Final Report – Appendices (June 10, 2019)

Potential Implications of Sea-Level Rise and Changing Rainfall for Communities in Florida using Miami-Dade County as a Case Study

Florida Department of Business and Professional Regulation
Florida Building Commission

And

Sea Level Solutions Center (SLSC), Florida International University (FIU)

Table of Contents

List of Figures

Appendix I. Development of average May-October groundwater level maps under future sea level rise scenarios (Task I)

This appendix describes the work performed to develop average May-October groundwater level maps under future sea level rise scenarios as part of Task I. The Urban Miami-Dade (UMD) MODFLOW model was used for this purpose. It is a peer-reviewed model developed by the USGS (Hughes and White, 2016) which uses MODFLOW-NWT and the Surface-Water Routing (SWR1) Process to simulate surface water stages, discharges, and surface water/groundwater interaction. It also uses the Sea Water Intrusion (SWI2) Package to simulate saltwater intrusion into the surficial aquifer.

As part of this task, we performed two main future scenario runs and three additional sensitivity runs using the calibrated Miami-Dade MODFLOW model developed by the USGS. This appendix starts by describing each of the model data files that were modified from the original UMD model for use in these future scenario and sensitivity runs. This is followed by a discussion on the model setup for these five runs, model results, and finally a list of model limitations and recommendations.

Future land use

The future scenarios previously simulated by the USGS using the Urban Miami-Dade (UMD) MODFLOW model used 2008 land use data to develop direct surface-water runoff, agricultural water demand, recreational irrigation, and monthly crop coefficient values (Hughes and White, 2016). However, for this project, we were able to obtain 2030 predicted land use from the Adopted 2020-2030 Comprehensive Development Master Plan (CDMP) for Miami-Dade County (Jerry H. Bell, Department of Regulatory and Economic Resources, Planning Division, pers. comm.). The map can be found at: https://www.miamidade.gov/planning/library/reports/planning-documents/cdmp/cdmp-land-use-map-2020-2030.pdf.

Jenifer Barnes at the South Florida Water Management District (SFWMD) assisted with cross-walking the land use categories in the 2030 CDMP into Florida Land Use, Cover, and Forms Classification System (FLUCCS) codes. In addition, since the 2030 CDMP does not provide detailed land use in natural and agricultural areas, Ms. Barnes added more details in these areas by intersecting the dataset with the SFWMD's 2018 permitted land use dataset. Subsequently, we added the 2018 permitted extent of quarry lakes from a shapefile provided by SFWMD which was used in the 2018 base case scenario for their RSMGL model. The FLUCCS codes were subsequently re-classified into the South Florida Water Management District's basic land use types (BLU) (Figure 1) as described in Table 1 of Hughes and White (2016). Finally, each MODFLOW grid cell was assigned its predominant BLU code (Figure 2). Table 1 (Figure 4) below shows the percentage of cells on the onshore part of the model that had a predominant land use equal to a given BLU category in the year 2008 (Figure 3) and 2030 (Figure 2). Note that this table differs from Table

2 in Hughes and White (2016) in that their table shows overall polygon areas in each BLU category as opposed to percentage of cells with that BLU as predominant shown here.

Table 1. Percentage of cells on the onshore part of the model with predominant land use in each basic land use category.

Basic Land Use Code	Description	Year 2008	Year 2030
1	Low Dens. Urban (LDU)	3.58%	3.97%
3	Med. Dens. Urban (MDU)	23.45%	24.48%
11	High Dens. Urban (HDU)	12.25%	13.58%
2	Citrus (CIT)	5.31%	4.90%
7	Row Crops (ROW)	4.39%	3.92%
8	Sugar Cane (SUG)	0.00%	0.00%
9	Irrigated pasture (IRR)	0.19%	0.11%
6	Shrubland (SHR)	2.24%	1.83%
18	Marl Prairie	4.60%	4.50%
4	Sawgrass	25.55%	25.72%
15	Cattail	0.00%	0.00%
19	Mix Cattail/Sawgrass	0.00%	0.00%
5	Wet Prairie	2.19%	1.77%
12	Forested Wetland	3.18%	2.72%
16	Forested Upland	0.39%	0.26%
13	Mangroves	3.85%	3.72%
14	Melaleuca	2.73%	2.33%
20	Open Water	5.55%	3.83%
30	Offshore	0.31%	0.31%
31	Rock Quarries	0.24%	2.06%
Total onshore area in square miles = 1094			

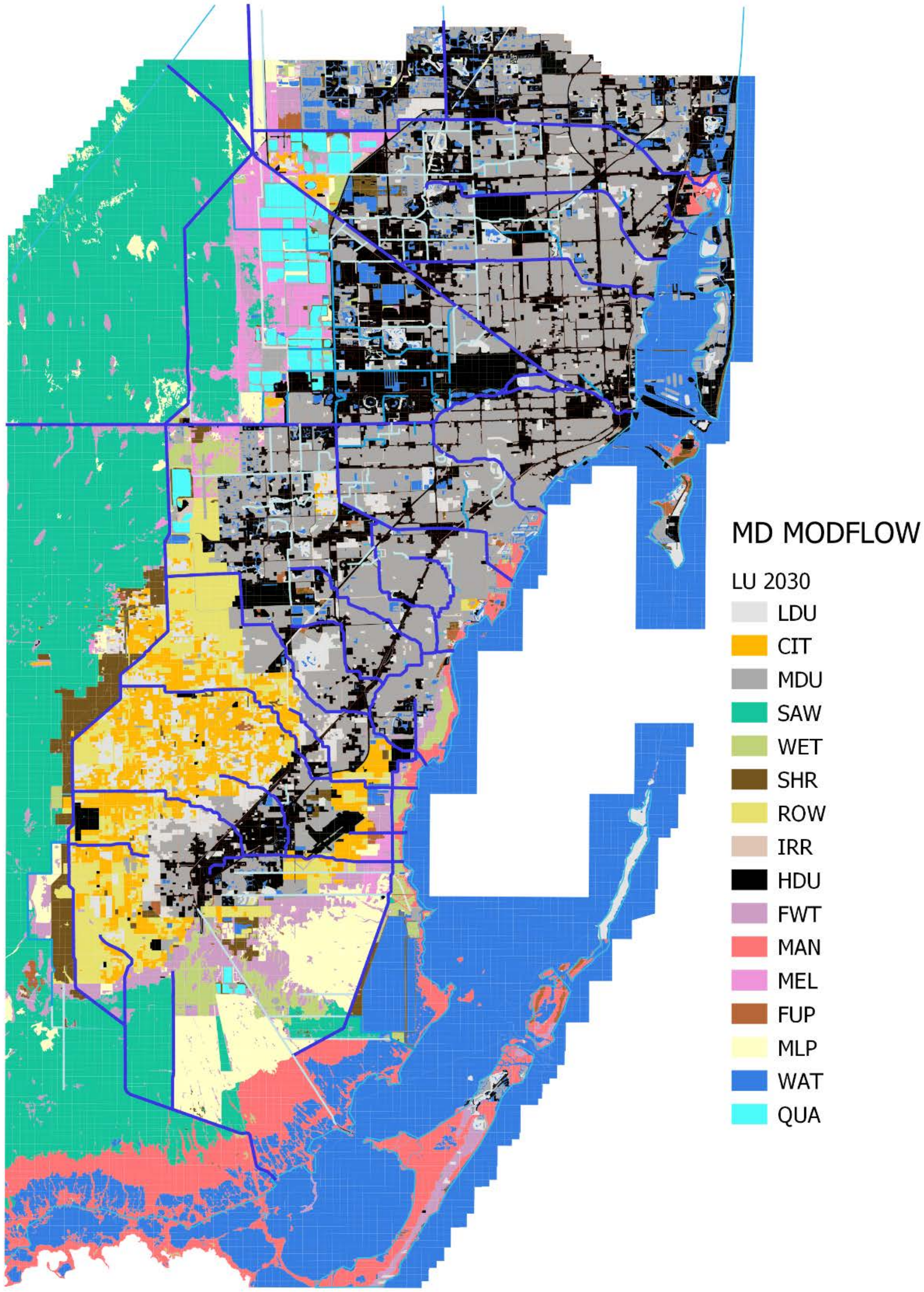


Figure 1. Polygon-scale basic land use codes for 2030.

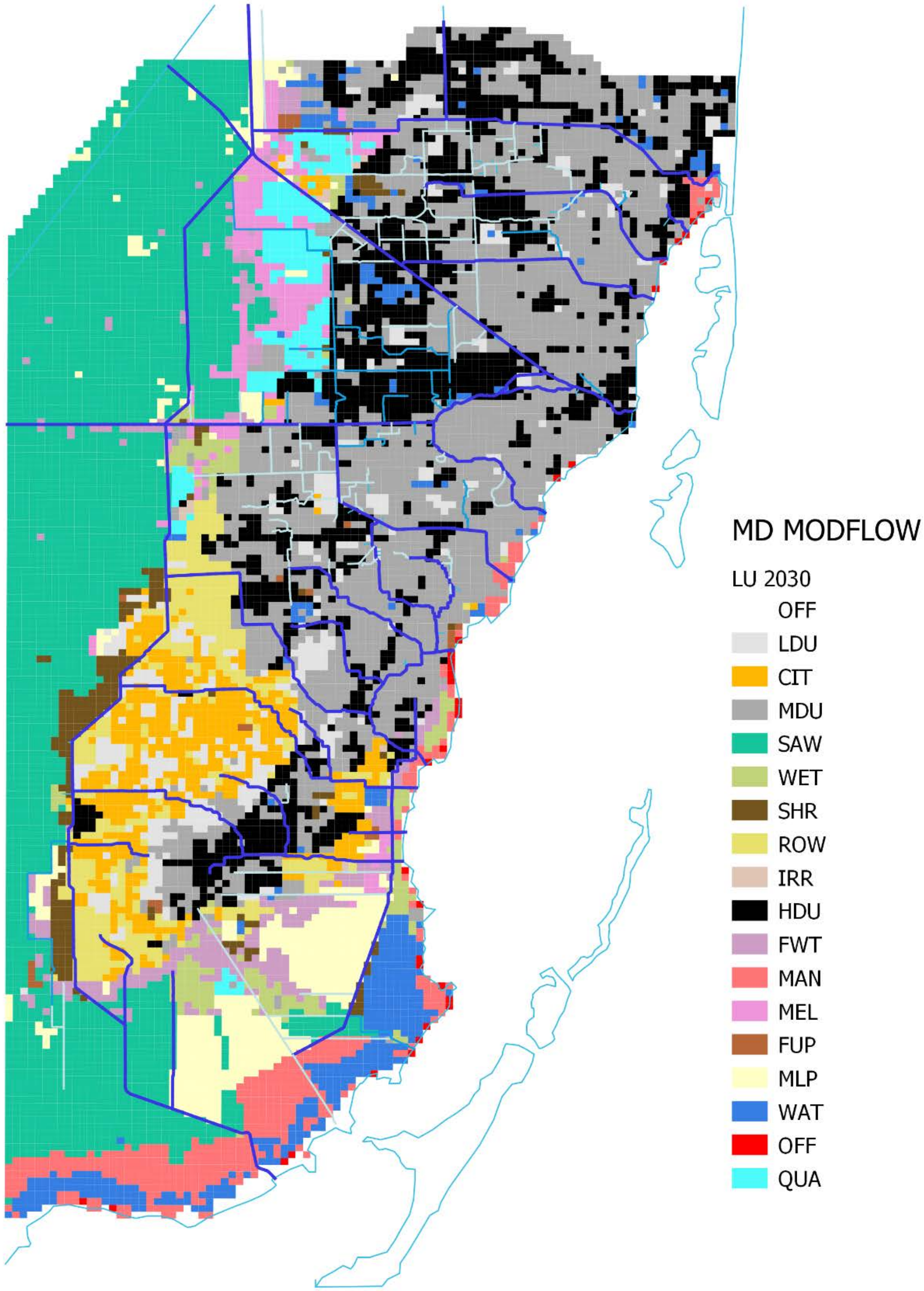


Figure 2. Predominant basic land use codes for model grid cells in 2030.

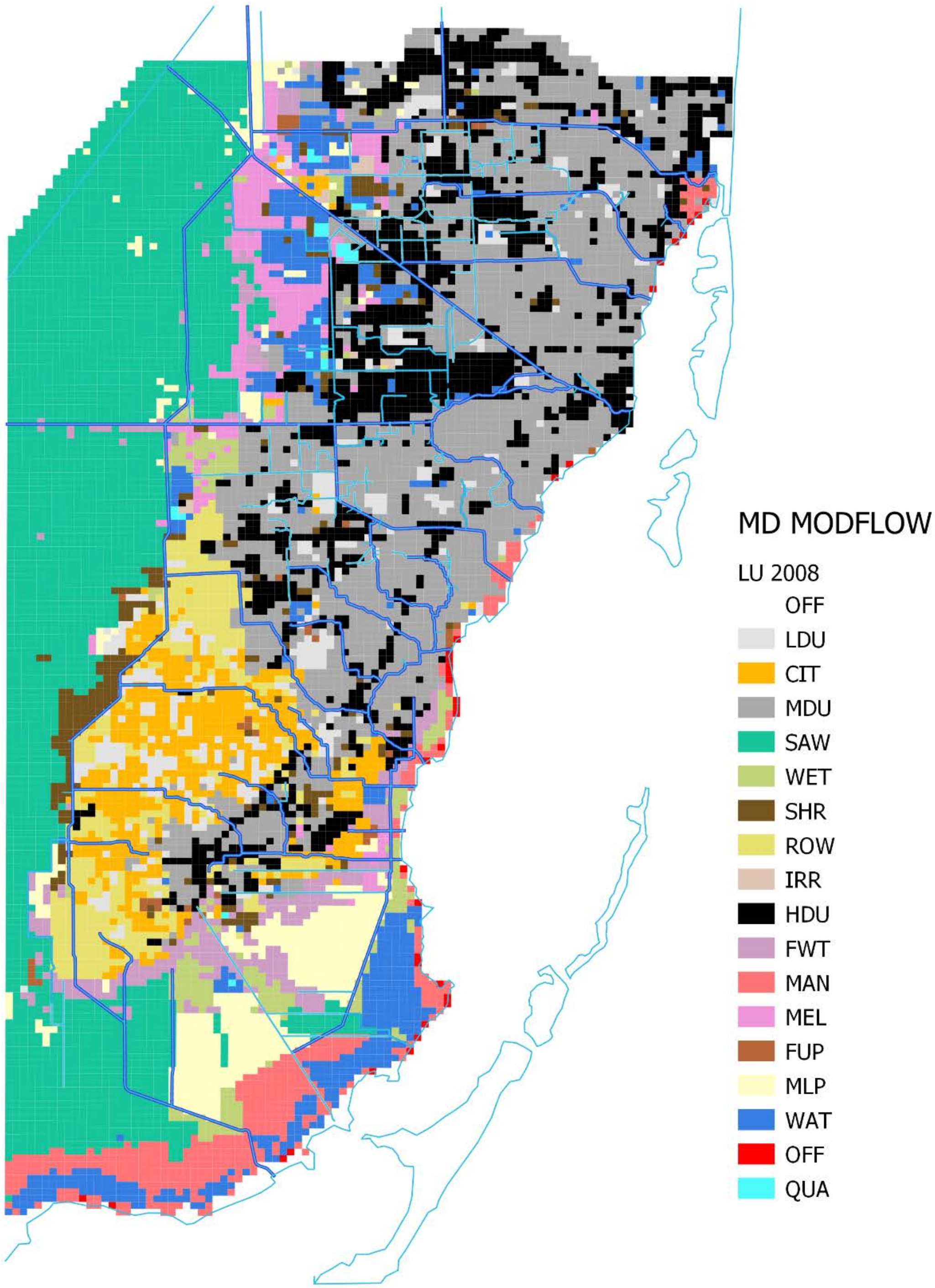


Figure 3. Predominant basic land use codes for model grid cells in 2008.

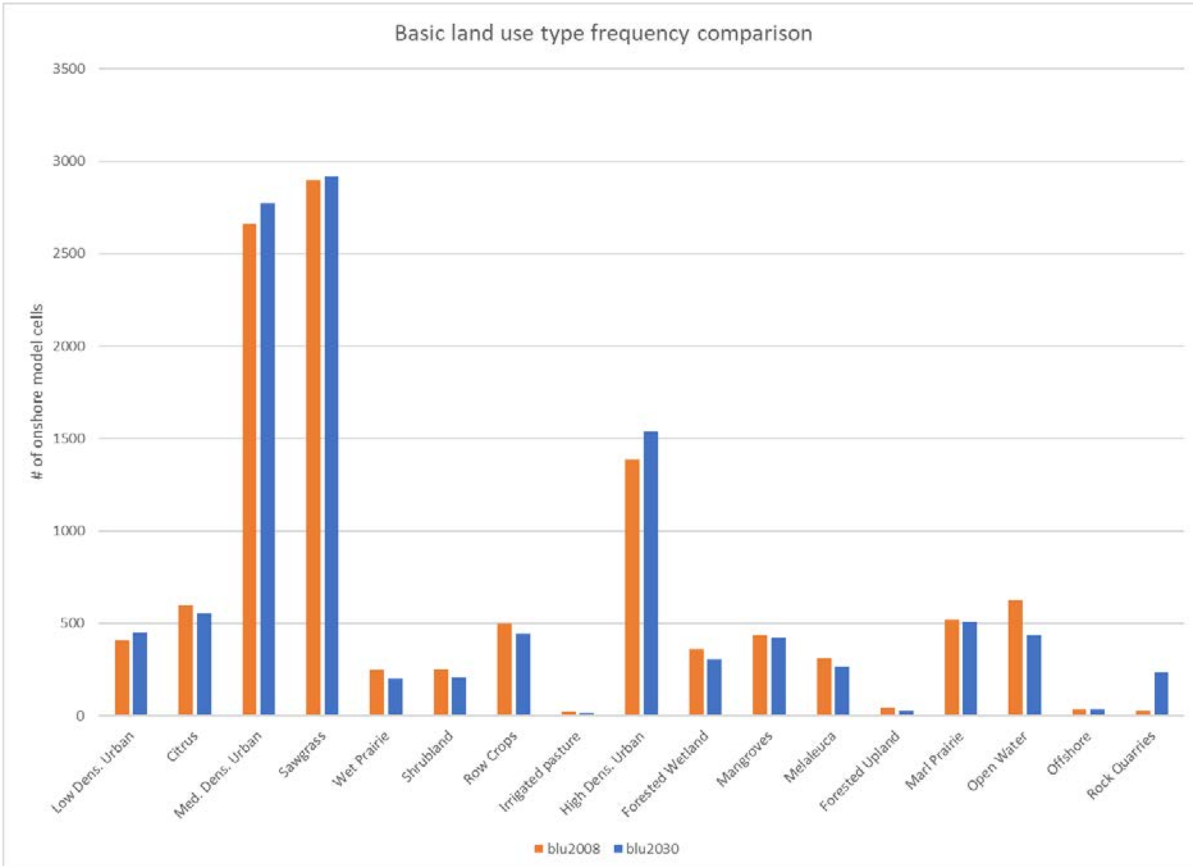


Figure 4. Number of onshore model cells with predominant basic land use category as given.

Future directly-connected impervious areas

Keith and Schnars (2004) provides a table of total impervious area (TIA) for each Level II/III land use code from the Miami-Dade County Department of Planning. After converting these land use codes into the equivalent FLUCCS codes, the TIA values were used to develop maps of directly-connected impervious area fractions (DCIA) based on a methodology similar to that of Hughes and White (2016). First, each land use polygon (Figure 1) was assigned a TIA based on its FLUCCS code (Table 2). Then DCIA was assumed to be 25% of TIA. Finally, an area-weighted average DCIA value was computed for each model grid cell based on the area of each grid cell occupied by each land use polygon on the grid cell. All open water, agricultural, and natural land uses were assigned TIA and DCIA values of 0.

Comparisons of DCIA fractions between the 2008 and 2030 land uses (Figure 5 and Figure 6) show that DCIA is expected to increase in the future as urban areas expand and densify. In the Miami-Dade MODFLOW model, DCIA * Rainfall is sent to the closest surface water feature in the SWR1 package (Figure 7), while (1 – DCIA) * Rainfall recharges the surficial aquifer.

Table 2. Total impervious area by FLUCCS code.

TIA (%)	FLUCCS CODES									
	1630	1660	1900	1920	2000	2110	2120	2140	2150	2160
	2210	2230	2240	2410	2420	2430	2500	2510	2540	2610
	3100	3200	3220	3300	4110	4200	4220	4240	4270	4340
0	4370	4410	5000	5110	5120	5200	5300	5410	5420	5430
	5710	6110	6111	6120	6170	6172	6180	6191	6210	6215
	6216	6250	6300	6410	6411	6420	6430	6440	6500	6510
	7200	7400	7430	7470	1650	1810				
3	1480	1820	1850	1860	1890	8115				
27	1460	8110	8113	8200	8300	8310	8320	8330	8340	8360
56.1	1700	1710	1730	1760						
64.62	1100	1110	1130	1180						
69.63	1330	1340								
70.5	1200	1210	1220	1230	1300	1310	1320	1350		
74.55	1500	1550	1560	1620						
74.71	1423	1840	8350							
78.42	1830	1870								
79.73	8120	8140	8150							
82.6	1400	1410	1411	8100						

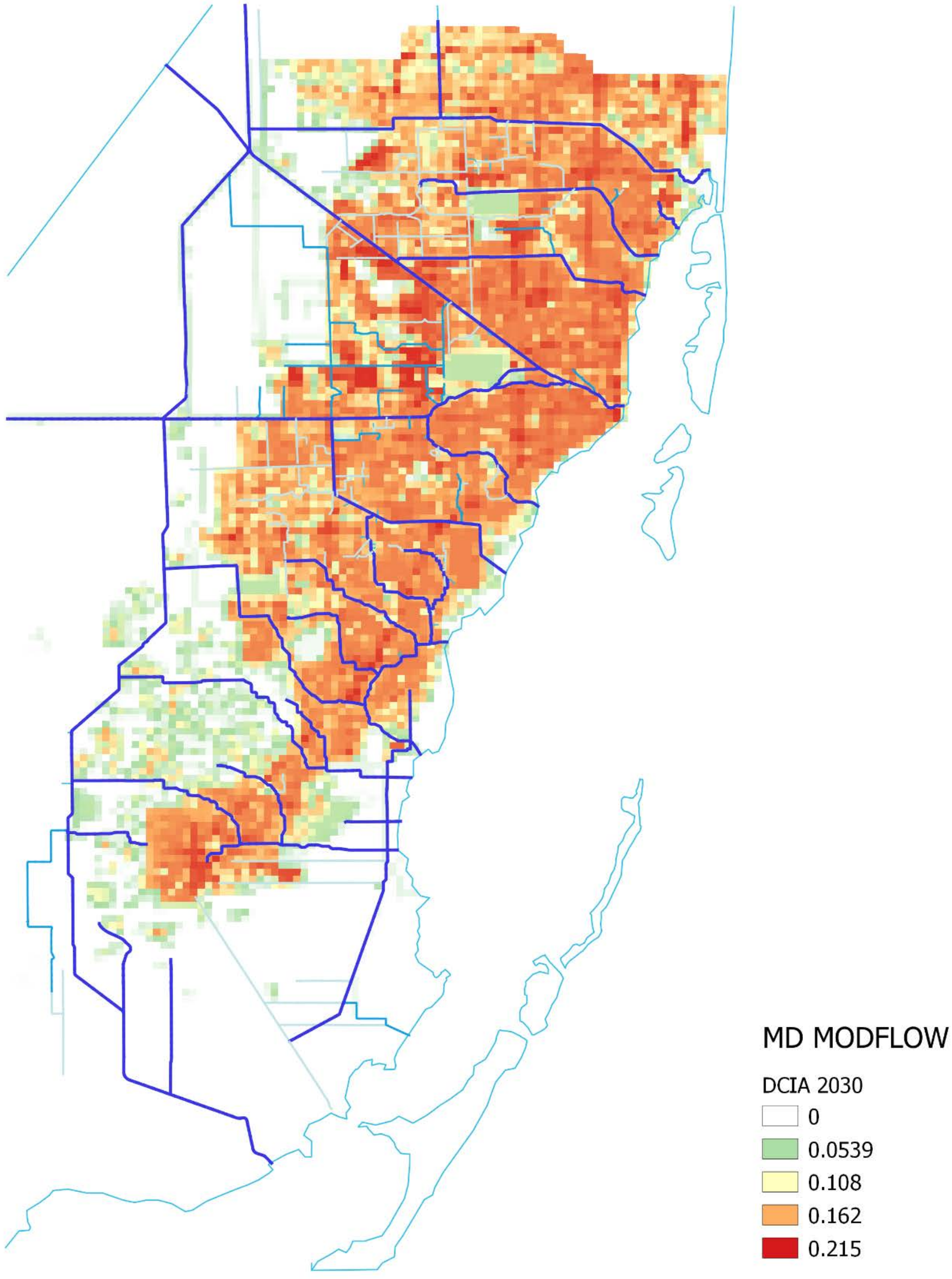


Figure 5. DCIA fractions for model grid cells in 2030.

