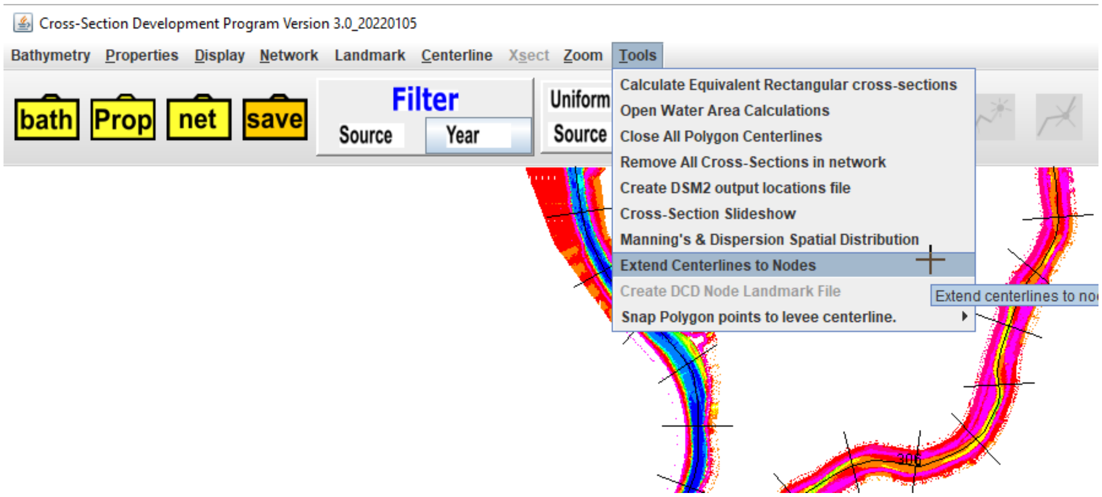
Ver. Datum Units: USSurveyFeet
Northing (UTM): 4232710.7
Top Width:
Properties Filename:

2.8.1.3 Creating the Centerline Channels Connected to Nodes WKT file

To create a Centerline Channels Connected to Nodes WKT file, you must first have CSDP bathymetry, network, and node landmark files loaded into the CSDP. Select *Tools-Extend Centerlines to Nodes* (Figure 2-10a). If you have not already loaded a DSM2 channel connectivity file, you will be prompted to load one (Figure 2-10b). After loading the file, the CSDP will automatically add line segments to the upstream and downstream ends of each centerline, connecting each to their respective nodes, and display a confirmation dialog (Figure 2-10c). If you do not see the confirmation dialog, there may be a problem with the channel connectivity file.

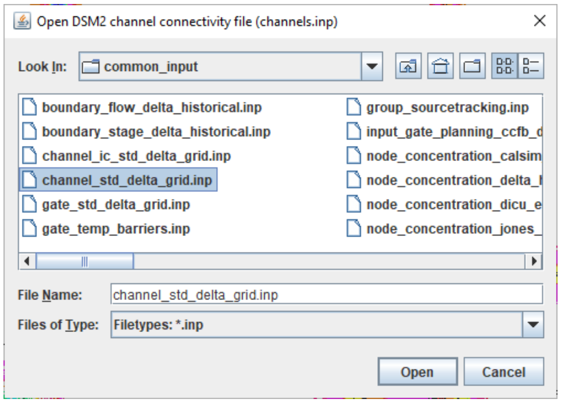
Create the WKT file using the procedure described at the end of section 2.8.1.1.

Figure 2-10a Creating the Centerline Channels Connected to Nodes layer by extending centerlines to nodes


Cross-Section Development Program Version 3.0_20220105

Bathymetry
Properties
Display
Network
Landmark
Centerline
Xsect
Zoom
Tools
bath
Prop
net
save
Filter
Source
Year
Uniform
Source
Calculate Equivalent Rectangular cross-sections
Open Water Area Calculations
Close All Polygon Centerlines
Remove All Cross-Sections in network
Create DSM2 output locations file
Cross-Section Slideshow
Manning's & Dispersion Spatial Distribution
Extend Centerlines to Nodes
Create DCD Node Landmark File
Snap Polygon points to levee centerline.
Extend centerlines to no
306

Figure 2-10b Using the file selector dialog to load a DSM2 channel connectivity file


Open DSM2 channel connectivity file (channels.inp)

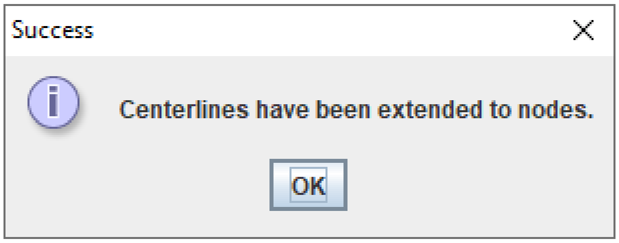
Look In: common_input

boundary_flow_delta_historical.inp
boundary_stage_delta_historical.inp
channel_ic_std_delta_grid.inp
channel_std_delta_grid.inp
gate_std_delta_grid.inp
gate_temp_barriers.inp
group_sourcetracking.inp
input_gate_planning_ccfb_d
node_concentration_calsim
node_concentration_delta_l
node_concentration_dicu_e
node_concentration_jones_

File Name: channel_std_delta_grid.inp

Files of Type: Filetypes: *.inp

Figure 2-10c This dialog confirms that centerlines have successfully been extended to nodes

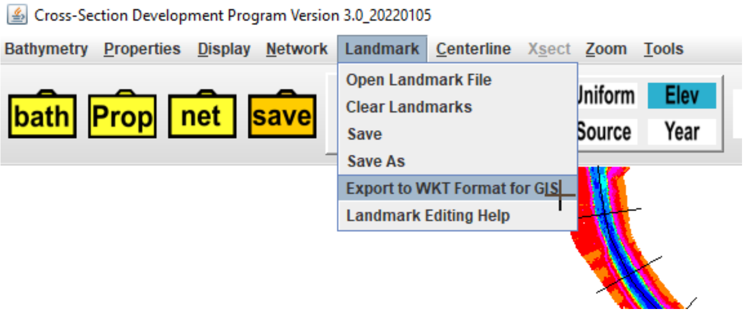

Success

Centerlines have been extended to nodes.

2.8.2 Creating DSM2 Node WKT files

DSM2 nodes are stored in CSDP landmark files. After loading a CSDP landmark file using *Landmark-Open Landmark File*, select *Landmark-Export to WKT Format for GIS* (Figure 2-11). In the file selector dialog that appears, enter the name of the WKT file you wish to create.

Figure 2-11 Exporting Landmark data to a WKT file



2.8.3 Creating DSM2 Gate WKT files

DSM2 gate coordinates are stored in CSDP landmark files. Follow the instructions in Section 2.8.2 to load a CSDP gate landmark file and to create a WKT file.

2.8.4 Creating DSM2 Reservoir WKT files

DSM2 reservoir coordinates are stored in CSDP landmark files. Follow the instructions in Section 2.8.2 to load a CSDP gate landmark file, and to create a WKT file.

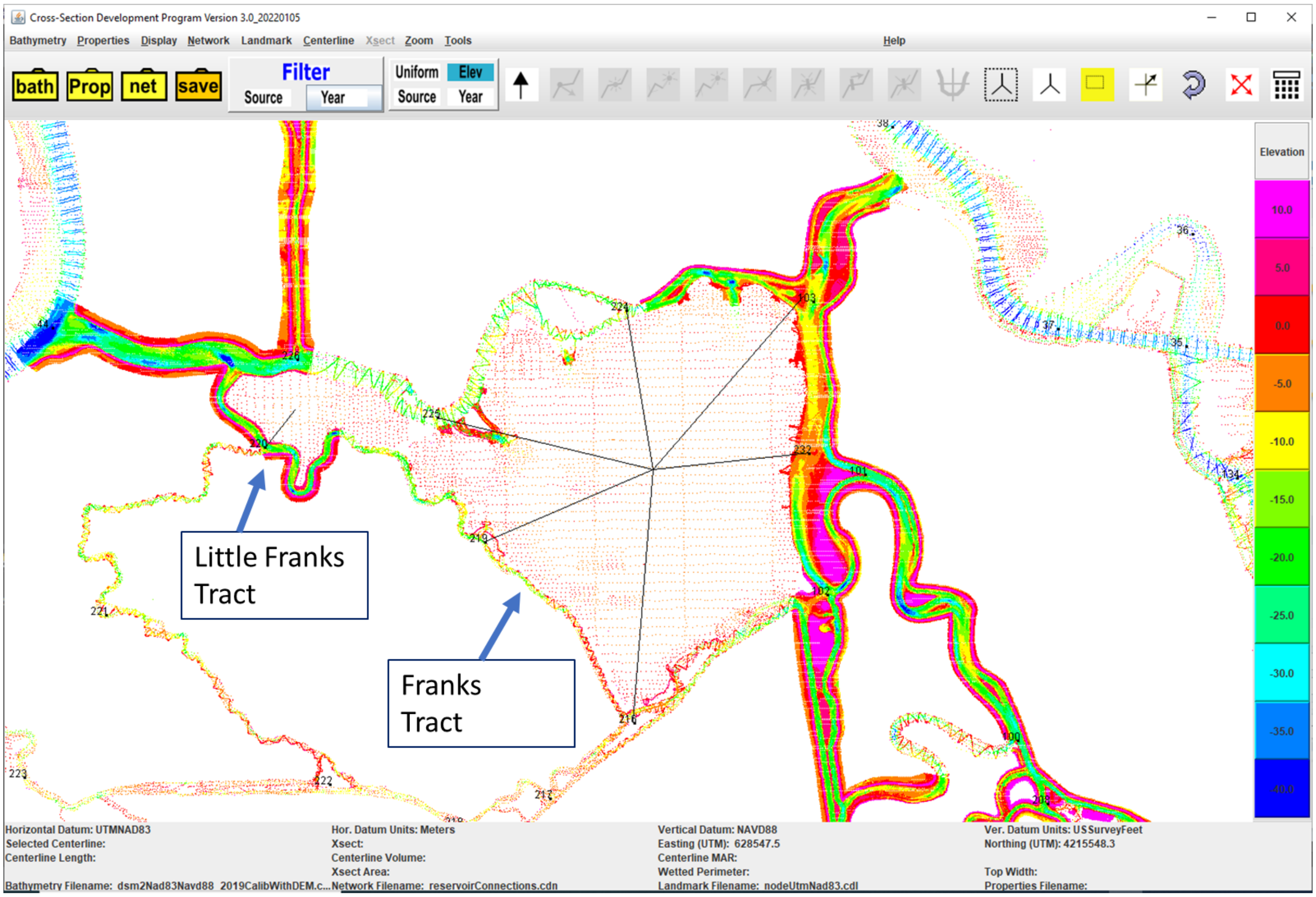
2.8.5 Creating DSM2 Reservoir Connection WKT files

Unlike other CSDP files, the CSDP reservoir connection data used to create the DSM2 georeferenced grid maps are not created from DSM2 input or from a file that is used to create DSM2 input. The CSDP reservoir connecting file is a CSDP network file, similar to the network file used to store DSM2 channel information.

To create a CSDP reservoir connection network file, load a bathymetry file and a node landmark file in the CSDP, and begin creating centerlines connecting the approximate reservoir location to the location of each node (Figure 2-12).

Create the WKT file using the procedure described at the end of section 2.8.1.1.

Figure 2-12 Creating reservoir connection lines in CSDP for Franks Tract and Little Franks Tract



Horizontal Datum: UTMNAD83
Selected Centerline:
Centerline Length:
Bathymetry Filename: dsm2Nad83Navd88 2019CalibWithDEM.c...

Hor. Datum Units: Meters
Xsect:
Centerline Volume:
Xsect Area:
Network Filename: reservoirConnections.cdn

Vertical Datum: NAVD88
Easting (UTM): 628547.5
Centerline MAR:
Wetted Perimeter:
Landmark Filename: nodeUtmNad83.cdl

Ver. Datum Units: USSurveyFeet
Northing (UTM): 4215548.3
Top Width:
Properties Filename:

2.8.6 Creating Monitoring Station WKT file

Monitoring station locations are stored in a CSDP landmark file. The monitoring station location data used to create the DSM2 georeferenced grid maps are not created from DSM2 input or from a file that is used to create DSM2 input. Follow the instructions in Section 2.8.2 to load a CSDP gate landmark file and to create a WKT file.

2.8.7 Converting WKT files to GIS Shapefiles

QGIS is a free open-source GIS application that can be used to create GIS shapefiles from the WKT files described in Section 2.8.6.

In QGIS, select *Layer-Add Layer-Add Delimited Text Layer…* (Figure 2-13a). In the *Data Source Manager | Delimited Text* dialog that appears (Figure 2-13b), click the Browse button in the upper right corner. In the *Choose A Delimited Text File to Open* dialog (Figure 2-13c), select the WKT file for which you would like to create a GIS shapefile. Click the *Add* button in the *Data Source Manager | Delimited Text* dialog that appears (Figure 2-13d). In the *Select Transformation for <layer name>* dialog (Figure 2-13e), select a coordinate transformation, then click *Add* to accept. Your layer will now be displayed in the main QGIS window (Figure 2-13f). Right click on the layer and select *Export-Save Features As…*. In the *Save <layer type> Layer as* dialog that appears (Figure 2-13g), click the browse button (three dot icon) next to the Filename field. In the file selector dialog that appears (Figure 2-13h), specify the name and location for the new GIS shapefile that you wish to create. Click the *Open* button (Figure 2-13i), and if all goes well, your new GIS shapefile will be displayed in the main QGIS window underneath the layer created by importing the WKT file (Figure 2-13j).

Figure 2-13a Import a WKT file into QGIS

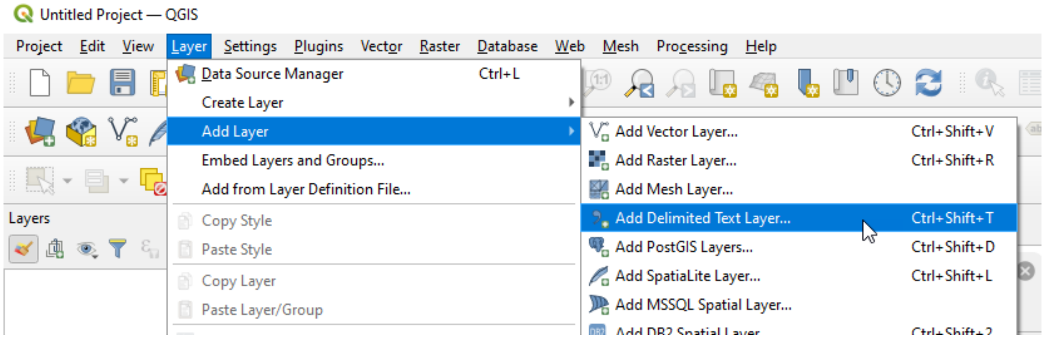
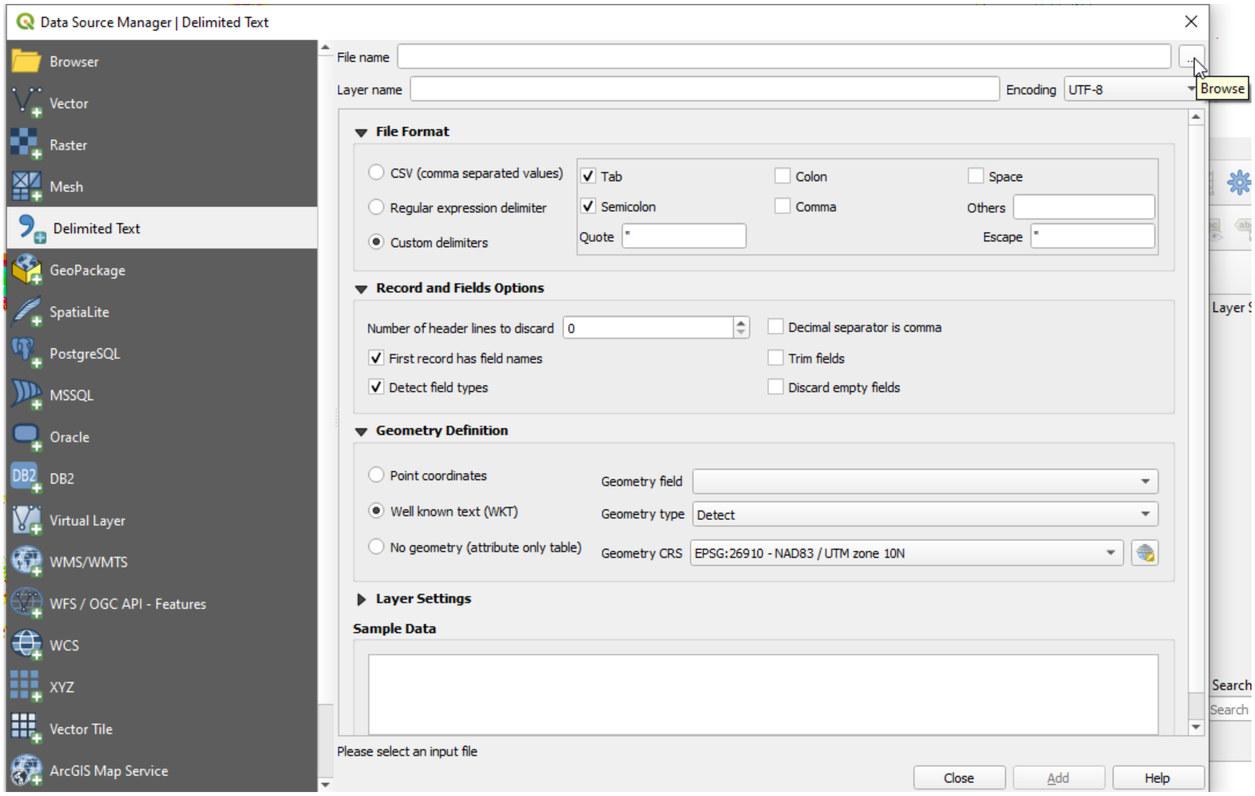

Untitled Project — QGIS
Project
Edit
View
Layer
Settings
Plugins
Vector
Raster
Database
Web
Mesh
Processing
Help
Data Source Manager Ctrl+L
Create Layer
Add Layer
Embed Layers and Groups...
Add from Layer Definition File...
Copy Style
Paste Style
Copy Layer
Paste Layer/Group
Add Vector Layer... Ctrl+Shift+V
Add Raster Layer... Ctrl+Shift+R
Add Mesh Layer...
Add Delimited Text Layer... Ctrl+Shift+T
Add PostGIS Layers... Ctrl+Shift+D
Add SpatiaLite Layer... Ctrl+Shift+L
Add MSSQL Spatial Layer...
Add DB2 Spatial Layer Ctrl+Shift+2
Layers

Figure 2-13b Click the *Browse* (3 dots) icon to load a WKT file



File name

Layer name

Encoding UTF-8

File Format

○ CSV (comma separated values) ☑ Tab □ Colon □ Space

○ Regular expression delimiter ☑ Semicolon □ Comma Others

● Custom delimiters Quote " Escape "

Record and Fields Options

Number of header lines to discard 0 □ Decimal separator is comma

☑ First record has field names □ Trim fields

☑ Detect field types □ Discard empty fields

Geometry Definition

○ Point coordinates Geometry field

● Well known text (WKT) Geometry type Detect

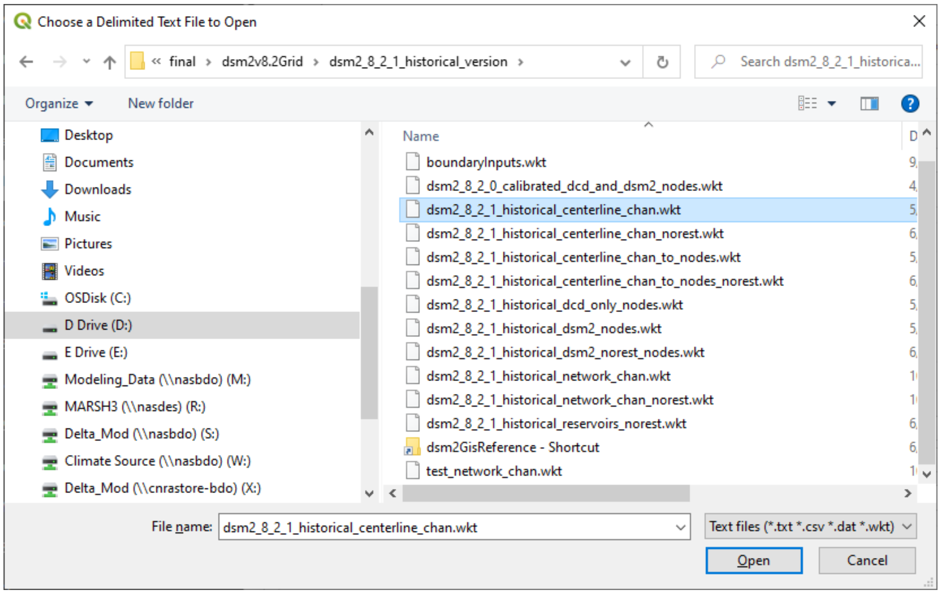
○ No geometry (attribute only table) Geometry CRS EPSG:26910 - NAD83 / UTM zone 10N

Layer Settings

Sample Data

Please select an input file

Figure 2-13c Use the file selector dialog to select a WKT file to load


Choose a Delimited Text File to Open
« final › dsm2v8.2Grid › dsm2_8_2_1_historical_version ›
Search dsm2_8_2_1_historica...
Organize
New folder
Desktop
Documents
Downloads
Music
Pictures
Videos
OSDisk (C:)
D Drive (D:)
E Drive (E:)
Modeling_Data (\\nasbdo) (M:)
MARSH3 (\\nasdes) (R:)
Delta_Mod (\\nasbdo) (S:)
Climate Source (\\nasbdo) (W:)
Delta_Mod (\\cnrastore-bdo) (X:)
Name
D
Open
Cancel

boundaryInputs.wkt

dsm2_8_2_0_calibrated_dcd_and_dsm2_nodes.wkt

dsm2_8_2_1_historical_centerline_chan.wkt

dsm2_8_2_1_historical_centerline_chan_norest.wkt

dsm2_8_2_1_historical_centerline_chan_to_nodes.wkt

dsm2_8_2_1_historical_centerline_chan_to_nodes_norest.wkt

dsm2_8_2_1_historical_dcd_only_nodes.wkt

dsm2_8_2_1_historical_dsm2_nodes.wkt

dsm2_8_2_1_historical_dsm2_norest_nodes.wkt

dsm2_8_2_1_historical_network_chan.wkt

dsm2_8_2_1_historical_network_chan_norest.wkt

dsm2_8_2_1_historical_reservoirs_norest.wkt

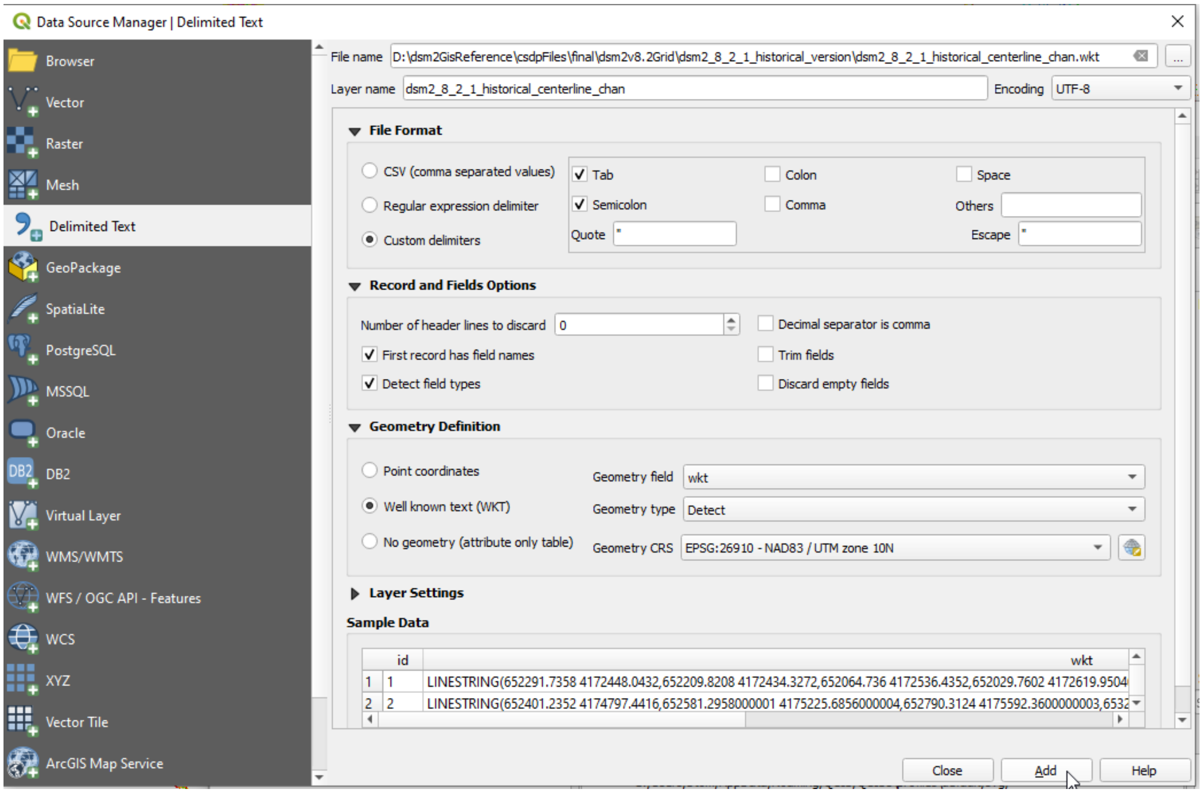
dsm2GisReference - Shortcut

test_network_chan.wkt

File name: dsm2_8_2_1_historical_centerline_chan.wkt

Text files (*.txt *.csv *.dat *.wkt)

Figure 2-13d Click *Add* to add a layer to the map, created from the WKT file


Data Source Manager | Delimited Text

File name D:\dsm2GisReference\csdpFiles\final\dsm2v8.2Grid\dsm2_8_2_1_historical_version\dsm2_8_2_1_historical_centerline_chan.wkt

Layer name dsm2_8_2_1_historical_centerline_chan Encoding UTF-8

File Format

○ CSV (comma separated values) ☑ Tab □ Colon □ Space
○ Regular expression delimiter ☑ Semicolon □ Comma Others
● Custom delimiters Quote " Escape "

Record and Fields Options

Number of header lines to discard 0 □ Decimal separator is comma
☑ First record has field names □ Trim fields
☑ Detect field types □ Discard empty fields

Geometry Definition

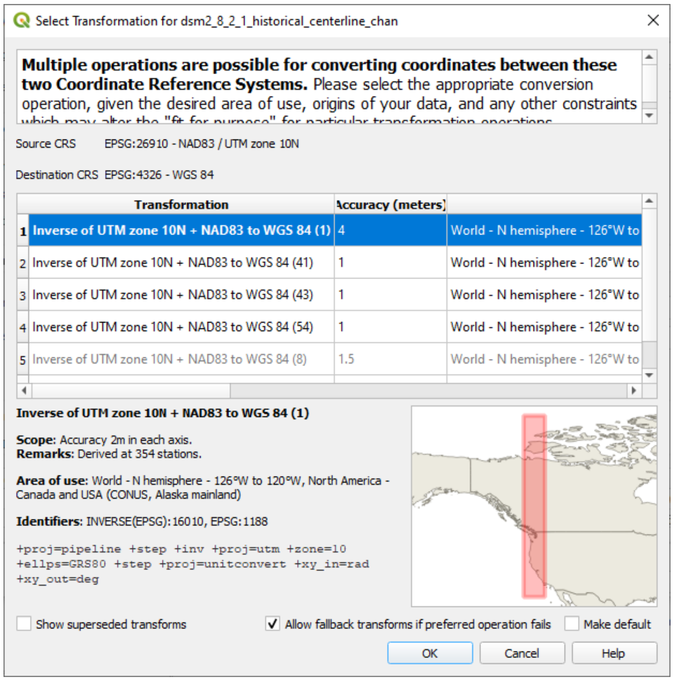
○ Point coordinates Geometry field wkt
● Well known text (WKT) Geometry type Detect
○ No geometry (attribute only table) Geometry CRS EPSG:26910 - NAD83 / UTM zone 10N

Layer Settings

Sample Data

	id	wkt
1	1	LINESTRING(652291.7358 4172448.0432,652209.8208 4172434.3272,652064.736 4172536.4352,652029.7602 4172619.9504
2	2	LINESTRING(652401.2352 4174797.4416,652581.2958000001 4175225.6856000004,652790.3124 4175592.360000003,6532

Figure 2-13e Select a coordinate transformation to use when creating a GIS layer from a WKT file


Select Transformation for dsm2_8_2_1_historical_centerline_chan

Multiple operations are possible for converting coordinates between these two Coordinate Reference Systems. Please select the appropriate conversion operation, given the desired area of use, origins of your data, and any other constraints which may alter the "fit for purpose" for particular transformation operations.

Source CRS EPSG:26910 - NAD83 / UTM zone 10N

Destination CRS EPSG:4326 - WGS 84

	Transformation	Accuracy (meters)	
1	Inverse of UTM zone 10N + NAD83 to WGS 84 (1)	4	World - N hemisphere - 126°W to
2	Inverse of UTM zone 10N + NAD83 to WGS 84 (41)	1	World - N hemisphere - 126°W to
3	Inverse of UTM zone 10N + NAD83 to WGS 84 (43)	1	World - N hemisphere - 126°W to
4	Inverse of UTM zone 10N + NAD83 to WGS 84 (54)	1	World - N hemisphere - 126°W to
5	Inverse of UTM zone 10N + NAD83 to WGS 84 (8)	1.5	World - N hemisphere - 126°W to

Inverse of UTM zone 10N + NAD83 to WGS 84 (1)

Scope: Accuracy 2m in each axis.
Remarks: Derived at 354 stations.

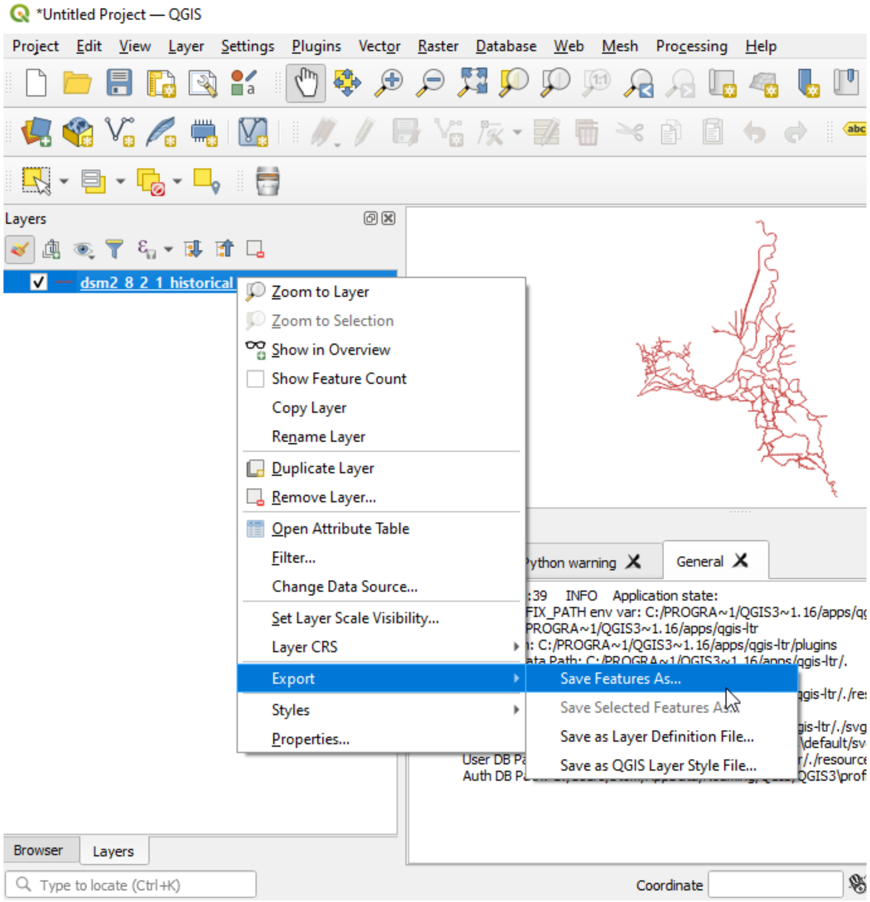
Area of use: World - N hemisphere - 126°W to 120°W, North America - Canada and USA (CONUS, Alaska mainland)

Identifiers: INVERSE(EPSG):16010, EPSG:1188

```
+proj=pipeline +step +inv +proj=utm +zone=10
+ellps=GRS80 +step +proj=unitconvert +xy_in=rad
+xy_out=deg
```

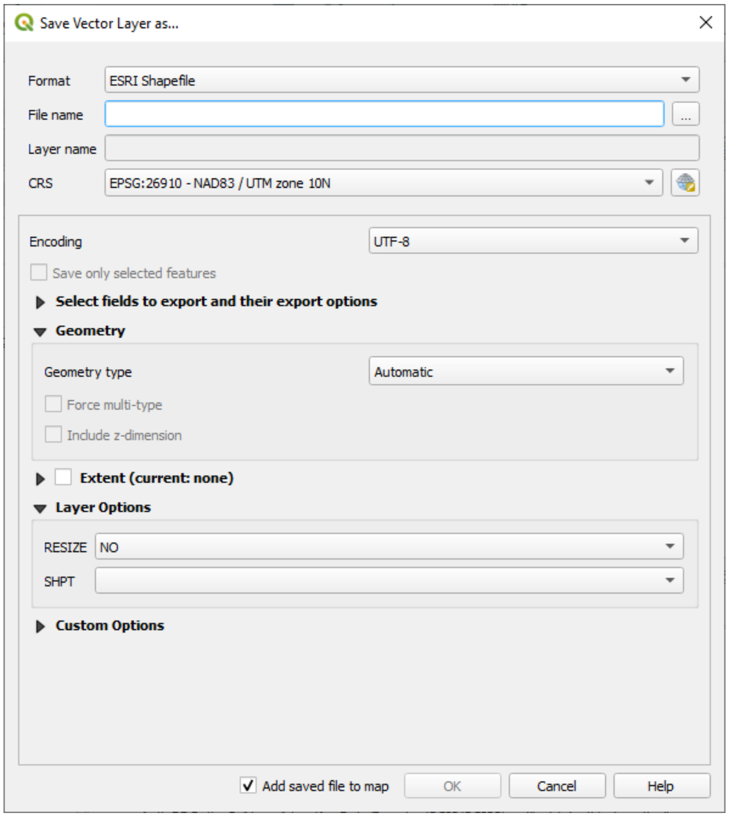
□ Show superseded transforms ☑ Allow fallback transforms if preferred operation fails □ Make default

Figure 2-13f Exporting the layer



```
:39    INFO    Application state:
FIX_PATH env var: C:/PROGRA~1/QGIS3~1.16/apps/qg
PROGRA~1/QGIS3~1.16/apps/qgis-ltr
: C:/PROGRA~1/QGIS3~1.16/apps/qgis-ltr/plugins
ata Path: C:/PROGRA~1/QGIS3~1.16/apps/qgis-ltr/.
qgis-ltr/./res
gis-ltr/./svg
\default/sv
r/./resource
User DB Pa
Auth DB Pa                                        QGIS3\prof
```

Figure 2-13g Click the *Browse* (3 dots) icon to specify a name and location for the shapefile you are creating



Save Vector Layer as...

Format: ESRI Shapefile

File name:

Layer name:

CRS: EPSG:26910 - NAD83 / UTM zone 10N

Encoding: UTF-8

☐ Save only selected features

▶ Select fields to export and their export options

▼ Geometry

Geometry type: Automatic

☐ Force multi-type

☐ Include z-dimension

▶ ☐ Extent (current: none)

▼ Layer Options

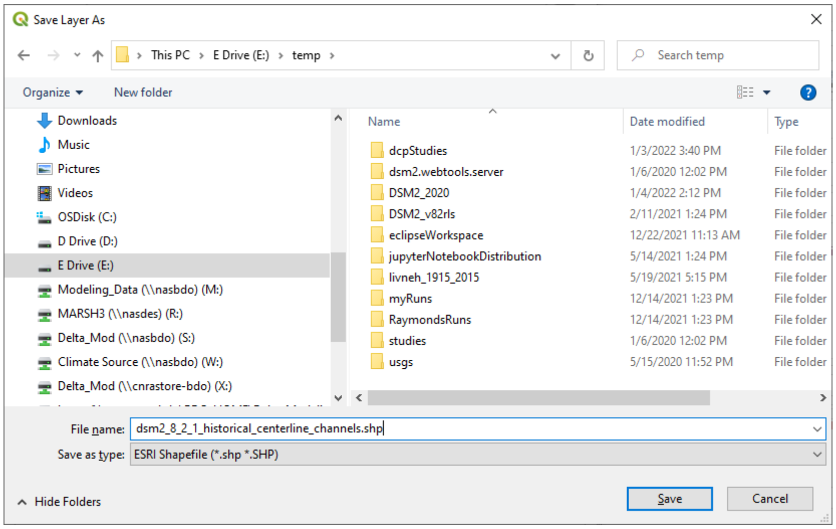
RESIZE: NO

SHPT:

▶ Custom Options

☑ Add saved file to map

Figure 2-13h Use the file selector dialog to specify the name and location of the shapefile



Save Layer As

This PC › E Drive (E:) › temp ›

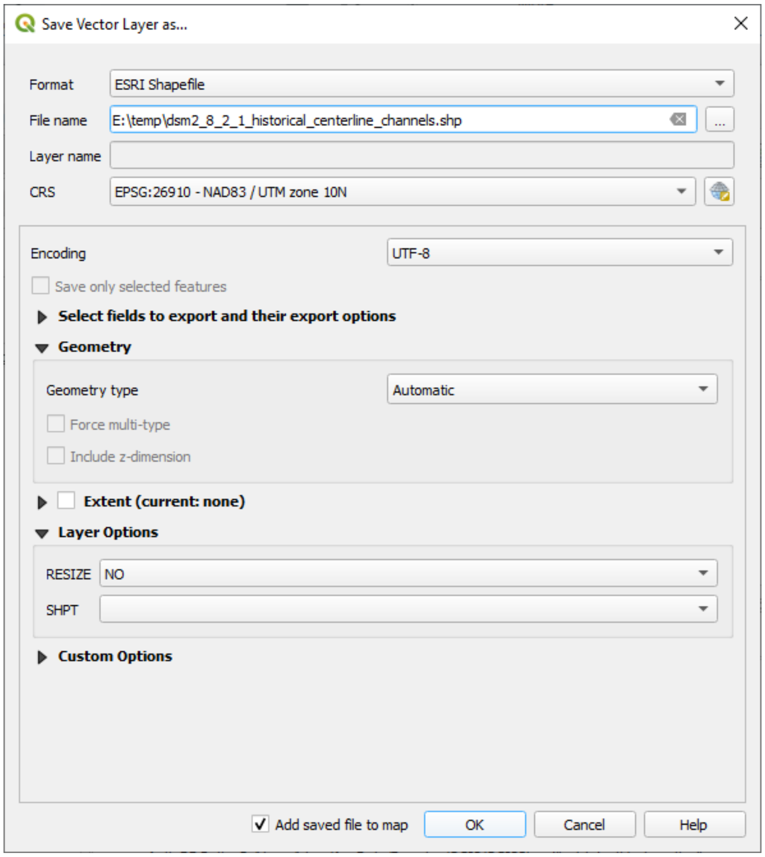
Search temp
Organize
New folder
Downloads
Music
Pictures
Videos
OSDisk (C:)
D Drive (D:)
E Drive (E:)
Modeling_Data (\\nasbdo) (M:)
MARSH3 (\\nasdes) (R:)
Delta_Mod (\\nasbdo) (S:)
Climate Source (\\nasbdo) (W:)
Delta_Mod (\\cnrastore-bdo) (X:)

Name	Date modified	Type
dcpStudies	1/3/2022 3:40 PM	File folder
dsm2.webtools.server	1/6/2020 12:02 PM	File folder
DSM2_2020	1/4/2022 2:12 PM	File folder
DSM2_v82rls	2/11/2021 1:24 PM	File folder
eclipseWorkspace	12/22/2021 11:13 AM	File folder
jupyterNotebookDistribution	5/14/2021 1:24 PM	File folder
livneh_1915_2015	5/19/2021 5:15 PM	File folder
myRuns	12/14/2021 1:23 PM	File folder
RaymondsRuns	12/14/2021 1:23 PM	File folder
studies	1/6/2020 12:02 PM	File folder
usgs	5/15/2020 11:52 PM	File folder

File name: dsm2_8_2_1_historical_centerline_channels.shp

Save as type: ESRI Shapefile (*.shp *.SHP)

Figure 2-13i Click the *OK* button to create the shapefile



Save Vector Layer as...

Format: ESRI Shapefile

File name: E:\temp\dsm2_8_2_1_historical_centerline_channels.shp

Layer name:

CRS: EPSG:26910 - NAD83 / UTM zone 10N

Encoding: UTF-8

☐ Save only selected features

▶ Select fields to export and their export options

▼ Geometry

Geometry type: Automatic

☐ Force multi-type

☐ Include z-dimension

▶ ☐ Extent (current: none)

▼ Layer Options

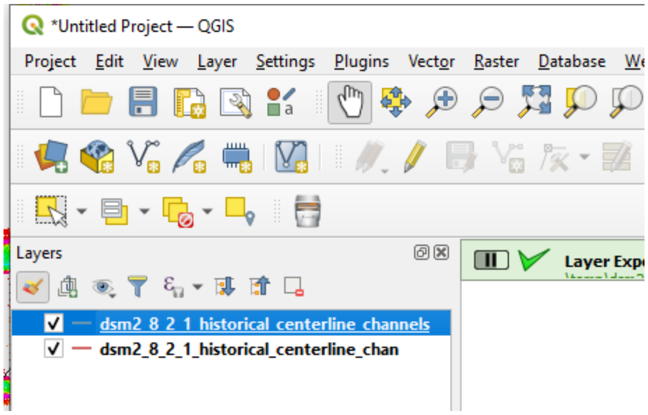
RESIZE: NO

SHPT:

▶ Custom Options

☑ Add saved file to map

Figure 2-13j The new shapefile is displayed underneath the layer created from the WKT file



*Untitled Project — QGIS

☑ dsm2_8_2_1_historical_centerline_channels

☑ dsm2_8_2_1_historical_centerline_chan

2.9 Products

The DSM2 georeferenced grid maps are available on the CNRA Open Data web site at https://data.cnra.ca.gov/dataset/dsm2-georeferenced-model-grid, in the following formats:

1. Adobe Portable Document Format (PDF): The PDF version of the grid map is recommended for most users. The PDF documents are created with ArcGIS Pro, and when viewed with Adobe Acrobat, retain the ability to toggle the display of layers and their symbols using the Layers panel, accessed by clicking the Layers icon (Figure 2-14a). Adobe Acrobat also has a search feature that some users may find useful (Figure 2-14b). Select *Edit-Find* and enter text describing the feature for which you are searching. The text could be the name of a body of water, road, city, or other feature names. If Adobe Acrobat is able to find the text anywhere in the layers, the map view will be panned to the text that it finds. The PDF version of the grid map is available in two formats:
    a. Single zoom level PDF grid map (Figure 2-14c): In this format, the entire grid map is displayed at the same zoom level.
    b. Multiple zoom level PDF grid map (Figure 2-14d): In this format, the grid map is broken up into sections. This grid map may be a better choice for printing.
2. GIS Shapefiles: A zip archive containing just the GIS shapefiles.
3. Map packages for ArcGIS Pro and ArcGIS Desktop: Map packages contain shapefiles and symbology, and both load easily into ArcGIS Pro and ArcGIS Desktop. In ArcGIS Pro or Desktop, you can right click on a layer and view the attribute table for that layer (Figure 2-15a). Double clicking on an item in the attribute table will pan the map view to the corresponding feature (Figure 2-15b).

Figure 2-14a Clicking the *Layers* button in Adobe Acrobat opens the *Layers* panel, enabling you to toggle the display of layers and their corresponding symbols

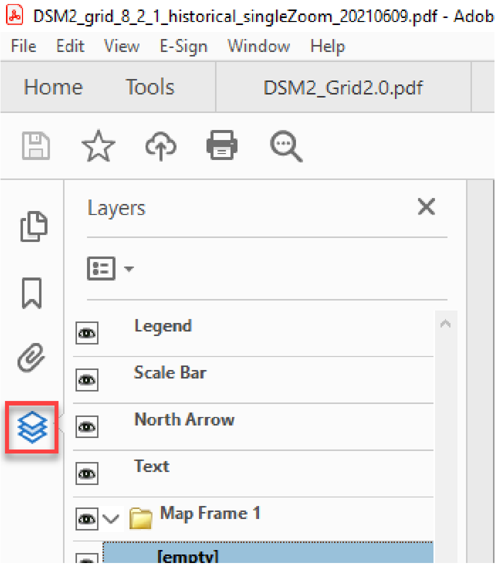


Figure 2-14b Searching for the text "Rio Vista" in the grid map using Adobe Acrobat

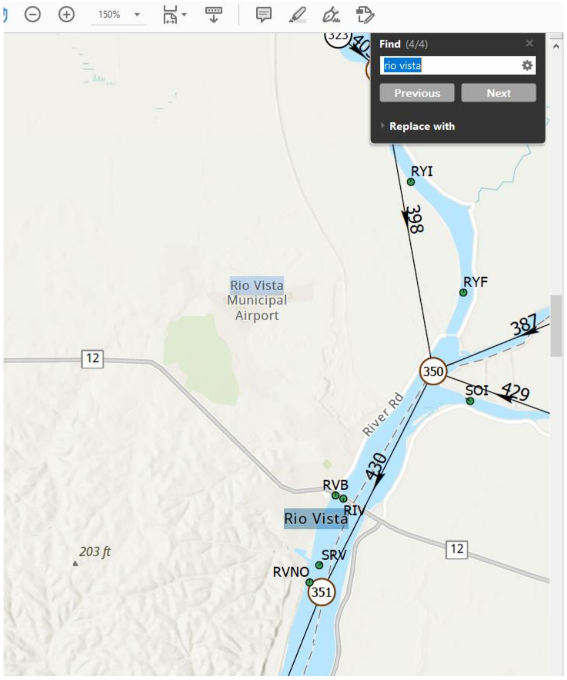


Figure 2-14c A single zoom level version of the PDF grid map

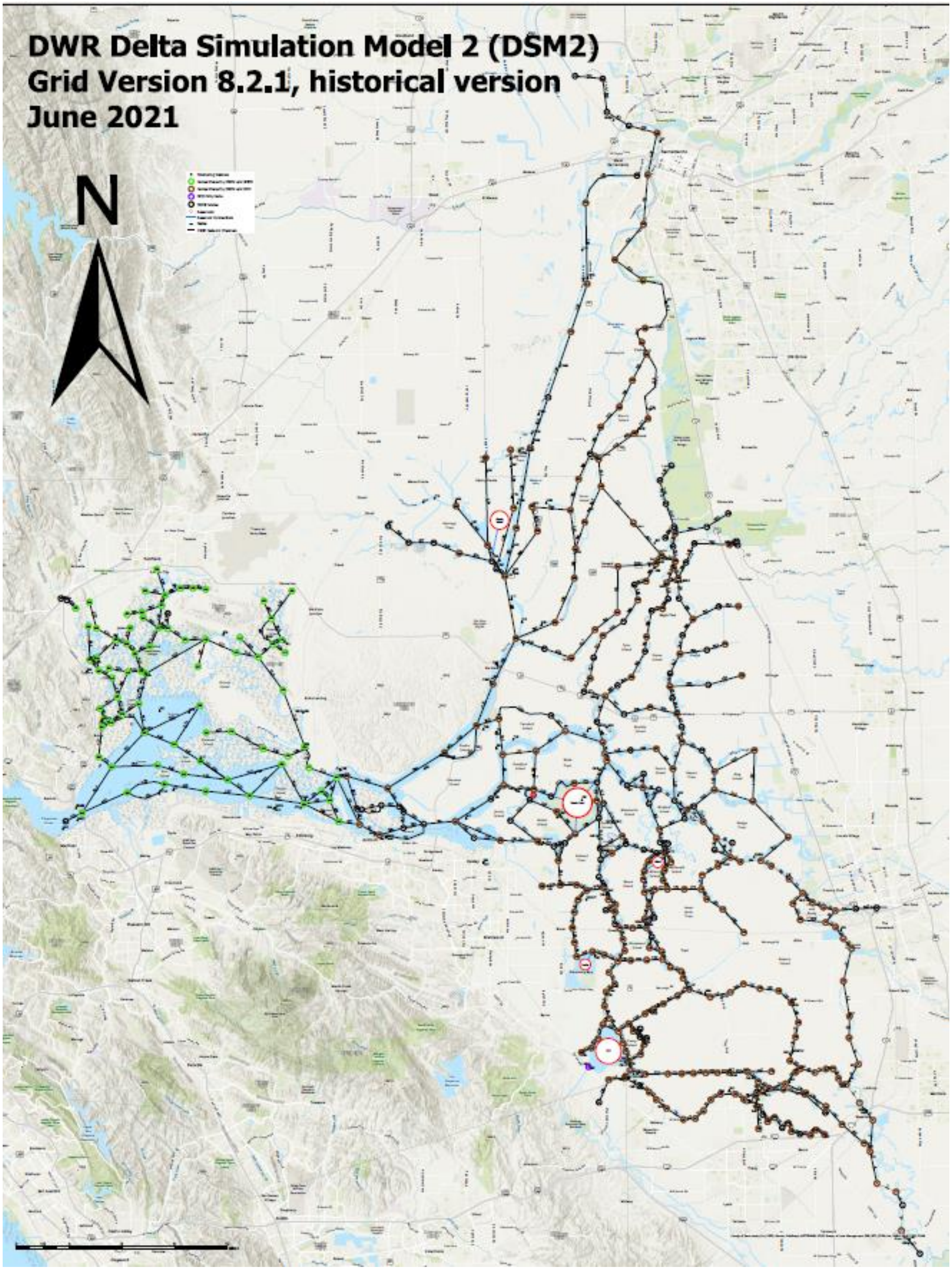

DWR Delta Simulation Model 2 (DSM2)
Grid Version 8.2.1, historical version
June 2021
N