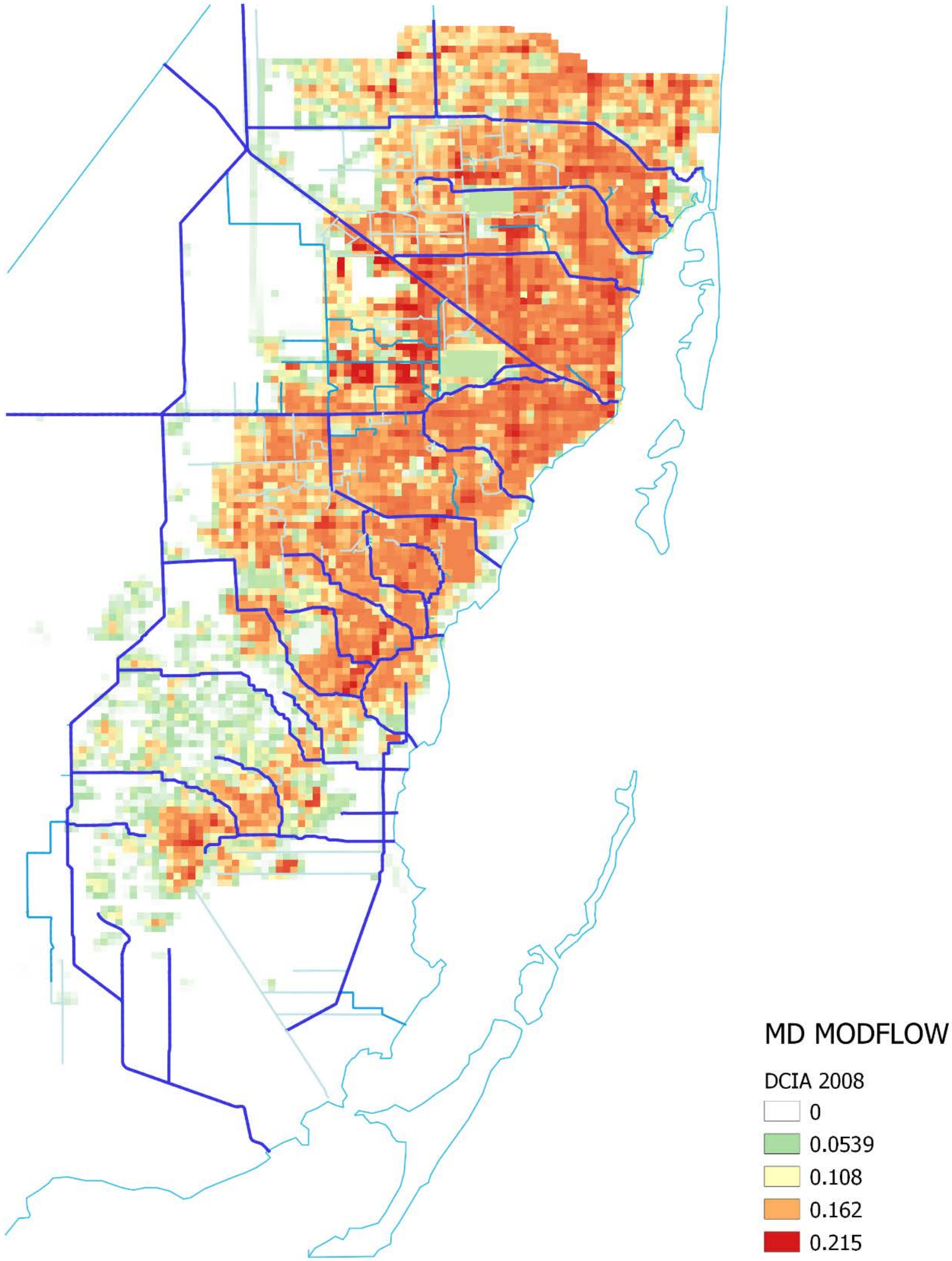


Figure 6. DCIA fractions for model grid cells in 2008.

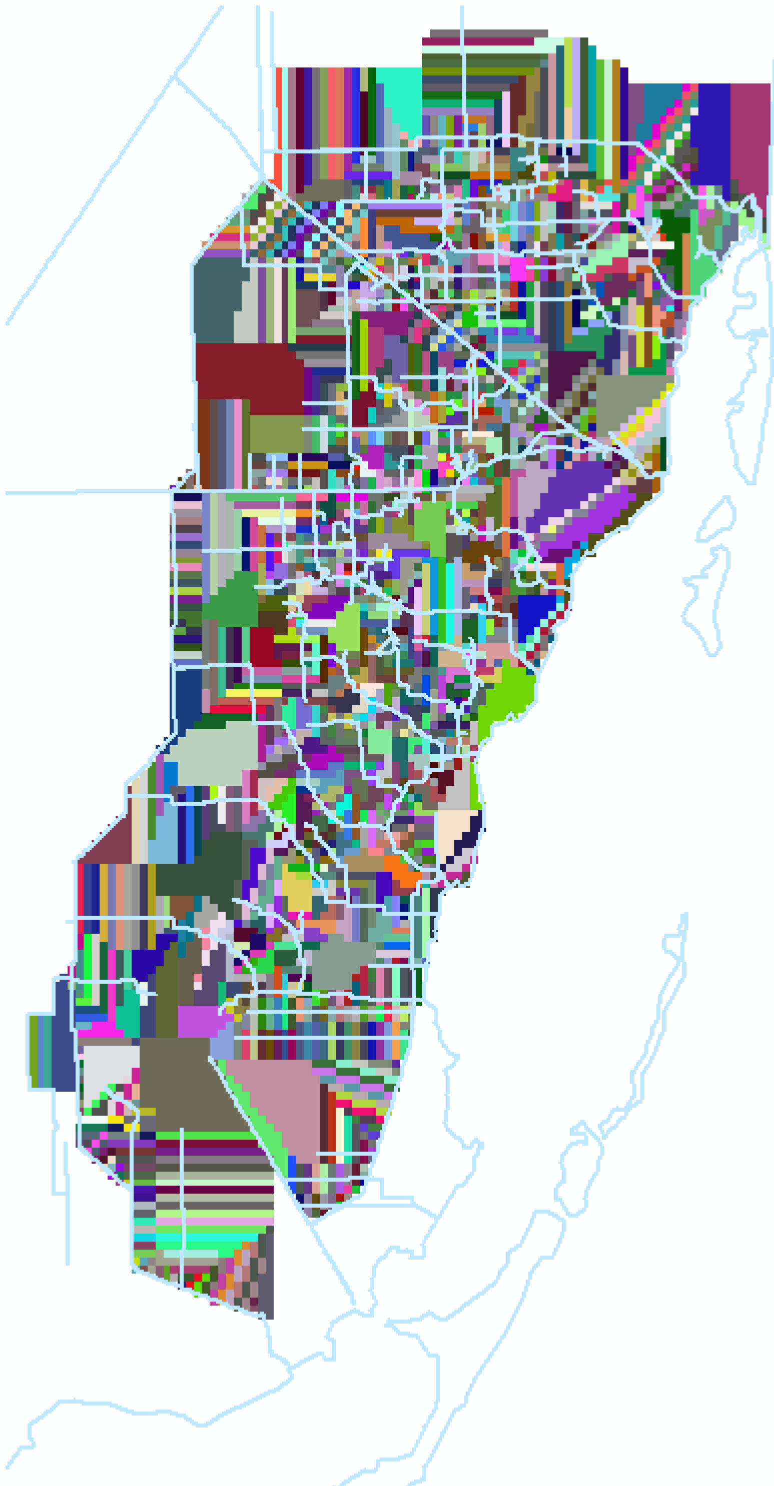


*Figure 7. Cell-->SWR1 mapping. Cells with the same color have their DCIA * Rainfall routed to the same SWR1 reach.*

Future groundwater properties

Due to the existence of additional quarry lakes in 2030 land use (based on 2018 permitted quarry lake coverage) compared to the 2008 land use (which assumed 1999 quarry lake coverage), the groundwater properties at quarry lake cells have to be modified for the future scenario model. Based on the Miami-Dade MODFLOW model documentation, it appears that the groundwater properties such as specific yield (Sy), specific storage (Ss), and hydraulic conductivity (Kh=Kv) for model layers 1-3 were first calibrated using PEST with pilot points and subsequently modified to reflect the presence of quarry lakes as of the year 1999. The following equation was likely used in estimating the effective Sy and Ss for model grid cells with some portion occupied by quarry lakes:

$$S_{y_{eff}} = \frac{(A_{cell} - A_{lake}) * S_{y_{cell}} + A_{lake} * 1}{A_{cell}}$$

Equation 1

Where$S_{y_{eff}}$ is the effective specific yield for the cell, $S_{y_{cell}}$ is the calibrated cell specific yield, 1 is the assumed quarry lake specific yield, A_{cell} is the total cell area, and A_{lake} is the surface area occupied by quarry lakes. By defining $Lakefrac = {}^{A_{lake}}/_{A_{cell}}$, the fraction of the cell occupied by quarry lakes, the above equation simplifies to the following:

$$S_{y_{eff}} = S_{y_{cell}} * (1 - Lakefrac) + Lakefrac$$

Equation 2

The variable Lakefrac (based on 1999 quarry lake coverage) is given in file umd_frclake.ref and is shown in Figure 8.

Solving the above equation for $S_{y_{cell}}$, one can back-calculate the original calibrated specific yield for the cell for cells with Lakefrac smaller than 1:

$$S_{ycell} = \frac{S_{yeff} - Lakefrac}{(1 - Lakefrac)}$$

Equation 3

The resulting 1999 $S_{y_{cell}}$ map showed the bullseye pattern typical of pilot points. In areas with quarry lakes, the pattern seems consistent with nearby cells (Figure 10) and among the different model layers. Therefore, the above methodology seems reasonable in estimating S_{ycell}.

A total of 28 cells had Lakefrac equal to 1. Therefore, $S_{y_{cell}}$ was undefined for these 28 cells (white cells in Figure 10). However, 24 of these 28 cells also have a 2018 permitted Lakefrac of 1.0 (Figure 9); therefore, the 1999 $S_{y_{eff}}$ can still be used for these 24 cells. For cells with a 2018 permitted Lakefrac of 1, $S_{y_{cell}}$ is set to 1 for 2018. For cells with a 2018 permitted Lakefrac between 0 and 1 (exclusive), the 2018 $S_{y_{eff}}$ was computed based on Equation 2 using the 2018 Lakefrac.

For the 4 cells with a 1999 Lakefrac value of 1 (shown with '+' marks in Figure 10), and a 2018 Lakefrac value different from 1, $S_{y_{cell}}$ was averaged for surrounding cells and used in Equation 2 to obtain the 2018 $S_{y_{eff}}$. Inspection of recent aerial imagery shows no mining lakes present in two of these cells (R17C43, R17C44 on Figure 11). The two remaining cells (R34C40, R34C41) had 2018 Lakefrac of 0.92 and 0.96, respectively. These are different from 1 due to accounting for berms and rock washing facilities as separate from deep mining areas.

The above Equation 1-Equation 3 were used for Sy and Ss for all layers. A similar methodology was used for the hydraulic conductivity with the following modifications (in units of ft/d), although it is unclear whether this type of equation was used by the original model developers. Alternatively, an equation based on assuming strata (cell/lake) being perpendicular or parallel to predominant groundwater flow direction may have been used by the original model developers. This approach would require detailed layer thickness/length information and due to its complexity was likely not used.

$$K_{eff} = \frac{(A_{cell} - A_{lake})K_{cell} * + A_{lake} * \max(3.3 * 10^5, K_{cell})}{A_{cell}}$$

Equation 4

Equations equivalent to Equation 2-Equation 3 were derived and applied to estimate the 2030 hydraulic conductivity starting from Equation 4. As discussed in the MODFLOW model documentation, the minimum hydraulic conductivity for lakes was set as 3.3E5 ft/d. Therefore, this minimum limit remained in the derivation. Missing values for the 4 cells identified above were filled in a similar manner but using Equation 4. The resulting 1999 K_{cell} map shows consistency with nearby cells in areas with quarry lakes (Figure 12).

Please note that the Miami-Dade MODFLOW model uses metric units, but the documentation and the figures presented herein are in English units.

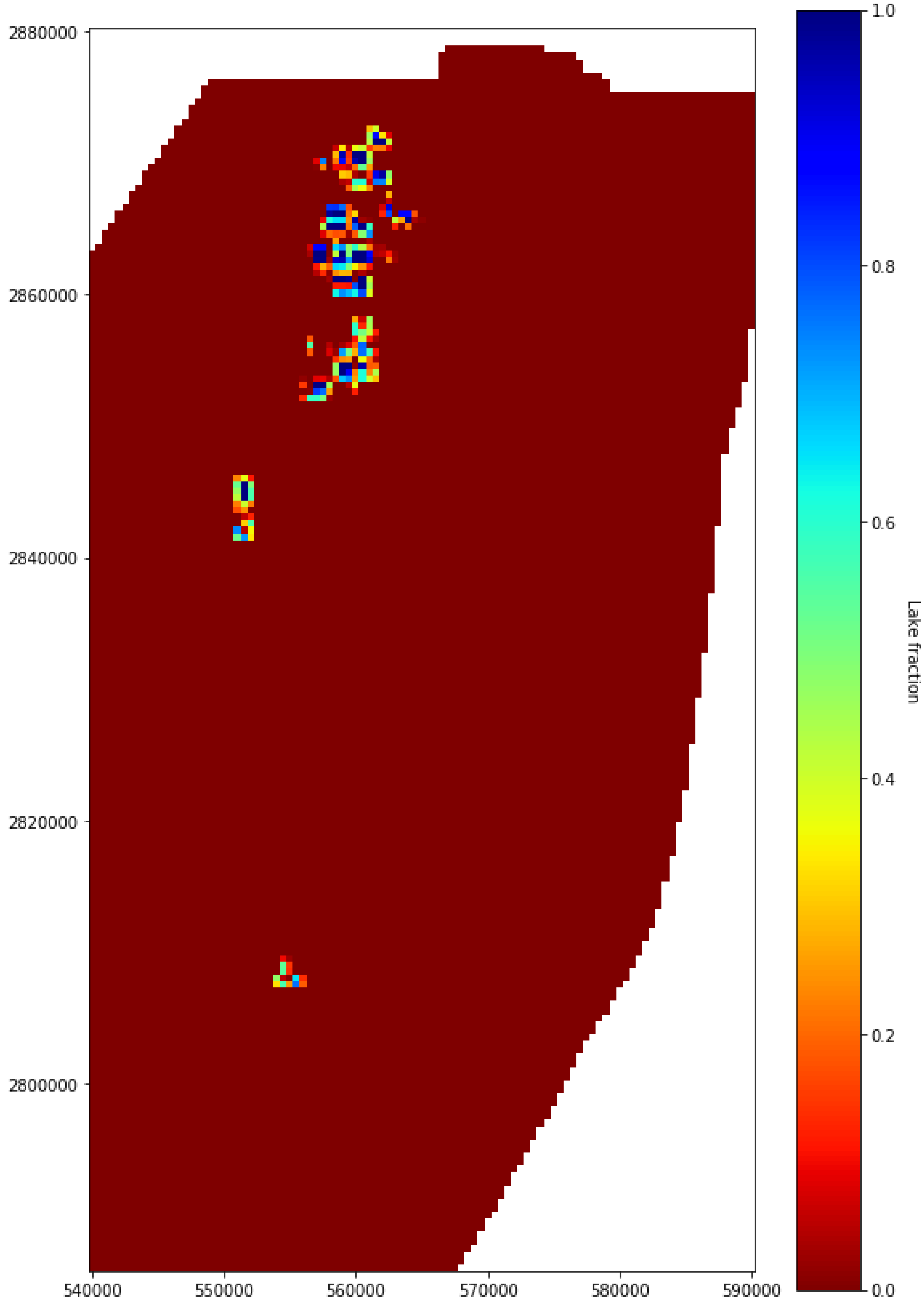


Figure 8. 1999 Lake fraction

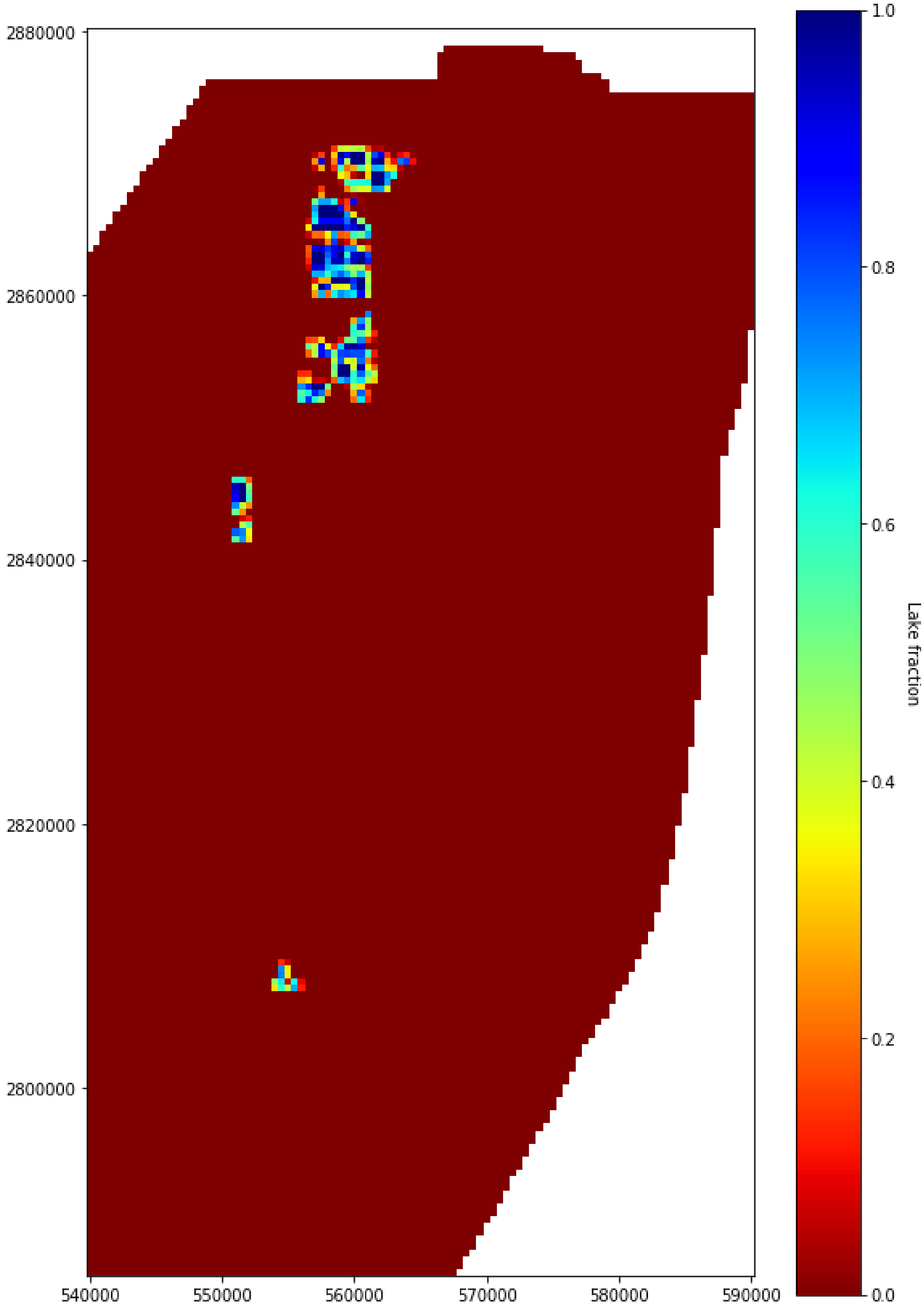


Figure 9. 2030 Lake fraction

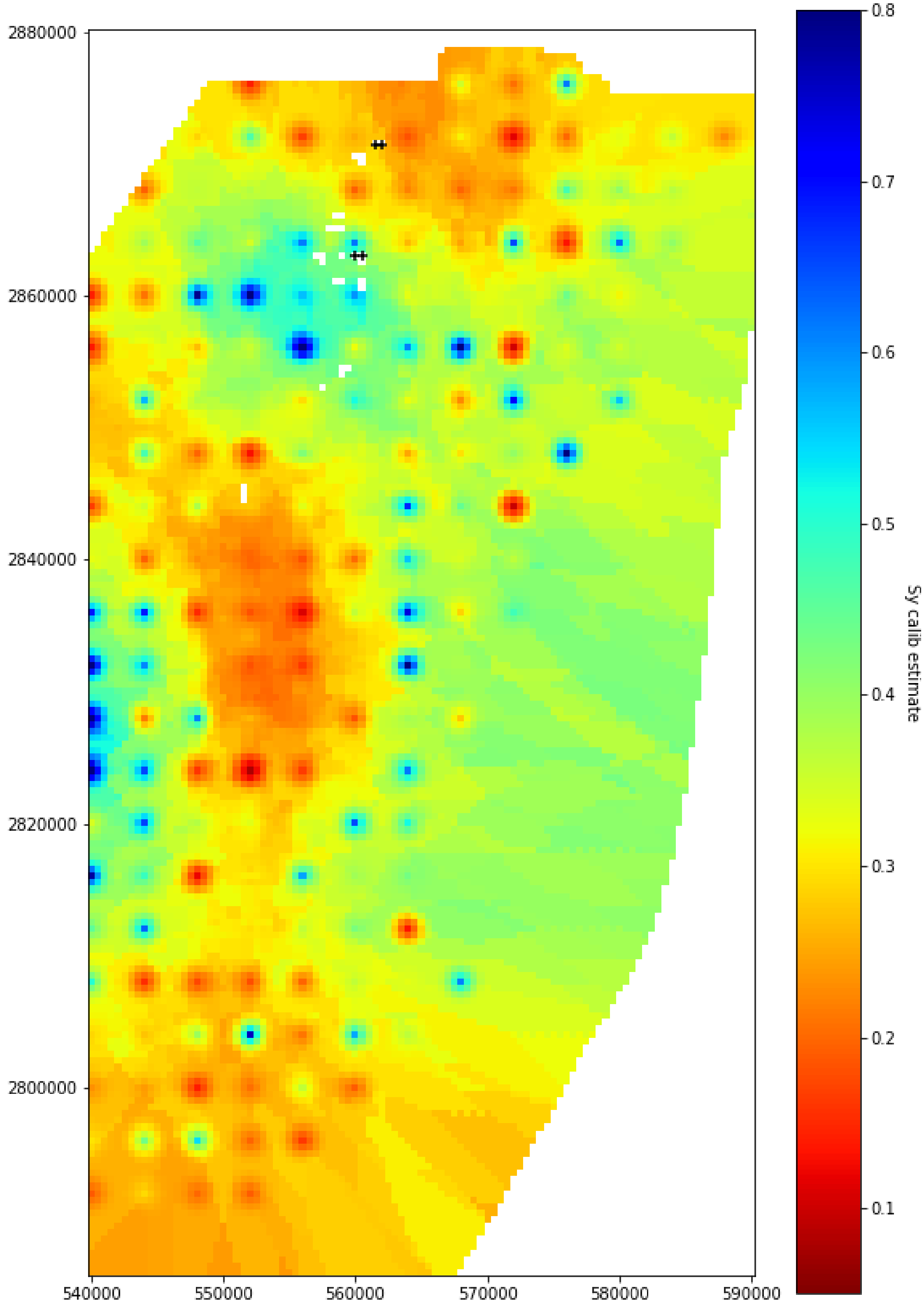


Figure 10. Back-calculated calibrated specific yield Sy_cell.

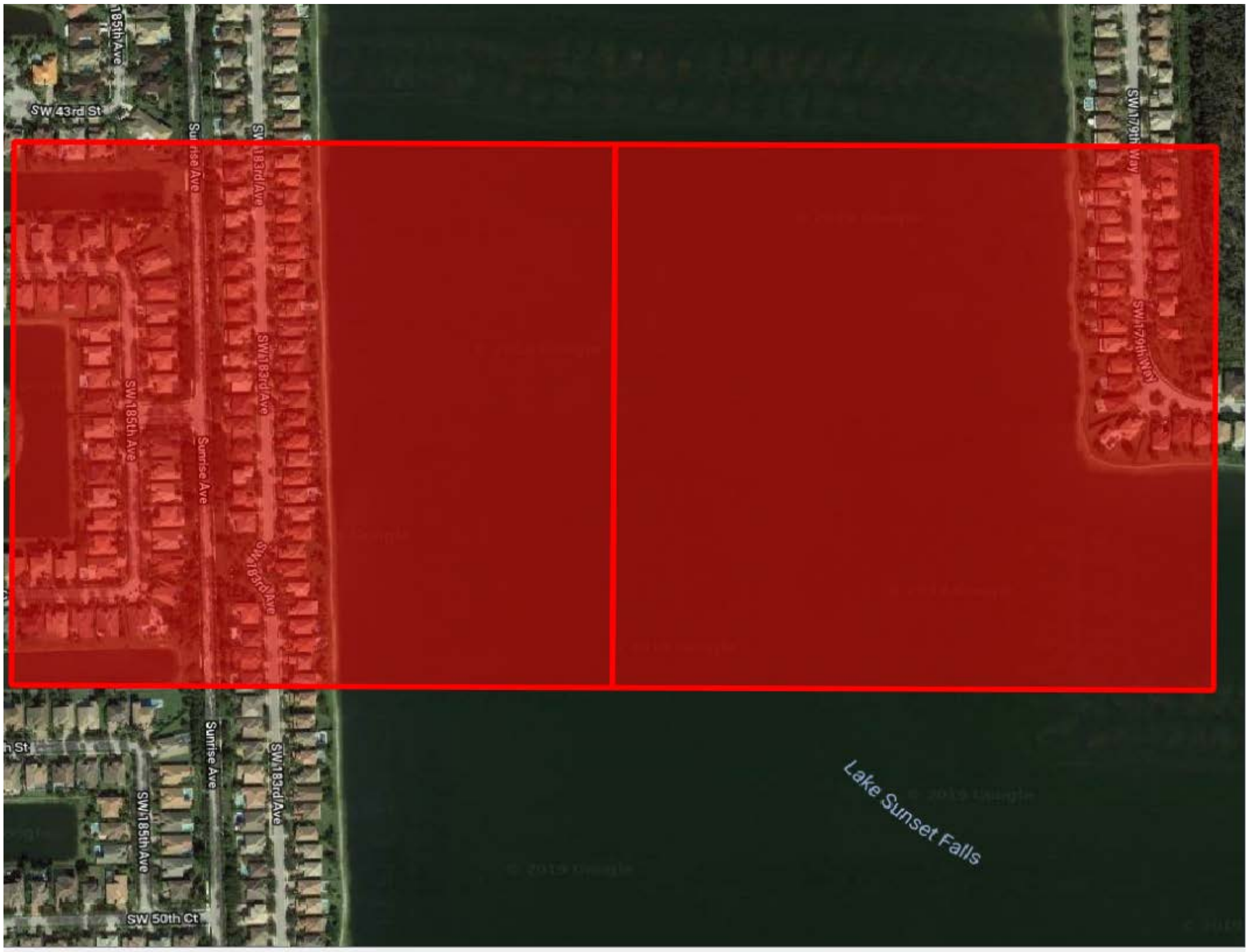


Figure 11. Model grid cells R17C43 and R17C44 (row from north, column from west) are occupied by an urban lake and an urban development. This was confirmed by inspection of mine boundary maps at the FLDEP Mining and Mitigation Program website (https://ca.dep.state.fl.us/mapdirect/?focus=bmr).