Figure 2-14d A multiple zoom level version of the PDF grid map, designed for printing

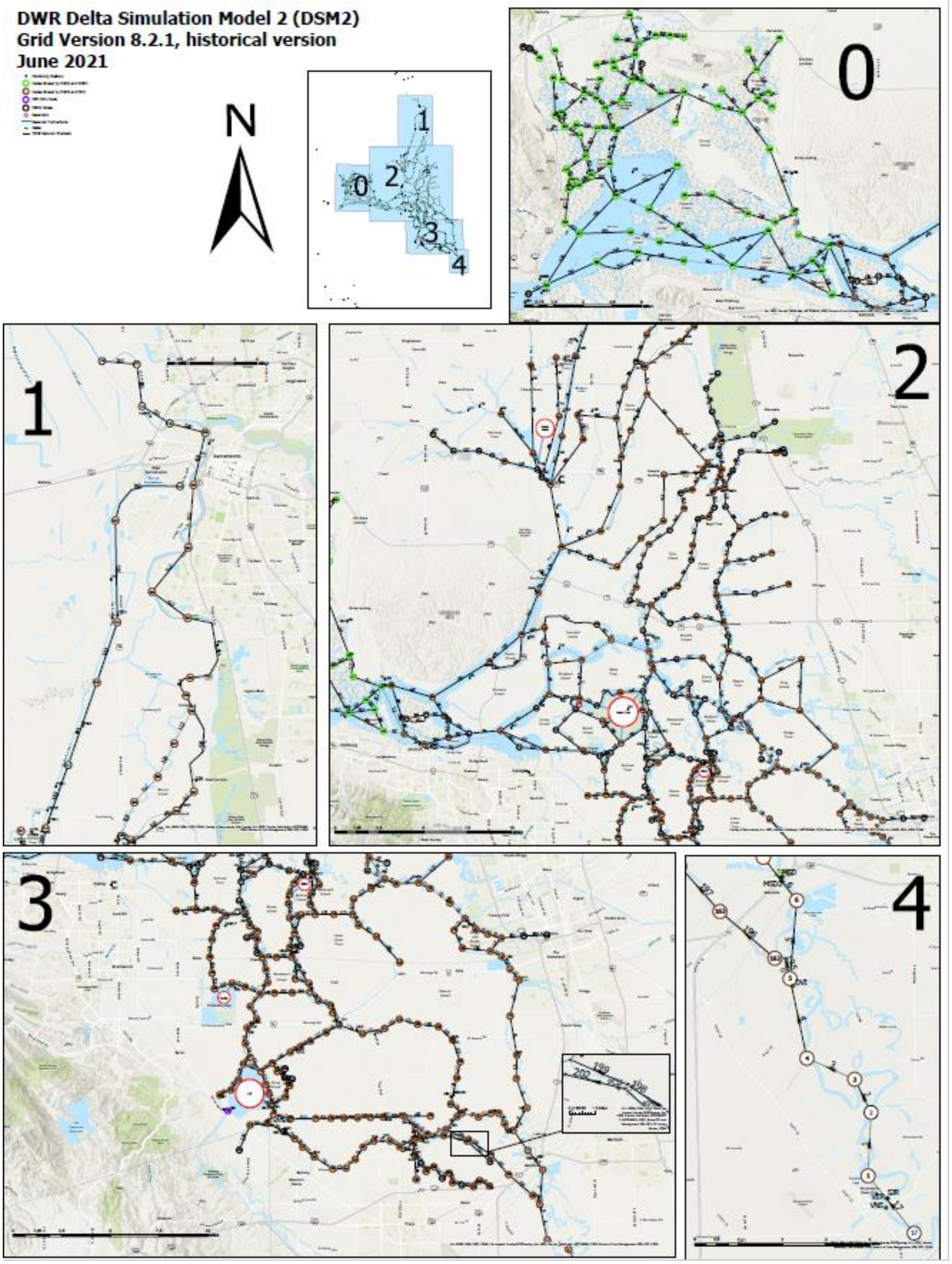

DWR Delta Simulation Model 2 (DSM2)
Grid Version 8.2.1, historical version
June 2021
N
0
1
2
3
4

Figure 2-15a To view an attribute table for a layer in ArcGIS, right click on the layer and select *Attribute Table*

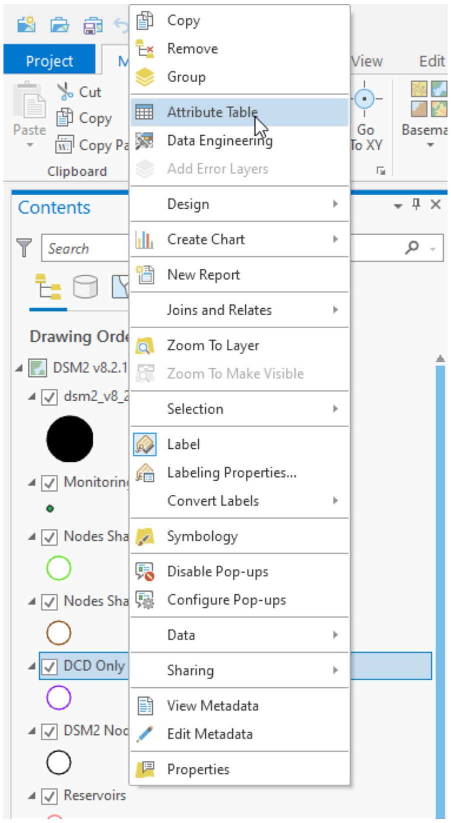
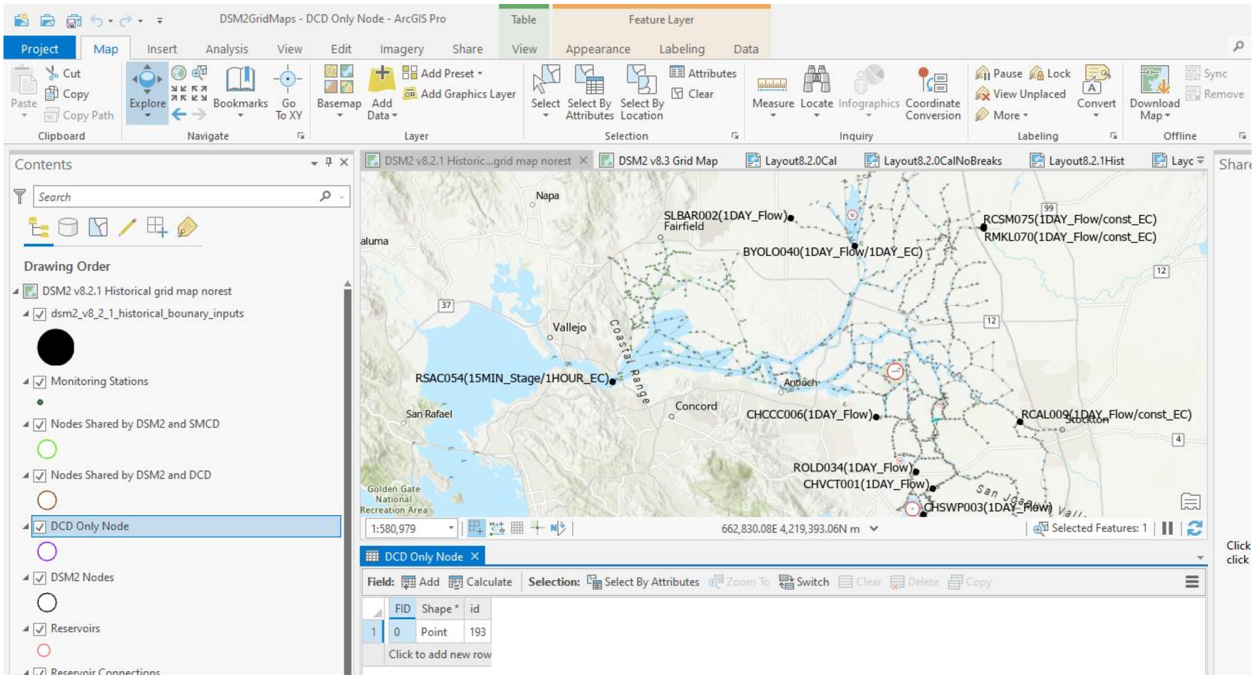


Figure 2-15b DSM2 grid map in ArcGIS, with attribute table displayed on the lower right



	FID	Shape *	id
1	0	Point	193

2.10 Grid Map Limitations

The accuracy of the georeferenced information used to create DSM2 input is limited. Features, such as channel centerlines and nodes, are typically created by clicking on the CSDP plan view, at a zoom level similar to that shown in Figure 2-12 (one of the figures above showing the CSDP plan view). Using a higher zoom level might in some cases result in more accurate coordinates, but this is not considered to be necessary for DSM2. One reason is that some DSM2 features, such as nodes and channel centerline endpoints, do not always represent exact locations. Nodes are placed in locations that are determined visually by the CSDP user, and their location is intended to be the middle of the channel or junction.

Here is a list of assumptions that users should not make regarding the information in the grid maps:

- For reasons mentioned above, none of the locations of the grid map features, such as nodes, channels, reservoirs, and gates should be assumed to have a high level of accuracy.
- For DSM2 v8.2 and prior versions, the grid maps should not be used to determine channel lengths or cross-section locations. DSM2 channel

lengths and some cross-sections in v8.2 and all prior versions have been determined without using the CSDP files. There are a number of discrepancies between these lengths and the CSDP centerline lengths. Future releases of DSM2 will resolve these discrepancies.

- The locations of gates in the Grid Location layer are adjusted to prevent the gate symbol from overlapping the node symbol.
- The locations of gates in the Actual Location layer are also approximate, not used by DSM2, and are just for reference.

2.11 References

Tom B, He M, Sandhu N. 2000. “Chapter 2: DSM2 GIS Reference.” In: Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 41st Annual Progress Report to the State Water Resources Control Board. California Department of Water Resources.

Tom B. 1998. “Chapter 6: Cross-Section Development Program.” In: Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 19th Annual Progress Report to the State Water Resources Control Board. California Department of Water Resources.

43rd Annual Progress Report
June 2022

Chapter 3
DSM2 Water Temperature Modeling Input Extension: 1922 – 2015

Authors: Minxue He, Yu Zhou, Han Sang Kim, Parviz Nader-Tehrani, and Nicky Sandhu
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Contents

Figures

Tables

Chapter 3 DSM2 Water Temperature Modeling Input Extension: 1922–2015

3.1 Background

The water quality module of the Delta Simulation Model II (DSM2 QUAL) was previously calibrated and validated during the period from 1990–2008 to simulate water temperature in the Sacramento-San Joaquin Delta (Delta) in 2011 (Resources Management Associates 2011). In a follow-up study, the simulation period was extended to 2012 (Resources Management Associates 2015). Recently, the Delta Modeling Section (DMS) was tasked to extend the water temperature simulation period to water years 1922–2015 to align with the current simulation period used by DWR's water resources planning model, CalSim3.

This document describes the input data requirements for modeling Delta water temperature via DSM2 QUAL and the methods applied to assemble or derive these data for the extended period.

3.2 Data requirement

DSM2 QUAL requires meteorological, water temperature, and flow boundary condition data to simulate temperature across the Delta. Five meteorological inputs are needed at a single location where conditions are representative of the Delta. In this case, a location in the central Delta at 38.0 N latitude and 121.5 W longitude was selected. Flow and water temperature are needed at three boundary locations of the Delta: Sacramento River at Freeport (Freeport), San Joaquin River at Vernalis (Vernalis), and Martinez. Effluent flow and water temperature are needed at 12 boundary locations. Figure 3-1 illustrates these locations as well as the DSM2 model boundary.

Figure 3-1 Approximate locations of temperature boundaries, effluent boundaries, and meteorological inputs for DSM2 QUAL water temperature modeling

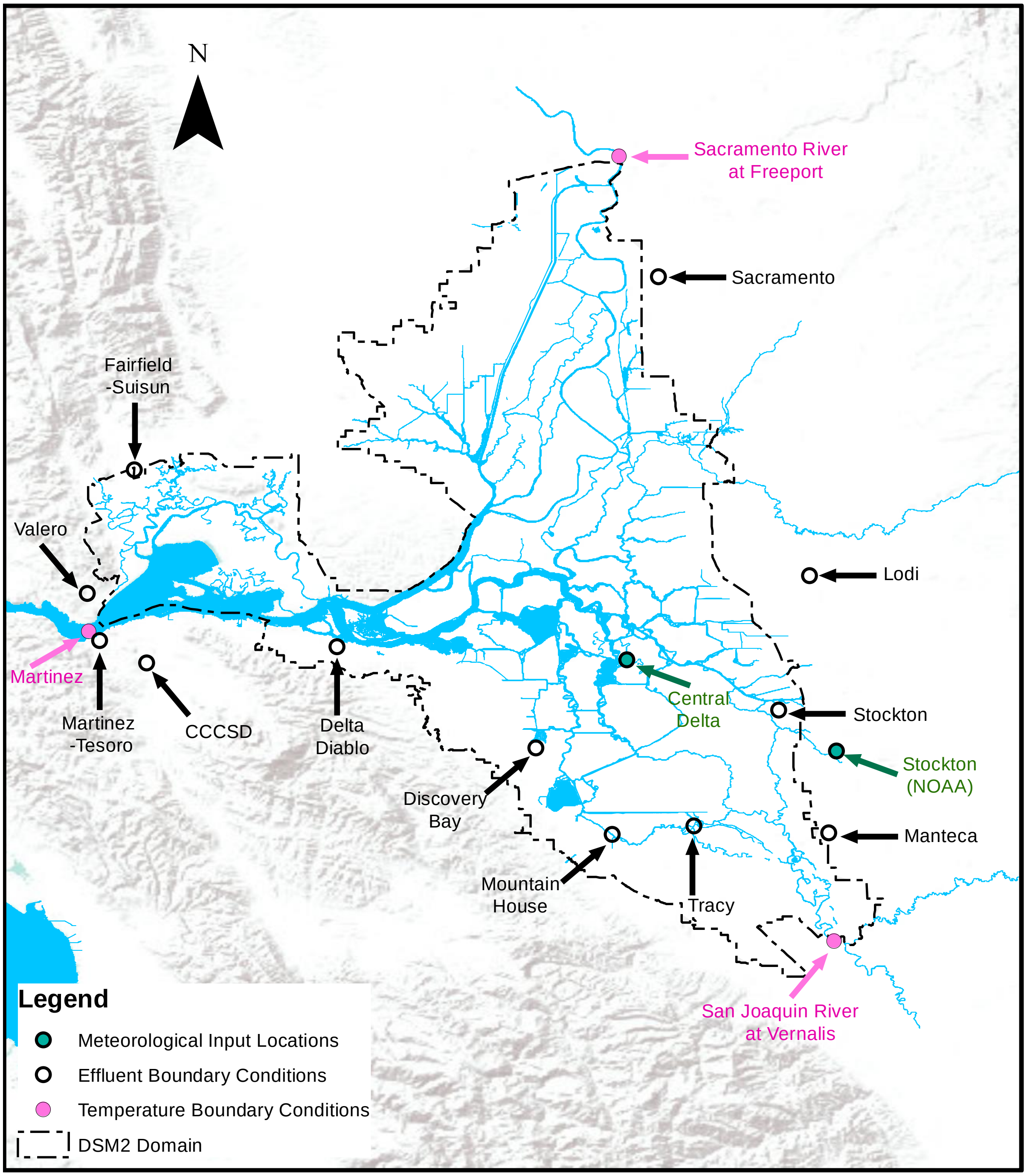


Table 3-1 lists five meteorological input variables and the location and sources of these data for the extended period. Data from gridded datasets Livneh and NOAA-CIRES-DOE 20th Century Reanalysis V3 reanalysis (NOAA reanalysis) are applied to derive these meteorological inputs. To our knowledge, these two datasets are the only readily available and widely used datasets with desirable temporal resolution (i.e., daily or finer) and record period (i.e., at least covering the extended simulation period from 1922–2015). The Livneh hydrometeorological dataset consists of daily precipitation and maximum and minimum air temperature at 1/16-degree spatial scale (approximately 6km by 6km) for the continental US, southern Canada, and Mexico from 1950–2013 (Livneh et al. 2015). These gridded temperature and precipitation data are interpolated from observations at about 20,000 weather stations. The developers later extended the dataset to cover the period from 1915 to 2015. The National Oceanic and Atmospheric Administration (NOAA) reanalysis dataset consists of a wide range of meteorological variables, including wind speed and atmospheric pressure at a three-hourly temporal scale and 1 degree (approximately 100 km by 100 km) spatial scale from 1836 to 2015 (Slivinski et al. 2019). For each variable at a specific time at a specific location, the reanalysis dataset contains 80 individual ensemble members. This study uses the mean value of these ensemble members. In the Delta, the wind speed varies widely, while the atmospheric pressure is relatively uniform across different locations (Resources Management Associates 2011). Because of this, NOAA wind speed analysis data were bias corrected using wind speed observations recorded at a NOAA ground station located at Stockton, while no corrections were applied to NOAA atmospheric pressure reanalysis. The sources of these data are provided in the "Data Sources" Section.

Table 3-1 Meteorological inputs required for Delta water temperature modeling

Input variables	Location	Source	Time Step
Dry Bulb temperature	(38.0N, 121.5W)	Livneh dataset	Hourly
Wet Bulb temperature	(38.0N, 121.5W)	Derived	Hourly
Cloud cover	(38.0N, 121.5W)	Derived	Hourly
Atmospheric pressure	(38.0N, 121.5W)	NOAA reanalysis	Hourly
Wind speed	(38.0N, 121.5W)	NOAA reanalysis	Hourly

The boundary conditions are tabulated in Table 3-2. The Daily water temperature during the extended period (i.e., 1922–2015) at three DSM2 boundary locations was derived via artificial neural networks (ANNs). The process is detailed in Section 3.2. Monthly effluent flow and water temperature boundaries come from either historical data patterns or the CalSim3 model, which is further explained in Section 3.3.

Table 3-2 Boundary conditions required for Delta water temperature modeling

Input variables	Location	Source	Time Step
Water temperature	Freeport	Derived	Daily
Water temperature	Vernalis	Derived	Daily
Water temperature	Martinez	Derived	Daily
Water temperature	Stockton	Historical	Monthly
Water temperature	Sacramento	Historical	Monthly
Water temperature	Tracy	Historical	Monthly
Water temperature	Manteca	Historical	Monthly
Water temperature	Lodi	Historical	Monthly
Water temperature	CCCSD	Historical	Monthly
Water temperature	Fairfield-Suisun	Historical	Monthly
Water temperature	Valero	Historical	Monthly
Water temperature	Martinez-Tesoro	Historical	Monthly
Water temperature	Delta Diablo	Historical	Monthly
Water temperature	Discovery Bay	Historical	Monthly
Water temperature	Mountain House	Historical	Monthly
Flow	Stockton	CalSim3	Monthly
Flow	Sac Waste	CalSim3	Monthly
Flow	Tracy	CalSim3	Monthly
Flow	Manteca	CalSim3	Monthly
Flow	Lodi	CalSim3	Monthly
Flow	CCCSD	Historical	Monthly
Flow	Fairfield-Suisun	Historical	Monthly
Flow	Valero	Historical	Monthly
Flow	Martinez-Tesoro	Historical	Monthly
Flow	Delta Diablo	Historical	Monthly
Flow	Discovery Bay	Historical	Monthly
Flow	Mountain House	Historical	Monthly

3.3 Methodology

3.3.1 Meteorological data generation

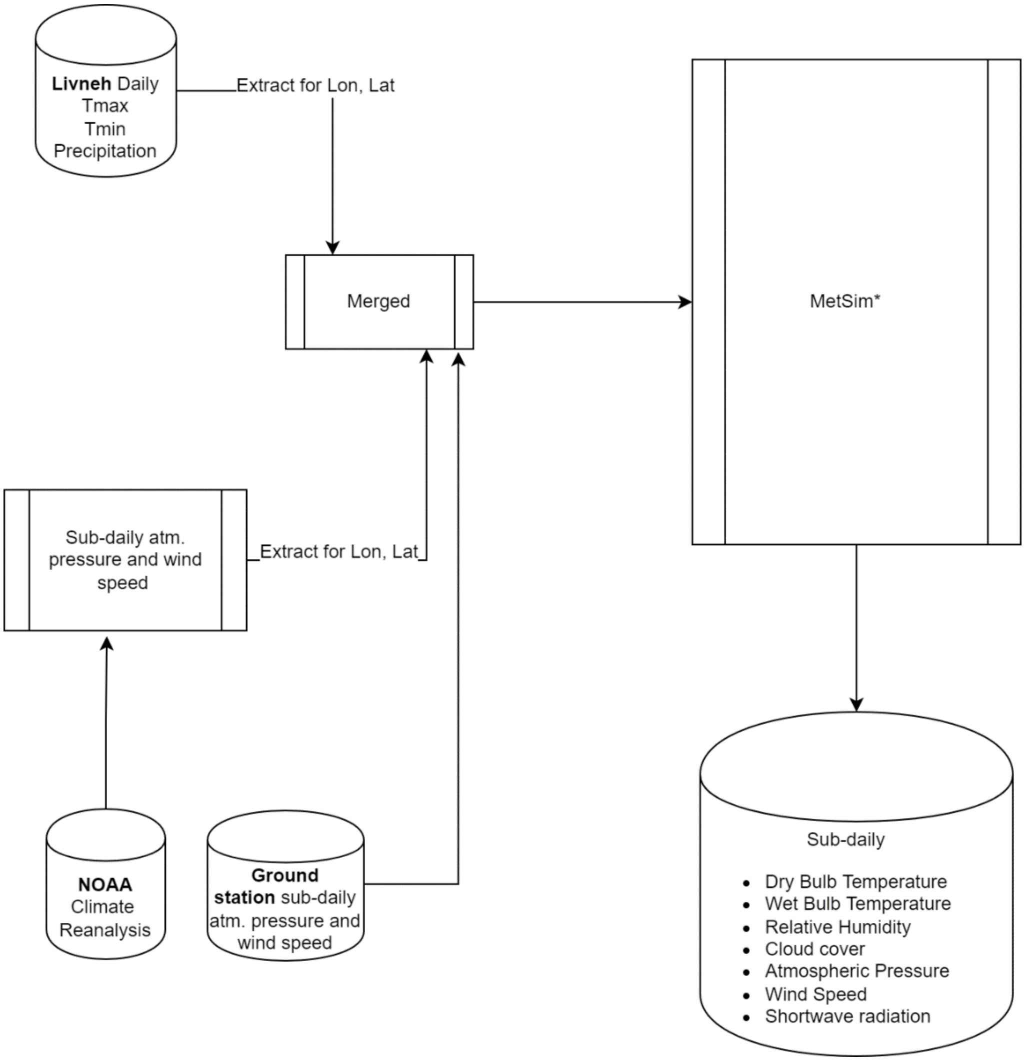
In this study, most of the meteorological inputs listed in Table 3-1 were generated via the meteorology simulator, MetSim (Bennett et al. 2020). MetSim contains three main modules that conduct solar geometry computation, meteorological simulation, and temporal disaggregation, respectively. The solar geometry module identifies the daylength, transmittance of the atmosphere, daily potential radiation, and the fraction of daily radiation received at the top of atmosphere. Output of the solar geometry module drives the meteorological simulation module along with the input forcings, which typically include daily precipitation and maximum and minimum air temperatures. The meteorological simulation module calculates daily mean temperature, vapor pressure, shortwave radiation, cloud cover fraction, and potential evapotranspiration, etc. Daily data from input forcings or the meteorological simulation module can be disaggregated down to sub-daily (e.g., positive factors of 24) values via the temporal disaggregation module. In addition to variables output from the meteorological simulation module, the disaggregation module also generates sub-daily relative and specific humidity, precipitation, longwave radiation, and wind speed. Temperature is disaggregated using a Hermite polynomial interpolation method with user-specified times when the daily maximum and minimum temperatures occur. Vapor pressure is disaggregated by linearly interpolating between the vapor pressure and the saturation vapor pressure at the times when the daily minimum temperature occurs. The shortwave radiation at a given timestep is calculated as a fraction of the total daily shortwave radiation (which is calculated from the solar geometry module). Air pressure disaggregation is based on the disaggregated temperature and a user-specified elevation value at the study location. Specific and relative humidity in each timestep is determined from the disaggregated temperature and air pressure data. Wind speed, if provided as part of the input forcings, is assumed to be constant during the day. Precipitation can be disaggregated uniformly throughout the day or via a triangle method. For more detailed explanation on MetSim, please see Bennett et al. (2020) as well as the references cited there.