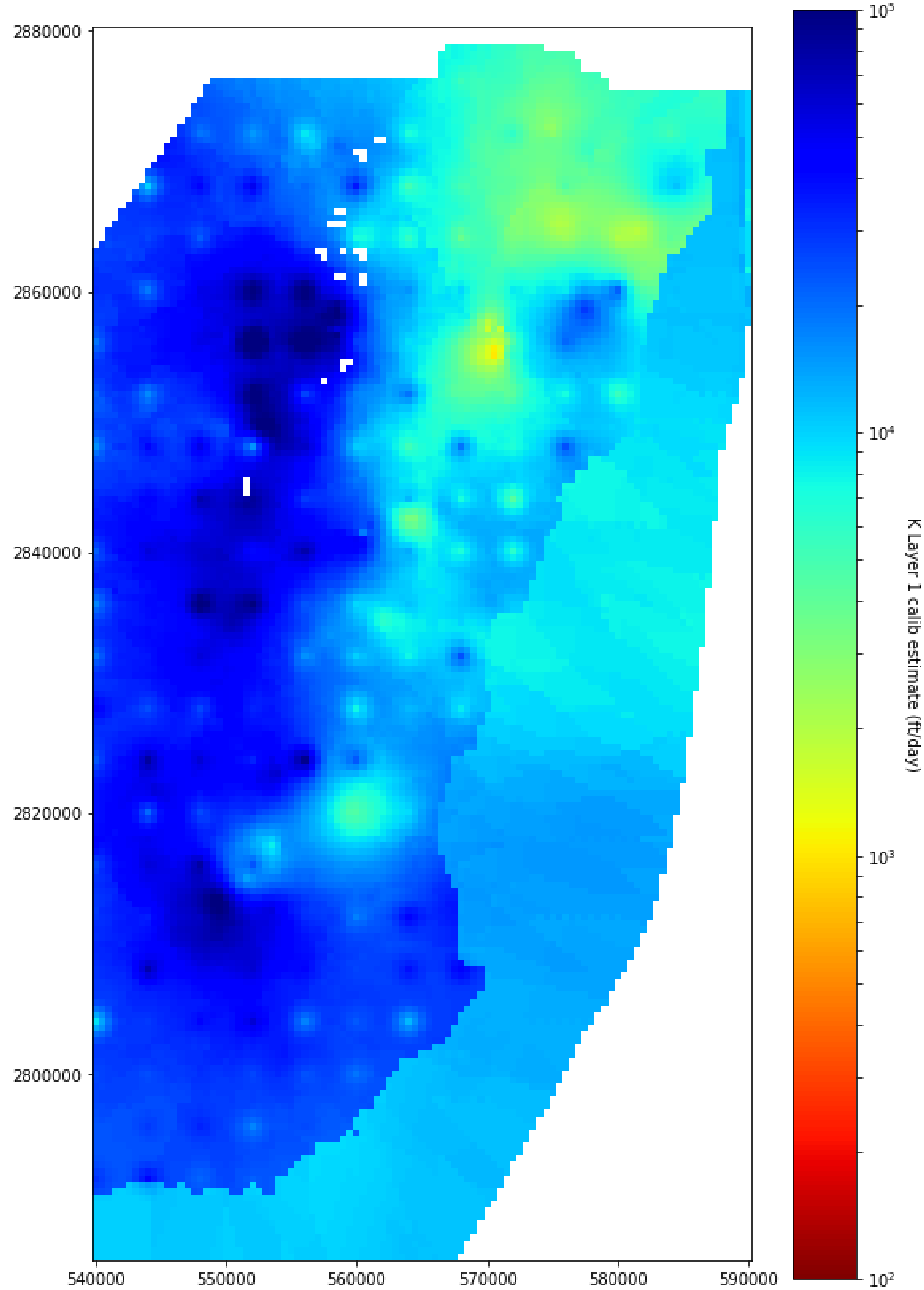


Figure 12. Back-calculated calibrated hydraulic conductivity for layer 1, K_cell.

Future ocean boundary condition

The Miami-Dade MODFLOW model uses daily historical water level data at Virginia Key (ft NAVD88) for the period 1996-2010 as the ocean boundary condition. The historical water level data includes both tidal forcing and sub-tidal forcing (i.e. meteorological effects, current effects, etc.), as well as land subsidence effects which are considered minimal in south Florida. We performed an initial investigation to assess the relative importance of wind-, ocean current-, and rain-driven departures of ocean water levels from the predicted tides on model results. That is, we wanted to know if using the predicted Virginia Key tide as a key model input would be adequate to simulate the historical response of the groundwater system. If so, then simpler predicted-tide-only projections of the future ocean water levels would be adequate for groundwater simulations of the future.

To asses this, we downloaded two sets of Virginia Key tide station data from the NOAA website: the observed water levels and the predicted water levels. The observed data were downloaded as hourly levels in feet (NAVD 88) and Local Standard Time (LST) zone from 1/1/1996 through 12/31/2018. We found that data were missing from 10/3/1997 through 11/8/1997 and from 2/13/2016 through 2/25/2016. We used predicted tidal data to fill these data gaps. The data are plotted along with a fitted linear trend in Figure 13.

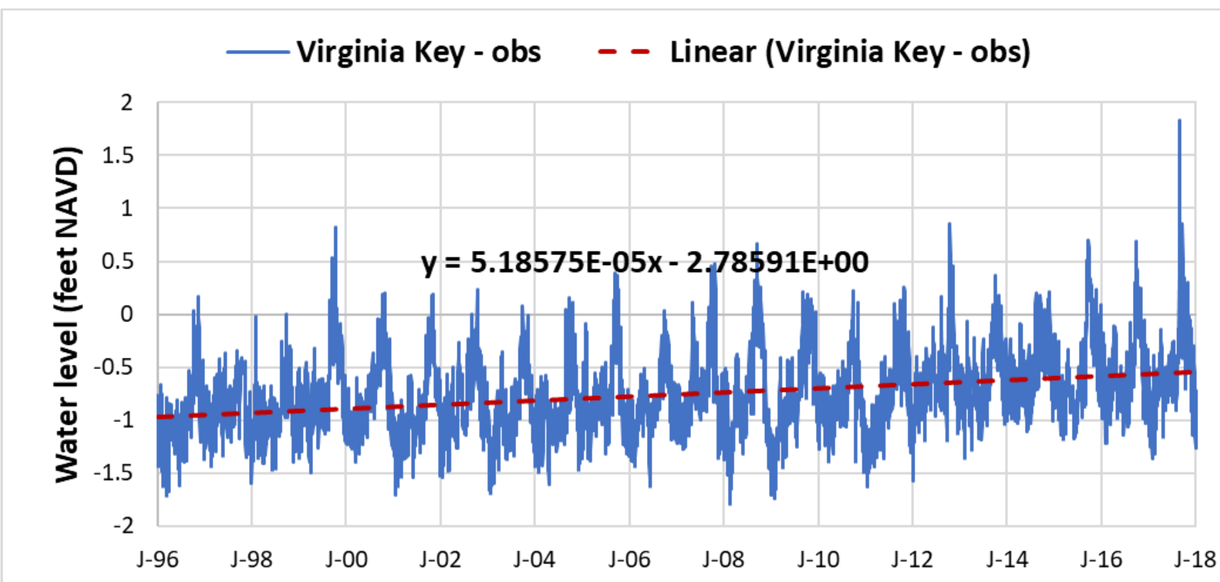


Figure 13. Daily observed data from Virginia Key tide station represented by blue line and linear trend represented by red line.

The linear trend is

Water level (feet NAVD) = 5.18575 * 10^{-5} feet/d * *x* + 2.78591 feet, where x is the number of days since 1/1/1900. This trend corresponds to a rate of 5.8 mm/year between January 1996 and December 2017.

The predicted tide water levels were also downloaded as hourly water levels in feet (NAVD 88) and LST time zone from 12/31/1995 through 1/1/2026. These predictions do not have a significant sea level rise trend (Figure 14, dashed black line). To include the sea level rise in the prediction, we added the sea level rise trend from the observed data. Because the NOAA tidal epoch for the predicted tide data begins in 1992, we began adding the sea level rise on 6/15/1992. The predicted trend-only stage for the 6/15/1992 date was -0.9628 feet, and the stage for 12/31/1995 date, where our chart and simulations begin, was -0.8952 feet. The difference d = 0.06752 feet corresponds to the amount of sea level increase between 6/15/1992 and 12/31/1995. This was added to the predicted tide on 12/31/1995 and to all subsequent dates along with the overall trend. Figure 15 shows the result. As expected, the predicted water level with the added trend has the same trend (dashed green line in Figure 15) as the observed sea level rise trend (dashed brown line in Figure 14).

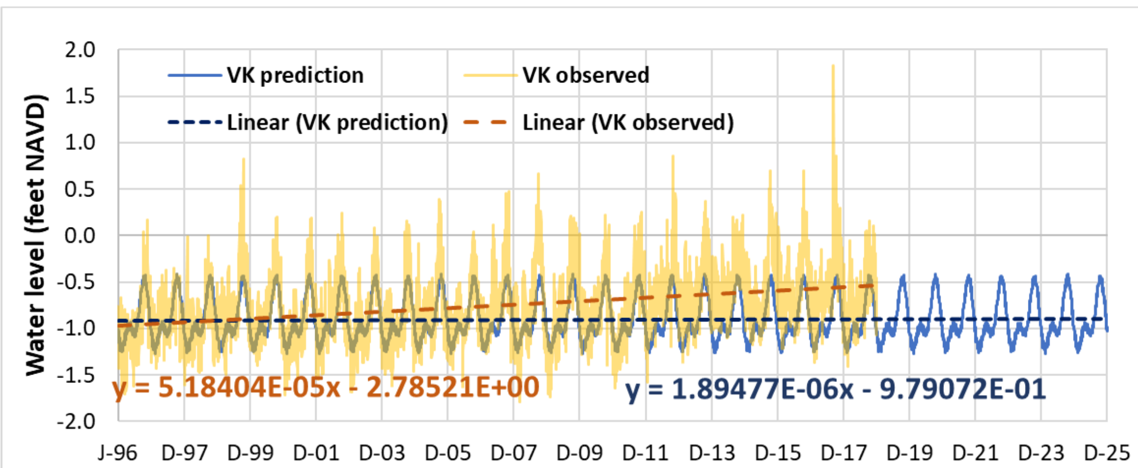


Figure 14. Blue line represents predicted water level at Virginia Key tide station and dashed black line represents its small trend. Green line represents water level with added sea level rise trend and the dotted red line represents its trend.

Figure 15 compares the observed tide data and the predicted tides plus SLR trend. This analysis was followed by an initial sensitivity analysis in order to determine whether the average May-October groundwater table elevations are sensitive to the exclusion of the sub-tidal forcing component. This sensitivity analysis will be discussed in the next section.

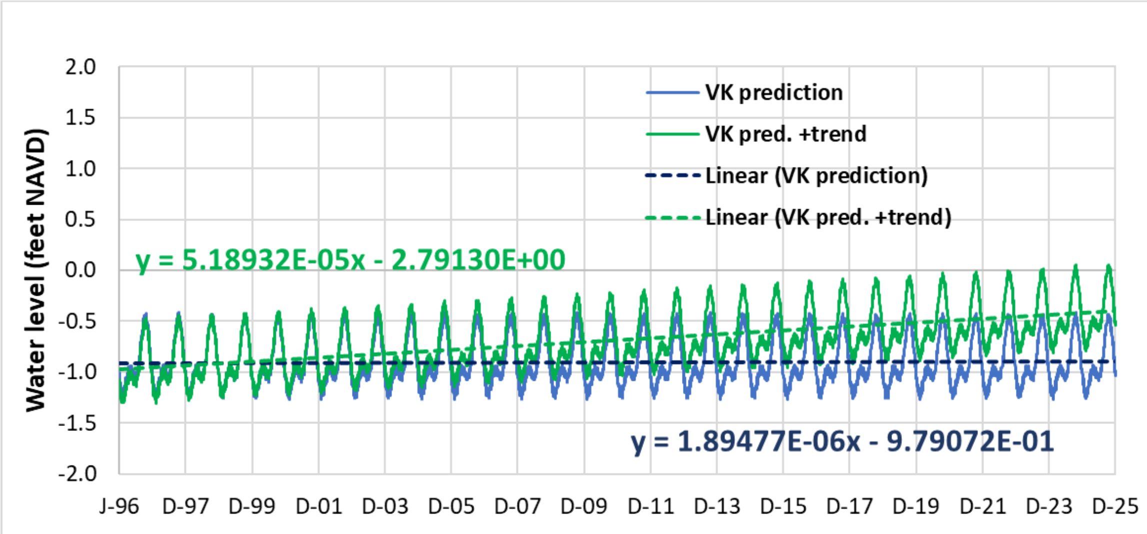


Figure 15. Comparison between observed Virginia Key tide data and predicted tide plus trend.

Sensitivity analysis of ocean boundary condition

For the model runs, we converted Virginia Key tide data to meters by multiplying them by 0.3048 m/ft and created .smp files starting with 1/1/1996 and ending with 12/31/2025. We had three sets of data: observed tide, predicted tide, and predicted tide with added sea level rise trend. Before running the model, we ran the UMD_scenario_BND python script. In the code we adjusted the paths, so it reads and creates outputs in the correct folders. This script created binary data for GHB and DRN model input files. This code also created umd_current.ghb and umd_current.drn files, which direct the model to read the binary files. We also created swr1 files for Virginia Key with the starting date 12/31/1995 and ending date 1/1/2026. In these files, the units are feet NAVD 88 because the model is set up to convert to meters NAVD 88. With these changes, we ran the model for the current sea level scenario. The results of the model runs are shown below.

Figure 16 shows 2017 simulated wet season heads in feet NAVD. The heads are based on observed tide water level input (left), predicted tide without trend input (center), and predicted tide with added trend (right). Overall, the maps show that the heads simulated based on the predicted tides without the added trend (center) are slightly lower than those simulated with the observed tides (left) and with the predicted tides plus the SLR trend (right). Comparing the simulations with the observed tides and with the predicted tides plus the SLR trend (left and right panels of Figure 17), indicates that heads simulated with the predicted tides plus the SLR trend are almost indistinguishable. This is reasonable considering the generally small difference in the 2017 wet season observations and predictions in Figure 16; a key exception is when Hurricane Irma and a weaker Florida Current led to higher observed water levels for a short time. Similarly, Hurricane Joaquin is likely responsible for the observed water level peak in 2015, and Hurricane Nicole – together with a weak Florida Current – likely contributed to the observed water level peak in October of 2016. The impact of these differences between predicted and observed tide water levels is the key issue we seek to address here. On the basis of these sensitivity simulations, it appears that the use of the predicted tides plus an SLR trend in the simulations captures the wet season heads adequately consistent with the original approach of the Urban Miami-Dade model of Hughes and White (2016), which used the observed tides.

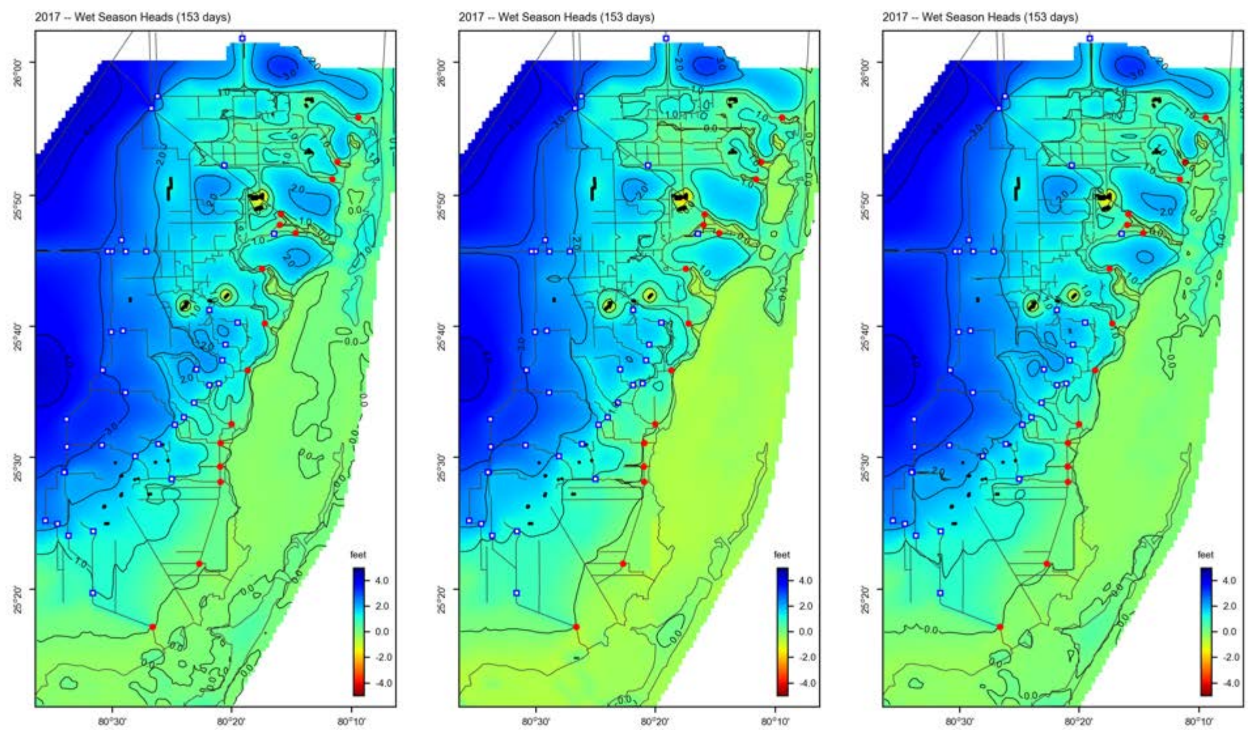


Figure 16. 2017 simulated average wet season heads in feet NAVD. Heads based on observed tide water level input (left), predicted tide without trend input (center), and predicted tide with added trend (right). Red dots are salinity control structures. White boxes with blue outlines are water supply/flood control structures.

Figure 17 shows the same results as Figure 16 displayed in terms of the depth to water. As expected, there are in general subtle increases in the simulated depth to water in case of the predicted tide without trend input (center), and the depths to water based on the observed tide water level input (left) and those based on the predicted tide with added trend (right) are nearly indistinguishable at this scale.

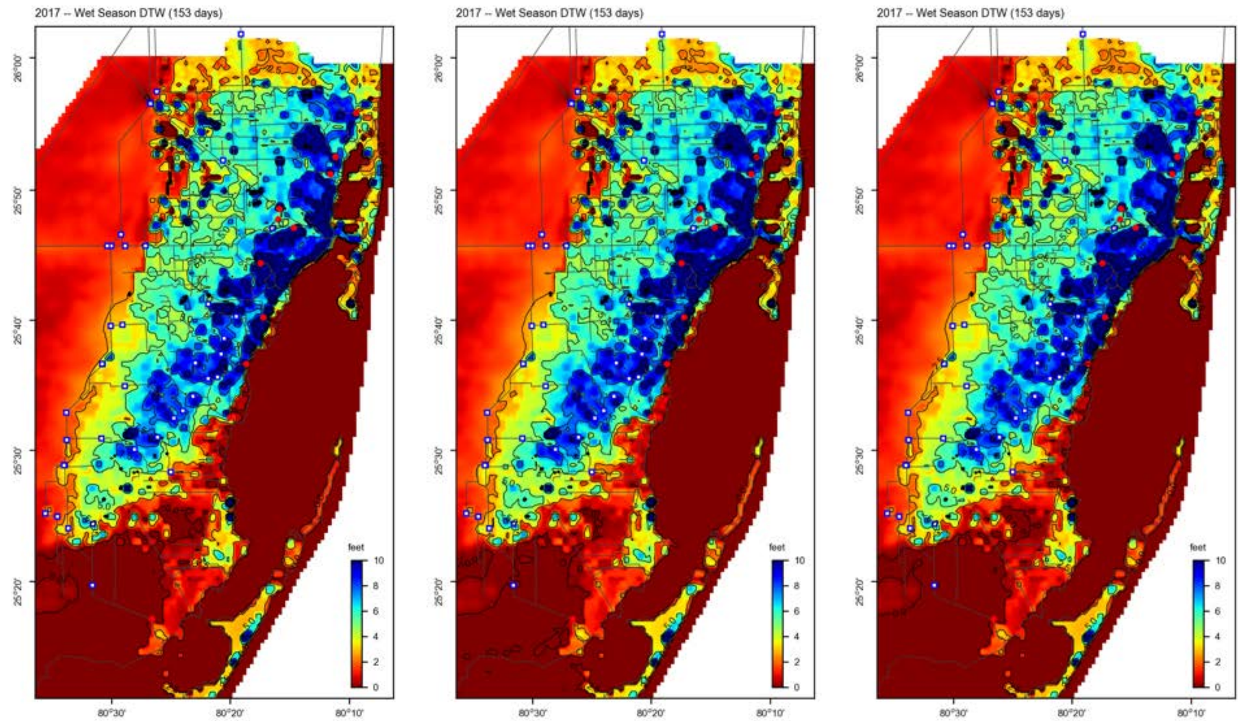


Figure 17. 2017 simulated average wet season depths to water in feet. Depths to water based on observed tide water level input (left), predicted tide without trend input (center), and predicted tide with added trend (right). Red dots are salinity control structures. White boxes with blue outlines are water supply/flood control structures.

Figure 18 gives time series of observed and simulated heads for select wells (same wells as Figure 5-3 of Hughes and White (2016)) from 1996 through 2018 based on observed-tide water level input (green), predicted-tide-without-trend input (red), and predicted-tide-with-added-trend input (blue). Wells G-3549 and G-3550 are closest to the tidal boundary and, as expected, show the greatest differences in the simulated heads as a function of the boundary type. The simulation based on the predicted tide-with-trend input tracks the simulation based on observations more closely. The deviation of the tide-without-trend input from the other simulations grows with time as the effect of the trend grows stronger. These results are consistent with the more generalized observations based on Figure 16.

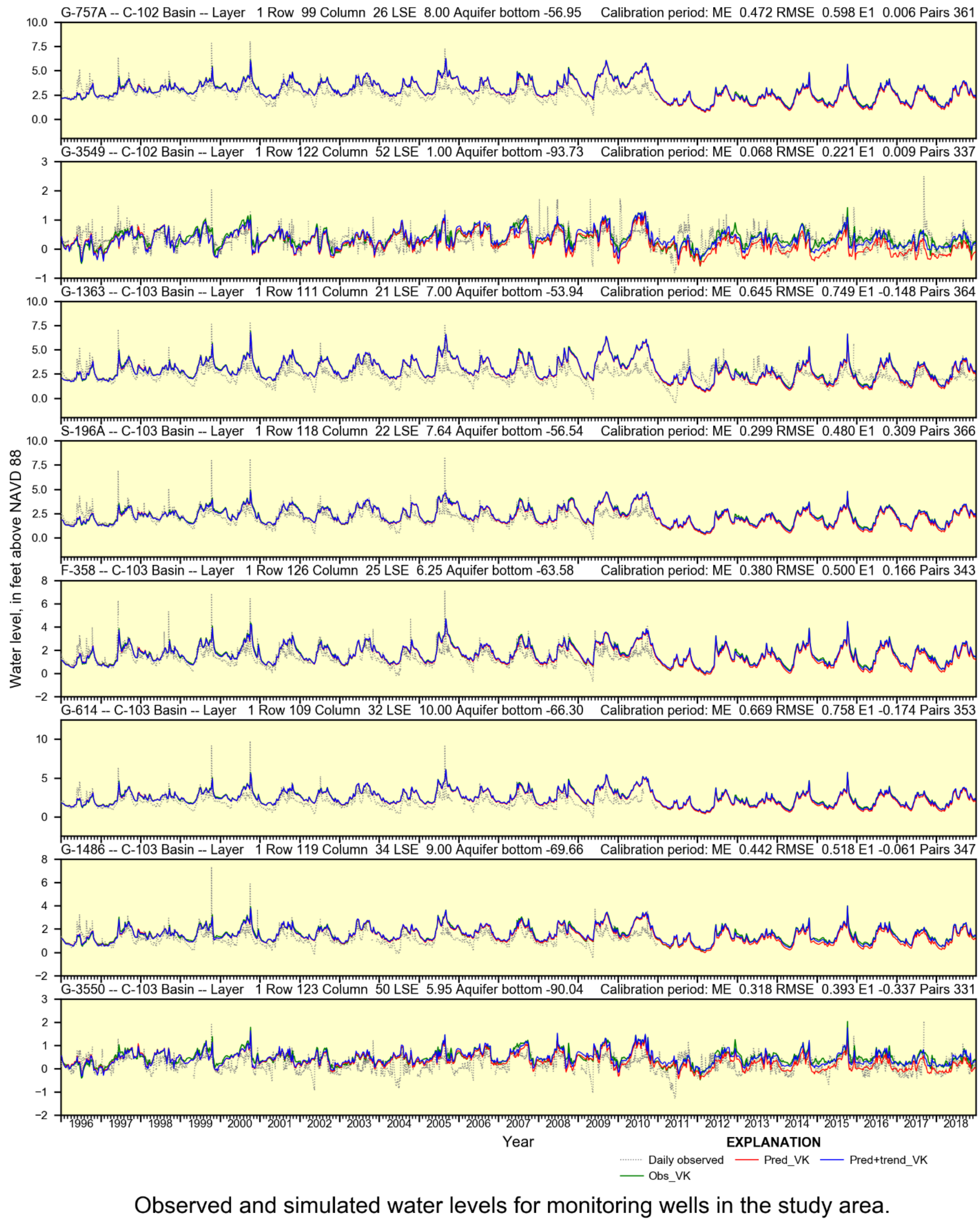


Figure 18. Time series showing observed (gray dotted) and simulated heads for select wells (same wells as Figure 5-3 of Hughes and White (2016)) from 1996 through 2018 based on observed tide water level input (green), predicted tide without trend input (red), and predicted tide with added trend input (blue). Wells G-3549 and G-3550 are closest to the tidal boundary. Remainder of wells show observations through 2010 only.

In conclusion, the simulations conducted for this ocean boundary sensitivity analysis suggest that the use of predicted tides with an added sea level rise trend as key model input is adequate for the purpose of this project. The use of the predicted tides plus an SLR trend in the simulations reproduces the wet season heads in a way that is adequately consistent with the original approach of the Urban Miami-Dade model of Hughes and White (2016), which used the observed tides as input.

Development of future ocean boundary condition timeseries

Future (2055-2069) ocean boundary conditions reflecting sea level rise for modeling were obtained from the Unified Sea Level Rise (SLR) Projections developed by the Southeast Florida Regional Climate Change Compact (2015) for both the IPCC AR5 RCP8.5 Median curve and the USACE High curve. These future conditions reflect the effect of sea level rise on the predicted tides (based on harmonic analysis and fitting) for the two selected SLR scenarios. In other words, sub-tidal forcing is neglected due to uncertainties in predicting this component of the total ocean water levels. The sensitivity analysis discussed in the previous section showed that the choice of excluding sub-tidal forcing from the ocean boundary condition has only minor effects and is adequate for the purpose of this project.

The procedure to derive projected daily projected tides at NOAA primary harmonic station 8723214 (Virginia Key) for 2055-2069 was as follows:

1. Hourly tide predictions for 1965-2016, based on harmonic analysis done by NOAA and a 1983-2001 National Tidal Datum Epoch (NTDE; with a mid-point in 1992) mean sea level (MSL = 0.67 ft NGVD29 = -0.90 ft NAVD88; Figure 22), were obtained from the NOAA website based on meters above the local MLLW datum and GMT time zone. In other words, the 1965-2016 data reflects mean sea level around the year 1992. This work was previously done by the sub-contractor at the South Florida Water Management District (SFWMD). This is reasonably close to the -0.808 ft NAVD88 average of calibration period (1996-2010; centered on ~2003) total water levels mentioned in the Miami-Dade MODFLOW documentation (p. 7). This gives a linear SLR rate of 0.00836 ft/yr (2.5 mm/yr), which is close to the SLR rate assumed in the model documentation on p. 94 (0.0073 ft/yr = 2.2 mm/yr).
2. Scripts were used to convert the hourly tide predictions from 1965-2016 to ft NGVD29. According to VDatum tool, at Virginia Key (lat: 25^{o} 43.9’ N, lon: 80^{o} 9.7’ W), the offset from MLLW datum to NGVD29 datum is 0.430 ft, and the offset from MLLW datum to NAVD88 datum is 1.994 ft. Therefore, Water level in MLLW – offset value (NGVD29 or NAVD88) = water level (in NGVD29 or NAVD88). This work was also previously done at the SFWMD.
3. The 1965-2016 hourly tide predictions in ft NGVD29 and GMT time zone were input into MATLAB/Octave script proj_allstas.m, which calls projecttides_new.m, both are included in Appendix A. MATLAB/Octave code for future tidal prediction. These programs do the following:
    a. First the tidal analysis and prediction code, UTIDE (Codiga, 2011), is run to fit harmonics to the hourly tide predictions at Virginia Key.

   b. Then the harmonics are used by UTIDE to predict the hourly tide for the period 2055-2069 (Figure 19) based on the 1983-2001 NTDE (i.e. based on a 1992 MSL). Predicting the hourly tides for 1965-2016 result in almost the same timeseries as the input timeseries, confirming that the harmonic fit is adequate for future prediction. Note that UTIDE is run with lunar nodal corrections which affect daily tidal range and diurnal inequalities. This is important for accurate future predictions. However, the effect of the lunar nodal cycle (LNC) on mean sea level (sinusoidal of 18.61-year period) is neglected.
   c. Finally, the hourly tide predictions for 2055-2069 based on 1992 MSL are shifted along a selected sea level rise curve. The SLR curve can be user-defined or based on one of the 4 curves defined in the Unified Sea Level Rise Projections by the Southeast Florida Regional Climate Compact (2015), which are based on sea level increase from a 1992 MSL and hardcoded in the program. For the Florida Building Commission Miami-Dade modeling effort, we selected two SLR curves: the IPCC AR5 RCP8.5 Median SLR curve (with linear coefficient (a) = 1.7 mm/yr, and quadratic acceleration coefficient (b) = 0.047 mm/yr^2; Figure 20), and the USACE High SLR curve (a = 1.7 mm/yr, b = 0.113 mm/yr^2; Figure 21). The final hourly tide predictions output by the code are in ft NGVD29, EST time zone and represent the sea level rise expected to occur from 2055-2069 under the two SLR scenarios.
4. The 2055-2069 hourly tide predictions in ft NGVD29 and EST time zone were imported into EXCEL and converted to daily averages using pivot tables. Finally, the datum was converted from ft NGVD29 to ft NAVD88 by subtracting 1.57 ft from elevations in ft NGVD29. The daily timeseries were saved as: Virg_Key_daily_tide_IPCC_AR5_Med_ftNAVD88.csv and Virg_Key_daily_tide_USACE_High_ftNAVD88.csv.
   a. The timeseries based on IPCC AR5 RCP8.5 Median SLR curve reflects MSL of 1.63 ft NGVD29 (0.06 ft NAVD88; Figure 23), for a SLR of 0.96 ft from 1/1/1992 to 1/1/2055. The MSL on 12/31/2069 is 2.04 ft NGVD29 (0.47 ft NAVD88; Figure 24), for a SLR of 1.37 ft from 1/1/1992 to 12/31/2069.
   b. The timeseries based on USACE High SLR curve reflects MSL of 2.49 ft NGVD29 (0.92 ft NAVD88; Figure 25), for a SLR of 1.82 ft from 1/1/1992 to 1/1/2055. The MSL on 12/31/2069 is 3.36 ft NGVD29 (1.79 ft NAVD88; Figure 26), for a SLR of 2.69 ft from 1/1/1992 to 12/31/2069.

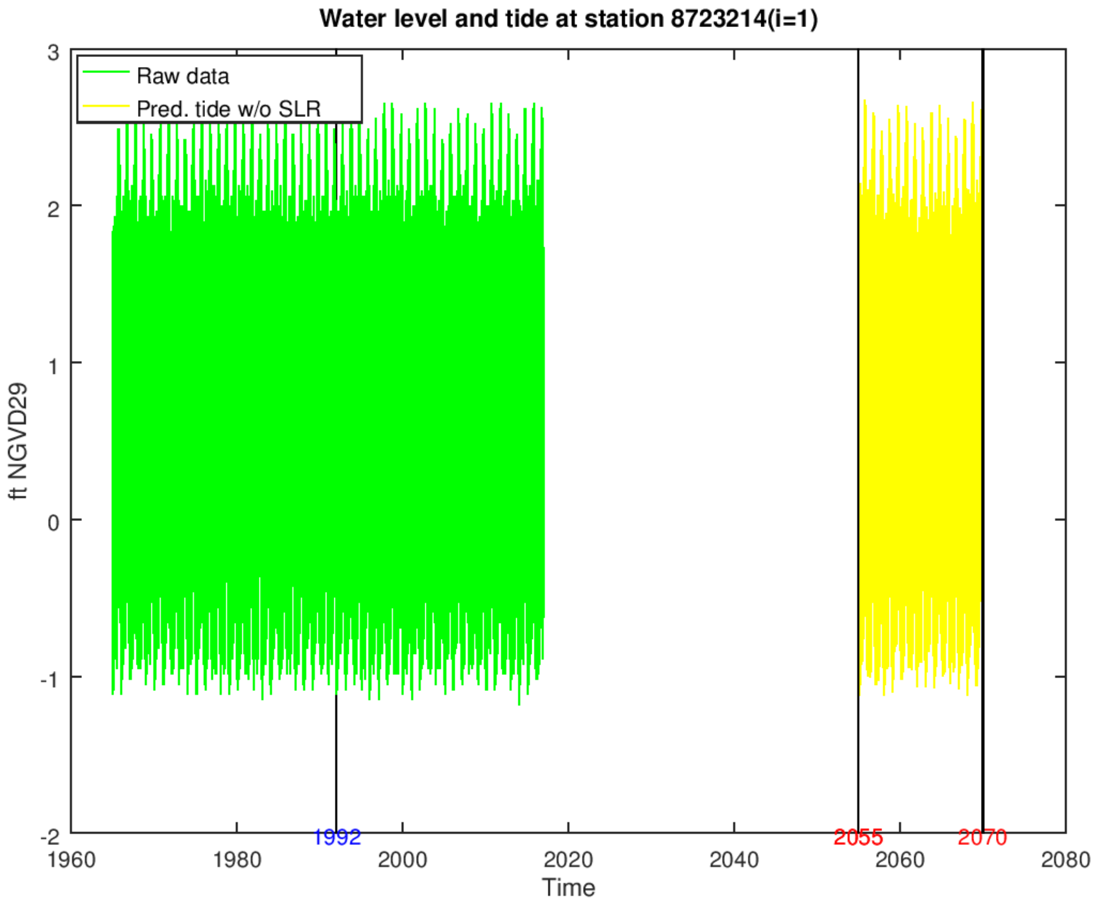


Figure 19. Raw hourly tidal predictions at Virginia Key for 1965-2016, and predicted tides for 2055-2069. Both are in ft NGVD29 and based on 1983-2001 NTDE (without SLR).

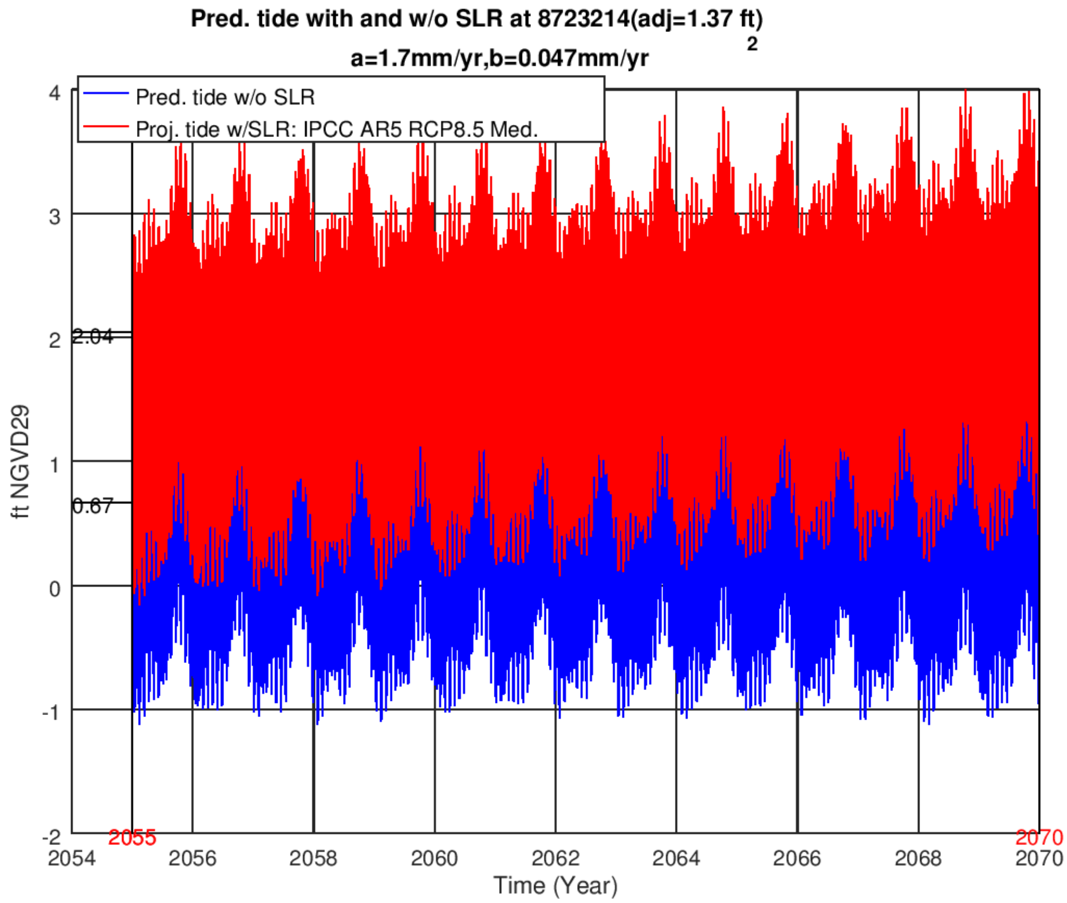


Figure 20. Hourly predicted tides (ft NGVD29) at Virginia Key for 2055-2069 without SLR (blue trace) and with IPCC AR5 RCP8.5 Medium SLR curve (red trace). Note: The amplitude of the blue trace is similar to that of the red trace, the data is just hidden behind the red. The 1983-2001 NTDE MSL at Virginia Key is 0.67 ft NGVD29, while the MSL at the end of 2069 is 2.04 ft NGVD29 after considering SLR. Daily averages were computed and the datum converted to ft NAVD88 prior to input into the model.

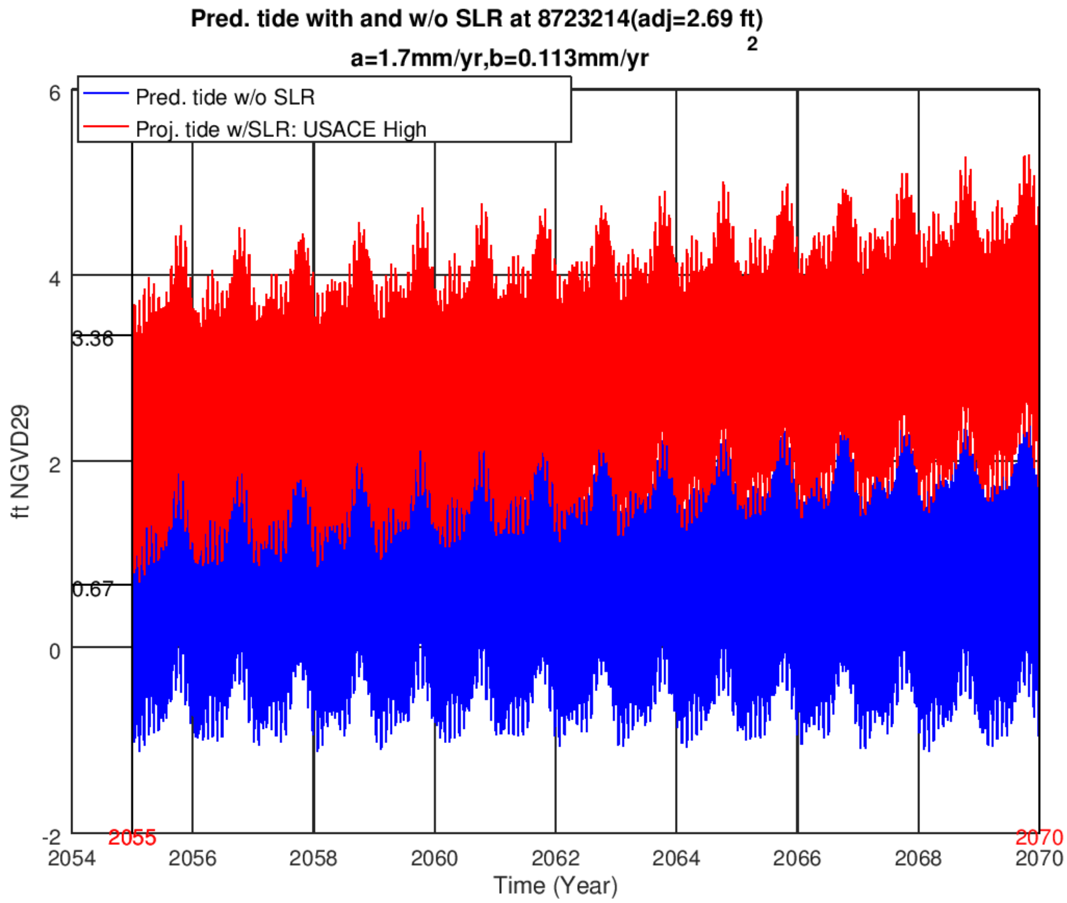


Figure 21. Hourly predicted tides (ft NGVD29) at Virginia Key for 2055-2069 without SLR (blue trace) and with USACE High SLR curve (red trace). Note: The amplitude of the blue trace is similar to that of the red trace, the data is just hidden behind the red. The 1983-2001 NTDE MSL at Virginia Key is 0.67 ft NGVD29, while the MSL at the end of 2069 is 3.36 ft NGVD29 after considering SLR. Daily averages were computed and the datum converted to ft NAVD88 prior to input into the model.

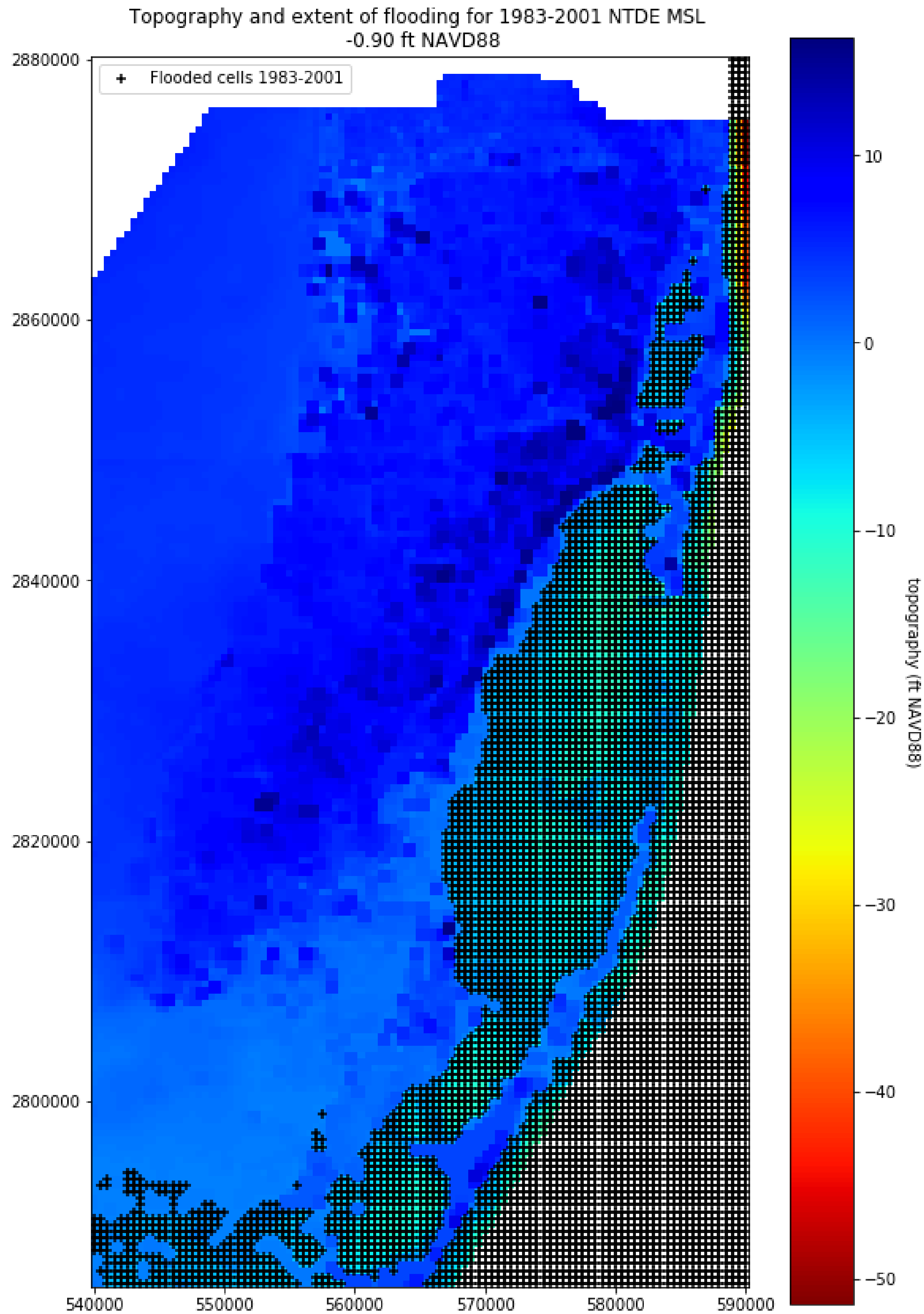


Figure 22. Topography and extent of flooding (cells with '+' black markers) based on 1983-2001 NTDE mean sea level at Virginia Key (-0.90 ft NAVD88). X and Y coordinates in meters, UTM17N, NAD83.

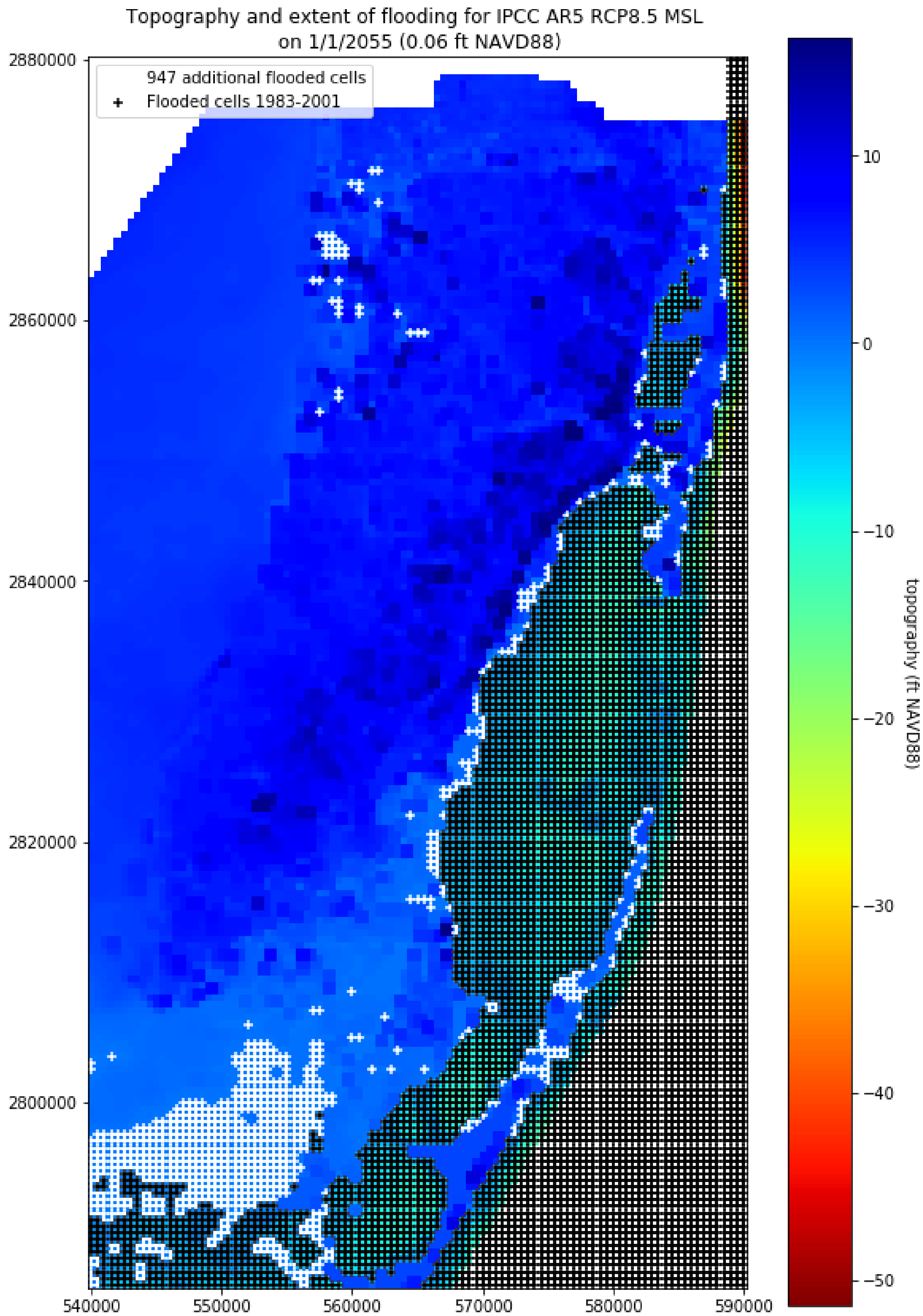


Figure 23. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 1/1/2055 at Virginia Key for the IPCC AR5 RCP8.5 scenario (0.06 ft NAVD88). A total of 947 additional model grid cells would be flooded compared to the cells flooded based on 1983-2001 NTDE MSL. X and Y coordinates in meters, UTM17N, NAD83.