To generate all necessary meteorological inputs for DSM2 QUAL at the desirable (hourly) time step, the current study made several modifications to MetSim (Figure 3-2). Firstly, MetSim does not calculate the wet-bulb temperature, so the following equation from Stull (2011) is added to MetSim to derive the wet-bulb temperature as a function of temperature and relative humidity:

$$\begin{aligned} T_{wb} = {} & Tatan\left(0.151977\sqrt{RH + 8.313659}\right) + \operatorname{atan}(T + RH) - \operatorname{atan}(RH - 1.6763331) \\ & + 0.00391838RH^{\frac{3}{2}}\operatorname{atan}(0.023101RH) \\ & - 4.686035 \qquad (1) \end{aligned}$$

where T and T_{wb} denote temperature and wet-bulb temperature in degrees Celsius (°C), and RH represents relative humidity (%), which is an output of the MetSim program. Yet, MetSim-derived relative humidity tends to have a dry bias (under-prediction). So before being applied to calculate the wet-bulb temperature, MetSim-derived relative humidity is bias-corrected. Secondly, instead of assuming constant wind speed throughout the day, a linear interpolation method is applied to disaggregate the bias-corrected NOAA reanalysis wind speed input. Thirdly, because MetSim-simulated cloud cover differs from the observed cloud cover, a module is added to bias-correct cloud cover simulation based on cloud cover recorded at Stockton station. When bias-correcting NOAA reanalysis wind speed, a constant ratio is applied every month to maintain consistency with the general approach applied in wind speed adjustment during the model calibration process (Resources Management Associates 2011, 2015). When bias-correcting relative humidity and cloud cover, a quantile mapping approach is employed.

Figure 3-2 Schematic of meteorological input generation process

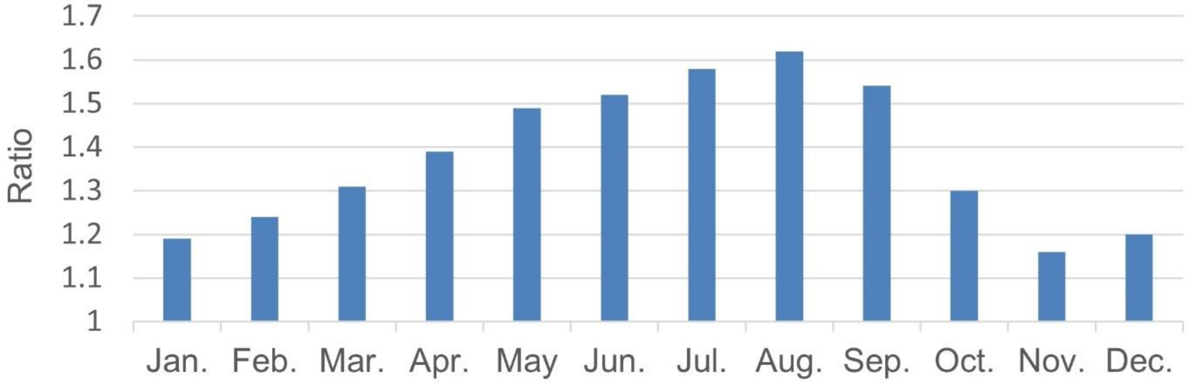


Note: The modified MetSim program is applied.

The process of bias-correcting wind speed is as follows:

- Firstly, during the period when wind speed observations are available (1973–2015), 3-hourly NOAA reanalysis wind speed is aggregated to monthly. In a similar way, observed wind speed at Stockton is aggregated to monthly. Long-term monthly mean values are calculated for both datasets.
- Secondly, a monthly ratio for each month is calculated as mean observed monthly wind speed divided by the mean reanalysis wind speed in that month, as illustrated in Figure 3-3 below.

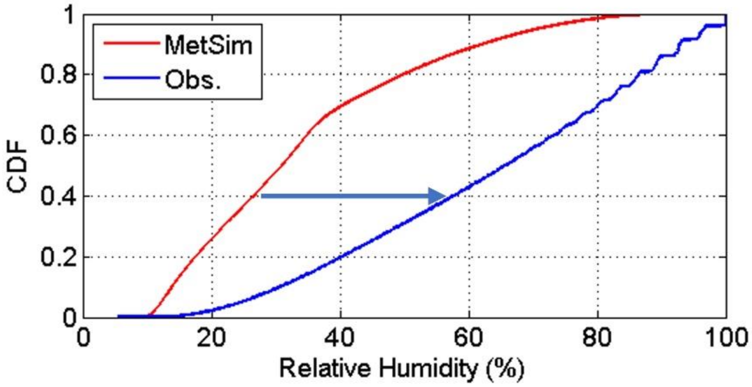
Figure 3-3 Monthly ratios determined for bias-correcting wind speed reanalysis data



- Thirdly, for reanalysis of wind speed in a specific month from water years 1915–2015, the monthly ratio corresponding to that month is applied to scale up or scale down all the wind speed reanalysis data.
- Finally, the adjusted 3-hourly reanalysis wind speed during 1915–2015 from the previous step is disaggregated into hourly wind speed using a linear interpolation method.

In comparison, the quantile mapping approach adjusts the simulated data based on its percentile rather than the month when it’s recorded. This approach first identifies the cumulative distribution functions (CDF) of both the observed and the simulated values of the selected variable (e.g., relative humidity). Second, for a specific percentile (e.g., 40 percent), a corresponding ratio is calculated as the observed value at that percentile divided by the simulated value at that percentile (Figure 3-4). Next, this ratio is applied to adjust MetSim simulations during the target period (1922–2015) at that percentile. This process is then repeated for all other percentiles.

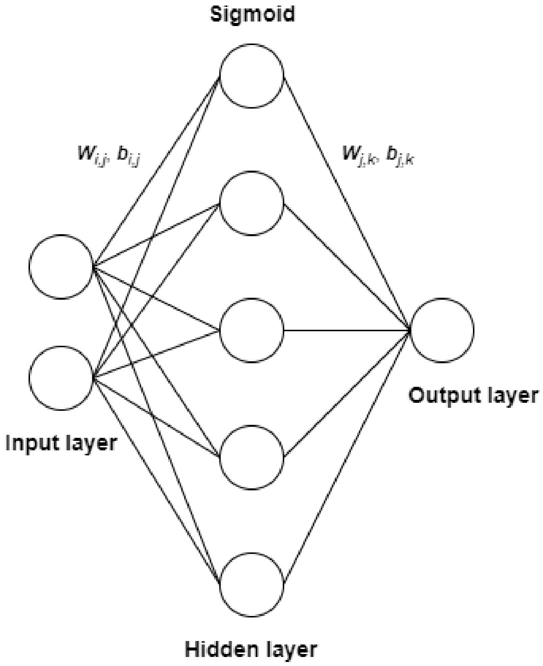
Figure 3-4 **Schematic illustrating the quantile mapping approach**



3.3.2 Water temperature boundary generation

Data-driven artificial neural networks (ANNs) are employed in deriving water temperature at three DSM2 boundary locations. An ANN utilizes a mathematical network structure to implicitly derive the relationships between the input variables (e.g., air temperature and solar radiation) and the output variables (e.g., water temperature). Among all ANN models developed and applied in the field of water resources engineering, multilayer perceptron (MLP) networks are probably the most popular (Maier et al. 2010). An MLP consists of an input layer, an output layer, and one or more hidden layer(s) (Figure 3-5). Each layer has one or multiple neurons. A neuron in a specific hidden layer obtains information from neurons in the previous layer and exports a transformation of the combined input information to neurons in the next layer. The connections between neurons in two adjacent layers are represented by linear weights. These weights are trainable parameters determined in the training process by minimizing the difference between network outputs and corresponding observations.

Figure 3-5 Schematic of the multilayer perceptron (MLP) structure applied in deriving water temperature at three Delta boundary locations



Note: $w_{i,j}$ and $b_{i,j}$ represent the linear weight and bias associated with the connection between two neurons in adjacent layers, respectively. Sigmoid is a common activation function that introduces non-linearity into the output of a neuron.

In the current study, a three-layer MLP is employed to derive water temperature boundaries. The hidden layer contains five neurons. Specifically, a separate MLP is developed for each of the three boundary locations (Freeport, Vernalis, and Martinez). During the training period (1990–2013), corresponding daily temperature observations from previous studies (Resources Management Associates 2011, 2015) and simulated solar radiation via empirical energy balance equations are utilized as inputs. During the simulation period (1922–2015), MetSim-generated temperature simulations (Section 3.1) and the simulated solar radiation are applied to drive the trained MLPs and generate daily water temperature at these three locations.

3.3.3 Effluent boundary generation

Effluent flow and temperature at 12 locations are required (Figure 3-1). Historical observations at these locations are sparse, so flow rate and flow temperature are assembled on a monthly time-scale. CalSim 3-simulated flows are directly utilized when produced at needed effluent locations (Table 3-2). At the remaining locations, flow discharge is assumed to follow a same monthly pattern every year from 1922–2015. For each month, the observed flow rates at that month during the historical period obtained in the Resources Management Associates (RMA) studies (Resources Management Associates 2011, 2015) are averaged to yield the corresponding flow rate. Flow temperatures at all 12 locations are derived in the same way.

3.4 Results

3.4.1 Meteorological inputs

Figure 3–6 to Figure 3-10 depict MetSim-generated (at the Central Delta location) versus observed (at Stockton) hourly meteorological variables, including dry-bulb temperature, wet-bulb temperature, air pressure, wind speed, and cloud cover. The results are shown for the times when the observations are available. It is evident that the simulated dry-bulb temperature (Figure 3-6), wet-bulb temperature (Figure 3-7), and air pressure (Figure 3-8) mimic the corresponding observations well. For each, the correlation between the simulated and observed is very high (i.e., over 0.9) while the bias between them is generally small (i.e., less than 1 percent).

In comparison, the correlation between simulated and observed wind speed is moderately strong (Figure 3-9). MetSim-derived wind speed generally under simulates the high range of wind speed (i.e., over 25 mph) and over simulates the low range of wind speed (i.e., less than 5 mph). Overall, though, the bias is still very small (1.3 percent).

For cloud cover (Figure 3-10), the correlation between the simulated and the observed is moderate (0.54). The bias between them is reasonably satisfactory (-5.1 percent). Notably, cloud cover observations contain only 10 values during the entire observation period, ranging from 0 to 1 in increments of 0.1. Therefore, the cumulative distribution function (CDF) curve of observed cloud cover has a stair-step shape (Figure 3-11). MetSim is able to generate continuous cloud cover values from 0 to 1. The corresponding CDF curve is smooth. Quantile-mapping a continuous CDF to a stair-step CDF limits the accuracy of the bias-correction.

Figure 3-6 MetSim-simulated and observed hourly dry-bulb temperature (T) from 1973–2015

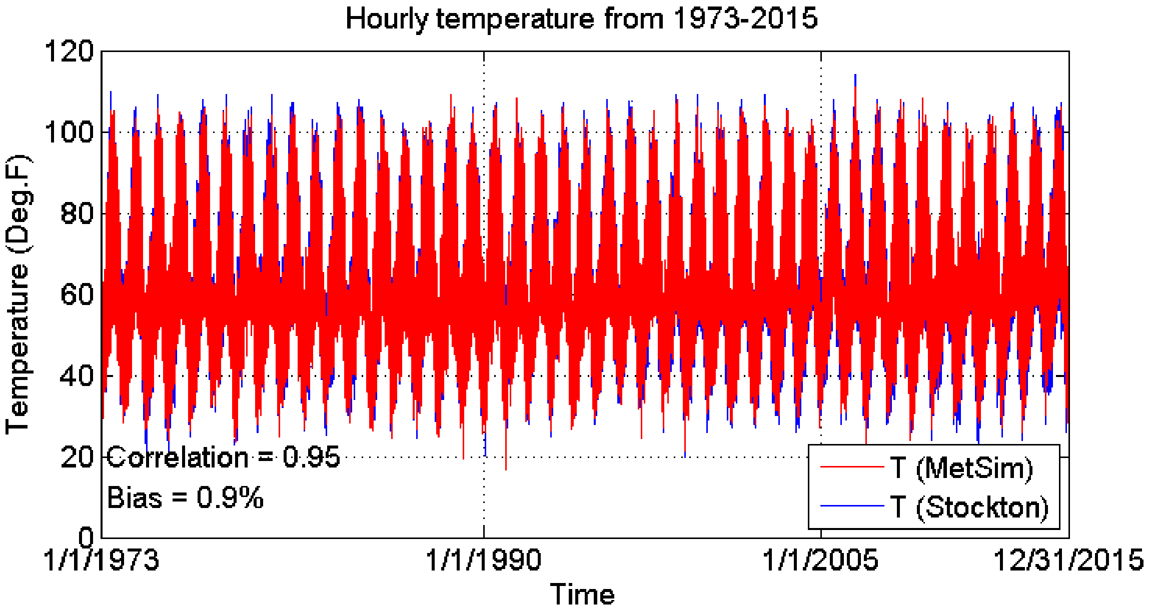


Figure 3-7 MetSim-simulated and observed hourly wet-bulb temperature (WBT) from 1973–2015

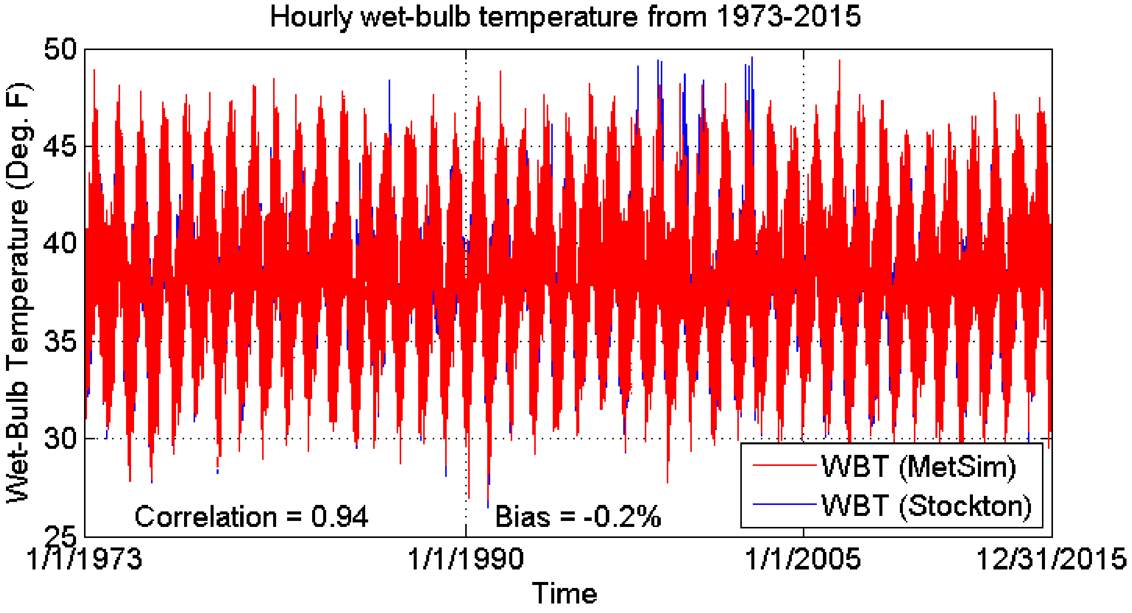


Figure 3-8 MetSim-simulated and observed hourly air pressure (AP) from 1973–2015

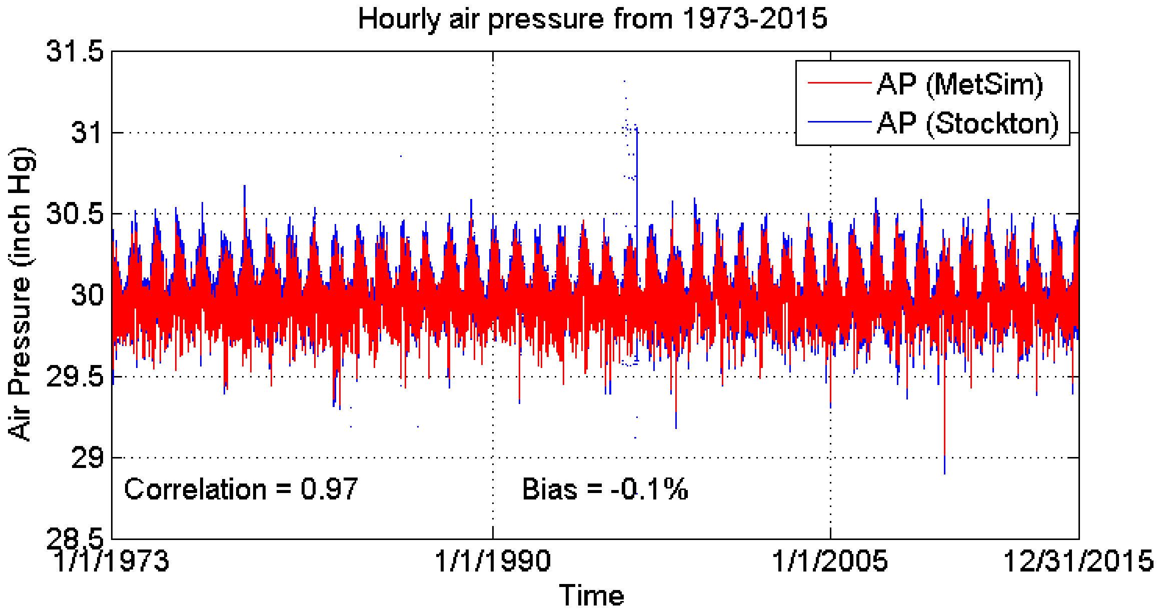


Figure 3-9 MetSim-simulated and observed hourly wind speed (WS) from 1973–2015

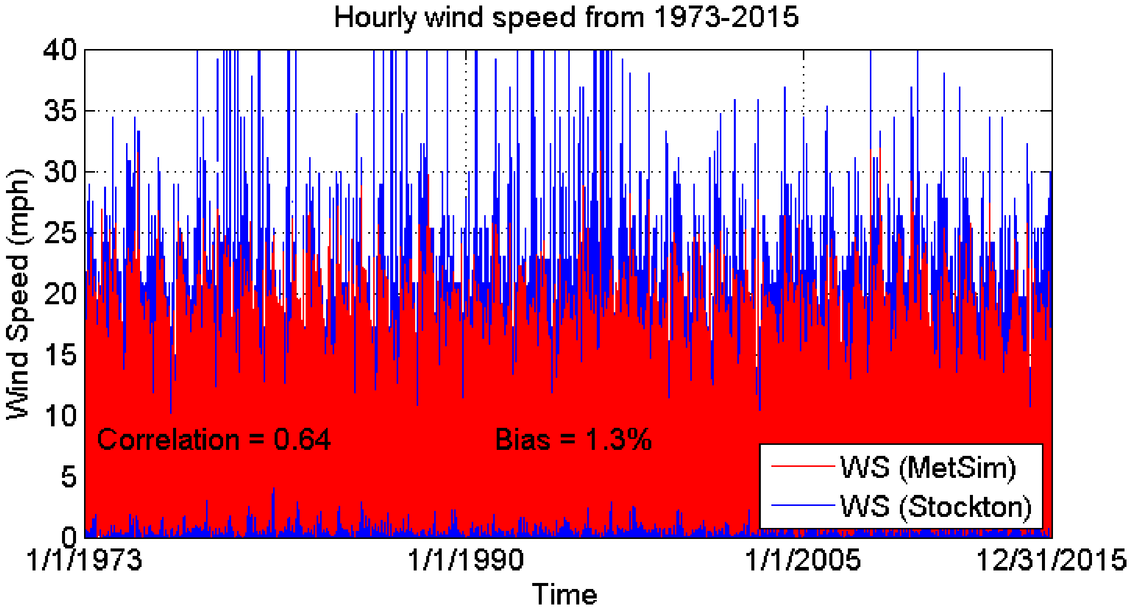


Figure 3-10 MetSim-simulated and observed hourly cloud cover (CC) from 1973–2015

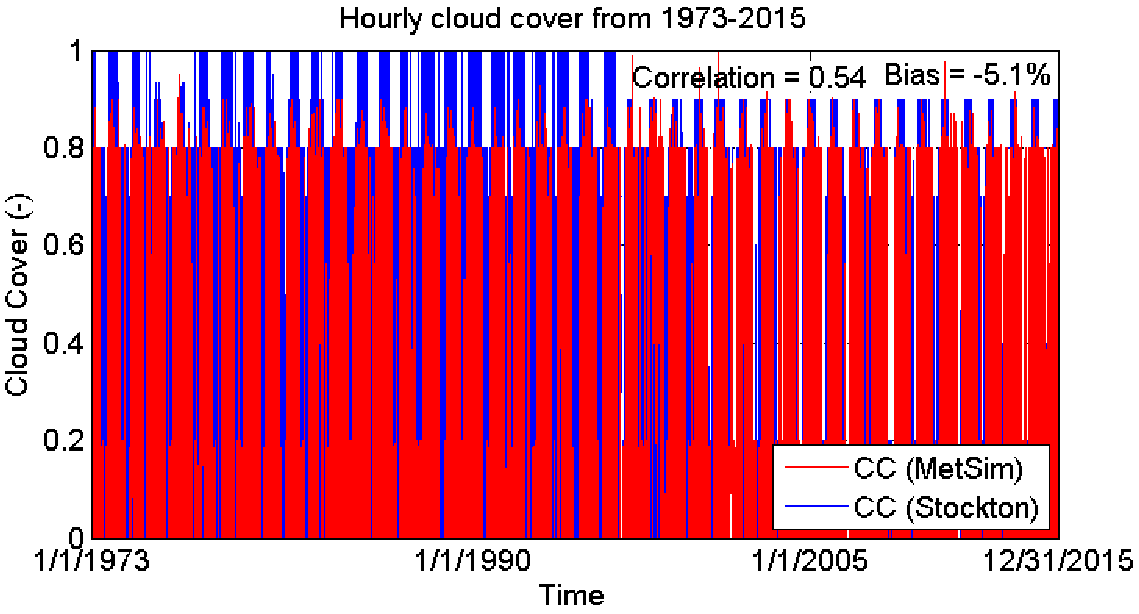
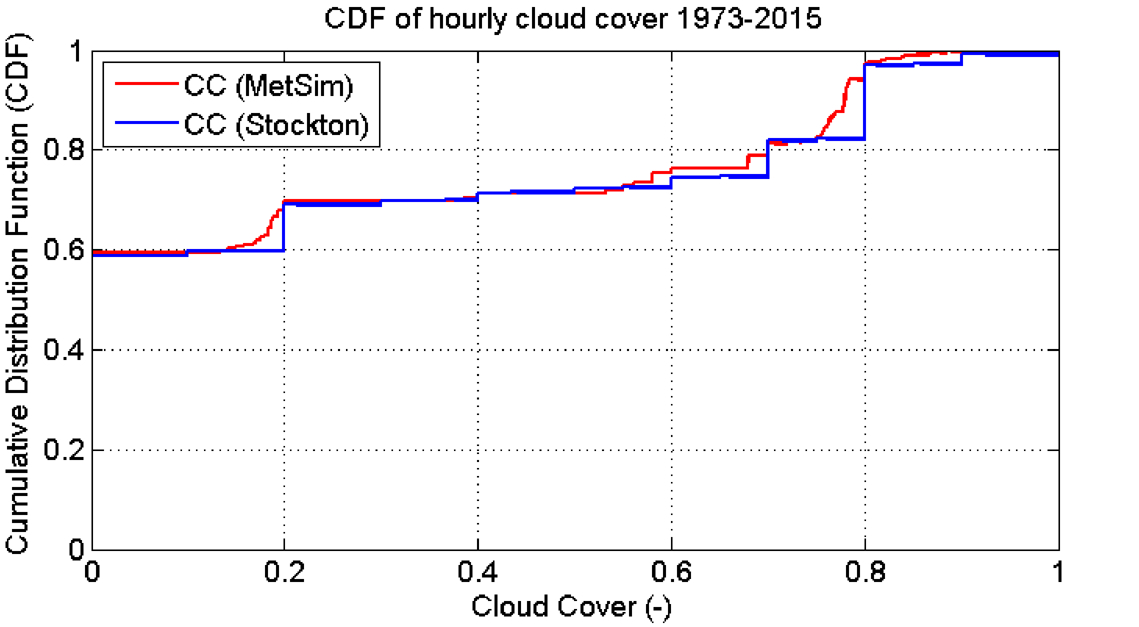