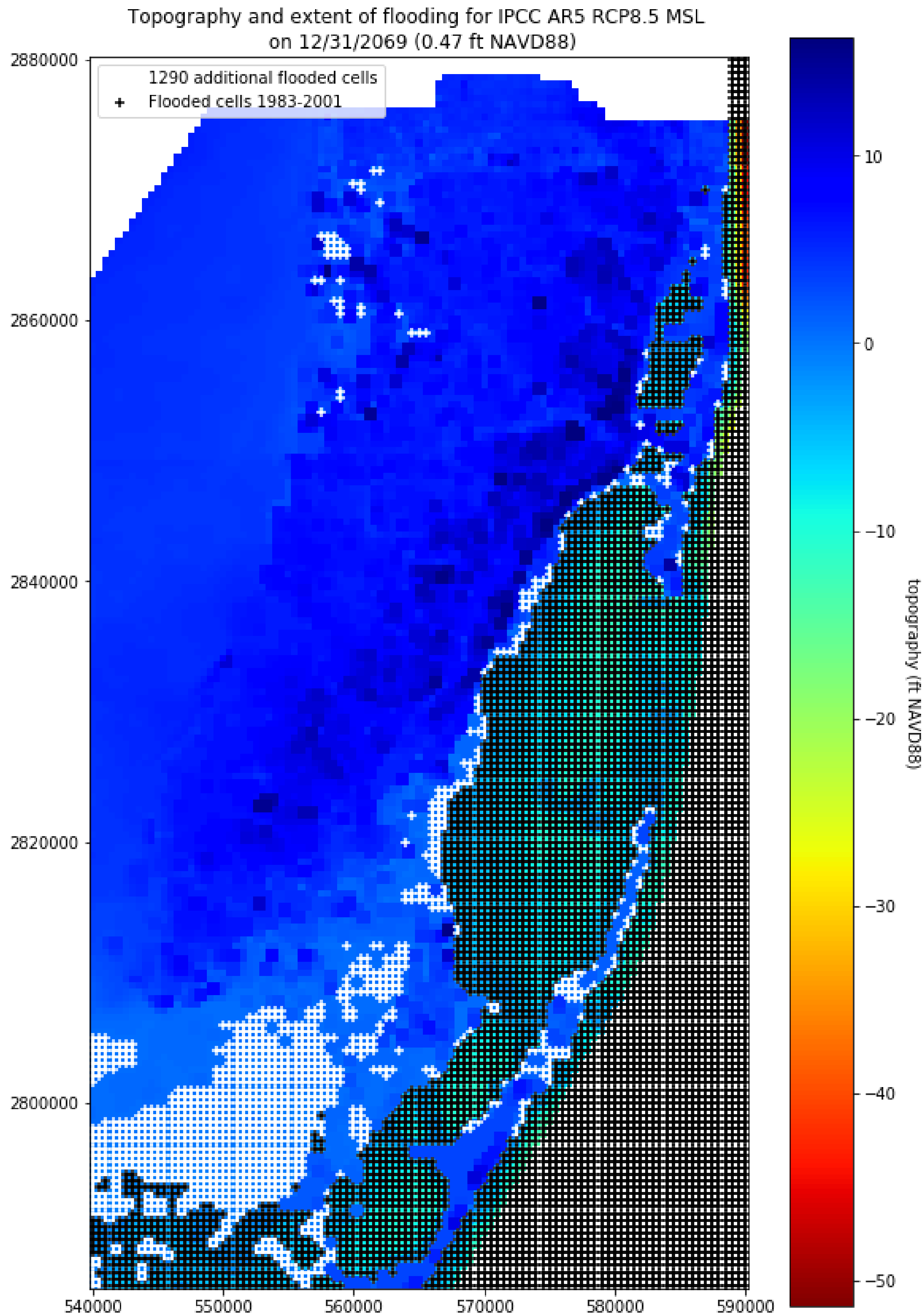


Figure 24. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 12/31/2069 at Virginia Key for the IPCC AR5 RCP8.5 scenario (0.47 ft NAVD88). A total of 1290 additional model grid cells would be flooded compared to the cells flooded based on 1983-2001 NTDE MSL. X and Y coordinates in meters, UTM17N, NAD83.

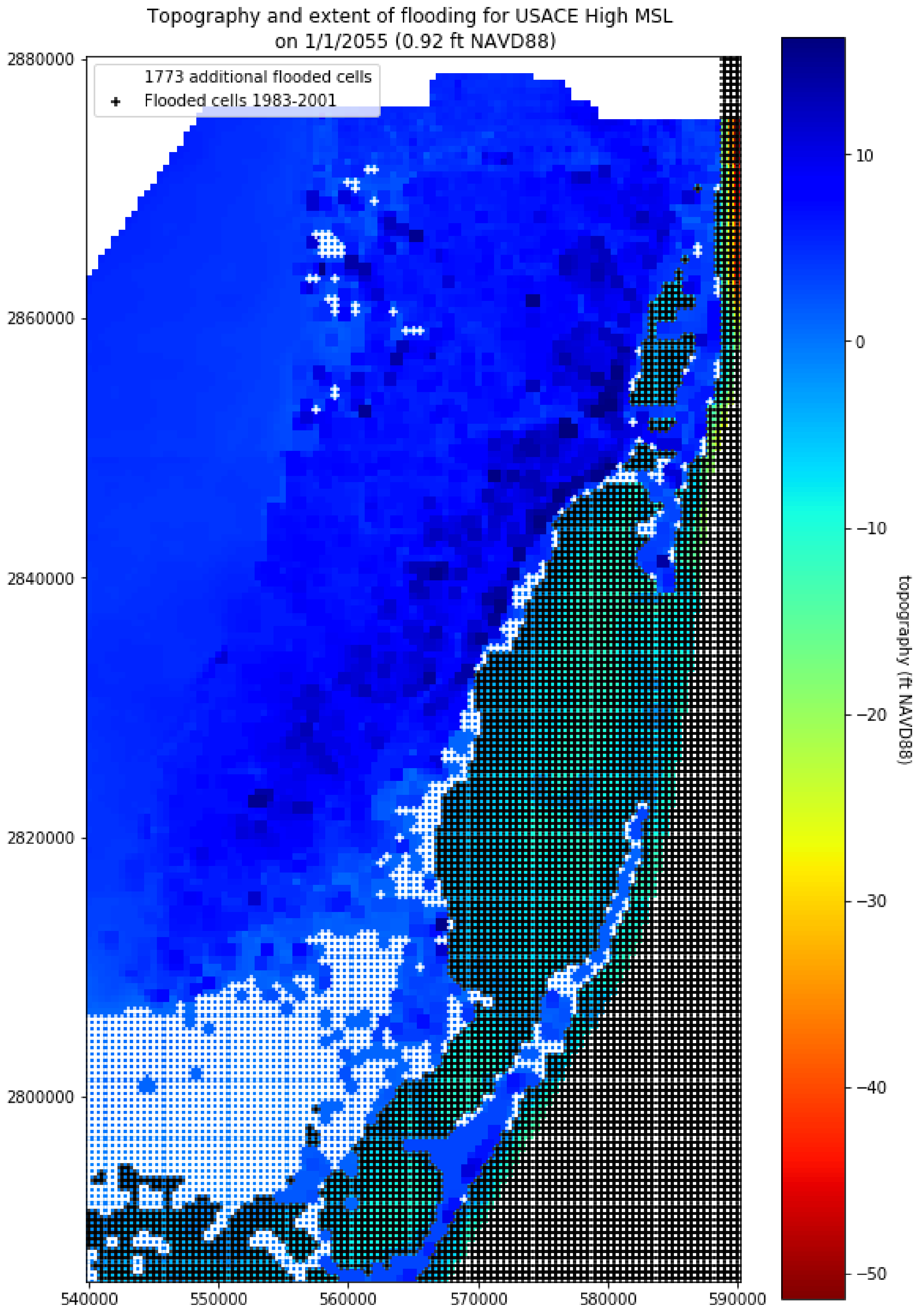


Figure 25. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 1/1/2055 at Virginia Key for the USACE High scenario (0.92 ft NAVD88). A total of 1773 additional model grid cells would be flooded compared to the cells flooded based on 1983-2001 NTDE MSL. X and Y coordinates in meters, UTM17N, NAD83.

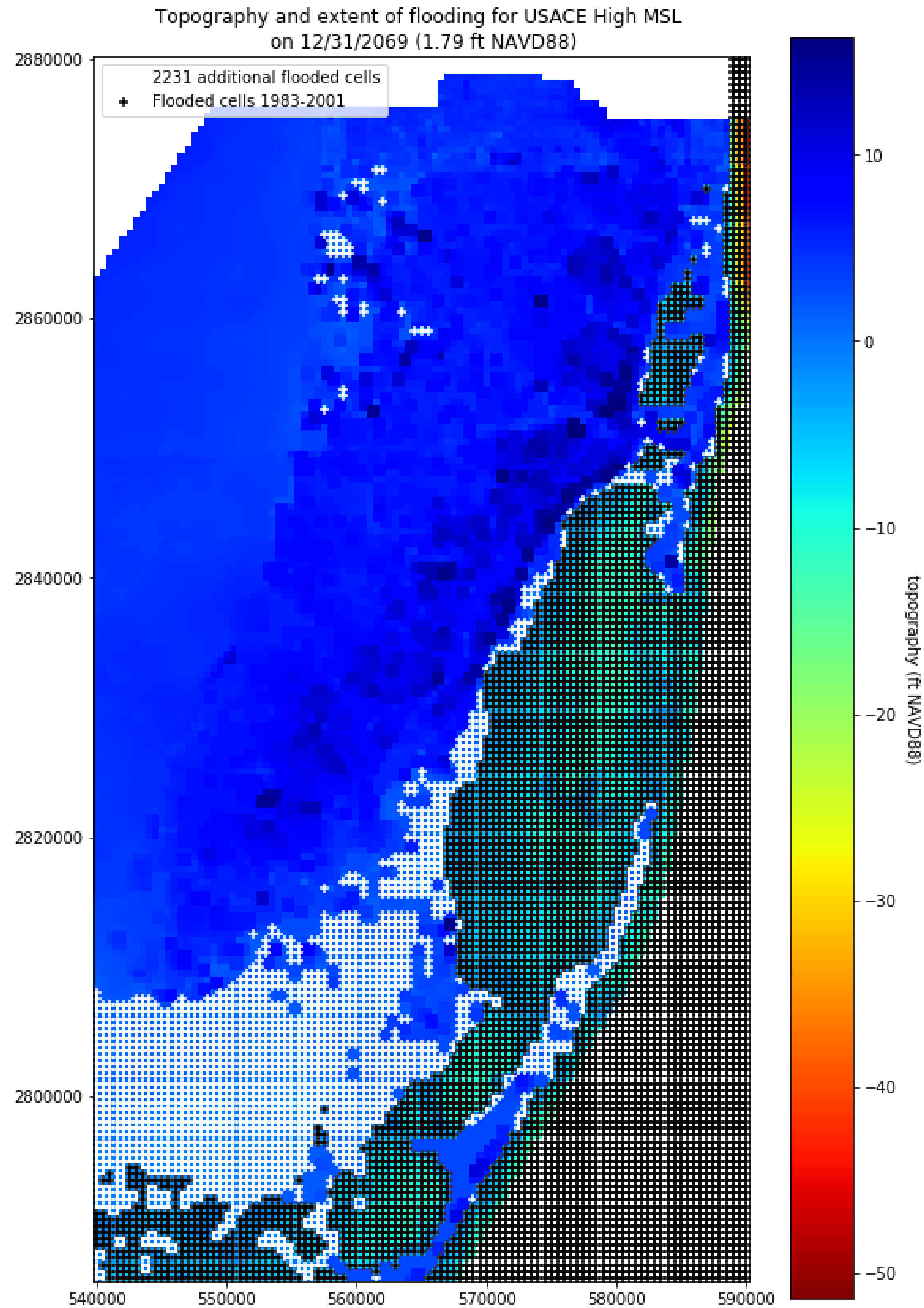


Figure 26. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 12/31/2069 at Virginia Key for the USACE High scenario (1.79 ft NAVD88). A total of 2231 additional model grid cells would be flooded compared to the cells flooded based on 1983-2001 NTDE MSL. X and Y coordinates in meters, UTM17N, NAD83.

Future rainfall

In the past, Irizarry evaluated the ability of various climate model outputs to capture the historical distribution of rainfall extremes for durations of 1-7 days in the state of Florida. In particular, the following statistically-downscaled data products, based on the World Climate Research Programme (WCRP) Coupled Model Intercomparison Project (CMIP) phase 5 (CMIP5), were evaluated:

- US Bureau of Reclamation bias-corrected and statistically-downscaled climate projections (Maurer et al., 2007; Reclamation, 2013), which used daily Bias-Correction Constructed Analogues (BCCA) technique for statistical downscaling.
- University of California (San Diego)'s Localized Constructed Analogs (LOCA) product (Pierce et al., 2014).

SFWMD staff had previously evaluated these same statistically downscaled data products in terms of their ability to capture historical rainfall temporal and spatial patterns in south Florida. As part of this work, it was observed that LOCA generally did a better job than BCCA at capturing rainfall patterns in the state (Irizarry et al., 2016). For this reason and also due to the fact that the LOCA product was used to guide the 4th US National Climate Assessment report (https://scenarios.globalchange.gov/), this dataset was chosen and further evaluated for this project. A description of the LOCA dataset is included below.

Localized Constructed Analogues technique (LOCA) is a statistical downscaling technique that uses past history to add improved fine-scale detail to global climate models (Pierce et al., 2014). First, a pool of candidate observed analog days is chosen by matching the model field to be downscaled to observed days over the region that is positively correlated with the point being downscaled, which leads to a natural independence of the downscaling results to the extent of the domain being downscaled. Then the one candidate analog day that best matches in the local area around the grid cell being downscaled is the single analog day used there.

Most grid cells are downscaled using only the single locally selected analog day, but locations whose neighboring cells identify a different analog day use a weighted combination of the center and adjacent analog days to reduce edge discontinuities. By contrast, existing constructed analog methods typically use a weighted average of the same 30 analog days for the entire domain. By reducing this averaging, LOCA produces better estimates of extreme days, constructs a more realistic depiction of the spatial coherence of the downscaled field, and reduces the problem of producing too many drizzle (light-precipitation) days.

The University of California at San Diego has used LOCA to downscale 32 global climate models from the CMIP5 archive at a 1/16th degree (approx. 4.3 miles; 6.9 km) spatial resolution, covering North America from central Mexico through Southern Canada. The historical period is 1950-2005, and there are two future scenarios available: RCP 4.5 and RCP 8.5 over the period 2006-2100 (although some models stop in 2099). RCP stands for representative concentration pathways (RCP). RCP 4.5 represents medium-low year 2100 radiative forcing (4.5 W/m^2), while RCP 8.5 represents high radiative forcing (8.5 W/m^2).

The variables currently available daily minimum and maximum temperature and relative humidity, and daily precipitation. In the future, they will be running the VIC hydrological model with the downscaled data, which will give many more variables, such as snow cover, soil moisture, runoff, and humidity, all at a 1/16th degree spatial resolution on a daily timescale. For more information see http://loca.ucsd.edu/.

Statistically downscaled daily rainfall timeseries from LOCA for 30 climate models were evaluated for mainland grid cells in Miami Dade County in terms of their ability to match historical data. The analysis described hereafter only pertains to MODFLOW model grid cells in the mainland. As described in the Miami-Dade MODFLOW model documentation, the model uses daily historical rainfall from NEXRAD at 2-km x 2-km for the calibration/verification period 1996-2010. In the model, a 1.05 multiplicative correction factor is applied to the NEXRAD rainfall due to previous evidence of bias compared to gage data. This data would have been ideal for evaluating and bias-correcting LOCA rainfall timeseries since the model was calibrated/verified to it. However, the LOCA historical period ends in 2005, which limits the comparison. For this reason, daily historical rainfall obtained from the South Florida Water Management Model (SFWMM) v4.7 binary file on a 3.2-km x 3.2-km (2-mi x 2-mi) grid for the period 1991-2005 was used in evaluating LOCA historical performance for the same period.

Two analyses of LOCA performance were done to cull the LOCA models. One analysis looked at the annual total rainfall and the other one was based on wet season (May-October) precipitation. The analysis was only based on MODFLOW model grid cells in the mainland. This was due to the fact that the LOCA and SFWMM grids did not fully cover active model grid cells in the barrier islands. The R code is included in Appendix B. R code for rainfall bias correction. Table 3 and Table 4 show the performance of each LOCA model run for the two periods of interest. It can be noticed that all models are negatively biased for annual total rainfall with biases ranging from -7.4 to -0.13 in/yr (average of -4.2 in/yr) over Miami-Dade County. In the wet season, 28/30 models are negatively biased with biases ranging from -6.4 in/season to +0.75 in/season (average of -3.1 in/season).

In order to account for these biases, the mean of future LOCA predictions for both the annual totals and wet season, were bias-corrected as follows:

$$R_{m-padj} = R_{m-p} * \frac{R_{o-c}}{R_{m-c}}$$

Equation 5

where R_{m-padj} is the adjusted future predicted annual total or wet season rainfall, R_{m-p} is the future predicted annual total or wet season rainfall, R_{o-c} is the observed current (historical) annual total or wet season rainfall, and R_{m-c} is the current (historical) predicted annual total or wet season rainfall. This equation is analogous to the Multiplicative Quantile Delta Mapping (MQDM) technique used for bias-correction of daily rainfall data, which will be discussed later on. It assumes that the mean will also change in a multiplicative way, which is not necessarily the case. However, performing MQDM on daily rainfall data for all LOCA runs was found to be too computationally-intensive and the above simplified equation seems adequate as a first approximation for model evaluation.

Figure 27 and Figure 29 show the distribution of changes from SFWMM historical (1991-2005) to future (2055-2069) bias-corrected annual total and wet season rainfall, respectively. It can be noticed that about half of the models project a decrease in rainfall, while the other half project an increase in rainfall. The median projected change in annual total rainfall is slightly less than 0 (approximately -1.2 in/yr). During the wet season, 70% of models predict a decrease in rainfall, while 30% predict an increase. This is consistent with previous studies by Kirtman and others (FIU Rainfall Workshop, May 16, 2019) who evaluated the US Bureau's BCSD data product and found that most models projected a drying of south Florida in the future. The median projected change in wet season rainfall is approximately -2.2 to -2.5 in/yr. Figure 28 and Figure 30 show scatterplots of the annual total and wet season rainfall pre and post bias-correction of the mean based on Equation 5.

It was decided that to be conservative in the estimation of average wet season water tables, a 90th-95th percentile of future bias-corrected rainfall (i.e. an increase from the baseline) should be selected for future modeling. Based on the analysis of annual total rainfall, the 95th percentile run was pr_MPI-ESM-MR_r1i1p1_rcp85 with 5.2 in/yr (9.0%) of additional rainfall in the future period compared to 1991-2005 SFWMM historical data. The run with the max change was pr_ACCESS1-0_r1i1p1_rcp45 with 8.7 in/yr (15.5%) extra rainfall. This run is shown with a red triangle marker on Figure 28 and Figure 30.

Based on the analysis of wet season rainfall, the 95th percentile run was pr_MPI-ESM-LR_r1i1p1_rcp45 with 3.6 in/season (8.2%) extra rainfall in the future period compared to 1991-2005 SFWMM historical data for wet season. The run with the max change was pr_MPI-ESM-MR_r1i1p1_rcp45 with 4.9 in/season (11.2%) extra rainfall.

As shown in Figure 28 and Figure 30, there are other runs showing similar future bias-corrected annual total and wet season rainfall, but which had much smaller biases in the historical period. In this figure, the closer a LOCA model run point is to the 1:1 line, the smaller its historical bias and the smaller the bias-correction of its future rainfall. LOCA run pr_MRI-CGCM3_r1i1p1_rcp85, shown as a yellow marker in these two figures, has similar bias-corrected annual total and wet season rainfall to the 95th percentile runs identified above. However, this run had a much smaller bias than the 95th percentile runs. Therefore, pr_MRI-CGCM3_r1i1p1_rcp85 was chosen for daily bias-correction for future scenario modeling. Figure 31 shows the spatial distribution of the observed and simulated means prior to and post bias-correction of the means using Equation 5. Despite the coarseness of the LOCA dataset, it is evident how the historical spatial pattern of precipitation is generally well-captured by this LOCA run. As a final check, Figure 32 shows that this LOCA model run does a decent job at capturing the historical seasonal cycle of rainfall in Miami-Dade County even prior to bias correction. It is evident that simulated variability is less in October, and the model overestimates variability in July and August.

Table 3. Comparison of LOCA run performance and predicted future annual total rainfall (in/yr). Note: SFWMM historical rainfall for 1991-2005 is 57.57 in/yr.

LOCA model run/RCP	Hist. (1991-2005) mean	Hist. bias	Fut. (2055-2069) mean	Fut. (2055-2069) B.C. mean	Fut. (2055-2069) B.C. mean minus hist. (1991-2005) mean	Percent change hist. to fut. B.C.	Percentile fraction
pr_ACCESS1-0_r1i1p1_rcp45	53.07	-4.5	61.06	66.23	8.66	15.04	1
pr_ACCESS1-0_r1i1p1_rcp85	53.07	-4.5	52.74	57.23	-0.34	-0.59	0.559
pr_ACCESS1-3_r1i1p1_rcp45	55.44	-2.13	52.86	54.99	-2.58	-4.48	0.457
pr_ACCESS1-3_r1i1p1_rcp85	55.44	-2.13	57.17	59.44	1.87	3.25	0.711
pr_CCSM4_r6i1p1_rcp45	52.54	-5.03	47.72	52.29	-5.28	-9.17	0.186
pr_CCSM4_r6i1p1_rcp85	52.54	-5.03	47.93	52.5	-5.07	-8.81	0.22
pr_CESM1-BGC_r1i1p1_rcp45	51.77	-5.8	50.71	56.4	-1.17	-2.03	0.491
pr_CESM1-BGC_r1i1p1_rcp85	51.77	-5.8	47.06	52.38	-5.19	-9.02	0.203
pr_CESM1-CAM5_r1i1p1_rcp45	52.33	-5.24	47.77	52.57	-5	-8.69	0.237
pr_CESM1-CAM5_r1i1p1_rcp85	52.33	-5.24	49.38	54.34	-3.23	-5.61	0.423
pr_CMCC-CMS_r1i1p1_rcp45	53.61	-3.96	54.6	58.7	1.13	1.96	0.661
pr_CMCC-CMS_r1i1p1_rcp85	53.61	-3.96	46.47	49.92	-7.65	-13.29	0.084
pr_CMCC-CM_r1i1p1_rcp45	53.22	-4.35	55.08	59.65	2.08	3.61	0.762
pr_CMCC-CM_r1i1p1_rcp85	53.22	-4.35	49.41	53.47	-4.1	-7.12	0.338
pr_CNRM-CM5_r1i1p1_rcp45	53.72	-3.85	56.24	60.28	2.71	4.71	0.83
pr_CNRM-CM5_r1i1p1_rcp85	53.72	-3.85	53.79	57.65	0.08	0.14	0.576
pr_CSIRO-Mk3-6-0_r1i1p1_rcp45	51.7	-5.87	53.82	59.92	2.35	4.08	0.813
pr_CSIRO-Mk3-6-0_r1i1p1_rcp85	51.7	-5.87	54.65	60.85	3.28	5.70	0.898
pr_CanESM2_r1i1p1_rcp45	54.45	-3.12	53.3	56.42	-1.15	-2.00	0.508
pr_CanESM2_r1i1p1_rcp85	54.45	-3.12	48.7	51.53	-6.04	-10.49	0.152
pr_EC-EARTH_r2i1p1_rcp85	53.86	-3.71	52.88	56.58	-0.99	-1.72	0.525
pr_EC-EARTH_r8i1p1_rcp45	53.86	-3.71	49.72	53.2	-4.37	-7.59	0.288
pr_FGOALS-g2_r1i1p1_rcp45	50.9	-6.67	52.56	59.48	1.91	3.32	0.728
pr_FGOALS-g2_r1i1p1_rcp85	50.9	-6.67	47.46	53.69	-3.88	-6.74	0.355
pr_GFDL-CM3_r1i1p1_rcp45	50.17	-7.4	53.32	61.23	3.66	6.36	0.915
pr_GFDL-CM3_r1i1p1_rcp85	50.17	-7.4	56.79	65.2	7.63	13.25	0.983
pr_GFDL-ESM2G_r1i1p1_rcp45	50.57	-7	46.25	52.64	-4.93	-8.56	0.254
pr_GFDL-ESM2G_r1i1p1_rcp85	50.57	-7	44.34	50.48	-7.09	-12.32	0.118
pr_GFDL-ESM2M_r1i1p1_rcp45	51.31	-6.26	47.51	53.38	-4.19	-7.28	0.322
pr_GFDL-ESM2M_r1i1p1_rcp85	51.31	-6.26	51.45	57.78	0.21	0.36	0.593
pr_GISS-E2-H_r2i1p1_rcp85	53.44	-4.13	50.58	54.55	-3.02	-5.25	0.44

LOCA model run/RCP	Hist. (1991-2005) mean	Hist. bias	Fut. (2055-2069) mean	Fut. (2055-2069) B.C. mean	Fut. (2055-2069) B.C. mean minus hist. (1991-2005) mean	Percent change hist. to fut. B.C.	Percentile fraction
pr_GISS-E2-H_r6i1p3_rcp45	53.44	-4.13	52.55	56.63	-0.94	-1.63	0.542
pr_GISS-E2-R_r2i1p1_rcp85	50.89	-6.68	52.6	59.5	1.93	3.35	0.745
pr_GISS-E2-R_r6i1p1_rcp45	50.89	-6.68	48.85	55.27	-2.3	-4.00	0.474
pr_HadGEM2-AO_r1i1p1_rcp45	56.95	-0.62	50.93	51.54	-6.03	-10.47	0.169
pr_HadGEM2-AO_r1i1p1_rcp85	56.95	-0.62	53.54	54.12	-3.45	-5.99	0.406
pr_HadGEM2-CC_r1i1p1_rcp45	52.47	-5.1	53.46	58.7	1.13	1.96	0.661
pr_HadGEM2-CC_r1i1p1_rcp85	52.47	-5.1	52.69	57.81	0.24	0.42	0.61
pr_HadGEM2-ES_r1i1p1_rcp45	53.42	-4.15	54.18	58.37	0.8	1.39	0.644
pr_HadGEM2-ES_r1i1p1_rcp85	53.42	-4.15	49.2	53.02	-4.55	-7.90	0.271
pr_IPSL-CM5A-LR_r1i1p1_rcp45	57.44	-0.13	53.88	54	-3.57	-6.20	0.389
pr_IPSL-CM5A-LR_r1i1p1_rcp85	57.44	-0.13	49.64	49.77	-7.8	-13.55	0.067
pr_IPSL-CM5A-MR_r1i1p1_rcp45	54.66	-2.91	50.51	53.2	-4.37	-7.59	0.288
pr_IPSL-CM5A-MR_r1i1p1_rcp85	54.66	-2.91	56.4	59.4	1.83	3.18	0.694
pr_MIROC-ESM-CHEM_r1i1p1_rcp45	54.71	-2.86	43.84	46.19	-11.38	-19.77	0.016
pr_MIROC-ESM-CHEM_r1i1p1_rcp85	54.71	-2.86	42.47	44.74	-12.83	-22.29	0
pr_MIROC-ESM_r1i1p1_rcp45	50.35	-7.22	50.9	58.2	0.63	1.09	0.627
pr_MIROC-ESM_r1i1p1_rcp85	50.35	-7.22	47.1	53.84	-3.73	-6.48	0.372
pr_MIROC5_r1i1p1_rcp45	51.99	-5.58	53.89	59.69	2.12	3.68	0.779
pr_MIROC5_r1i1p1_rcp85	51.99	-5.58	54.86	60.76	3.19	5.54	0.881
pr_MPI-ESM-LR_r1i1p1_rcp45	54.74	-2.83	57.63	60.6	3.03	5.26	0.864
pr_MPI-ESM-LR_r1i1p1_rcp85	54.74	-2.83	57.41	60.38	2.81	4.88	0.847
pr_MPI-ESM-MR_r1i1p1_rcp45	51.11	-6.46	57.03	64.25	6.68	11.60	0.966
pr_MPI-ESM-MR_r1i1p1_rcp85	51.11	-6.46	55.69	62.73	5.16	8.96	0.949
pr_MRI-CGCM3_r1i1p1_rcp45	56.68	-0.89	59	59.9	2.33	4.05	0.796
pr_MRI-CGCM3_r1i1p1_rcp85	***56.68***	***-0.89***	***60.94***	***61.9***	***4.33***	***7.52***	***0.932***
pr_NorESM1-M_r1i1p1_rcp45	56.48	-1.09	49.14	50.11	-7.46	-12.96	0.101
pr_NorESM1-M_r1i1p1_rcp85	56.48	-1.09	46.37	47.31	-10.26	-17.82	0.033
pr_bcc-csm1-1-m_r1i1p1_rcp45	56.41	-1.16	48.6	49.59	-7.98	-13.86	0.05
pr_bcc-csm1-1-m_r1i1p1_rcp85	56.41	-1.16	49.52	50.52	-7.05	-12.25	0.135

Table 4. Comparison of LOCA run performance and predicted future wet season rainfall (in/ wet season). Note: SFWMM historical rainfall for 1991-2005 is 43.61in/wet season.

LOCA model run/RCP	**Hist. (1991-2005) mean**	**Hist. bias**	**Fut. (2055-2069) mean**	**Fut. (2055-2069) B.C. mean**	**Fut. (2055-2069) B.C. mean minus hist. (1991-2005) mean**	**Percent change hist. to fut. B.C.**	**Percentile fraction**
pr_ACCESS1-0_r1i1p1_rcp45	41.66	-1.95	43.2	45.22	1.61	3.69	0.813
pr_ACCESS1-0_r1i1p1_rcp85	41.66	-1.95	38.55	40.38	-3.23	-7.41	0.389
pr_ACCESS1-3_r1i1p1_rcp45	41.66	-1.95	36.4	38.22	-5.39	-12.36	0.203
pr_ACCESS1-3_r1i1p1_rcp85	41.66	-1.95	39.02	40.96	-2.65	-6.08	0.44
pr_CCSM4_r6i1p1_rcp45	39.18	-4.43	36.21	40.3	-3.31	-7.59	0.355
pr_CCSM4_r6i1p1_rcp85	39.18	-4.43	35.23	39.2	-4.41	-10.11	0.288
pr_CESM1-BGC_r1i1p1_rcp45	39.29	-4.32	36.9	40.98	-2.63	-6.03	0.457
pr_CESM1-BGC_r1i1p1_rcp85	39.29	-4.32	33.99	37.78	-5.83	-13.37	0.186
pr_CESM1-CAM5_r1i1p1_rcp45	39.53	-4.08	33.31	36.76	-6.85	-15.71	0.084
pr_CESM1-CAM5_r1i1p1_rcp85	39.53	-4.08	35.86	39.57	-4.04	-9.26	0.305
pr_CMCC-CMS_r1i1p1_rcp45	40.57	-3.04	43.08	46.39	2.78	6.37	0.898
pr_CMCC-CMS_r1i1p1_rcp85	40.57	-3.04	34.28	36.87	-6.74	-15.46	0.118
pr_CMCC-CM_r1i1p1_rcp45	40.46	-3.15	43.37	46.84	3.23	7.41	0.915
pr_CMCC-CM_r1i1p1_rcp85	40.46	-3.15	39.52	42.62	-0.99	-2.27	0.593
pr_CNRM-CM5_r1i1p1_rcp45	41.55	-2.06	43.13	45.28	1.67	3.83	0.83
pr_CNRM-CM5_r1i1p1_rcp85	41.55	-2.06	41.38	43.44	-0.17	-0.39	0.677
pr_CSIRO-Mk3-6-0_r1i1p1_rcp45	39.76	-3.85	39.45	43.26	-0.35	-0.80	0.661
pr_CSIRO-Mk3-6-0_r1i1p1_rcp85	39.76	-3.85	38.96	42.74	-0.87	-1.99	0.627
pr_CanESM2_r1i1p1_rcp45	41.89	-1.72	38.17	39.81	-3.8	-8.71	0.322
pr_CanESM2_r1i1p1_rcp85	41.89	-1.72	34.49	35.96	-7.65	-17.54	0.067
pr_EC-EARTH_r2i1p1_rcp85	41.65	-1.96	38.32	40.14	-3.47	-7.96	0.338
pr_EC-EARTH_r8i1p1_rcp45	41.65	-1.96	36.86	38.63	-4.98	-11.42	0.237
pr_FGOALS-g2_r1i1p1_rcp45	39.16	-4.45	39.58	44.11	0.5	1.15	0.745
pr_FGOALS-g2_r1i1p1_rcp85	39.16	-4.45	36.67	40.84	-2.77	-6.35	0.423
pr_GFDL-CM3_r1i1p1_rcp45	37.92	-5.69	38.3	44.08	0.47	1.08	0.728
pr_GFDL-CM3_r1i1p1_rcp85	37.92	-5.69	40.04	46.1	2.49	5.71	0.881
pr_GFDL-ESM2G_r1i1p1_rcp45	37.2	-6.41	35.31	41.38	-2.23	-5.11	0.508
pr_GFDL-ESM2G_r1i1p1_rcp85	37.2	-6.41	32.1	37.63	-5.98	-13.71	0.169
pr_GFDL-ESM2M_r1i1p1_rcp45	38.88	-4.73	36.23	40.7	-2.91	-6.67	0.406
pr_GFDL-ESM2M_r1i1p1_rcp85	38.88	-4.73	39.72	44.6	0.99	2.27	0.779

LOCA model run/RCP	Hist. (1991-2005) mean	Hist. bias	Fut. (2055-2069) mean	Fut. (2055-2069) B.C. mean	Fut. (2055-2069) B.C. mean minus hist. (1991-2005) mean	Percent change hist. to fut. B.C.	Percentile fraction
pr_GISS-E2-H_r2i1p1_rcp85	40.18	-3.43	37.83	41.11	-2.5	-5.73	0.491
pr_GISS-E2-H_r6i1p3_rcp45	40.18	-3.43	38.95	42.31	-1.3	-2.98	0.542
pr_GISS-E2-R_r2i1p1_rcp85	38.27	-5.34	38.2	43.53	-0.08	-0.18	0.694
pr_GISS-E2-R_r6i1p1_rcp45	38.27	-5.34	35.38	40.34	-3.27	-7.50	0.372
pr_HadGEM2-AO_r1i1p1_rcp45	43.74	0.13	36.85	36.8	-6.81	-15.62	0.101
pr_HadGEM2-AO_r1i1p1_rcp85	43.74	0.13	37.56	37.47	-6.14	-14.08	0.152
pr_HadGEM2-CC_r1i1p1_rcp45	39.56	-4.05	38.47	42.47	-1.14	-2.61	0.576
pr_HadGEM2-CC_r1i1p1_rcp85	39.56	-4.05	38.75	42.73	-0.88	-2.02	0.61
pr_HadGEM2-ES_r1i1p1_rcp45	41.08	-2.53	41.98	44.53	0.92	2.11	0.762
pr_HadGEM2-ES_r1i1p1_rcp85	41.08	-2.53	35.01	37.17	-6.44	-14.77	0.135
pr_IPSL-CM5A-LR_r1i1p1_rcp45	43.37	-0.24	42.61	42.83	-0.78	-1.79	0.644
pr_IPSL-CM5A-LR_r1i1p1_rcp85	43.37	-0.24	38.81	39.04	-4.57	-10.48	0.254
pr_IPSL-CM5A-MR_r1i1p1_rcp45	41.16	-2.45	40.08	42.46	-1.15	-2.64	0.559
pr_IPSL-CM5A-MR_r1i1p1_rcp85	41.16	-2.45	44.77	47.42	3.81	8.74	0.966
pr_MIROC-ESM-CHEM_r1i1p1_rcp45	41.24	-2.37	30.18	31.96	-11.65	-26.71	0
pr_MIROC-ESM-CHEM_r1i1p1_rcp85	41.24	-2.37	30.95	32.76	-10.85	-24.88	0.033
pr_MIROC-ESM_r1i1p1_rcp45	38.06	-5.55	36.27	41.55	-2.06	-4.72	0.525
pr_MIROC-ESM_r1i1p1_rcp85	38.06	-5.55	35.87	41.08	-2.53	-5.80	0.474
pr_MIROC5_r1i1p1_rcp45	40.07	-3.54	40.15	43.71	0.1	0.23	0.711
pr_MIROC5_r1i1p1_rcp85	40.07	-3.54	41.38	45.04	1.43	3.28	0.796
pr_MPI-ESM-LR_r1i1p1_rcp45	41.04	-2.57	44.4	47.19	3.58	8.21	0.949
pr_MPI-ESM-LR_r1i1p1_rcp85	41.04	-2.57	42.85	45.55	1.94	4.45	0.847
pr_MPI-ESM-MR_r1i1p1_rcp45	38.87	-4.74	43.24	48.52	4.91	11.26	1
pr_MPI-ESM-MR_r1i1p1_rcp85	38.87	-4.74	42.79	48	4.39	10.07	0.983
pr_MRI-CGCM3_r1i1p1_rcp45	42.86	-0.75	44.83	45.6	1.99	4.56	0.864
pr_MRI-CGCM3_r1i1p1_rcp85	***42.86***	***-0.75***	***46.26***	***47.08***	***3.47***	***7.96***	***0.932***
pr_NorESM1-M_r1i1p1_rcp45	44.36	0.75	35.4	34.82	-8.79	-20.16	0.05
pr_NorESM1-M_r1i1p1_rcp85	44.36	0.75	33.26	32.75	-10.86	-24.90	0.016
pr_bcc-csm1-1-m_r1i1p1_rcp45	41.99	-1.62	37.66	39.08	-4.53	-10.39	0.271
pr_bcc-csm1-1-m_r1i1p1_rcp85	41.99	-1.62	36.89	38.31	-5.3	-12.15	0.22

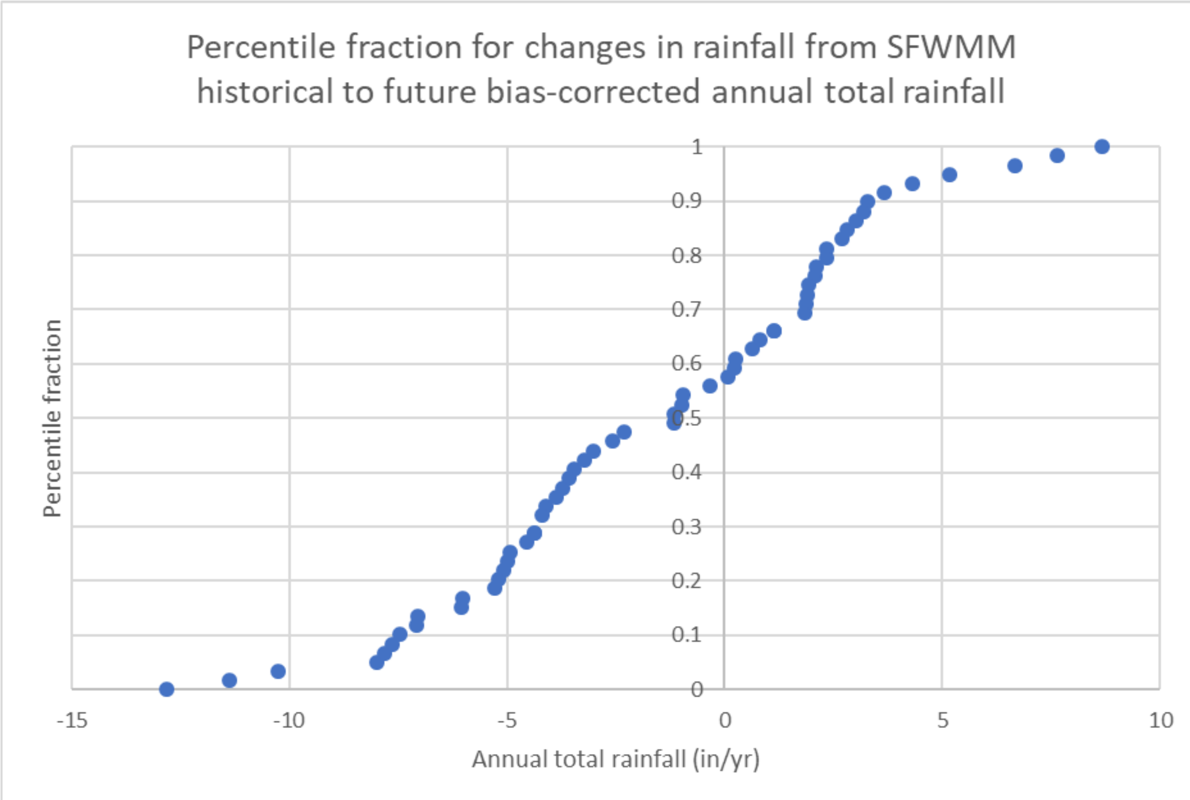


Figure 27. Distribution of changes from SFWMM historical to future bias-corrected annual total rainfall.

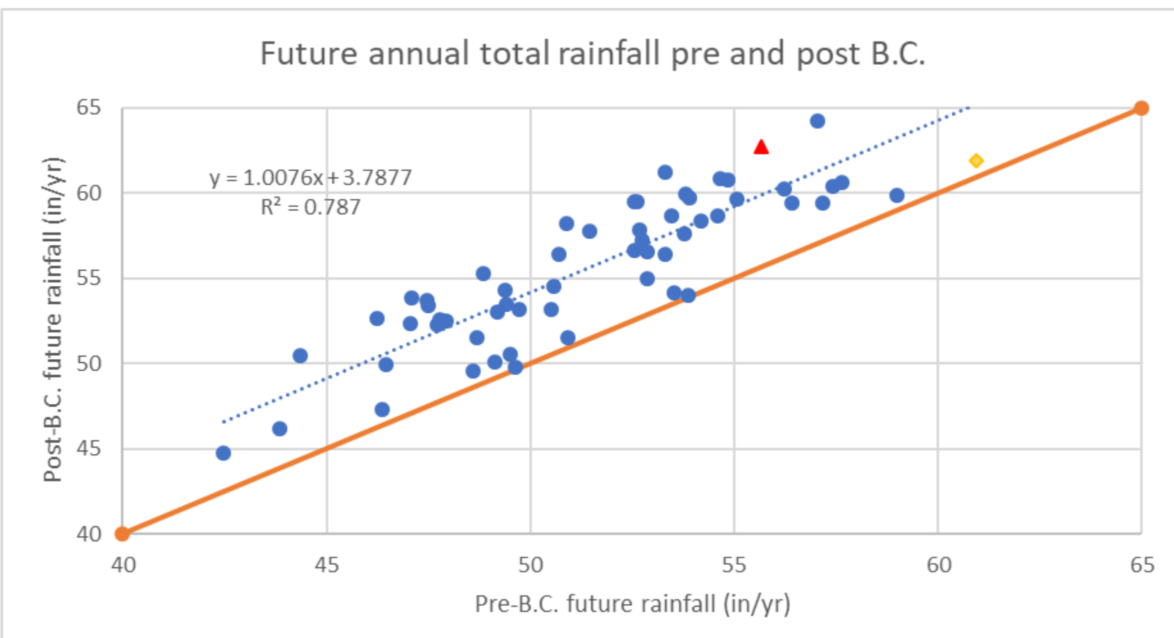


Figure 28. Scatterplot of future annual total rainfall pre and post bias-correction.

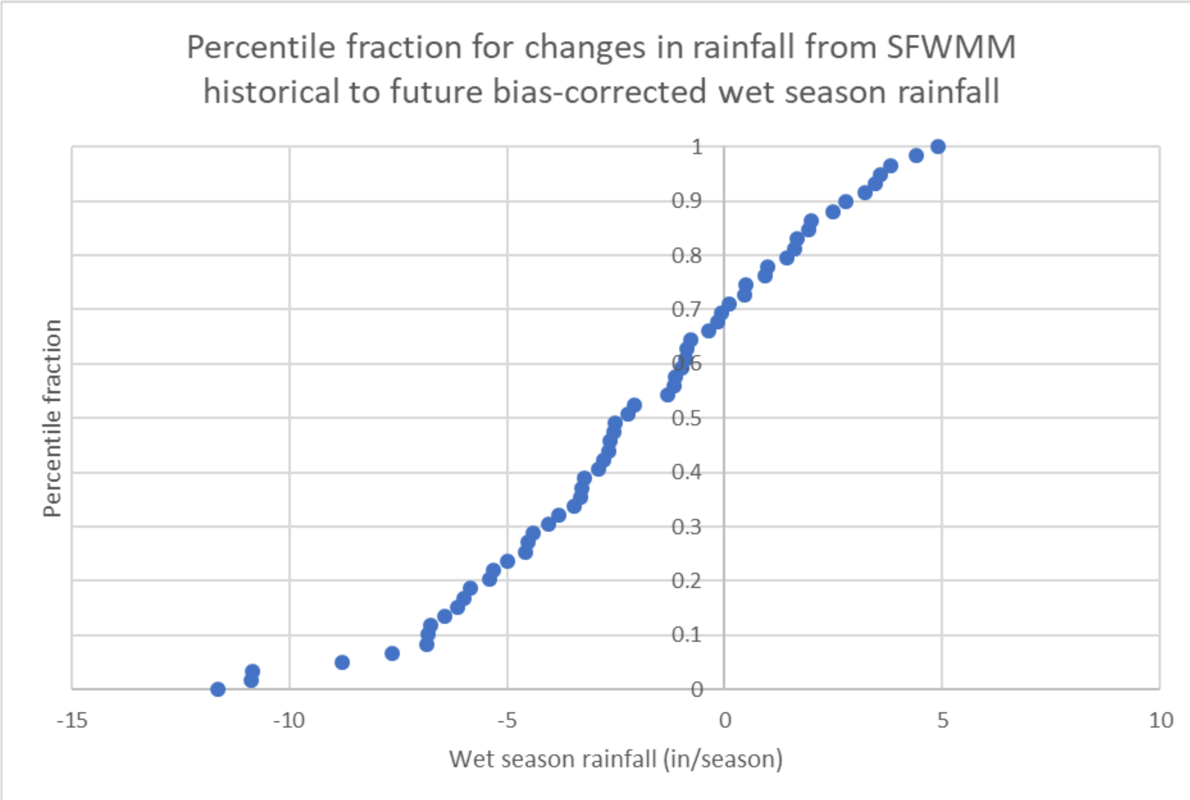


Figure 29. Distribution of changes from SFWMM historical to future bias-corrected wet season rainfall.

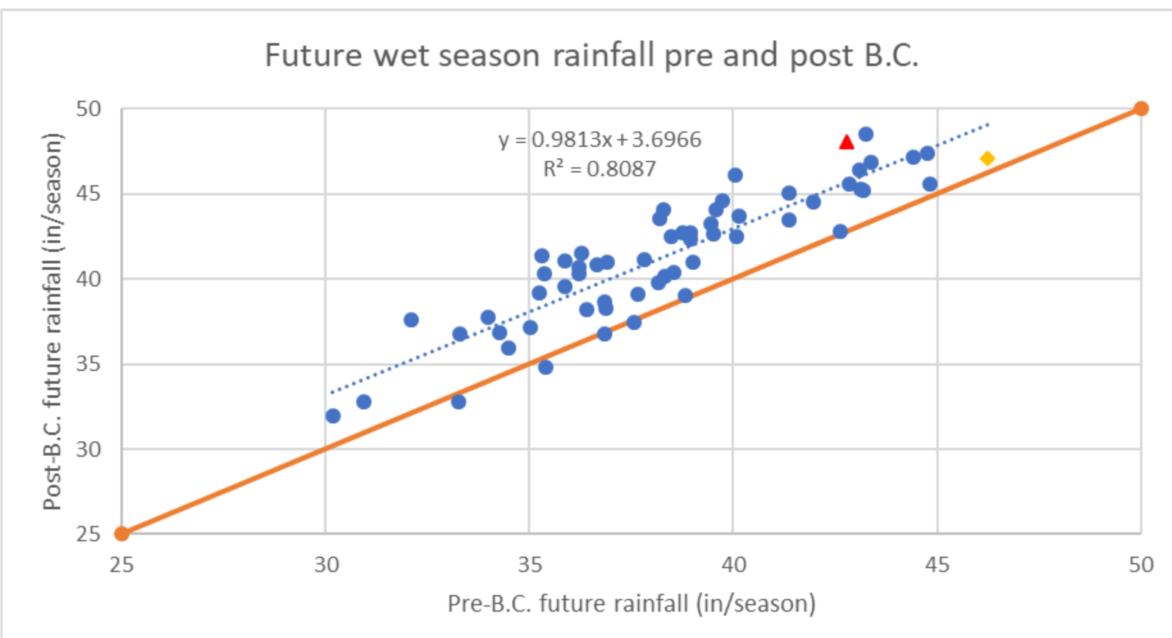
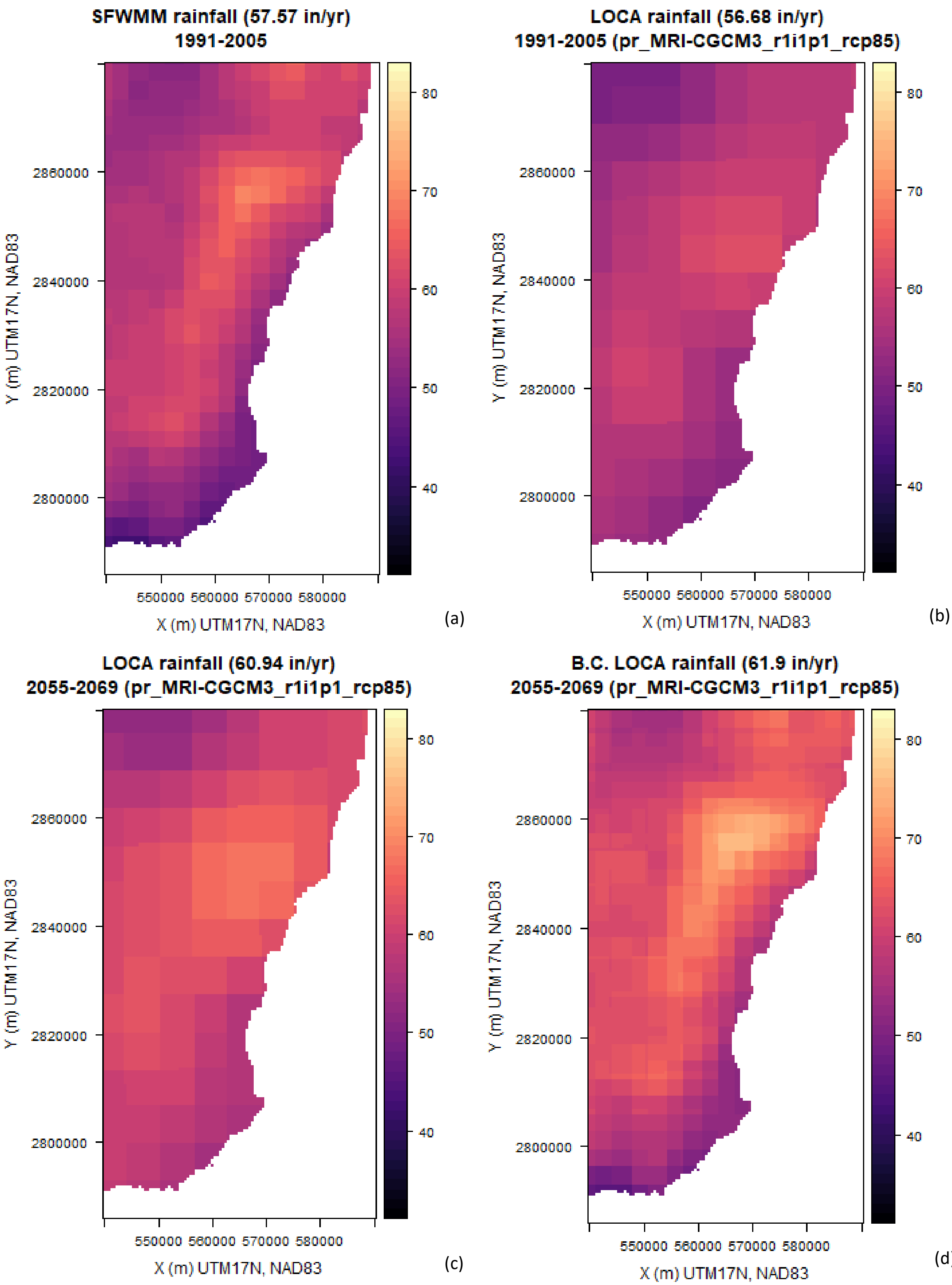


Figure 30. Scatterplot of future wet season rainfall pre and post bias-correction.