Figure 3-11 Cumulative distribution functions (CDFs) of MetSim-simulated and observed hourly cloud cover from 1973–2015

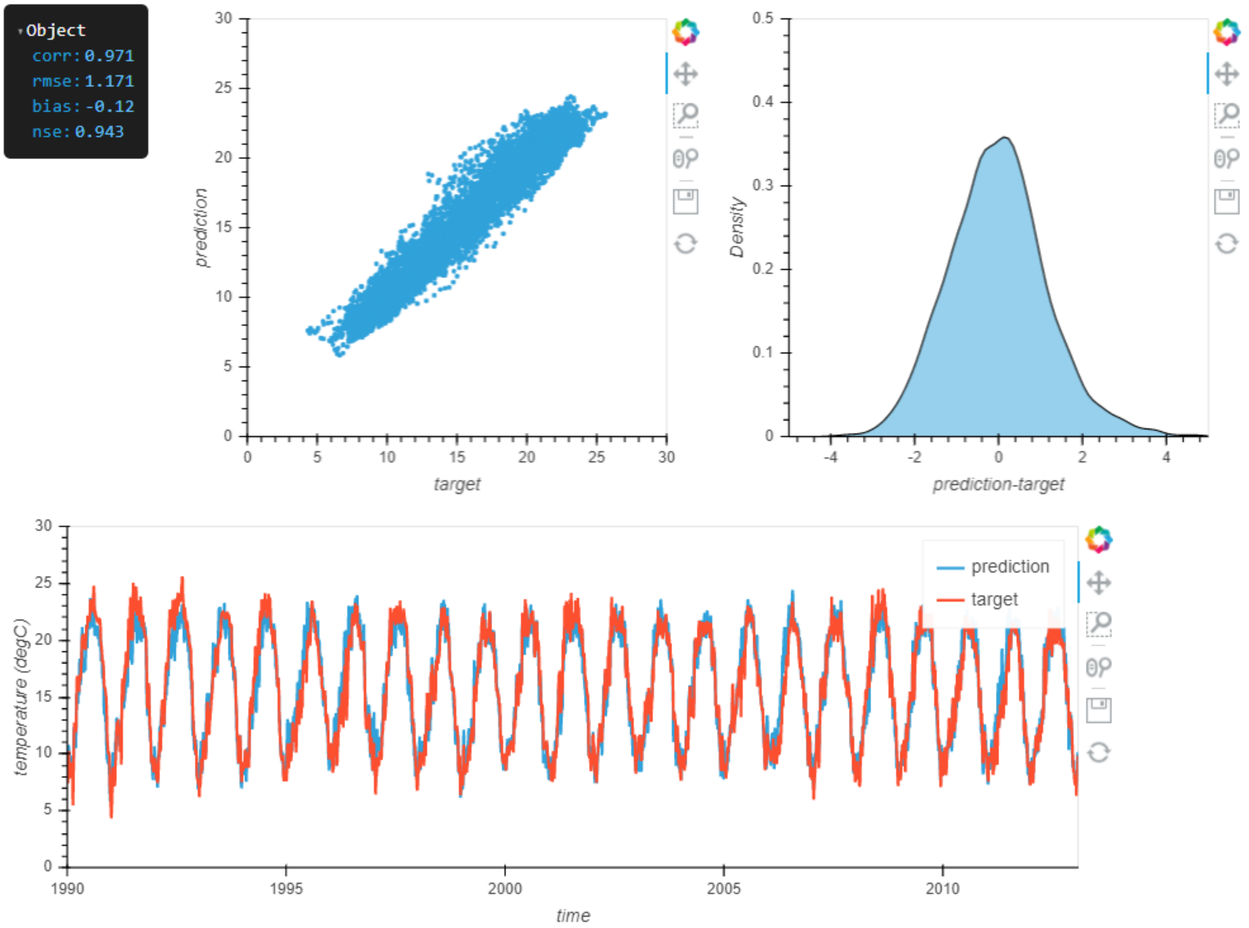


3.4.2 Water temperature boundary

This section presents the performance of the trained water temperature ANNs during the analysis period from 1990 to 2013. For that purpose, visual comparison of ANN simulations against the corresponding observations, as well as the discrepancies between them, is presented. Additionally, a set of statistical metrics are examined. These metrics include correlation coefficient, root-mean square error, percent bias (bias), and Nash-Sutcliffe efficiency coefficient. The correlation coefficient ranges from -1 to 1, with an absolute value closer to 1 indicating higher-end correlation between model simulations and the observations. The root-mean-square-error is a non-negative number, with smaller values designating better model performance. The metric takes the square root of the discrepancy between modeled and observed data. Consequently, it implicitly assigns relative higher weights to larger discrepancies. The percent bias measures the percent differences between model simulations and the corresponding observations. This shows how much the model under-simulates (negative bias) or over-simulates (positive bias) the observations on an average sense. The Nash-Sutcliffe efficiency coefficient is less than or equal to 1. A value closer to 1 suggests more satisfactory model performance.

ANN-derived water temperature simulations at three locations well mimic the observed values in terms of both variation pattern and magnitude (Figure 3-12 to Figure 3-14). Most of the discrepancies between modeled and observed values range from -1 to 1 °C. The correlations between them are consistently above 0.9 across three locations. The biases are generally small while the Nash-Sutcliffe efficiency values are high. Among three locations, the downstream boundary location Martinez has the most desirable metrics. Overall, the ANN models yield satisfactory water temperature simulations at these boundary locations.

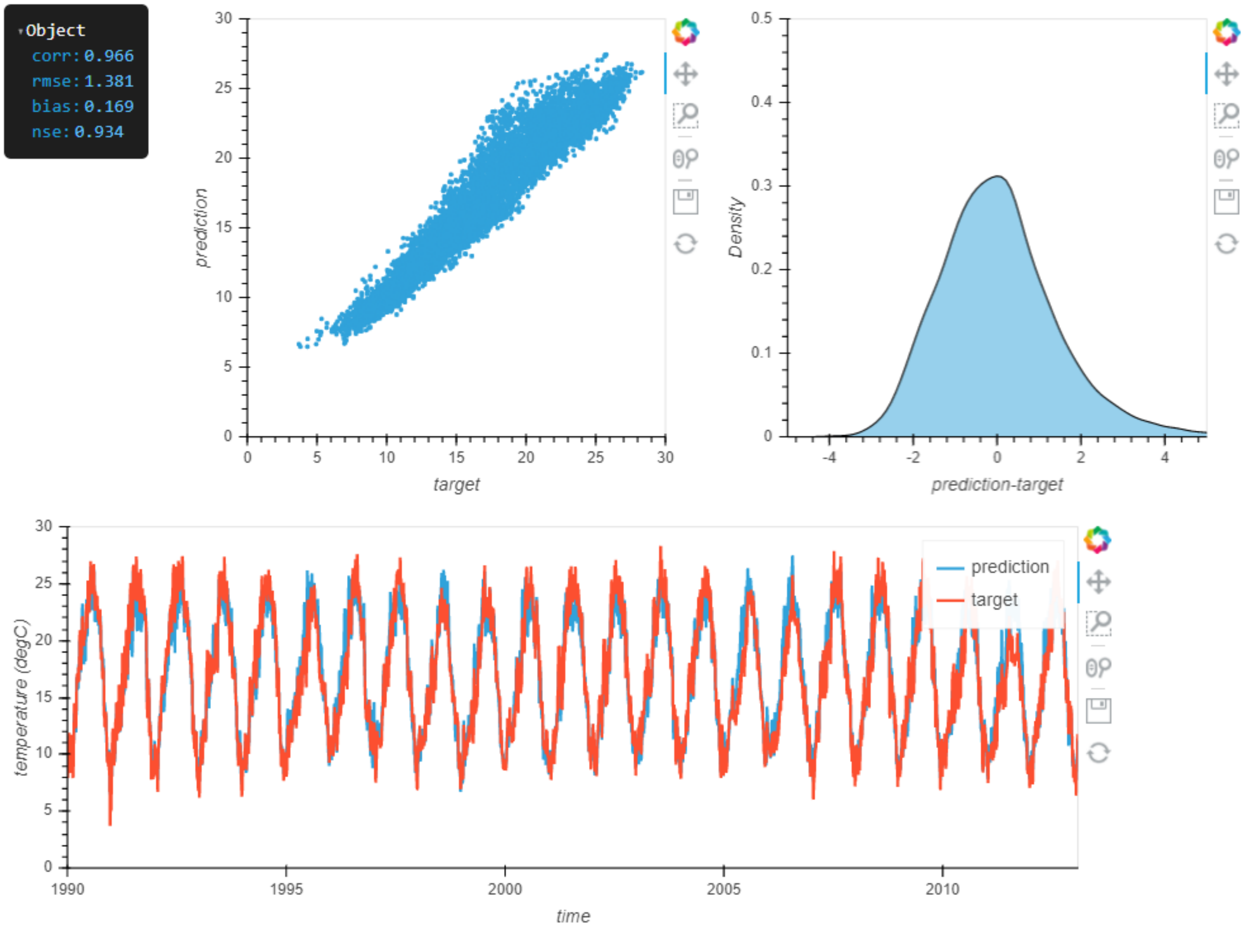
Figure 3-12 Performance of the trained water temperature ANN at Freeport



```
▾Object
  corr: 0.971
  rmse: 1.171
  bias: -0.12
  nse: 0.943
```

Note: The performance is illustrated by a scatter plot showing the observed (i.e., target) against the ANN-derived (i.e., prediction) water temperature values (upper left panel), a probability density plot of the residuals between them (upper right panel), and a time series of them during the training period (lower panel). Corresponding statistical metrics correlation (corr), root-mean square error (rmse), percent bias (bias), and Nash-Sutcliffe efficiency (nse) coefficient are also shown.

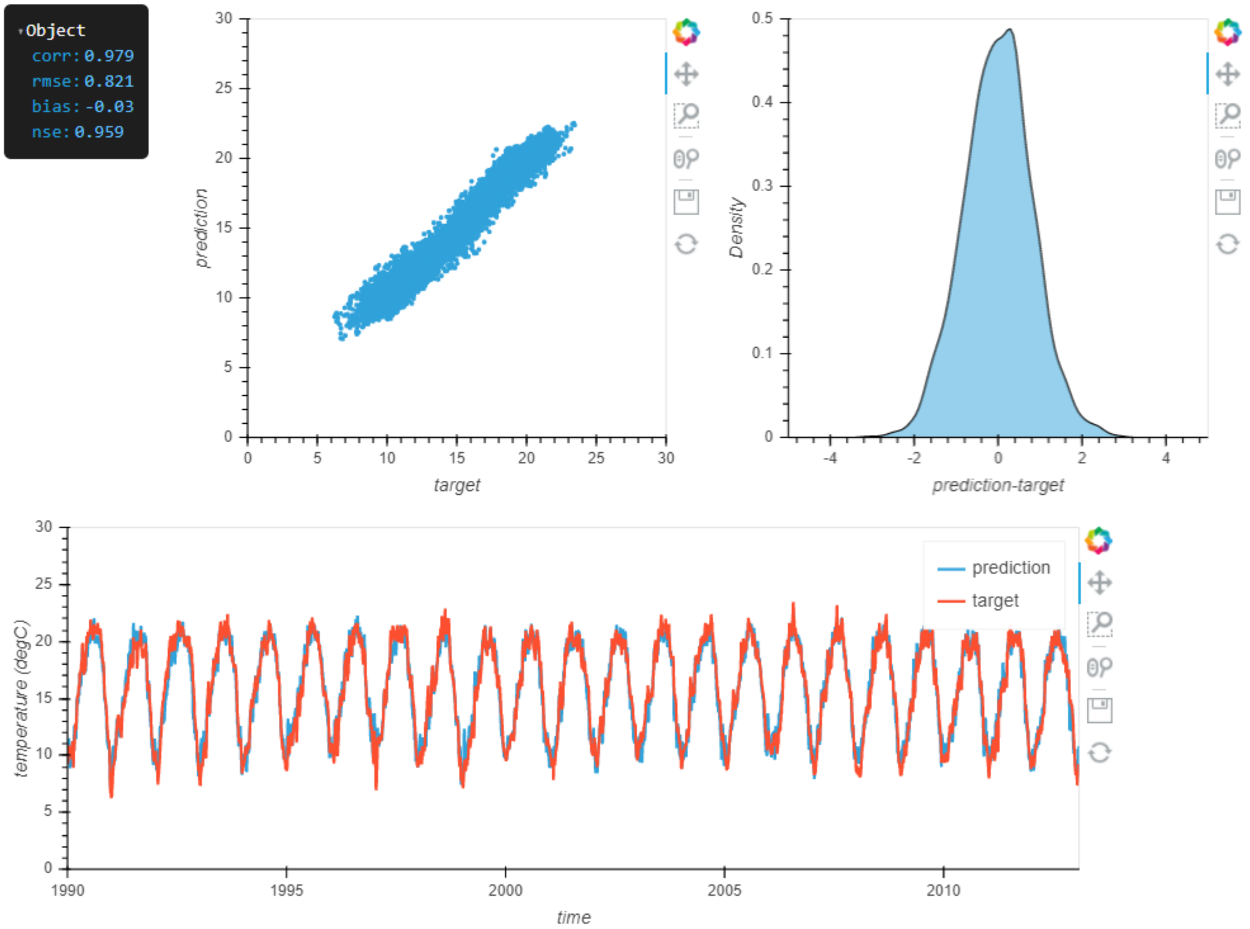
Figure 3-13 Performance of the trained water temperature ANN at Vernalis



```
▾Object
  corr: 0.966
  rmse: 1.381
  bias: 0.169
  nse: 0.934
```

Note: The performance is illustrated by a scatter plot showing the observed (i.e., target) against the ANN-derived (i.e., prediction) water temperature values (upper left panel), a probability density plot of the residuals between them (upper right panel), and a time series of them during the training period (lower panel). Corresponding statistical metrics correlation (corr), root-mean square error (rmse), percent bias (bias), and Nash-Sutcliffe efficiency (nse) coefficient are also shown.

Figure 3-14 Performance of the trained water temperature ANN at Martinez



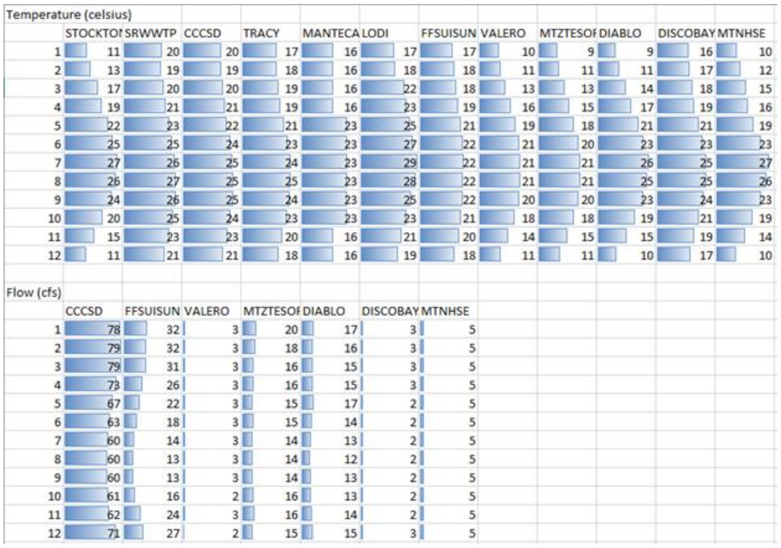
```
▾Object
  corr: 0.979
  rmse: 0.821
  bias: -0.03
  nse: 0.959
```

Note: The performance is illustrated by a scatter plot showing the observed (i.e., target) against the ANN-derived (i.e., prediction) water temperature values (upper left panel), a probability density plot of the residuals between them (upper right panel), and a time series of them during the training period (lower panel). Corresponding statistical metrics correlation (corr), root-mean square error (rmse), percent bias (bias), and Nash-Sutcliffe efficiency (nse) coefficient are also shown.

3.4.3 Effluent boundary

Figure 3-15 shows the monthly effluent flow temperature (the upper panel) at the 12 locations shown in Figure 3-1. A similar seasonal variation pattern is evident at each location. For a specific month, there are noticeable variations in temperature values among 12 locations. Monthly effluent flow rates at seven non-CalSim3 locations are also illustrated in Figure 3-15 (the lower panel). The effluent flow rates are generally small, particularly for Valero, Discovery Bay, and Mount House. The seasonal variation in flow rate is less evident compared to that of the effluent water temperature.

Figure 3-15 Monthly effluent temperature and flow discharge



Temperature (celsius)

	STOCKTOI	SRWWTP	CCCSD	TRACY	MANTECA	LODI	FFSUISUN	VALERO	MTZTESOI	DIABLO	DISCOBAY	MTNHSE
1	11	20	20	17	16	17	17	10	9	9	16	10
2	13	19	19	18	16	18	18	11	11	11	17	12
3	17	20	20	19	16	22	18	13	13	14	18	15
4	19	21	21	19	16	23	19	16	15	17	19	16
5	22	23	22	21	23	25	21	19	18	21	21	19
6	25	25	24	23	23	27	22	21	20	23	23	23
7	27	26	25	24	23	29	22	21	21	26	25	27
8	26	27	25	25	23	28	22	21	21	25	25	26
9	24	26	25	24	23	25	22	20	20	23	24	23
10	20	25	24	23	23	23	21	18	18	19	21	19
11	15	23	23	20	16	21	20	14	15	15	19	14
12	11	21	21	18	16	19	18	11	11	10	17	10

Flow (cfs)

	CCCSD	FFSUISUN	VALERO	MTZTESOI	DIABLO	DISCOBAY	MTNHSE
1	78	32	3	20	17	3	5
2	79	32	3	18	16	3	5
3	79	31	3	16	15	3	5
4	73	26	3	16	15	3	5
5	67	22	3	15	17	2	5
6	63	18	3	15	14	2	5
7	60	14	3	14	13	2	5
8	60	13	3	14	12	2	5
9	60	13	3	14	13	2	5
10	61	16	2	16	13	2	5
11	62	24	3	16	14	2	5
12	71	27	2	15	15	3	5

3.5 Summary

This document describes the methods utilized to derive necessary input data for DSM2 QUAL to simulate water temperature across the Delta for water years 1922 through 2015. These inputs contain five meteorological variables at a location in the central Delta, water temperature values at three boundary locations, and effluent flow temperature and flow rate at 12 locations. A modified meteorological processor, MetSim, is applied to derive those meteorological variables based on two research datasets with long records. Artificial neural networks (ANNs) were developed to generate water temperature at three DSM2 boundary locations. Effluence flow temperature and flow rate are derived from limited observations from relevant previous studies or directly from CalSim3 simulations. This document further presents MetSim and ANN results and compares them with the corresponding field observations. These comparisons indicate that both MetSim and ANN-based inputs are able to yield simulations that effectively mimic the observations. The methods described herein can be adapted to generate input data for DSM2 QUAL water temperature modeling under different climate scenarios.

3.6 Acknowledgements

The authors would like to thank Chilmakuri (State Water Contractors) for providing guidance on selecting approaches applied in the study. The authors also like to thank their colleagues Hari Rajbhandari and Bradley Tom for reviewing an early version of the chapter.

3.7 Data sources

- Livneh temperature and precipitation dataset ftp://livnehpublicstorage.colorado.edu/public/Livneh.2016.Dataset/Meteorology.netCDF/
- NOAA reanalysis dataset ftp://ftp2.psl.noaa.gov/Datasets/20thC_ReanV3
- CIMIS data ftp://ftpcimis.water.ca.gov/pub2/annual/

3.8 References

Bennett A, Hamman J, and Nijssen B. 2020. MetSim: A Python package for estimation and disaggregation of meteorological data. Journal of Open Source Software, 5(47), 2042.

Bohn T, Livneh B, Oyler J et al. 2013. Global evaluation of MTCLIM and related algorithms for forcing of ecological and hydrological models, Agricultural and Forest Meteorology.

Livneh B, Bohn T, Pierce D et al. 2015. A spatially comprehensive, hydrometeorological data set for Mexico, the U.S., and Southern Canada 1950–2013. Scientific Data, 150042.

Resources Management Associates (RMA). 2011. Modeling the Fate and Transport of Nutrients Using DSM2: Calibration/Validation Report, Technical Report, p668.

Resources Management Associates (RMA). 2015. Modeling the Fate and Transport of Nutrients Using DSM2: Calibration/Validation Report, Technical Report, p226.

Slivinski L, Compo G, Whitaker J et al. 2019. Towards a more reliable historical reanalysis: Improvements for version 3 of the Twentieth Century Reanalysis system, Quarterly Journal of the Royal Meteorological Society.

Stull R. 2011. Wet-Bulb Temperature from Relative Humidity and Air Temperature. Journal of Applied Meteorology and Climatology, 2267–2269.

43rd Annual Progress Report
June 2022

Chapter 4
South Delta Salinity-Constituent Conversion via Machine Learning

Authors: Peyman Namadi and Minxue He
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Contents

Figures

Tables

Chapter 4 South Delta Salinity-Constituent Conversion via Machine Learning

4.1 Introduction

Electrical Conductance (EC) is a water quality metric typically used to represent the salinity level. It can also be used as the predictor for other ion constituents, including Total Dissolved Solids (TDS), dissolved chloride (Cl^-), dissolved sulfate ($SO4^{2-}$), dissolved sodium (Na^+), dissolved calcium (Ca^{2+}), dissolved magnesium (Mg^{2+}), dissolved nitrate ($NO3^-$), dissolved potassium (K^+), dissolved bromide (Br^-), dissolved boron (B), Alkalinity, and water hardness in the Delta. These ion constituents are typically treated as water quality indicators and can be measured by standard laboratory methods. Regression models have also been developed and applied to simulate the concentrations of these ion constituents in the Delta (Jung 2000; Suits 2002; Hutton 2006; Denton 2015). Most recently, the North Central Region Office (NCRO) used parametric quadratic regression equations to estimate the concentrations of these 12 ion constituents, using EC as the predictor. That study, intended to identify and investigate sources of local salt loading in south Delta channels, collected and used grab sample data from 2018–2020 at seven key locations in the south Delta (California Department of Water Resources North Central Region Office 2021). The goal of the current study is to develop machine learning models to emulate the regression equations in the NCRO study to simulate ion constituents. The results indicate that machine learning models can provide simulations comparable or superior to the regression equations.

4.2 Methodology

4.2.1 Study Locations and Study Dataset

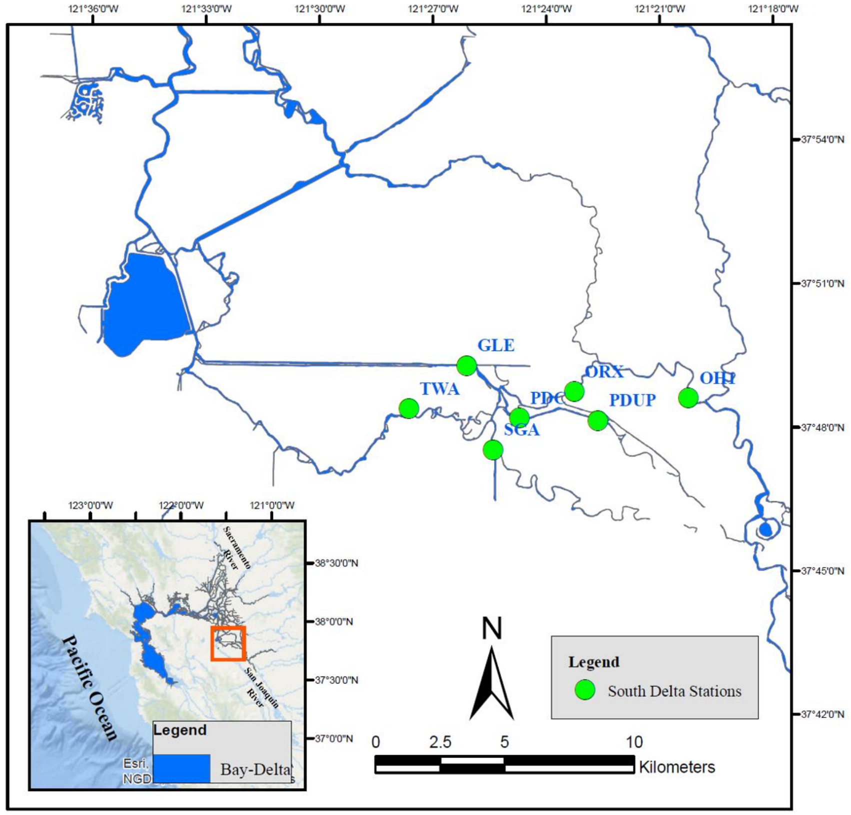
From 2018 to 2020, the Water Quality Evaluation Section (WQES) of NCRO collected standard ion samples at seven stations (Figure 4-1) co-located with continuous water quality equipment measuring salinity conditions in the south Delta. These stations (Figure 4-1, Table 4-1) were selected to track the water quality effects in south Delta channels resulting from possible discharges into Paradise Cut and Sugar Cut (California Department of Water Resources North Central Region Office 2021). Samples were collected on a near-monthly basis for ion analysis at 1-meter depth using a Van Dorn sampler. The California Department of Water Resources' (DWR's) Bryte Laboratory used 0.45-micron filter grab samples to determine the concentrations of the aforementioned 12 ion constituents (California Department of Water Resources North Central Region Office 2021). The sampled data were used in this study to train and test proposed machine learning models.

Table 4-1 Station information including Water Data Library discrete water sample station I.D. and geographic coordinates in WGS 84*

Station Name	ID	Latitude	Longitude
Grant Line Canal East	GLE	37.820	-121.435
Old River above Doughty Cut	ORX	37.811	-121.387
Old River at TWA	TWA	37.803	-121.457
Old River near Head	OH1	37.808	-121.331
Paradise Cut	PDC	37.802	-121.412
Paradise Cut Upstream	PDUP	37.801	-121.373
Sugar Cut at Golden Anchor	SGA	37.793	-121.421

* Adapted from (California Department of Water Resources North Central Region Office 2021).

Figure 4-1 Map showing seven study stations in the south Delta.



Note: The insert map shows the location of the San Francisco Bay and Sacramento-San Joaquin Delta (Bay-Delta), containing the South Delta study area (highlighted in the red rectangle).

4.2.2 Model Development

Four nonparametric supervised machine learning (ML) techniques, Generalized Additive Model (GAM), Regression Trees (RT), Random Forest (RF), and Artificial Neural Networks (ANNs), were employed to estimate ion constituents after given the EC at these seven study stations. Because of the combination of a complex channel network and bathymetry in the south Delta and varying impacts on local hydrodynamics from ocean tides, channel diversions, island drainage, Banks Pumping Plant pumping, and San Joaquin River inflow, the source of the water and thus the proportions of water quality constituents at each station can differ.

For this reason, in the first scenario, station names were employed as categorical variables as part of the input data fed into the ML models. Considering that machine learning algorithms cannot use categorical variables in the numerical calculation, an encoding technique was implemented to convert the seven stations' names to numerical values. In encoding the categorical variables, a number is assigned to each variable (1 to 7 in this case). The numbers have no quantitative value, and the order does not matter (Potdar et al. 2017). In the second scenario, the month and water year type (when a specific sample was taken) were added as additional input features to assess their potential impacts on the model outcome. In the first scenario, ML models were trained for the entire dataset to maintain consistency with the training method applied in developing the quadratic regression models. The input-output datasets were randomly split into two groups for training (80 percent of the dataset) and testing (20 percent of the dataset) in the second scenario. The performance of four ML models was evaluated using two criteria, R^2 (Equation 1) and Mean Absolute Error (MAE) (Equation 2). R^2 ranges from 0 to 1, with a value close to 1 meaning that model simulations capture most of the variability in the observed data. MAE is a positive number, with a value close to 0 meaning that the model-simulated values are very close to observed values. A brief overview of the nonparametric supervised machine learning techniques used in this study is provided as follows.

$$R^2 = 1 - \frac{\text{SSE}}{\text{SSTotal}} \tag{1}$$

SSE: Sum of squared error (or residuals). $SSE = \sum_i (y_i - \hat{y}_i)^2$

SSTotal: Sum of squared deviations from the mean $\bar{y}$ (total variation of y without model adjustment). $\text{SSTotal} = \sum_i (y_i - \hat{y}_i)^2 + \sum_i (\hat{y}_i - \bar{y})^2$

$y_i = observed\ values$

$\hat{y}_i = simulated\ values$

$\bar{y} = mean\ of\ observed\ values$

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\text{n}} \tag{2}$$

4.2.2.1 Generalized Additive Model

The Generalized Additive Model (GAM) provides a general framework for improving standard linear models by allowing for non-linear relationships between each feature and the response (Hastie and Tibshirani 1986; James et al. 2013). GAM replaces each linear component with a (smooth) non-linear function $f_j(x_{ij})$ and calculates a separate $f_j(x_{ij})$ for each predictor when others remain fixed. GAM divides the variation range of each environmental predictor into distinct regions. It fits a polynomial function in each region with the limitation that the polynomial function in each region needs to join smoothly to the polynomial in the next region. Equation 3 shows the general form of the GAM model. y_i represents the targets that are ion constituents in our study. Also, $f_1(EC)$ is unspecified smooth ("nonparametric") function of EC. The individual functions of the GAM model were developed using the *mgcv* estimation package (Wood 2017) in the R statistical computing environment (R Core Team 2021) and the field sampling data.

$$\boldsymbol{y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i = \beta_0 + f_1(EC) + \epsilon_i} \tag{3}$$

4.2.2.2 Decision Trees

Decision trees are popular machine learning methods that can be applied to both regression and classification problems. This method stratifies the predictor space into several rectangular regions and assigns a mean of each region to all observed data included in that specific region (Loh 2011; James et al. 2013). Tree-based ML models are useful for interpretation, as their results indicate the importance of predictors, and the split points suggest the best threshold for each predictor.

The first step in each decision tree is finding the best split predictor and cutpoint at each node of the decision tree. The model implements the recursive binary splitting method that splits the dataset into two new branches. The decision tree considers all predictors and all possible cutpoints for each predictor and then chooses the predictor and cutpoints for which the Residual Sum of Squares (RSS) is the minimum (James et al., 2013). Equation 4 shows the RSS criteria to be minimized at each splitting point, where R_1 and R_2 are the two new branch regions after each splitting process, j is the predictor indicator, and S is the cutpoint. y_i represents the targets that are ion constituents in this study. The individual functions of the RT model were determined by using the *rpart* package (Therneau and Atkinson 2019) in the R statistical computing environment.

$$RSS = \sum_{i:x_i \in R_1(j,s)} \left(y_i - \hat{y}_{R_1}\right)^2 + \sum_{i:x_i \in R_2(j,s)} \left(y_i - \hat{y}_{R_2}\right)^2 \qquad (4)$$

4.2.2.3 Random Forest

Random Forest (RF) has demonstrated strong predictive performance in addressing a wide range of classification and regression analysis problems (Liaw and Wiener 2002). It incorporates multiple decision trees in conjunction with the bootstrap technique to decrease the variance of a statistical learning method. This allows for the production of new populations from the primary population by resampling data (James et al. 2013). RF cumulates the results of all decision trees that were produced by the bootstrapping technique. In other words, if β separate training datasets were produced by the bootstrapping method, $\hat{f}^1(x), \hat{f}^2(x), \dots, \hat{f}^\beta(x)$ will be the result of each decision tree. Equation 5 shows the final result of the RF method, the average of all of the decision trees, which generates a single low-variance statistical learning model with more accuracy. The individual functions of the

RF model were determined by using the "*randomForest*" package in the R statistical computing environment.

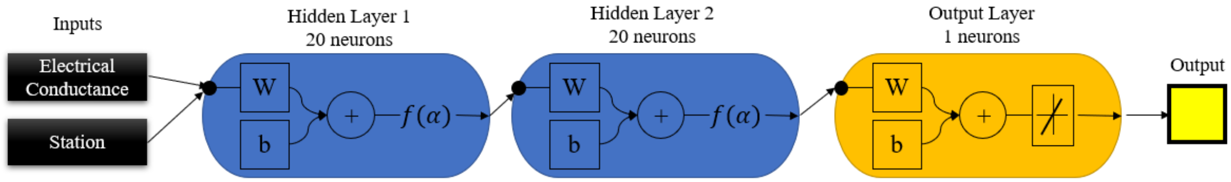
$$\hat{f}_{avg}(x) = \frac{1}{\beta}\sum_{b=1}^{\beta}\hat{f}^{b}(x) \quad (5)$$

4.2.2.4 Artificial Neural Network (ANN)

Artificial intelligence-based neural network (ANN) models are alternative predictive models that have been widely adopted for model identification, analysis, and forecasting. The ANN has been proven to be an effective method for developing non-linear relationships between a dependent variable and independent variables (Hopfield 1988; Zhang et al. 2015).

A typical ANN model consists of three primary layers: an input layer, a hidden layer, and an output layer. In this study, the ANN consists of four layers: an input layer, two hidden layers, and an output layer. The input layers contain two input variables, electrical conductance (EC) and station name (as a categorical variable). The number of neurons in hidden layers and their activation functions were determined after experimenting with multiple iterations until maximum simulation accuracy can be obtained. The number of neurons in each hidden layer was determined to be 20, and the Rectified linear function $f(\alpha) = \max(o, \alpha)$ was selected as the activation function. The loss function is the Mean Squared Error (MSE). Figure 4-2 shows the Artificial Neural Network architecture. The individual functions of the ANN model were determined by using the open source "H2O" package in the R statistical computing environment (Candel et al. 2016).

Figure 4-2 Artificial Neural Network architecture



4.3 Results

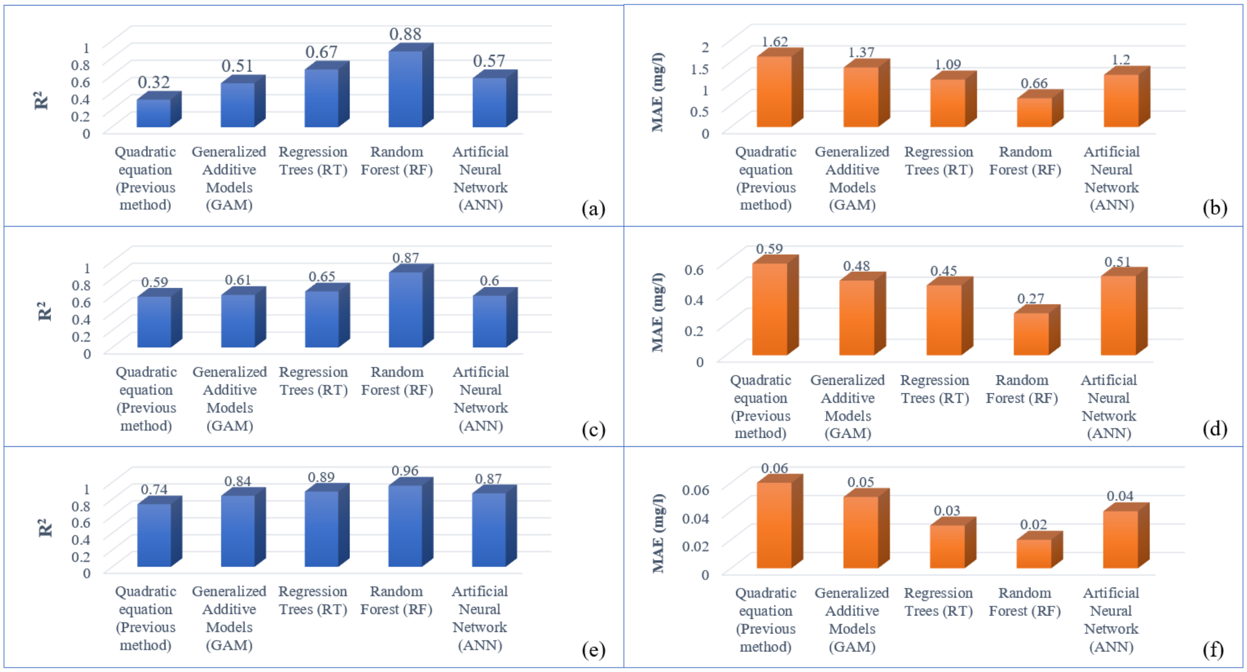
This section first presents the training performance of the four ML models while simulating three ion constituents (nitrate, potassium, and boron) for which the regression equations in the NCRO study (California Department of Water Resources North Central Region Office 2021) have relatively poor performance. This section presents the selection and evaluation of the ML model with the most desirable performance. Finally, the performance of the selected model on the remaining nine ion constituents is illustrated.

4.3.1 Simulation of Nitrate ($NO3^-$), Potassium (K^+), and Boron (B)

The performance of ML models on simulating the concentration of nitrate, potassium, and boron is first evaluated using two metrics, R^2 and Mean Absolute Error (MAE) (Figure 4-3 (a–f)). The findings are summarized as follows:

- For the nitrate simulation, the R^2 values are calculated as 0.32, 0.51, 0.67, 0.88, 0.57 for the quadratic equation (benchmark), GAM, RT, RF, and ANN, respectively. The MAE values are determined as 1.62, 1.37, 1.09, 0.66, and 1.2 milligrams per liter (mg/l) for these five models, respectively. The training results for nitrate show that the RF model yields the highest R^2 (Figure 4-3a) and lowest MAE (Figure 4-3b). Compared to the quadratic equation, the R^2 for RF model increases by 175 percent and the MAE decreases by 59 percent.
- The training results for potassium show that the RF model again yields the highest R^2 (Figure 4-3c) and lowest MAE (Figure 4-3d). The R^2 values are 0.61, 0.65, 0.87, 0.60 for GAM, RT, RF, and ANN, respectively. The RF model shows the largest improvement (47 percent) on R^2 over the benchmark quadratic equation. The quadratic equation also has a poor MAE value (0.59 mg/l). The MAE values for these four ML models are 0.48, 0.45, 0.27, and 0.51 mg/l, respectively. The RF model also has the largest improvement in MAE (54 percent) over the quadratic equation.
- The training results for boron show that the RF model again yields the highest R^2 (Figure 4-3e) and lowest MAE (Figure 4-3f). Compared to the quadratic equation, the R^2 for RF model increases by 28 percent and the MAE decreases by 64 percent. Consistent with nitrate and potassium, the GAM, RT, RF, and ANN models all outperform the Quadratic Equation model.

Figure 4-3 Comparison between performance of five ion simulation models based on R² (first column, panels (a), (c), and (e)) and MAE (second column, panel (b), (d), and (f))

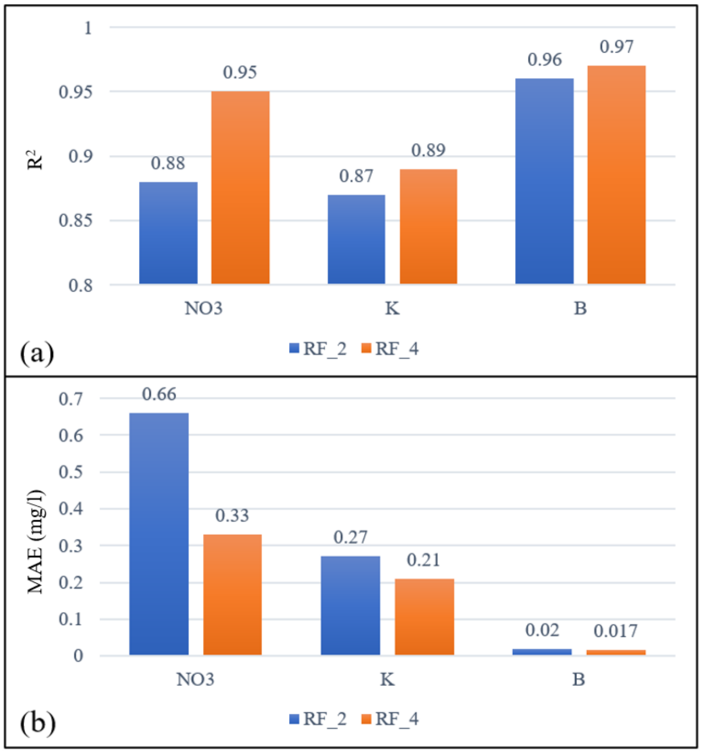


Note: The first, second, and third rows show performance of nitrate, potassium, and boron, respectively.

4.3.2 Model selection and testing under a second scenario

The results above show that the Random Forest (RF) model has the best performance during training based R^2 and MAE. The RF model was therefore chosen to be tested under a second scenario which contains month and water year type as inputs. The grab samples in the dataset covered all 12 months and three water year types (below normal for 2018, wet for 2019, and dry for 2020). Figure 4-4 a–b compares the two scenarios' performances based on R^2 (Figure 4-4a) and MAE (Figure 4-4b). RF based on four inputs (RF_4) outperforms RF based on only two inputs, increasing R^2 by 8 percent, 2.3 percent, and 1 percent for $NO3^-$, K^+, and B, respectively, and decreasing MAE by 50 percent, 22 percent, and 15 percent for $NO3^-$, K^+, and B, respectively.

Figure 4-4 RF model performance on simulating the concentrations of nitrate, potassium, and boron under two scenarios (scenario 1, RF_2, with two predictors consisting of EC and station; scenario 2, RF_4, with four predictors consisting of EC, station, month, and water year type) based on (a) R^2 and (b) MAE



4.3.3 Model assessment

This subsection now assesses the prediction error (i.e., test error or generalization error) of the RF model on new ion data and discusses the potential issue of overfitting the model. The generalization performance of a model developed via a learning method is based on its ability to predict test data not used in training. Assessment of this performance guides model selection and measures the usefulness of the chosen model. Test error is the model prediction error over a test sample of data not used in training the model. One of the best approaches for first training and then testing a model is to randomly divide the data into two parts: training data and test data. The training data are used to fit or develop the models. The test data are used to assess the model generalization error by comparing simulated ion concentrations to observed values not used in the model development. Determining the number of observations in the training and test datasets depends on the signal-to-noise ratio in the data and the training sample size. In this study, because of the limitation of data samples (183 samples), the data were divided such that 80 percent (146 samples) are used for training and 20 percent (37 samples) are used for testing. The models are then tested with data not used in model training, which helps to avoid overtraining (or overfitting) the models.

A concern of the development of any model that is based on observed data is fitting the model to the observed data so closely that it then fails to provide meaningful results for other conditions. Random Forest models rely on an ensemble of other models called *decision trees*. Decision trees can capture complex interaction structures in the data. When trees grow deep, the final model has low bias and high variance that cause an overfitting problem. Also, decision trees use Gini impurity (a measurement of the likelihood of an incorrect classification of a new instance of a random variable if that new instance were randomly classified according to the distribution of class labels from the data set) to split each node. Gini impurity restricts the decision tree; consequently, there is no guarantee of using all features during training. Random Forest is a substantial modification of bagging that builds a large collection of de-correlated trees. In fact, being able to choose these random subsets of features allows us to explore many different aspects of the entire feature space. Random Forest creates subsets of randomly picked features at each potential split. Because of this, the developer of the RF algorithm claimed that "Random forests does not overfit" (Breiman 2001). Hastie (2018) further ascertained that increasing the number of trees in RF does not cause the RF sequence to overfit after a certain number of trees, because: (a) different random selections don't reveal any more information; and (b) different random selections are simply duplicating trees that have already been created. Therefore, in theory, overfitting while training an RF model shouldn't be a concern. The study results shown in Figure 4-5 look at this issue directly.

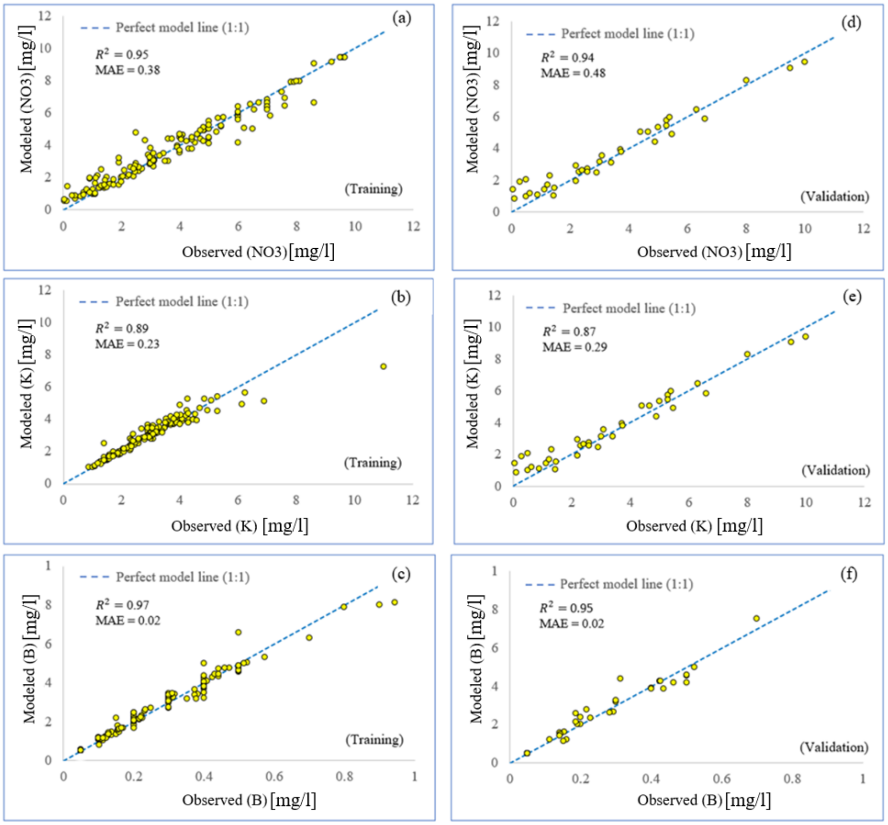
Figure 4-9 a–f shows how well the RF model performs with training and independent test datasets by comparing observed $NO3^{-}$, K^{+}, and B levels and their counterparts simulated via the RF models. The x-axis shows the observed data, and the y-axis shows the simulated f-data. The dashed line in the graphs is the 1:1 line that shows a perfect model that can simulate the observations without any errors. The quantitative performance of the RF models with the training and independent testing datasets is also summarized in Table 4-2. The R^2 are 0.95 and 0.95 for training and independent testing for $NO3^-$, respectively. The MAE is 0.39 and 0.48 mg/l for training and independent testing for $NO3^-$, respectively. The simulation model for potassium has close R^2 and MAE for training and independent testing. R^2 values are 0.88 and 0.86, and MAE values are 0.24 and 0.25 for training and independent testing, respectively. The performance of the boron simulation model for training and independent testing is satisfactory. Specifically, R^2 values are 0.97 and 0.97, and MAE values are 0.02 and 0.02 for training and independent testing, respectively. Overall, the results indicate that the model performance during the training process is fairly similar to its counterpart during the independent testing process. The possibility of model overfitting is remote.

Table 4-2 shows the performance of the RF-4 model for all 12 ion constituents based on two criteria (R^2 and MAE) for training and independent testing. Overall, R^2 for training and independent dataset for other nine ion constituents are close to 1. Also, the MAE of the ion constituents are noticeably decreased when compared with benchmark models.

Table 4-2 Performance of RF prediction model with four predictors and benchmark model (quadratic equation)

Ion Constituents	Performance of the RF model				Performance of the benchmark model	
	R^2		MAE		R^2	MAE
	Training	Validation	Training	Validation		
B	0.97	0.97	0.02	0.02	0.75	0.06
Br^-	0.99	0.99	0.02	0.02	0.98	0.03
Ca^{2+}	0.997	0.998	1.33	1.29	0.99	2.71
Cl^-	0.998	0.998	3.28	3.14	0.99	6.25
Hardness	1	1	4.79	5.06	0.99	8.13
K^+	0.88	0.86	0.24	0.25	0.59	0.52
Mg^{2+}	1	1	0.49	0.61	0.99	0.75
$NO3^-$	0.95	0.95	0.39	0.48	0.32	1.59
Na^+	1	1	1.98	2.54	0.99	4.08
$SO4^{2-}$	1	1	2.38	2.36	0.98	5.76
TDS	1	1	7.14	8.25	0.99	8.02
Alkalinity	0.99	0.98	2.81	3.94	0.96	5.37

Figure 4-5 Observed (x-axis) RF model-simulated (y-axis) on the concentrations of nitrate (first row), potassium (second row), and boron (third row)

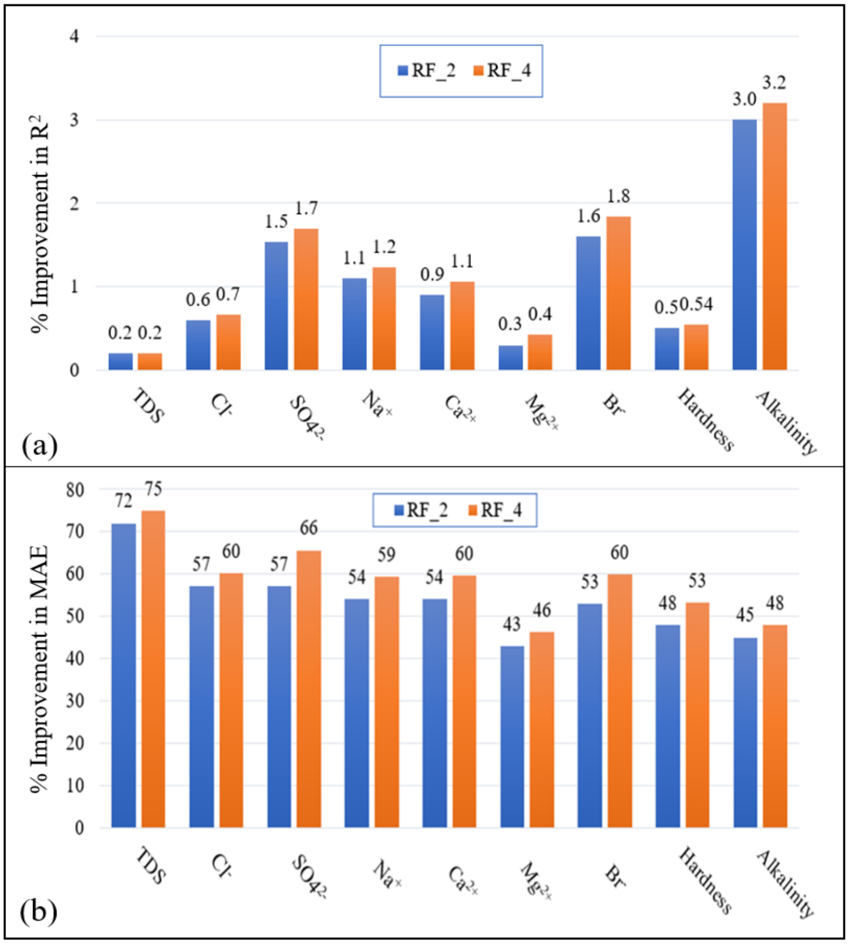


Note: The first column (panels (a), (b) and (c)) and second column (panels (d), (e), and (f)) show the training and validation results, respectively.