SFWMM rainfall (57.57 in/yr)
1991-2005
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
2800000
2820000
2840000
2860000
550000
560000
570000
580000
40
50
60
70
80
(a)
LOCA rainfall (56.68 in/yr)
1991-2005 (pr_MRI-CGCM3_r1i1p1_rcp85)
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
(b)
LOCA rainfall (60.94 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
(c)
B.C. LOCA rainfall (61.9 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
(d)

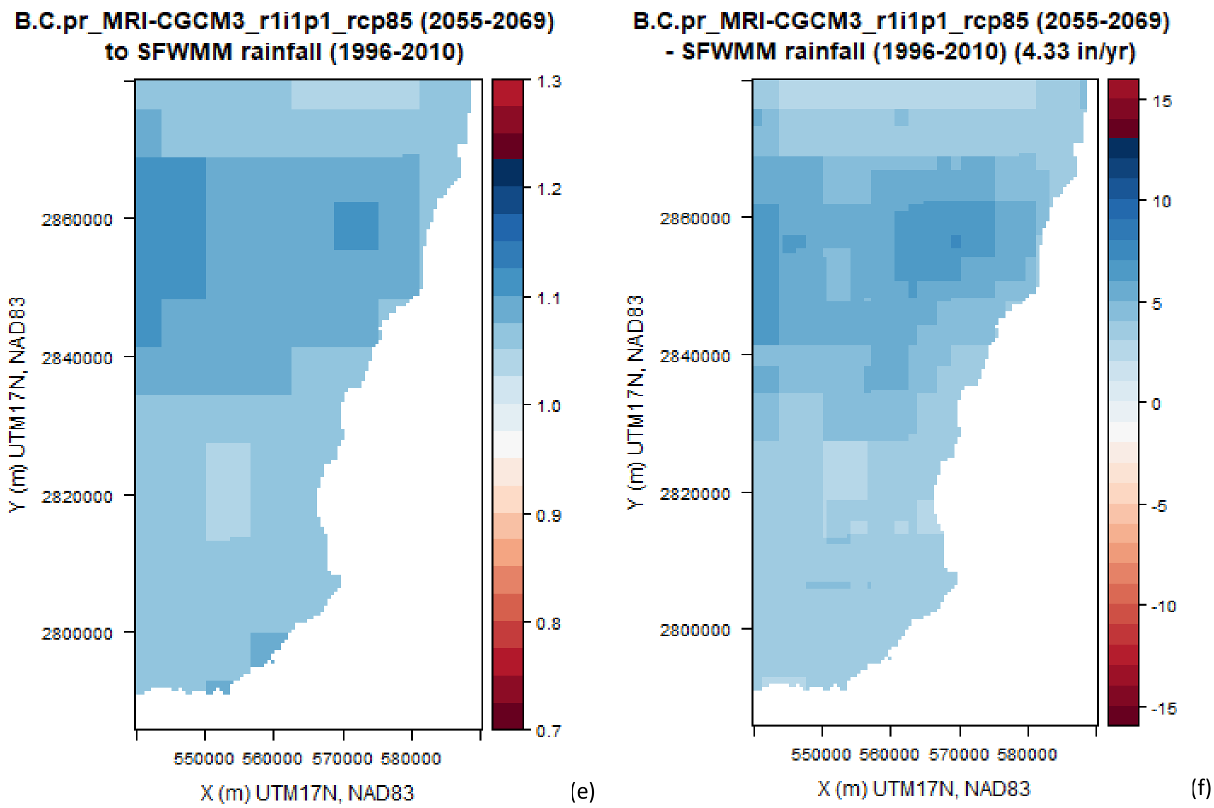


Figure 31. Spatial distribution of (a) SFWMM historical rainfall (1991-2005), (b) Simulated historical rainfall (1991-2005), (c) Simulated future rainfall (2055-2069), (d) Bias-corrected simulated future rainfall, (e) = (d)/(a), (f) = (d) - (a). Note: Here, bias correction has been done on the annual total rainfall (mean) by applying Equation 5.

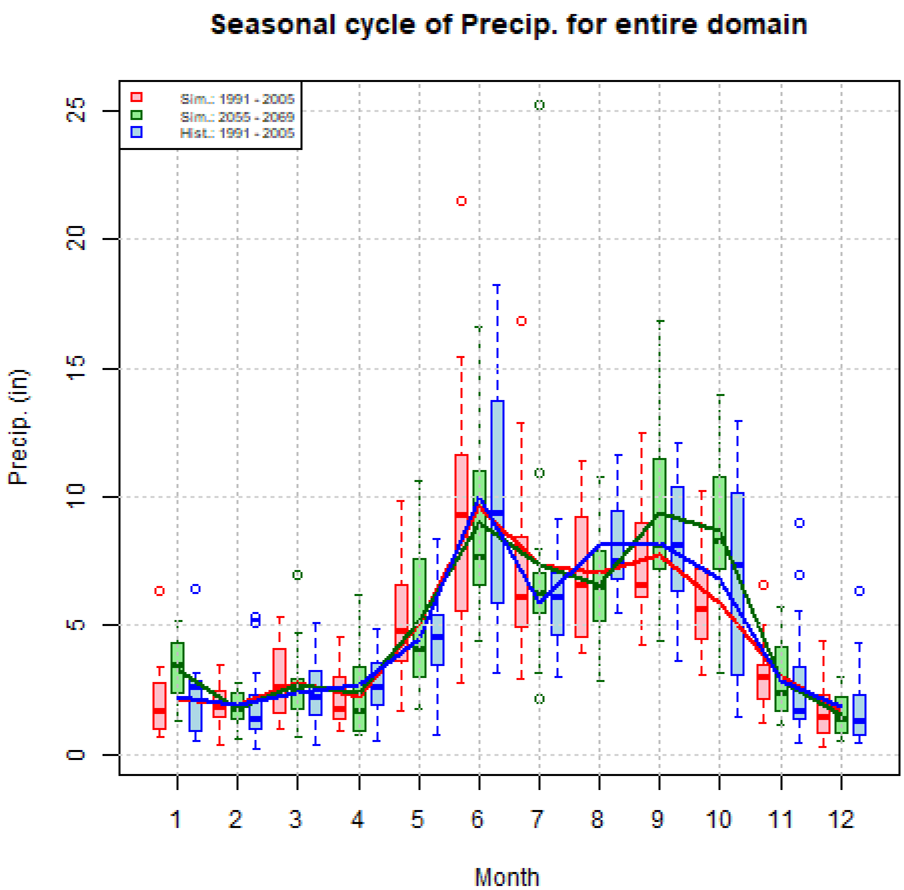


Figure 32. Seasonal cycle of domain-averaged monthly total precipitation for the historical period (1991-2005) and the future period (2055-2069). The historical SFWMM data is shown in blue, the simulated historical data for LOCA model pr_MRI-CGCM3_r1i1p1 is in red, and the simulated future data for LOCA model pr_MRI-CGCM3_r1i1p1_rcp85 is in green.

Quantile mapping (QM), a CDF matching method (Panofsky and Brier, 1968), has been typically applied to bias-correct precipitation timeseries from climate model simulations. To avoid some common limitations of QM, other methods have been developed such as Quantile Delta Mapping (QDM). As shown in Cannon et al. (2015), QM tends to inflate trends in precipitation extreme indices projected by GCMs, whereas QDM is not as prone to this problem. QDM preserves model-projected changes in quantiles, while simultaneously correcting for systematic biases across quantiles (Cannon et al., 2015). QDM also attempts to bridge the gap between point estimates for the observations vs. grid cell estimates in the model. However, it is important to note that changes in the mean may not be preserved by QDM.

QDM can be applied in additive form or multiplicative form. Multiplicative QDM (MQDM) is better suited to correcting variables like precipitation where preserving relative changes is important in order to respect the Clausius-Clapeyron equation which relates the amount of atmospheric moisture to temperature changes simulated by the models. MQDM was used to bias-correct daily precipitation data for LOCA run pr_MRI-CGCM3_r1i1p1_rcp85 at every MODFLOW model grid cell independently.

The Multiplicative QDM (MQDM) method is described by

$$\hat{x}_{m-padj.} = x_{m-p} * \{F_{o-c}^{-1}[F_{m-p}(x_{m-p})]/F_{m-c}^{-1}[F_{m-p}(x_{m-p})]\}$$

Equation 6

which is equivalent to:

$$\hat{x}_{m-padj.} = F_{m-padj.}^{-1}(G) = F_{m-p}^{-1}(G) * \{F_{o-c}^{-1}(G)/F_{m-c}^{-1}(G)\}$$

Equation 7

where $\hat{x}_{m-padj.}$ is the adjusted quantile for the LOCA model (m) projections (p) for the future period (2055-2069), F_{o-c} is the CDF of the SFWMM observations (o) in the current (1991-2005) baseline period (c), F_{m-c} is the CDF of the LOCA model (m) in the current (1991-2005) baseline period (c), F_{m-p} is the CDF for the LOCA model (m) projections (p) for the future period (2055-2069), and x_{m-p} is the quantile for the LOCA model (m) projections (p) in the future (2055-2069) baseline period. F^{-1} means the inverse of the CDF (i.e. the quantile function), *G* is the annual non-exceedance probability (CDF value) and is equal to *1-P*, *P* is the annual exceedance probability (AEP) which is related to the return period *T* by *1/P = T* (i.e. *G=1-1/T*). Figure 33 shows MQDM application for hypothetical data.

As part of this project, MQDM was applied to bias-correct future daily rainfall data for LOCA run pr_MRI-CGCM3_r1i1p1_rcp85 for each month of the year separately. In other words, twelve (12) CDFs were developed for each cumulative function shown in Equation 6 and Equation 7. The analysis was done for all model grid cells in the mainland. For active model grid cells in the barrier islands, data from the closest SFWMM or LOCA grid cell was used.

Figure 34 shows quantile-quantile plots of observed and simulated daily rainfall for the historical period (1991-2005) in the top panel, simulated historical (1991-2005) vs. future (2055-2069) projected daily rainfall in the middle panel, and the respective CDFs in the bottom panel, considering all daily values at a particular MODFLOW model grid cell together. As a reminder, MQDM was applied for different months of the year separately, but looking at all daily values lumped together is still helpful. It is evident that this LOCA run underestimates daily extremes at this location in the historical period, while it simulates an increase in future precipitation. Similar behavior is observed at most MODFLOW model grid cells in Miami-Dade County. Figure 35 shows how MQDM bias-correction fixes the extremes in the historical period (top panel), while still simulating increased precipitation in the future period (middle panel). The top panel in this figure shows the daily data points almost exactly on the 1:1 line; however, since MQDM is applied separately for different months of the year, there may be some small excursions from the 1:1 line at other model grid cells when plotting all daily values together.

Figure 36 shows the spatial distribution of the observed and simulated means prior to and post daily bias-correction using Equation 6. It is evident how daily bias correction corrects the mean historical spatial pattern to match that of the observed data (top 2 panels). The bias-corrected future precipitation (Figure

36d) has a similar spatial pattern to that of the historical period, as expected. The spatial pattern in Figure 36d is very similar to the case when the long-term mean was bias-corrected (Figure 31d). However, the magnitude of the rainfall is larger in the daily bias-correction case (63.9 in/yr or 11% higher than historical) than in the mean bias-correction case (61.9 in/yr or 7.5% higher than historical) due to non-linearities in the daily bias-correction process. Figure 37 shows the same maps as in Figure 36 but including the barrier islands.

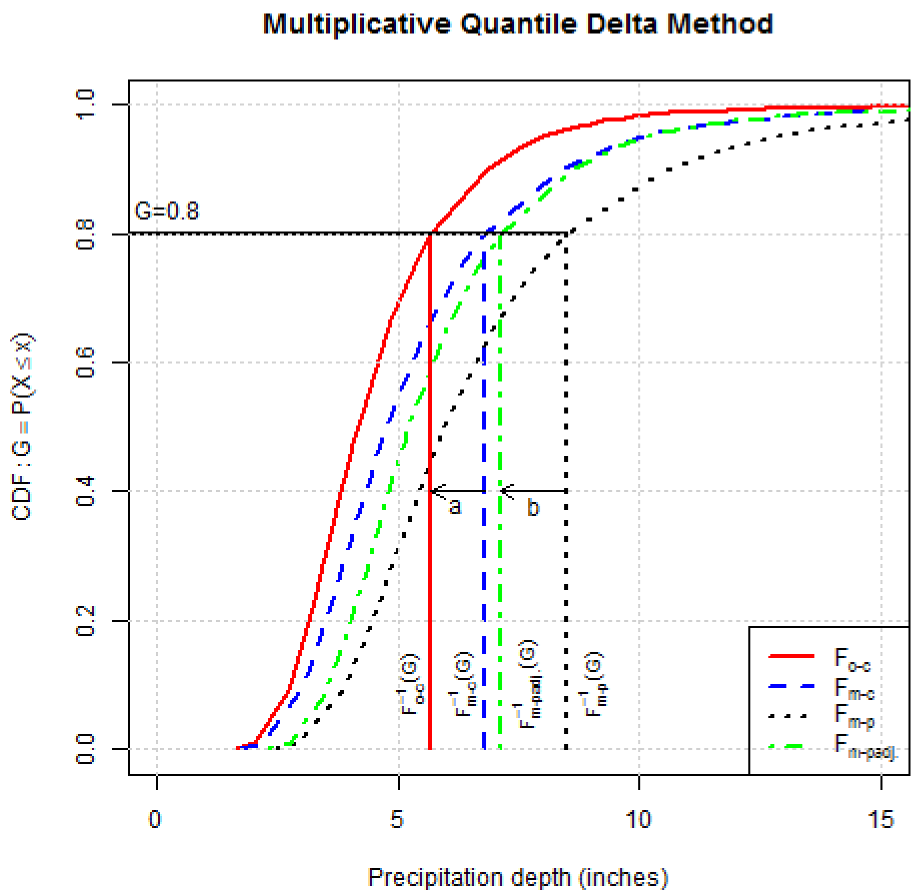


Figure 33. Diagram showing the Multiplicative Quantile Delta Method for hypothetical data.

F is the non-exceedance probability of interest. The quantiles corresponding to F are given by CDF1^{-1}: $F^{-1}_{o-c}(F)$ for the observed current baseline, CDF2^{-1}: $F^{-1}_{m-c}(F)$ for the model current baseline, CDF3^{-1}: $F^{-1}_{m-p}(F)$ for the model projected (future) period. The corresponding adjusted quantile for the model projected (future) period is CDF4^{-1}: $F^{-1}_{m-p\ adjust}(F) = F^{-1}_{m-p}(F)*\{F^{-1}_{o-c}(F)/F^{-1}_{m-c}(F)\}$. The distances a and b are different in MQDM due to the use of a ratio in the bias correction equation. However, a and b would be equal in Additive Quantile Delta Method.

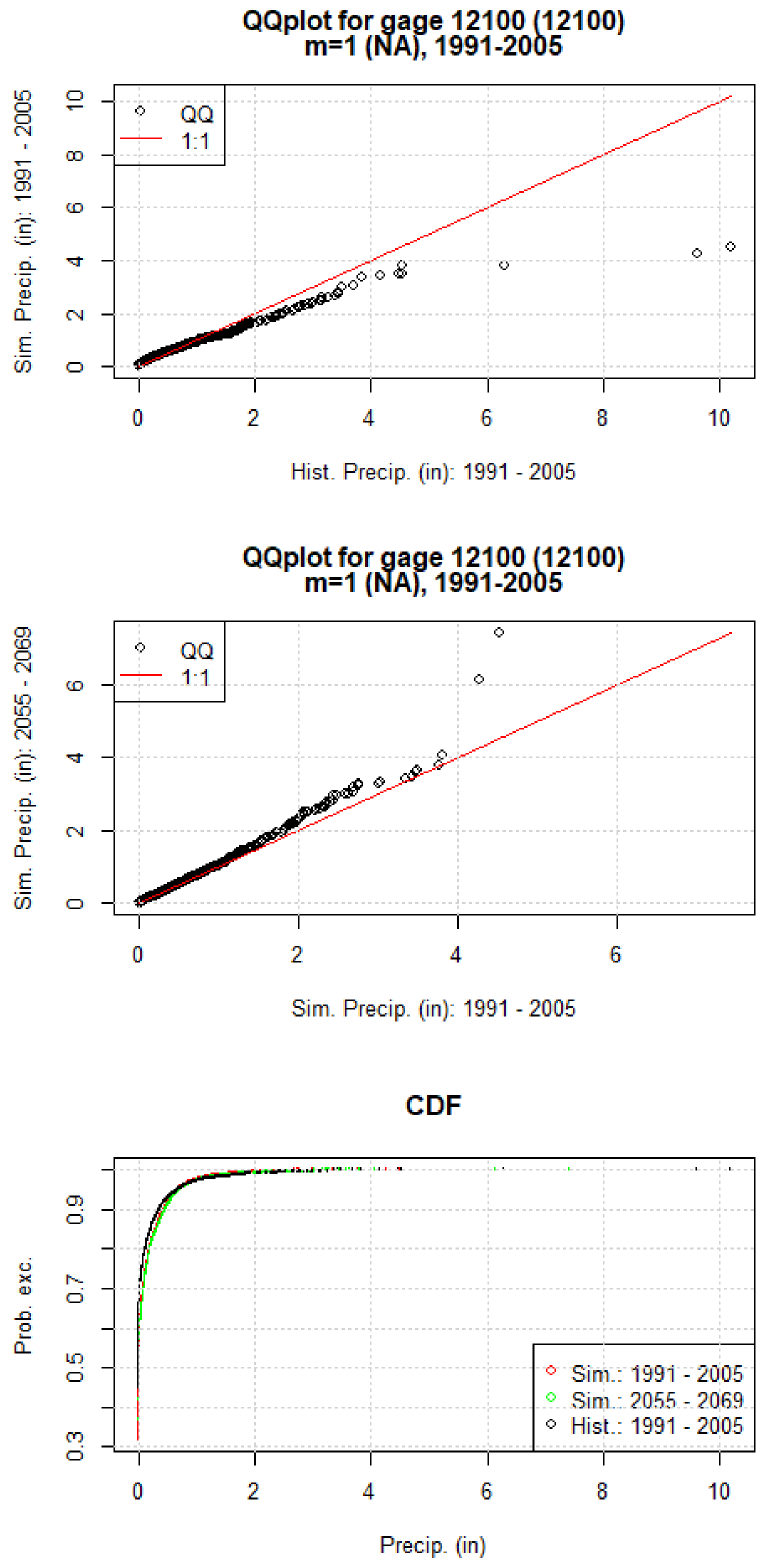


Figure 34. Pre-bias correction: (a) Quantile-quantile plot of daily observed vs. simulated precipitation for the historical period (1991-2005), (b) Quantile-quantile plot of simulated historical (1991-2005) vs. simulated future (1991-2005) precipitation, (c) Cumulative distribution functions.

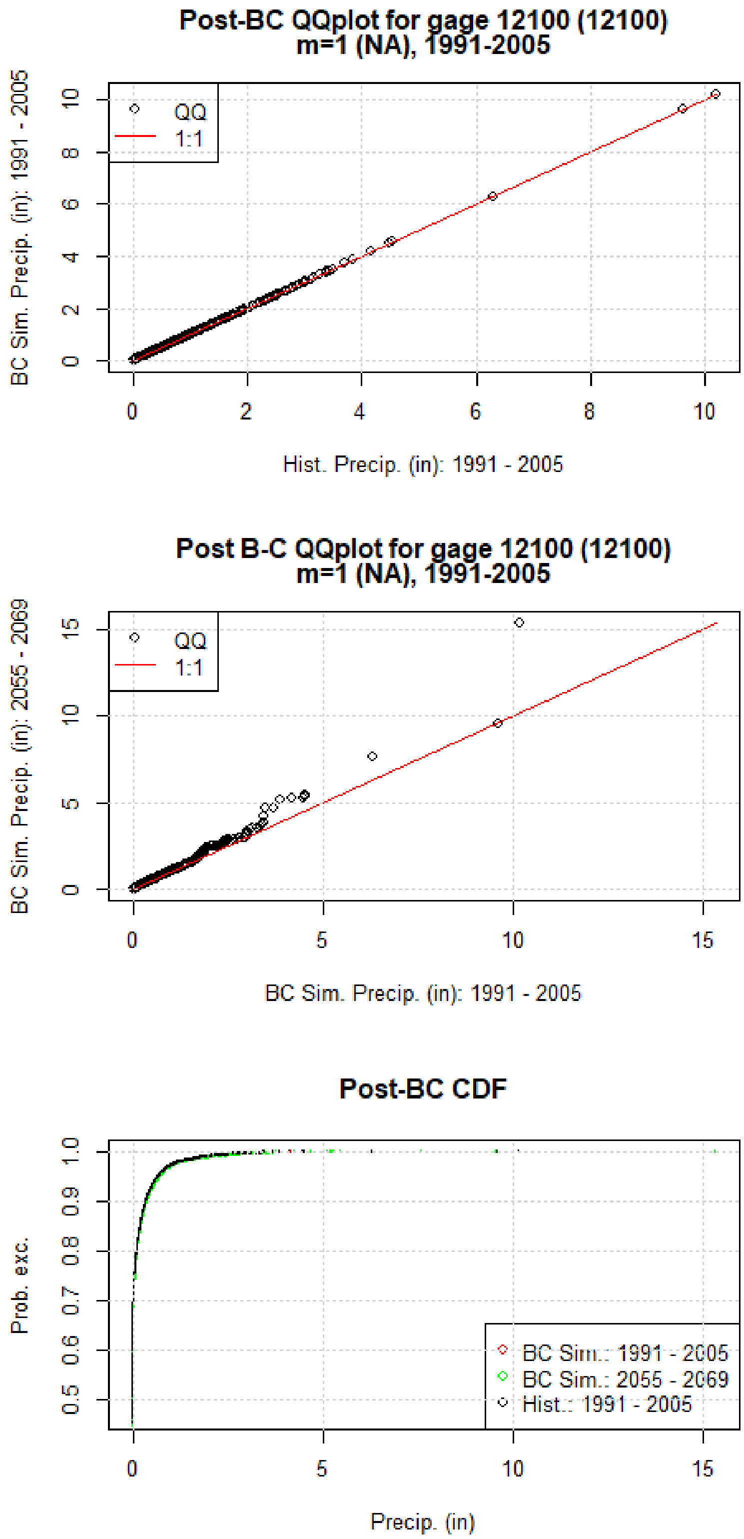
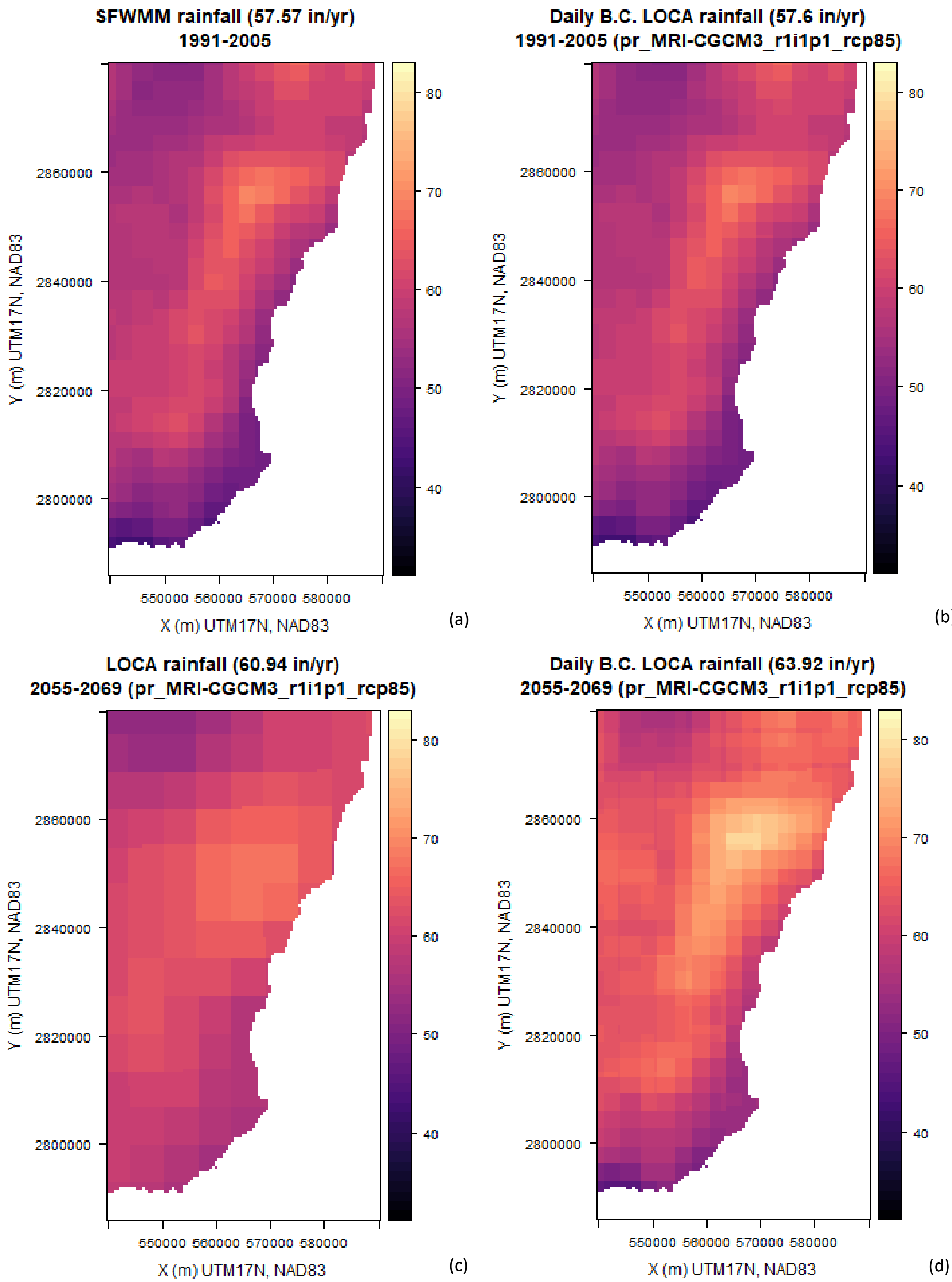
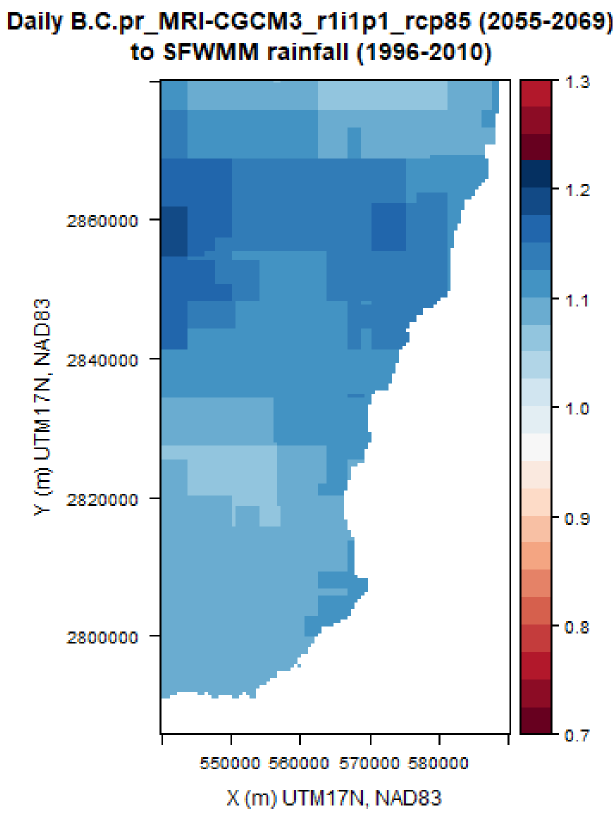
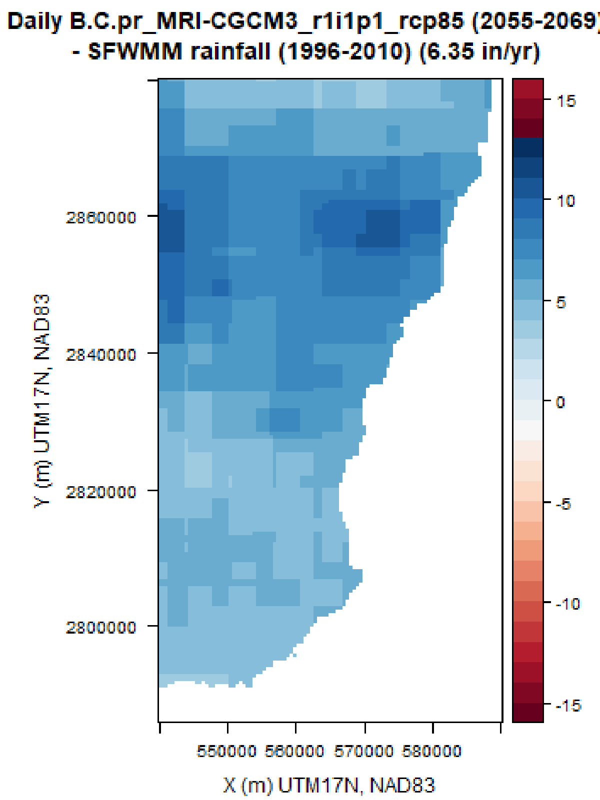