4.3.4 Testing the selected model on other ion constituents

The RF_2 and RF_4 scenarios are tested on nine other water quality parameters to evaluate the performance of the selected ML models on all water quality parameters. For the purpose of illustration, Figure 4-6 presents the percent improvement of the RF_2 and RF_4 models when compared with the benchmark model (quadratic equation) based on R^2 and MAE, respectively. The improvement in R^2 is between 0.2 percent to 3.2 percent for nine ion constituents. The RF models do not significantly improve the accuracy based on R^2 because the benchmark models already yield satisfactory R^2 for these nine constituents (Table 4-2). In contrast, the improvement in MAE is remarkable. For instance, RF_4 increases R^2 by 0.2 percent, but it reduces MAE by 75 percent over the quadratic equation for simulating TDS. Moreover, RF_4 improves MAE by 60, 66, 59, 60, 46, 60, 53, and 48 percent for Cl$^-$, SO4^{2-}, Na$^+$, Ca^{2+}, Mg^{2+}, Br$^-$, Hardness, and Alkalinity, respectively. These observations indicate that though the quadratic questions can yield fairly reasonable simulations on these nine constituents, the RF models (particularly with four predictors) can yield even better simulations with notably smaller errors (measured by MAE).

Figure 4-6 RF model performance on simulating the concentrations of nine ion constituents under two scenarios (RF_2 and RF_4) based on percent improvement from the benchmark model represented by (a) R^2 and (b) MAE



4.4 Summary and Future Work

This study developed four types of Machine Learning (ML) models (i.e., GAM, RT, RF, and ANNs) within the R statistical computing environment to simulate the concentrations of 12 ion constituents in the South Delta. The results are compared to those of the conventional quadratic regression equations previously developed. The key findings are summarized as follows:

- ML models showed comparable or better performance in simulating the concentrations of ion constituents than the conventional quadratic equations.
- Among all ML models, the RF models tended to yield the best performance metrics.
- Using additional input features including station name and the corresponding time (including month and the type of the year when the samples were collected) as categorical variables improved the performance of the RF models.
- RF models, by design, minimize the potential of model overfitting, which was confirmed in this study by testing the trained models using randomly selected independent datasets.

The newly developed machine learning models in this study were trained for seven different water quality stations in the south Delta. The application of these models is limited to these stations. Next, machine learning models will be trained for wherever sample data are available in the Delta. Moreover, clustering methods, such as K-means, and hierarchical methods will be applied Delta-wide to divide the Delta into sub-regions based on data patterns (i.e., stations with similar data patterns will be grouped into the same sub-region). Different machine learning models will be developed for different sub-regions. Furthermore, salinity and discharge information at Delta boundaries such as the Sacramento River (freshwater boundary), the seawater boundary, and the San Joaquin River (agricultural boundary) will be added to the model to increase model performance. In addition to sampled data, model simulations (e.g., DSM2-simulated salinity and flows) will also be utilized to train machine learning models.

4.5 Acknowledgements

We gratefully thank our colleagues in the Water Quality Evaluation Section of the North Central Region Office for collecting and analyzing the grab samples. We thank Raymond Hoang for reviewing an earlier version of the chapter and provided useful comments. We would also like to thank Nicky Sandhu and Kijin Nam for their management support on the work.

4.6 References

Hastie T, Tibshirani R. 1986. Generalized Additive Models. Statist. Sci. 1 (3) 297310,

Hopfield JJ. 1988. Artificial neural networks. IEEE Circuits and Devices Magazine, 4(5), 3-1

Jung Marvin. 2000. Revision of Representative Delta Island Return Flow Quality for DSM2 and DICU Model Runs. Prepared for the CALFED Ad-Hoc Workgroup to Simulate Historical Water Quality Conditions in the Delta by Marvin Jung and Associates, Inc. Consultant's Report to the Department of Water Resources Municipal Water Quality Investigations Program (MWQI-CR#3), December 2000.

Breiman L. 2001. Random Forests. Machine Learning 45, 5–32.

Suits B. 2002. Chapter 5, Relationships between Delta Water Quality Constituents as derived from Grab Samples. In DWR's "Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh." 23rd Annual Progress Report, June 2002.

Liaw A, Matthew W. 2002. Classification and regression by random Forest, R news 2, no. 3.

Hutton P. 2006. Validation of DSM2 Volumetric Fingerprints Using Grab Sample Mineral Data, Power Point presentation at CWEMF Annual Meeting, March 2006.

Loh Y. 2011. Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery, 1(1), 14-23.

James G, Witten D, Hastie T, Tibshirani R. 2013. An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

Zhang Z, Deng Z, Rusch K, Walker N. 2015. Modeling system for predicting enterococci levels at Holly Beach. 109: 140-47. Marine environmental research 109, 140-147.

Denton R. 2015. Delta Salinity Constituent Analysis. Richard Denton and Associates, prepared for the State Water Project Contractors Authority.

Wood SN. 2017. Generalized additive models: an introduction with R. CRC press.

Potdar K, Taher P, Chinmay P. 2017. A comparative study of categorical variable encoding techniques for neural network classifiers. International journal of computer applications.

Therneau T, Atkinson B. 2019. rpart: Recursive Partitioning and Regression Trees. R package version 4.1–15.

California Department of Water Resources North Central Region Office. 2021. South Delta Ion Report. Technical Report, California Department of Water Resources North Central Region Office, Sacramento, California, USA.

Team RC. 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.

43rd Annual Progress Report
June 2022

Chapter 5
Hotstart and Nudging Preprocessors for Bay-Delta SCHISM

Authors: Zhenlin Zhang and Eli Ateljevich
Delta Modeling Section
Bay-Delta Office
California Department of Water Resources

Contents

Figures

Tables

Chapter 5 Hotstart and Nudging Preprocessors for Bay-Delta SCHISM

5.1 Background

Bay-Delta SCHISM (Ateljevich et al. 2014) is an application of the Semi-implicit Cross-scale Hydroinformatics Simulation Model (SCHISM), a three-dimensional hydrodynamic and water quality model, to the Sacramento-San Joaquin River Delta. The model simulates flow, salinity, sediment transport, and other water quality processes. Data and examples described in this chapter are distributed publicly in the subdirectory of a Python preprocessing library on github called *schimpy* (https://github.com/CADWRDeltaModeling/schimpy.git).

Like any hydrodynamic and transport model, SCHISM requires an initial condition. An initial condition is a complete specification of the spatial field for all variables, and this is never available except for a continuation of a prior run, so simplifying assumptions, approximation, and interpolation are required. The concept of hot-starting SCHISM is to start the model with accurate or realistic initial states of temperature, salinity, and potentially other water quality constituents (e.g., suspended sediment or biogeochemical variables). This is an important capability for the Delta system, because the memory of the initial condition for tracers, particularly conservative tracers like salinity, can last for months. Starting from a more accurate initial state can greatly shorten the spin-up time of the model. A high-fidelity initial state normally results in better hindcast and future predictions in the short term — eventually the simulation degrades to what might be termed the “long term accuracy” of the model. For Delta Simulation Model 2 (DSM2) operational modeling, an optimization procedure (described in Ateljevich 2000) is sometimes used to achieve similar goals.

For completeness, an alternative way to begin a historical simulation is with a “cold start” in which a generic initial condition is used with the understanding that the simulation results for some time after the start will be in error. After a cold start, it may take months for the influence of the initial conditions of some water quality parameters to disappear. Additionally, in SCHISM, a cold start is often not physical — for instance, there may not be a reasonable constant to use for salinity that is applicable to the whole domain — and many innocuous-seeming assumptions can lead to strange horizontal salt gradients and associated shocks in the modeled flow field. Neither the computational cost of a long startup nor the baroclinic shocks happen in DSM2, which may indicate why cold starts are a more common practice for that model. Indeed, when SCHISM is used as a 2D or barotropic model without salinity transport (i.e., for flood modeling or as a warmup to generate reasonable ocean boundary velocities), the run is typically cold started using text files and simple assumptions, such as a flat water surface and zero water velocity.

After initialization, it is also possible to incorporate observations into the model over a longer period, a practice known as *data assimilation*. The simplest, *Newtonian relaxation*, often referred to as *nudging*, is a process of relaxing the model toward local observations, creating final merged fields that reflect both the model dynamics and observations.

There are numerous advantages to nudging:

- It improves initialization by bringing observational data in over a length of time. Likewise, it may mitigate a situation where the usual data for initializing the model are not available.
- It may help to characterize some local dynamics that cannot be fully captured by the model.
- It results in a more flexible way to apply the model boundary conditions than forcing the model to exactly adopt the pre-defined values at the boundaries. This is especially helpful when the boundary condition is uncertain, as is the case for the ocean boundary of the Bay-Delta SCHISM Delta model.
- It allows modelers to tune subdomains or subprocesses (which are not nudged) while holding far-field influences or unrelated processes closely pinned to field data. An example is sediment resuspension in the Suisun Marsh area — assimilating sediment information upstream on the Sacramento–San Joaquin River confluence allows practitioners to focus on work in the Suisun region in isolation.
- For some applications of models, such as hindcasts of habitat distribution, accurate reproduction of a single historical scenario is the

only goal. In these cases, bringing data and models together is an obvious choice.

There are also limitations to data assimilation, the most obvious being that assimilation is only possible when there are data. Nudging must stop at the bifurcation point of different planning situations or at the end of the available data stream. A second limitation is that care must be taken to assure stakeholders that data assimilation (except at the ocean boundary) does not intrude on model validation or assessment. In this regard, validation (where nudging could be perceived as cheating) is very different from calibration (where it can be a valuable tool).

The goal of the present work was to extend the algorithms and methods available for preparing initialization and nudging files for SCHISM. Before the project, Fortran preprocessors to prepare hotstart and nudging files were already disseminated with the original Bay-Delta SCHISM package. Unfortunately, these tools were hardwired to specific variables, data sources, and interpolation methods. The new Python tools fit in with the schimpy preprocessor and allow more flexibility of creating both hotstart and nudging files for SCHISM under more versatile conditions — a detailed description of these applications is given in this chapter.

5.2 Hotstart preprocessor

5.2.1 Introduction

The hotstart preprocessor (schism_hotstart.py) is a new module in schimpy. This preprocessor: (1) provides a consistent framework that merges the observed mooring and discrete sampling data from multiple sources in the Bay-Delta system, and (2) applies the merged fields to create 3D initial conditions of salinity, temperature, suspended sediment concentration, and biogeochemical variables for SCHISM. Input data for the hotstart preprocessor can be established in several ways. The initial field can be interpolated from various types of observations or from a prior model run with possibly different horizontal or vertical grids or with a different set of active modules.

The output of the preprocessor is a file in NetCDF format, an open-source high-performance binary software library and format specification widely used in estuary, oceanographic, and climate models. The name of a hotstart file is usually derivative of “hotstart.nc”.

5.2.2 Methods

To initialize a 2D or 3D field for a model variable, a range of settings needs to be specified, such as variable name, the names of files holding observational data to be used to initialize the model, and the spatial interpolation method to be applied. The hotstart preprocessor uses a YAML-formatted file as the input file. YAML (Yet Another Markup Language) is an intuitive, ubiquitous human-readable data-serialization language that can both be read as a text file and interpreted programmatically (e.g., via python). Nearly every schimpy capability is specified in this language.

Classes: `hotstart` and `VariableField` were created to meet the preprocessor’s required functionality. The hotstart class reads the input hotstart.yaml file (`hotstart.read_yaml`), initializes the output hotstart.nc file (`hotstart.initialize_netcdf`: defining wet-dry cells and the initial condition for turbulent mixing), loops through each of the variables defined in the YAML file to generate a dataset for each of the 3D fields (`hotstart.generate_3D_field`), and finally maps the dataset to the specific format required to hotstart SCHISM (`hotstart.map_to_schism`). Within the software, the data are stored in xarray format, a python library for managing multidimensional data, which also works well with NetCDF. The steps defined above are bundled in one function: `hotstart.create_hotstart`. The `VariableField` class is a major factor in handling the spatial interpolation and merging for the modeled variable fields. The class is reinitialized for each variable and generates a 3D field based on the settings defined in the hotstart.yaml for the variable.

After the hotstart file (hotstart.nc) is created, users can visualize the 3D fields generated for error detection and data presentation. The variable fields can either be plotted in 2D by `schism_mesh.plot_elems`, `schism_mesh.plot_nodes`, or `schism_mesh.plot_edges`, or be converted to the SCHISM output format (schout_hotstart.nc) by the function `hotstart_to_outputnc` in order to be viewable in either 2D or 3D in VisIt. To visualize schout_hotstart.nc, VisIt, a plug-in created by DWR, needs to be installed. The source code for the plug-in is available at https://github.com/schism-dev/schism_visit_plugin, along with a user guide with the VIMS schism documentation and links to Windows binaries. Note that `hotstart_to_outputnc`, rather than being imbedded in the hotstart class, is written as an independent function. Consequently, hotstart files generated by other means (such as directly from SCHISM) can also be converted to schout_hotstart.nc and visualized in VisIt.

5.2.3 Hotstart usage

The hotstart processor can be invoked as a standalone utility, create_hotstart, which is automatically installed by conda or programmatically through python.

The simplest invocation is typing the following in a python prompt:

create_hotstart --input hotstart.yaml --modules TEM,SAL,SED

Additional processing and plotting options are available. A complete listing is available by typing:

create_hotstart --help

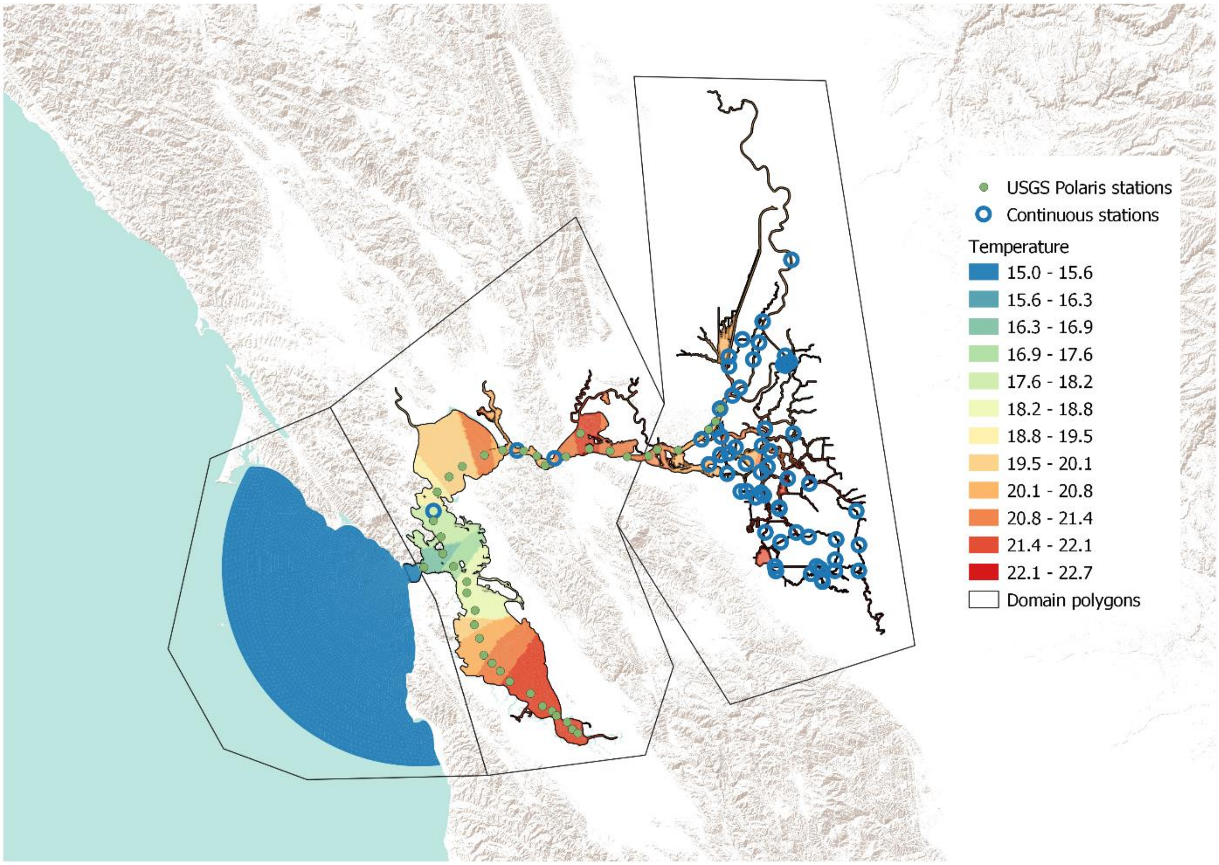
The hotstart options are described in the input yaml file (hotstart.yaml in the above example). The command line option "modules" allows you to choose a subset of the defined variables and create a smaller file. Additional options and decisions are described in the following sections.

The second approach is to invoke the hotstart processor programmatically within python. An example script to generate the hotstart.nc file for a sediment transport run, convert it to schout.nc file, and visualize the generated 2D surface temperature in the Delta is given below.

```
from schism_hotstart import hotstart
h = hotstart('hotstart.yaml',modules=['TEM','SAL','SED'])
v1 = h.create_hotstart()
coll = h.mesh.plot_elems(v1['tr_el'].values[:,-1,0],clim=(15,22))
cb = plt.colorbar(coll)
plt.axis('off')
plt.title('Regional Temperature')
plt.tight_layout(pad=1)
```

A map of the interpolated surface temperature field from the generated hotstart file is presented in Figure 5-1.

Figure 5-1 A spatial map of surface water temperature generated by merging observations from both USGS Polaris cruise and continuous observational data throughout the SCHISM domain



5.2.3.1 Example hotstart YAML files

A list of hotstart YAML files with various purposes and initialization methods is presented in Table 5-1.

Table 5-1 A list of example hotstart YAML files for various application and initialization methods

Application	Modules	Tracers	Example YAML file
Basic baroclinic run	NA	TEM, SAL	tracer_age/hotstart.yaml
Sediment transport	SED	TEM, SAL, SED_1, SED_2, SED_3	sed/hotstart.yaml

Application	Modules	Tracers	Example YAML file
hotstart based on a previous hotstart file with AGE and TRACER modules turned on	GEN, AGE	TEM, SAL, GEN_1, AGE_1	tracer_age/hotstart_from_prev_hotstart_GEN_AGE.yaml
hotstart with biogeochemical options: CoSiNE and ICM only	COSINE, ICM	TEM, SAL, COS_i (i varies from 1 to13), ICM_i (i varies from 1 to 25)	bio/hotstart_cosine_icm.yaml

Note: The path for the example files is: https://github.com/CADWRDeltaModeling/schimpy/tree/master/examples/schism_hotstart

5.2.3.2 Projection

Any spatial projection may be used for the hotstart generator, but the UTM (Universal Transverse Mercator) coordinate system for the Bay Delta grid is the default projection system, and the user does not need to supply a projection for the hotstart class if all the input data has the same projected coordinates. It is preferrable that the hotstart initialization operates within a consistent coordinate system.

The applications where input data and SCHISM mesh file (i.e., hgrid.gr3 file) differ in coordinate systems are not supported. When the destination mesh is in Latitude (Lat) and Longitude (Lon) coordinates (usually when the wind-wave model is invoked), a projected mesh is used first to create a hotstart file, and then the hotstart file will be used together with the desired hgrid file (Lat-Lon based) for SCHISM.

If the SCHISM input horizontal grid file (hgrid.gr3) is based on World Geodetic System, 1984 revision (WGS 84), it can be easily converted to UTM coordinate system using the following example script:

```
from schimpy.schism_mesh import read_mesh, write_mesh
import schimpy.schism_hotstart as sh
mesh =  read_mesh('hgrid.gr3',
                   proj4='EPSG:4326')          # WGS 84
mesh_new = sh.project_mesh(mesh,'EPSG:32610')  # UTM Zone 10N.
write_mesh(mesh_new,'hgrid_new.gr3')
```

5.2.3.3 Modules and variable names

The default model variables that must be initialized for any SCHISM runs are elevation, temperature (TEM), salinity (SAL), velocity_u, velocity_v, and velocity_w, where only temperature and salinity are tracer variables. In the hotstart.yaml file, the names of these variables should use these names. When the sediment module (‘SED’) is turned on, a few extra tracer variables and model variables related to a sediment bed are also needed. The python function `schism_hotstart.describe_tracers` can be applied to the main control input file for schism (param.nml) to produce a complete list of tracer variables (including temperature and salinity) for any combination of add-on modules in SCHISM. For example, the list for the sediment transport module of:

```
ntracers,ntrs,irange_tr,tr_mname=describe_tracers(“param.nml”,
modules=[’SED’])
```

will parse param.nml to output a complete list of tracer variables `tr_mname = [ ‘TEM’,’SAL’,'SED_1', 'SED_2', 'SED_3']` that need to be initialized to run the model given the specified enabled modules. Note that either ‘temperature’ or ‘TEM’ and ‘salinity’ or ‘SAL’ can be used as variable names for the hotstart.yaml file, and they are by default required tracer variables in a hotstart file unless the “modules” variable is defined as an empty list (modules=[]). A hotstart without tracer variables is useful to initialize a SCHISM run in barotropic mode where no tracer transport is modeled.

It is helpful to know the list of tracer variables because, in the final stage of generating a hotstart.nc, all tracer variables in the `tr_mname` list will be combined in order to create the three tracer variables required to initialize

SCHISM: tr_el (all tracers on element [elem, nVert, ntracers]), tr_nd (all tracers on node [node, nVert, ntracers]), and tr_nd0 (==tr_nd). The tracer input data can be specified either on node (by .ic file) or on element (by .prop file). If not defined on node, tracer data will first be interpolated to node to generate tr_nd0 and tr_nd and then to element center to generate tr_el by averaging all node values attached to the element.

Additional sediment bed variables must be initialized representing sediment bed properties: SED3D_dp, SED3D_rough, SED3D_bed, and SED3D_bedfrac. These variables are not 3D tracer variables (and will not be combined into tr_* in the hotstart.nc) and therefore are not listed in tr_mname. Instead, they will be saved as independent variables in the hotstart.nc file.

The currently available modules in SCHISM are listed below. Those implemented in schism_hotstart.py and tested in SCHISM runs are indicated by "Done".

! 1: T (default, Done)
! 2: S (default, Done)
! 3: GEN (Done)
! 4: AGE (Done)
! 5: SED3D or SED (Done)
! 6: EcoSim or ECO (In testing)
! 7: ICM: ICM and/or ICM_PH (In testing)
! 8: CoSINE: COSINE (Done)
! 9: Feco: FIB (not sure if the tracers can be initialized in a hotstart file)
! 10: TIMOR (not implemented)
! 11: FABM (not implemented)

5.2.3.4 Centering

The centering refers to where a model variable is specified relative to the mesh topology. For instance, in the schism algorithm, some data are defined on node and tracers are generally defined on prism (cell) center. Options include node, edge, and elem (element) in the horizontal direction and on the whole- or half-level in the vertical direction. Each model variable is tied to a unique centering option. A list of centerings and the corresponding model variables is presented in Table 5-2. These locations are automatically assigned according to the variable names in the hotstart preprocessor. Note that all tracer variables must be prism-centered (element center at half-level).

Table 5-2 Centering of typical model variables

Centering	Position	Example model variables
node2D	node at surface or bed	Elevation (elev), SED3D_rough, SED3D_dp
node3D	node at whole level	All variables related to total kinetic energy (TKE). The only available options are: (1) 0 (default); and (2) using the values provided by an existing hotstart.nc file.
edge	edge center at whole level	velocity_u, velocity_v
elem	element center at whole level	velocity_w
prism	prism center or element center at half level	tracers (salinity, temperature, etc.)
bed	element center at sediment layer	SED3D_bed
bedfrac	element center at sediment layer with 3 sediment properties	SED3D_bedfrac

5.2.3.5 Initializers

The initializers define the methods chosen to initialize the variable fields. Six options are currently available, described below:

- text_init: Text initialization is based on 2D map input from either *.prop (values defined on elem) or *.ic (on node) files, which are reduced-dimension formats that are native to SCHISM and described in the SCHISM manual (http://ccrm.vims.edu/schismweb/schism_manual.html). One input text file for each variable is required.
- simple_trend: Can either be a number or an equation that depends on the projected x, y, and z in native coordinates such as degrees or meters. One particularly common example is when the initial surface is put just below the bed in dry areas, which is given by max (-z-0.1, 0.97), where 0.1 m is the distance below the bed (z represents bed depth, which is positive downwards), and 0.97 sets the initial elevation (in North American Vertical Datum of 1988 [NAVD88] meters) for elements that are wet — in other words, a bed below 0.97m elevation.
- extrude_casts: 3D spatial interpolation based on transect data (data points collected along a trajectory), such as boat cruising along a transect taking vertical “casts” of data at multiple horizontal points.

This category of data includes USGS cruise data from the USGS Polaris and its successor, the RV Peterson (Schraga et al. 2020). Two input files are required: a CSV file that defines the UTM locations for all the stations (with columns "Station", "x", and "y") and a CSV file that contains the data (with columns "Station", "Depth (m)", and a user-defined variable name). A simple nearest-neighborhood method in the horizontal direction and a linear interpolation-extrapolation method in the vertical direction are currently applied to create a 3D tracer field based on the observed data.

- obs_points: 2D spatial interpolation based on time series of continuous observations across multiple stations. One CSV file is required for the option (with columns "x", "y", and a user-defined variable name). A 2D horizontal field is interpolated from the observational data, and only an inverse-distance-weighing method is currently available to perform the interpolation task. The variable field is assumed to be uniform in the vertical direction.
- patch_init: Regional based method. A polygon shape file or YAML file that divides the mesh into different regions is required, and a function geo_tools.partition_check will be called to check that the computational domain divided by regions (defined in the shapefile or YAML file) is unique and complete. If a cell is not within the geographic boundary of any defined regions (orphaned cells), it will be assigned to the nearest region. If a cell is positioned within multiple regions, it will be assigned to only one region to satisfy uniqueness requirement. The region names defined in the hotstart.yaml must match the "region" attribute for each polygon defined in the shapefile or YAML file. All other initializers (text_init, simple_trend, extrude_casts, obs_points and hotstart_nc) can be applied to each individual region and stitched together to generate the final tracer field for the entire domain.
- hotstart_nc: Initialization option using a source hotstart file (hotstart.nc) produced by a previous run. The source grid can be different from the destination grid, and a spatial interpolation will be applied to map the values from the old grid to the new one. Note that performing vertical interpolation between the two full meshes can be very slow. There are two ways to make the process faster if the two meshes are mostly the same except for a small percentage of cells.

- patch_init: In the following code, only the cells within the polygon region “edb” are interpolated vertically; all the other cells are assigned a region name ‘other’ and the nearest cell values from hotstart_source.nc are used (no vertical interpolation).

```
salinity:
    centering: prism
    initializer:
        patch_init:
            smoothing: False
            # the attribute 'region' in the shapefile
            # needs to match with the region values below.
            regions_filename: edb_polygon.shp
            regions:
              - region: edb
                initializer:
                    hotstart_nc:
                        data_source: hotstart_it=2688000.nc
                        source_hgrid: hgrid.gr3
                        #if the source hgrid is different
                        source_vgrid: vgrid.in.3d
                        #if the source vgrid is different
                        vinterp: True
              - region: other
                initializer:
                    hotstart_nc:
                        data_source: hotstart_it=2688000.nc
                        source_hgrid: hgrid.gr3
                        source_vgrid: vgrid.in.3d
                        vinterp: False
```

- Define distance_threshold: When distance_threshold is defined, the interpolation method will only be applied to the new cells (those cells with distance to the nearest cells greater than a predefined distance_threshold). The distance_threshold has many uses, but an important one is the case where parts of the source and destination mesh are supposed to be very similar both horizontally and vertically. In this case values are copied over rather than interpolated.

Two methods are available to create variable fields for the new cells:

3. The “nearest” option will use the vertical profiles from the nearest horizontal cell of the source mesh to interpolate vertically to the new cells.
4. Define a function of “x”, “y”, or “z”.

```
distance_threshold: 10     # the unit here is meter for UTM grid.

method: nearest            # method: nearest interpolate vertically
method: np.maximum( -(z+0.1), 0.97)  # a simple function (for flooded
island case)
```

   - option: vinterp = False

   if distance_threshold is not defined, an additional key “vinterp” can be set to switch on/off vertical interpolation. The default value of “vinterp” is False, so if no vinterp value is given, the script assumes that no vertical interpolation will be performed unless the number of the vertical layers between the two meshes is different. When the number of vertical layers is different, vertical interpolation of the entire mesh must be applied. Note that performing vertical interpolation can be very slow and should not be turned on for the entire domain unless necessary.

5.2.4 The auxiliary functions

A few new auxiliary functions were created in schimpy. Although they were developed for the hotstart preprocessor, they could also be used to support other efforts. A list of the most useful functions is explained and presented below.

In class schism_mesh (schism_mesh.py)

- to_geopandas (self,feature_type='polygon',proj4=None,shp_fn=None, node_values=None, value_name=None, create_gdf=True)

  A function that converts schism horizontal mesh into polygon or points as geopandas data frames and saves them as shapefiles. This step takes advantage of the powerful spatial analysis and domain manipulation tools provided by geopandas PYTHON library.

- plot_elems(self,var=None,ax=None,inpoly=None,**kwargs)

  Plot a 2D map of *var* (which should be input as a 1D array of values on elem, e.g., modeled surface salinity field at one time step) on SCHISM computational mesh. If input var is None, only the mesh grid itself will be plotted.

- plot_nodes(self,var,ax=None,inpoly=None,**kwargs)

  Similar to the above, but plot a 2D map of variables on node.

- plot_edges(self,var,ax=None,size=500,inpoly=None,**kwargs)

  Similar to the above, but plot a 2D map of variables on edge.

- schism_mesh.compare_mesh()

  Function designed to compare two meshes and map each node of mesh2 to mesh1.

- schism_hotstart.project_mesh()

  Function to project the nodes of a mesh from one coordinate system to another one.

In geo_tools.py

- partition_check(mesh,poly_fn,centering='nodes')

  Check if the schism mesh division by the polygon features in poly_fn is unique and complete.

  The input poly_fn can either be a shape file or a YAML file that specifies the boundary coordinates of the polygons.

  The partition check performed for the SCHISM horizontal mesh is based on either node (centering='node') or element (centering='elem'), and the function checks:

  (1) if there are any orphaned nodes/elems (nodes/elems that do not fall within any polygon defined above). If so, the nearest polygon will be applied to the orphaned nodes/elems.

(2) if any nodes/elems were assigned to multiple polygons. If so, only the last assigned polygon will be applied to categorize the nodes/elems.

- ll2utm(lonlat,proj4=None) and utm2ll(utm_xy,proj4=None)

  Conversion between WGS 84 (Lat, Lon) and UTM coordinate systems. No input for proj4 is required if the input or output Lat-Lon coordinate is based on WGS84.

In schism_hotstart.py

- read_param_in(nml_fn)

  Read param.nml or other input files (e.g., sediment.nml) and generate a dictionary object with (key, value) pairs for all the parameters.

- describe_tracers(param_in,modules=['TEM','SAL'])

  This function returns the total number of tracers (ntracers), the number of tracers for each add-on module, the starting and ending indices in the tracer list for the tracer items in each module, and a list of tracer names based on the input list of modules and param.nml. More details about this function are in Section 5.2.3.2.

- hotstart_to_outputnc(hotstart_fn,init_date,hgrid_fn = 'hgrid.gr3',vgrid_fn = 'vgrid.in',outname="hotstart_out.nc")

  This function converts hotstart.nc to schism output NetCDF file format that can then be read and visualized by VisIt. The hotstart.nc can be generated by schism_hotstart.py or SCHISM itself.

5.3 Nudging preprocessor

5.3.1 Introduction

The nudging preprocessor (schism_nudging.py) is another new tool in schimpy. The script performs two tasks: (1) providing boundary conditions of temperature and salinity off the California coast based on the simulated ROMS modeling results by CenCOOS (https://www.cencoos.org/), and (2) providing nudging values and weights in the interior of the model domain to nudge the model state closer to observations.

5.3.2 Methods

5.3.2.1 How nudging is implemented in SCHISM.

In SCHISM source code (hydro/schism_steps.f90), nudging is achieved by the following equations:

$$nu\_el_{j,k,i} = \left(\sum_{k=k-1}^{k} \sum_{ei=1}^{i34(i)} nu\_nd_{j,k,elnode(i)_{ei}}\right)/2/i34(i) \qquad (1)$$

$$tr\_el_{j,k,i} = tr\_el_{j,k,i} * (1 - trnu) + tr\_nu_{j,k,i} * nu\_el_{j,k,i} \qquad if\ \ nu\_el_{j,k,i} > -99.0 \qquad (2)$$

where:

$nu\_el_{j,k,i}$ is the nudging value for tracer *j*, vertical level *k*, and element *i*, and it is calculated by averaging the nudging values on node ($nu\_nd_{j,k,elnode(i)}$) and over the two adjacent vertical levels (*k* and *k+1*). $nu\_nd_{j,k,elnode(i)}$ for each tracer is defined in the *_nu.nc file (see Section 5.3.2.2), where the * is a wildcard referring to one tracer such as SAL or TEM.

$elnode(i)$ is a function that finds the indices of nodes attached to element *i; ii* iterates all nodes that belong to the element *i*, which have a range from 1 to i34. For SCHISM meshes, either triangular (*i34*=3) or quadrangular (*i34*=4) grids are allowed.

$tr\_el_{j,k,i}$ on the RHS of equation (1) represents the modeled tracer fields, and that on the LHS means a combination of modeled and nudging tracer fields. Weight factor $tr\_nu_{j,k,i}$ controls how much the combined fields relax toward the nudging fields and must lie between 0 and 1. A weight factor $tr\_nu_{j,k,i}$=0 means no nudging (combined fields are equal to the modeled fields), and $tr\_nu_{j,k,i}$=1 means that the combined fields are equal to the nudging fields (i.e., the modeled fields are discarded).

$tr\_nu_{j,k,i}$ is a weight factor that combines a horizontal weight ($tr\_nu\_h_{j,i}$) and vertical weight (vnf).

$$tr\_nu_{j,k,i} = (tr\_nu\_h_{j,i} + vnf_{j,k,i}) * dt \qquad (3)$$

dt is the computational time step.

$tr\_nu\_h_{j,i}$ is the horizontal weight on element *i* for tracer *j* calculated by averaging all weight factors on node that belong to the element.

$$tr\_nu\_h_{j,i} = \left(\sum_{ii=1}^{i34(i)} tr\_nudge_{j,elnode(i)_{ii}}\right)/i34(i) \tag{4}$$

where *i34(i)* is the number of nodes (3 or 4) in the element *i*. tr_nudge are the horizontal weights on node defined in *_nudge.gr3 for each tracer (see Section 5.3.2.2), where again * is a wildcard representing the particular model variable such as SAL or TEM.

In schimpy/schism_nudging.py, for point observations, the horizontal weight (tr_nudge) is defined as a Gaussian-shaped function (with the peak 1 at the observation location) divided by a nudging time scale (t_{second}). The Gaussian weights are calculated by the node distance from a predefined central point and a length scale (L).

$$tr\_nudge_{j,ni} = e^{-\frac{(x_{ni}-x_0)^2+(y_{ni}-y_0)^2}{2L^2}} \Big/ t_{second} \tag{5}$$

Where (x_{ni}, y_{ni}) and (x_0, y_0) are the coordinates of node *ni* and a predefined central point, respectively. (x_0, y_0), L, and t_{second} are all defined in a nudging YAML file, which is a master input file for the nudging preprocessor (see Section 5.*3.2.3* for more details).

The vertical weight $vnf_{j,k,i}$is dependent on depth and given by:

$$vnf_{j,k,i} = \begin{cases} vnf1, & Ze_{k,i} > -vnh1 \\ vnf1 + (vnf2 - vnf1) * \frac{Ze_{k,i}+vnh1}{vnh1-nvh2}, & -vnh1 \geq Ze_{k,i} \geq -vnh2 \\ vnh2, & Ze_{k,i} < -vnh2 \end{cases} \tag{6}$$

where vnf1, vnf2, vnh1, and vnh2 are vertical relaxation factors defined in the master input file param.nml.

This creates depth-dependency in which the nudging factor can gradually strengthen or weaken with depth. The current default values for vnf1 and vnf2 are 0, so the depth-dependence weight is 0.

5.3.2.2 The design of the preprocessor

A class "nudging" in schism_nudging.py was created to generate the nudging files (both *_nu.nc and *_nudge.gr3) required to apply nudging to SCHISM. *_nu.nc stores the 3D nudging values on node and whole vertical levels for the corresponding tracer, *_nudge.gr3 stores the 2D horizontal weights on node for the corresponding tracer, and the asterisk (*) here is a generic stand-in for a tracer name (e.g., TEM for temperature and SAL for salinity). A more detailed description on how the nudging values and weights influence the modeled field is given in Section 5.3.2.1. The computational domain is assumed to be divided into multiple polygons (regions), with different nudging options to be applied to the different polygons. The script iterates through the polygons, creates nudging weights and values according to the defined method for each polygon (`nudging.create_region_nudging`), organizes the nudging data, and finally concatenates the nudging data (`nudging.concatenate_nudge`) to create a combined dataset for the entire domain. A function that bundles all the above steps is `nudging.create_nudging`.

5.3.2.3 Nudging files required to run SCHISM and schism_nudging.py.

For each nudging variable, SCHISM requires two nudging files: *_nudge.gr3 and *_nu.nc.

*_nudge.gr3 (e.g., SAL_nudge.gr3 or TEM_nudge.gr3) defines the nudging weights to be applied to each node in SCHISM. When node weight is equal to or less than zero, no nudging is applied to the node.

*_nudge.gr3 (e.g., SAL_nu.nc or TEM_nu.nc) defines the nudging values to be applied to each node at each vertical level.

schism_nudging.py script creates both *_nudge.gr3 and *_nu.nc input files required by SCHISM.

5.3.3 nudging usage

Similar to the hotstart preprocessor, the nudging processor can be invoked as a standalone utility or programmatically through python.

To invoke it as a standalone utility, type the following command in a python prompt (such as miniconda):

```
create_nudging --input nudging.yaml
```

A complete listing of available options can be viewed by typing:

```
create_nudging --help
```

To invoke the nudging processor programmatically through python, type the following commands in a python editor or write a script that includes these commands.

```
from schimpy import schism_nudging

yaml_fn = 'nudging.yaml'

nudging = schism_nudging.nudging(yaml_fn,proj4 ='EPSG:32610') #proj does not need to
be defined if the grid is already in utm coordinates.

nudging.read_yaml()

nudging.create_nudging()
```

An example YAML file (https://github.com/CADWRDeltaModeling/schimpy/blob/master/examples/schism_nudging/nudge.yaml) is presented to provide an example of generating combined weights and nudging fields from a combination of ROMS modeling results at the ocean boundary, a single-point observation at GZL (Grizzly Bay), and multi-point observations from various observational sites across the Delta. The combined temperature nudging values and weights are shown in Figure 5-2. A close-up view of the nudging salinity compared with the modeled ROMS temperature and salinity at the ocean boundary is shown in Figure 5-3.

Detailed options to set up nudging.yaml are presented in the following sections.

Figure 5-2 The left panels show the nudging values (nu_salt) for salinity and the right panels show the product of nudging values and weights

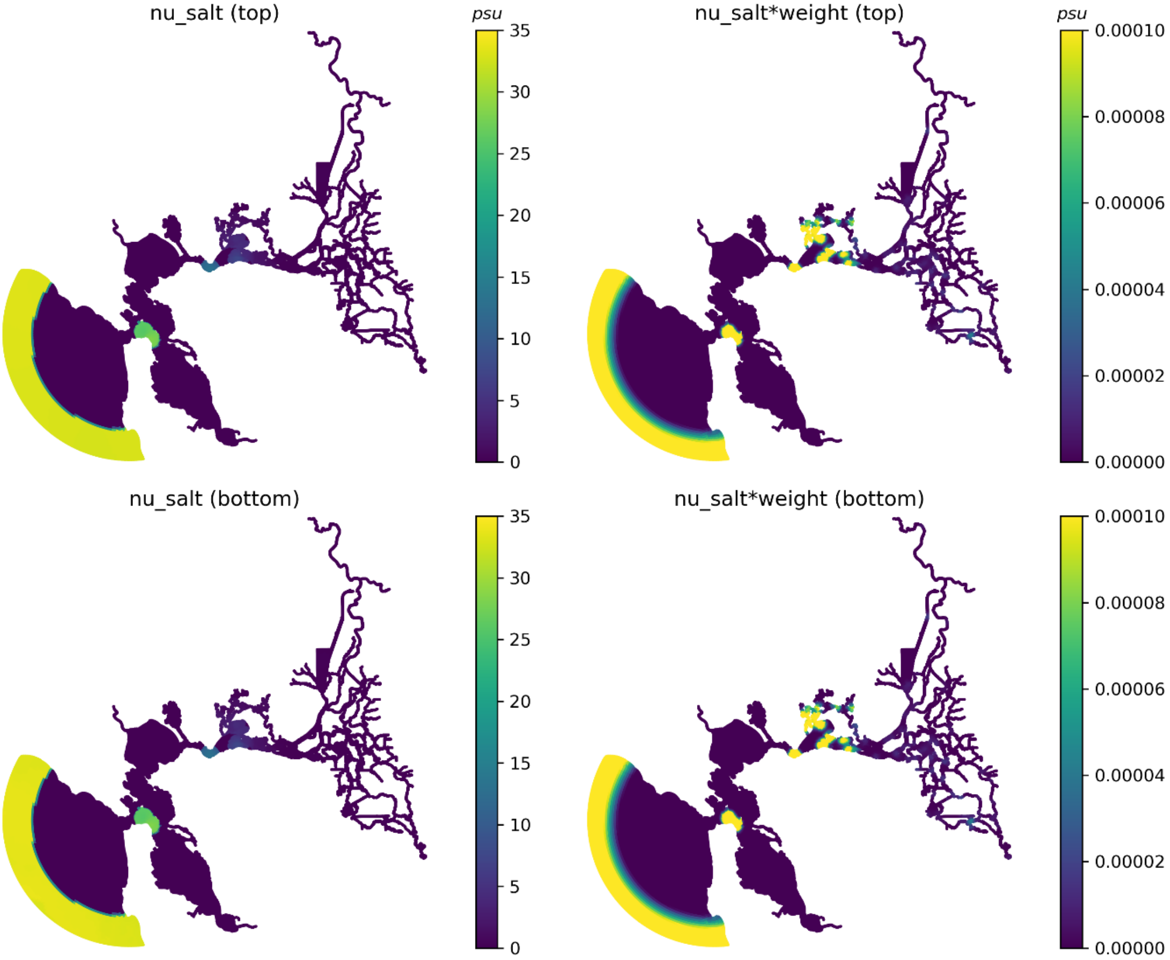
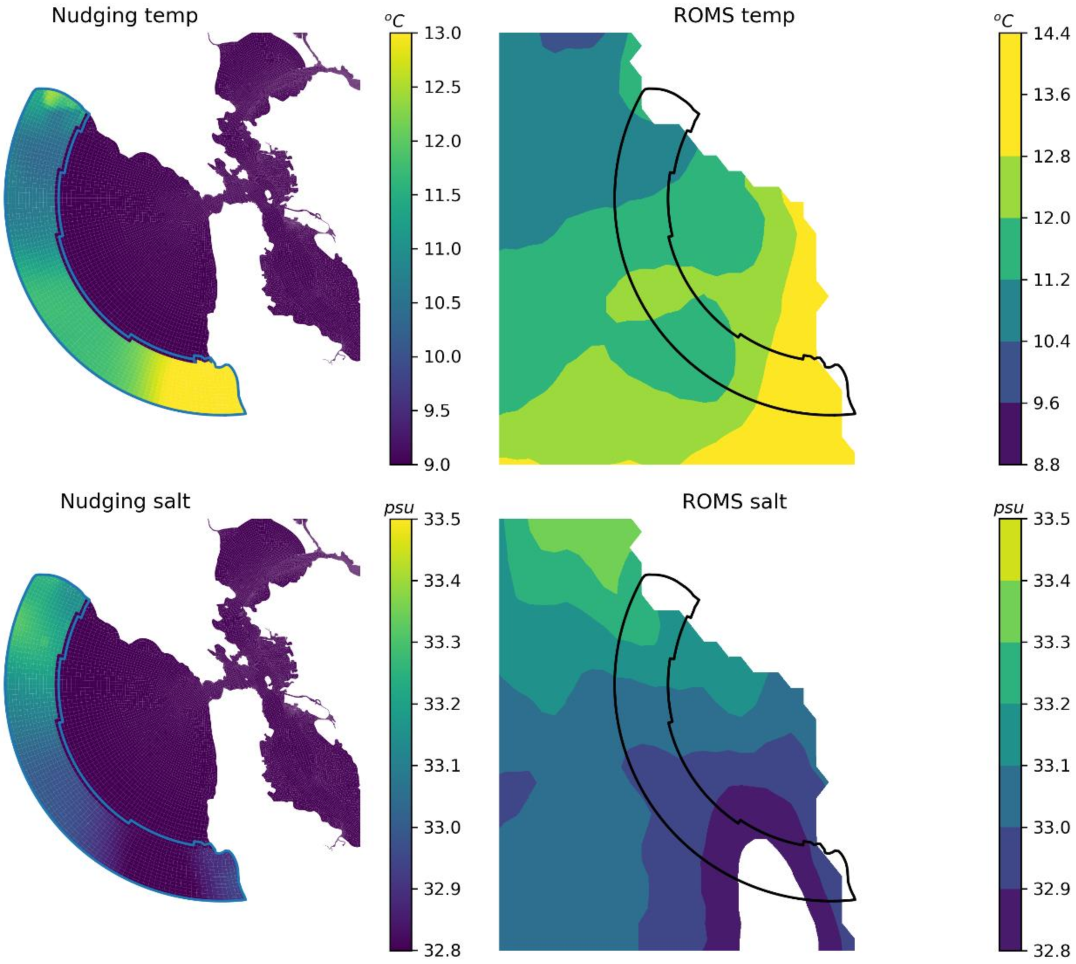


Figure 5-3 The left panels show the nudging values for temperature (top) and salinity (bottom) implemented in SCHISM and the right panels show the modeled temperature (top) and salinity (bottom) from ROMS



5.3.4 nudging options

Two nudging options are currently available: roms and obs.

5.3.4.1 roms option

This option was designed to generate ocean open boundary condition of temperature and salinity from the simulated ROMS modeling results from the CenCOOS program (http://thredds.cencoos.org/thredds/catalog.html?dataset=CENCOOS_CA_ROMS_DAS). "Vertices" define the boundary of the region that this option will be applied to.

5.3.4.2 obs option

The obs option nudges model results to observation points in a polygon region defined by "vertices." When the nudging area is not defined (verticies: None), this option of nudging will only be applied to nodes where the nudging weights are greater than 0. "Attribute" defines the method used to generate the weights; currently only the gaussian method is available. The coordinates (x and y) of the observational points, length_scale and time_scale, need to be defined for the attribute. Note that when the weights are set to zero when they become less than 10^{-3} of the maximum Gaussian weight. "Interpolant" defines the method to generate spatial nudging values, the data source (in either *.csv or *.nc format), and the variables to be interpolated. Only the nearest and inverse distance weighing methods are available for spatial interpolation.

References

Ateljevich E. 2000. "DSM2-QUAL Initialization." In In: Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 21st Annual Progress Report to the State Water Resources Control Board. Chapter 11. Sacramento (CA): Bay-Delta Office. Delta Modeling Section. California Department of Water Resources.

Ateljevich E, Nam K, Zhang Y, Wang R, Shu Q. 2014. "Bay-Delta SELFE calibration overview." In: Methodology for Flow and Salinity Estimates in the Sacramento-San Joaquin Delta and Suisun Marsh. 35th Annual Progress Report to the State Water Resources Control Board. Chapter 7. Sacramento (CA): Bay-Delta Office. Delta Modeling Section. California Department of Water Resources.

Schraga TS, Nejad ES, Martin CA, and Cloern JE. 2020. USGS measurements of water quality in San Francisco Bay (CA), beginning in 2016 (ver. 3.0, March 2020): U.S. Geological Survey data release, https://doi.org/10.5066/F7D21WGF.