Figure 35. Pre-bias correction: (a) Quantile-quantile plot of daily observed vs. simulated precipitation for the historical period (1991-2005), (b) Quantile-quantile plot of simulated historical (1991-2005) vs. simulated future (1991-2005) precipitation, (c) Cumulative distribution functions.


SFWMM rainfall (57.57 in/yr)
1991-2005
Daily B.C. LOCA rainfall (57.6 in/yr)
1991-2005 (pr_MRI-CGCM3_r1i1p1_rcp85)
LOCA rainfall (60.94 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
Daily B.C. LOCA rainfall (63.92 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
2800000
2820000
2840000
2860000
550000 560000 570000 580000
40
50
60
70
80
(a)
(b)
(c)
(d)

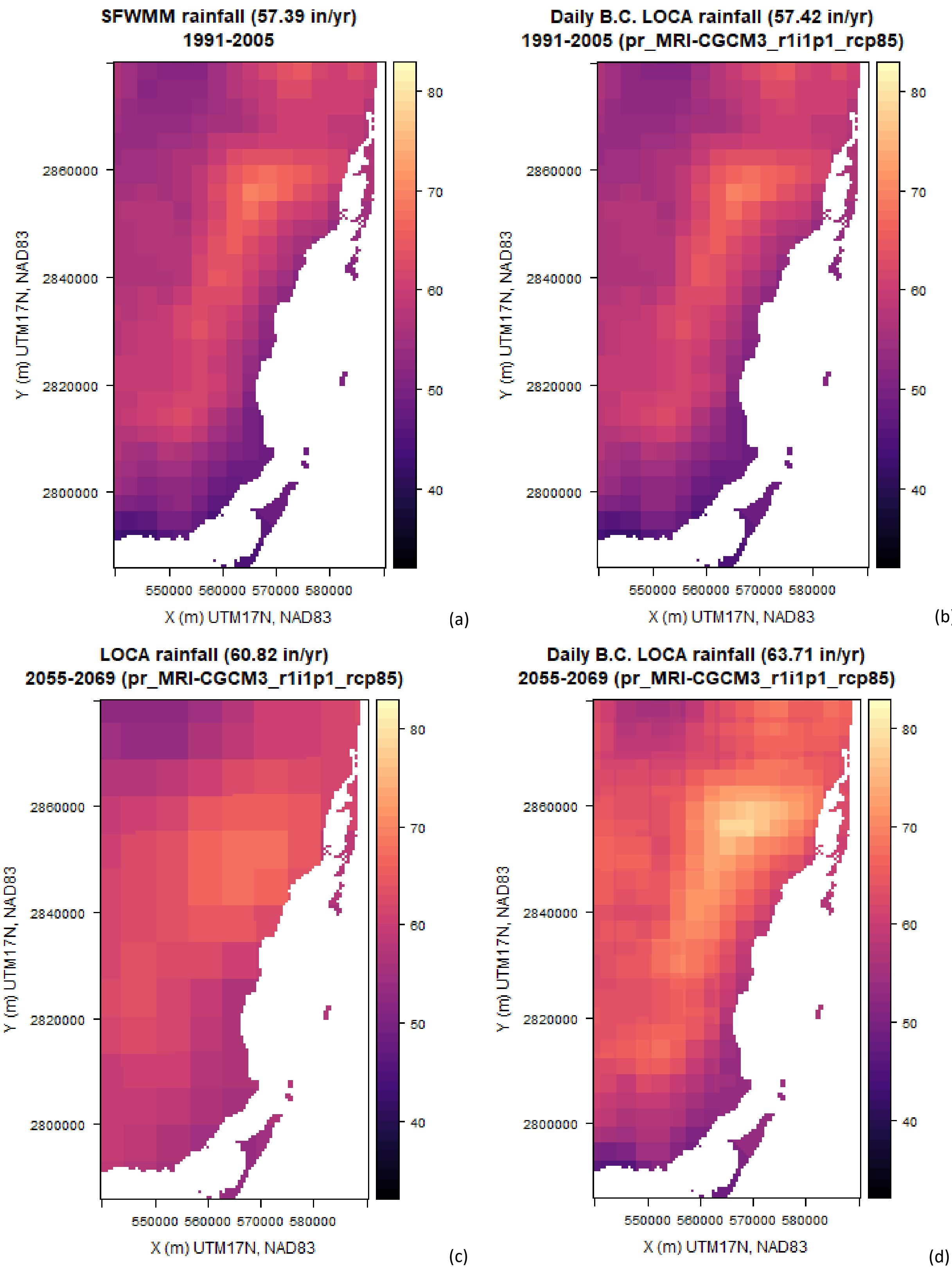


(e)



(f)

Figure 36. Spatial distribution of (a) SFWMM historical rainfall (1991-2005), (b) Bias-corrected simulated historical rainfall (1991-2005), (c) Simulated future rainfall (2055-2069), (d) Bias-corrected simulated future rainfall, (e) = (d)/(a), (f) = (d) - (a). Note: Here, bias correction has been done on the annual total rainfall (mean) by applying Equation 6.


SFWMM rainfall (57.39 in/yr)
1991-2005
Y (m) UTM17N, NAD83
X (m) UTM17N, NAD83
2860000
2840000
2820000
2800000
550000 560000 570000 580000
80
70
60
50
40
(a)
Daily B.C. LOCA rainfall (57.42 in/yr)
1991-2005 (pr_MRI-CGCM3_r1i1p1_rcp85)
(b)
LOCA rainfall (60.82 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
(c)
Daily B.C. LOCA rainfall (63.71 in/yr)
2055-2069 (pr_MRI-CGCM3_r1i1p1_rcp85)
(d)

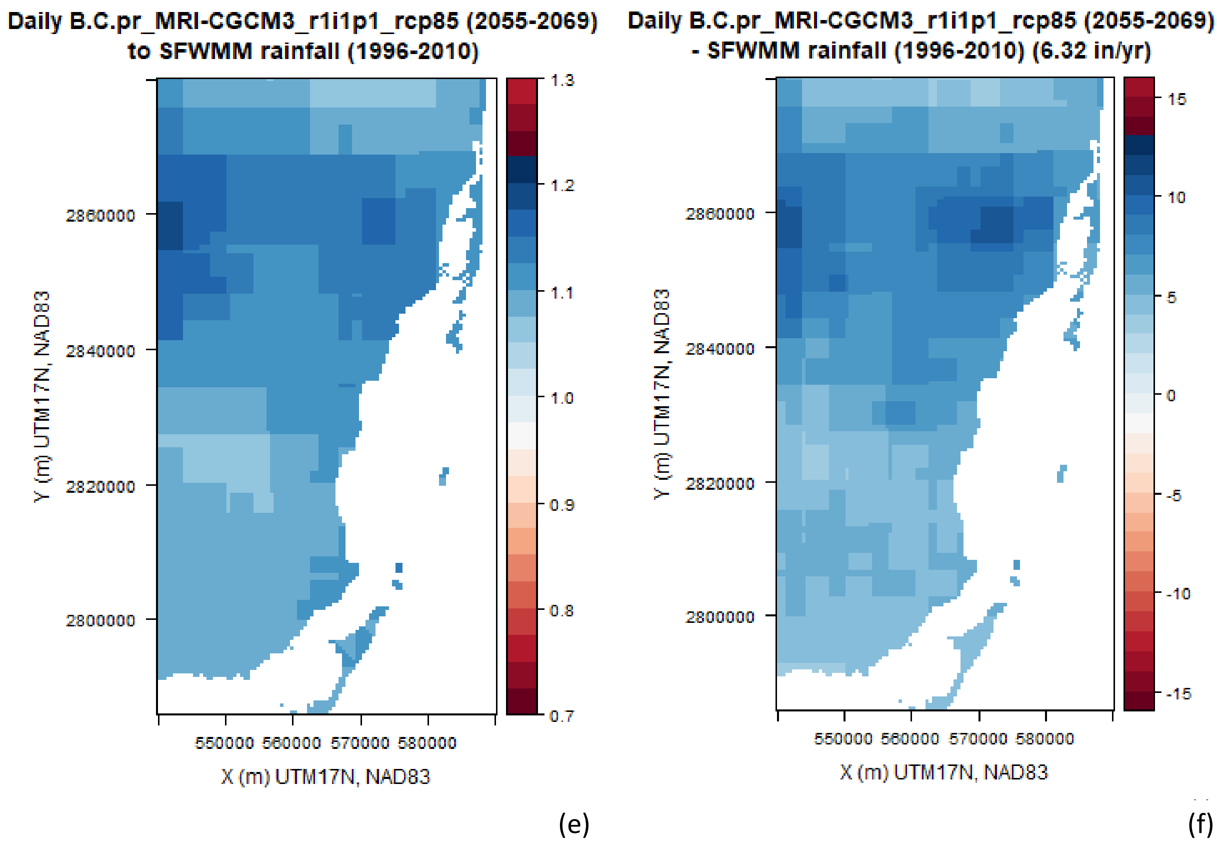


Figure 37. As in Figure 36, but including the barrier islands.

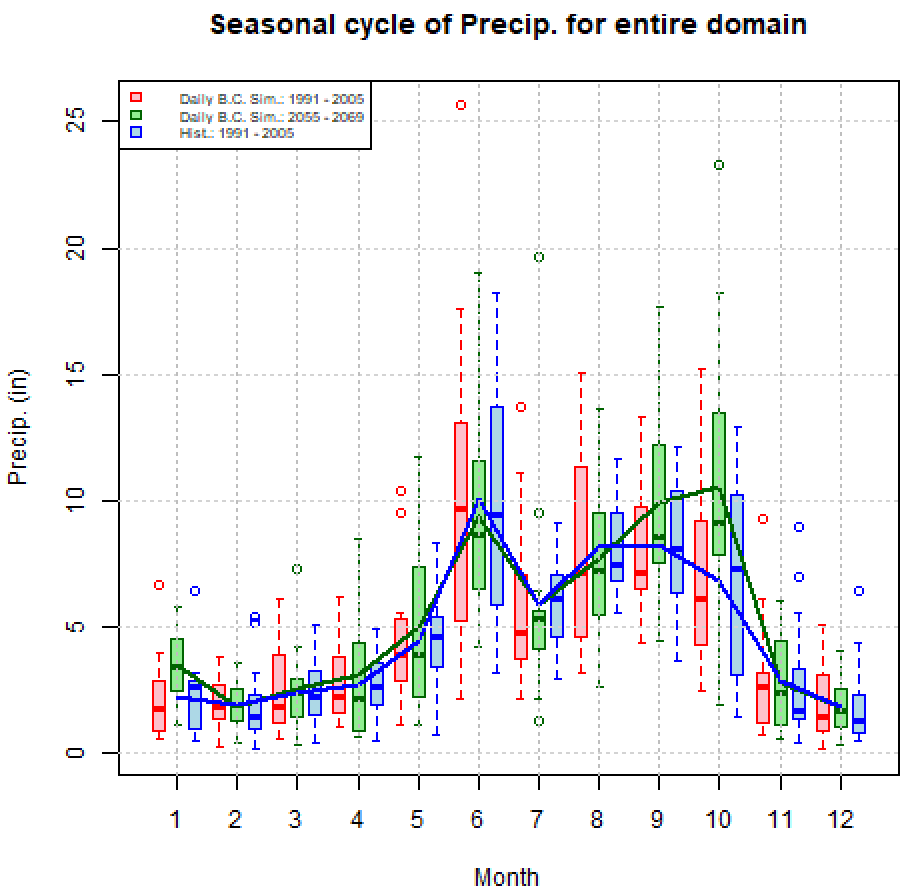


Figure 38. Seasonal cycle of domain-averaged monthly total precipitation for the historical period (1991-2005) and the future period (2055-2069). The historical SFWMM data is shown in blue, the simulated historical data for LOCA model pr_MRI-CGCM3_r1i1p1 after daily bias-correction is in red, and the simulated future data for LOCA model pr_MRI-CGCM3_r1i1p1_rcp85 after daily bias correction is in green.

Figure 38 shows the seasonal cycle of domain-average rainfall after daily bias-correction for the historical and future periods. It is evident how daily-bias correction makes the domain-average simulated monthly mean rainfall match that of the observations. However, the method is not always able to improve the monthly variability. For example, the variability of the bias-corrected historical simulated rainfall improves during the month of June, but is even larger than prior to bias-correction during August and October (Figure 32). This is due to the fact that the daily MQDM bias-correction method corrects only one grid cell at a time without considering the spatial and temporal variability of rainfall. This limitation is addressed by other bias-correction methods such as Bias-Corrected Stochastic Analogs (BCSA; Hwang and Graham, 2013). Comparison of Figure 38 with Figure 32 shows that the pattern of increased future precipitation at the end of the wet season (September-October) and slight decreases in January and August precipitation remain after daily bias-correction.

Future Everglades water levels

Future water levels in the Everglades are expected to be different from historical due to implementation of the Comprehensive Everglades Restoration Plan (CERP) and possibly due to increased rainfall as a result of climate change. For that purpose, simulated water levels in the Everglades for two modeling scenarios simulated by the South Florida Water Management Model (SFWMM) were evaluated: (1) the updated full-CERP implementation (CERP0 scenario; Figure 39), which uses projected future land use, historical rainfall, and includes CERP restoration components such as partial decompartmentalization of Water Conservation Area 3 (WCA3) and Everglades National Park (ENP), Water Preserve Areas (Lakebelt Storage), etc., and (2) A current baseline scenario with 2010 land use and a 10% increase in rainfall.

The stage data for these two modeling scenarios were obtained from the South Florida Water Management District (J. Barnes, pers. comm.). The stage data was provided at all SFWMM 2-mile x 2-mile grid cells across the MODFLOW model domain, and interpolated to the MODFLOW model resolution using bilinear interpolation tool from R's {akima} package (Appendix C. R code for calculating average Everglades water levels by Julian day). Due to the coarse resolution of the SFWMM, the boundary between WCA-3B/ENP and the urban areas east of the Lower East Coast (LEC) protective levee is represented by the SFWMM in a way that water levels in the LEC were affecting interpolated stages in MODFLOW cells representative of WCA3/ENP. Therefore, as shown in Figure 40, MODFLOW grid cells located in WCA3/ENP, but east of the SFWMM representation of the WCA3/ENP-LEC boundary (blue line on this figure) were assigned the interpolated stage data from the closest MODFLOW grid cell west of the SFWMM representation of the WCA3/ENP-LEC boundary.

Figure 41 shows the average historical stages used in the original USGS MODFLOW model for the calibration/verification period 1996-2010 (ft NGVD29). The data is from the Everglades Depth Estimation Network (EDEN) database (USGS, 2012) and was interpolated to the MODFLOW model grid using bi-linear interpolation as described in Hughes and White (2016). Figure 42 and Figure 43 show the average simulated stages for the CERP0 and the 10% increased rainfall scenarios, respectively.

Figure 44 shows the differences between the average simulated stages in the two SFWMM scenarios. The differences reflect not only changes in rainfall, but also differences in the configuration of the water resource management system. For example, the reduction in stages in WCA-3A and WCA-3B, and the increased stages in northeastern Everglades National Park are a result of the partial Decompartmentalization component of CERP0. Figure 45 and Figure 46 show the differences between the simulated stages in the two scenarios (1965-2005) and the historical stages from the EDEN network (1996-2010). A SFWMM scenario run with both CERP0 features and 10% increased rainfall (CERP0+10%RF) performed by Obeysekera et al. (2010; Figure 15 in that publication) showed only slightly increased average water levels in the Everglades and Water Conservation Areas in the order of about 0.1 ft or less. This small difference is likely due to CERP0 management features, such as rain-driven operations, being able to compensate for the rainfall increase and move water out of the system. Output from this CERP0+10%RF run was not available; therefore, water levels from the CERP0 scenario with historical rainfall was chosen for inclusion in future (2055-2069) modeling scenarios. It is notable that the CERP0 simulation assumes historical predicted tides as boundary conditions and does not reflect the expected increases in sea level rise in the future. This is a limitation of using this model run to provide boundary conditions for this modeling effort.

The CERP0 modeling scenario simulates 1965-2005 water levels based on historical rainfall. However, the future modeling scenarios that will be simulated using the Miami-Dade MODFLOW model as part of this project encompass the period 2055-2069 (with the first 5 years thrown out). Therefore, a decision was made to use average simulated water levels from the CERP0 run for each day of the year (1-365) at each Everglades/WCA model grid cell, and repeat this 365-day timeseries for every year of the future simulation period (2055-2069). This would provide a more reasonable future Everglades water level boundary condition than simply using the historical stage time series.

The resulting timeseries of water levels at each model grid cell was converted from ft NGVD29 to m NAVD88, which are the vertical datum and units used in the MODFLOW model. The U.S. Army Corps of Engineers Corpscon 6.0.1 software with vertcon05 was used to obtain offsets from ft NGVD29 to ft NAVD88 at each MODFLOW model grid cell (Figure 47). The resulting offsets were corroborated by comparison against offsets published by the South Florida Water Management District (https://www.arcgis.com/home/item.html?id=4ffd84bc93ce4862bcd642bdb023668e). The final dataset was saved in netCDF format in R and read by a Python notebook to generate daily binary files of water levels in the format required by the model (e.g. CERP0_stage_20101209.bin).

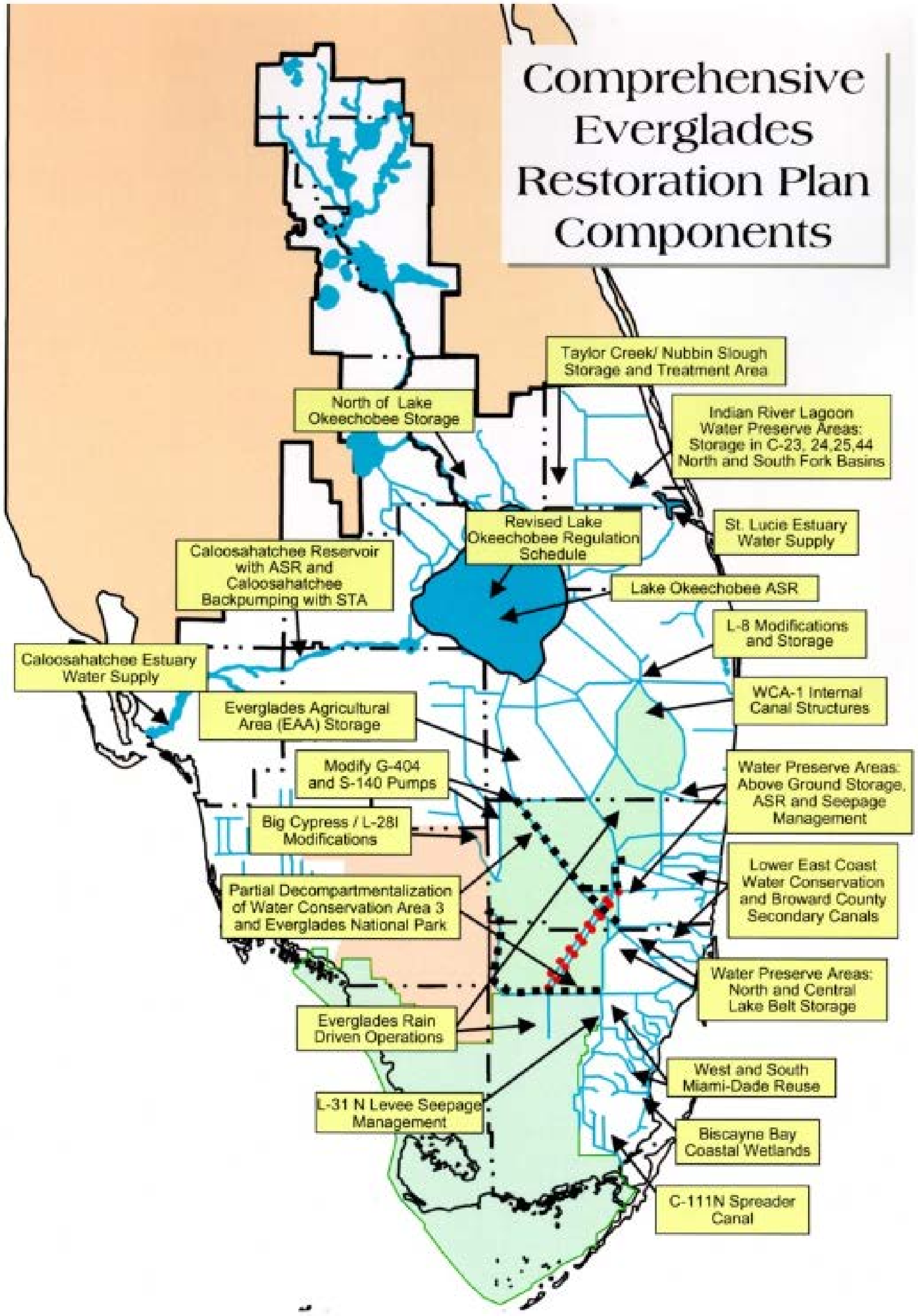


Figure 39. Comprehensive Everglades Restoration Plan components. Source: https://www.sfwmd.gov/sites/default/files/documents/roq_scenariodev_2010_0127.pdf

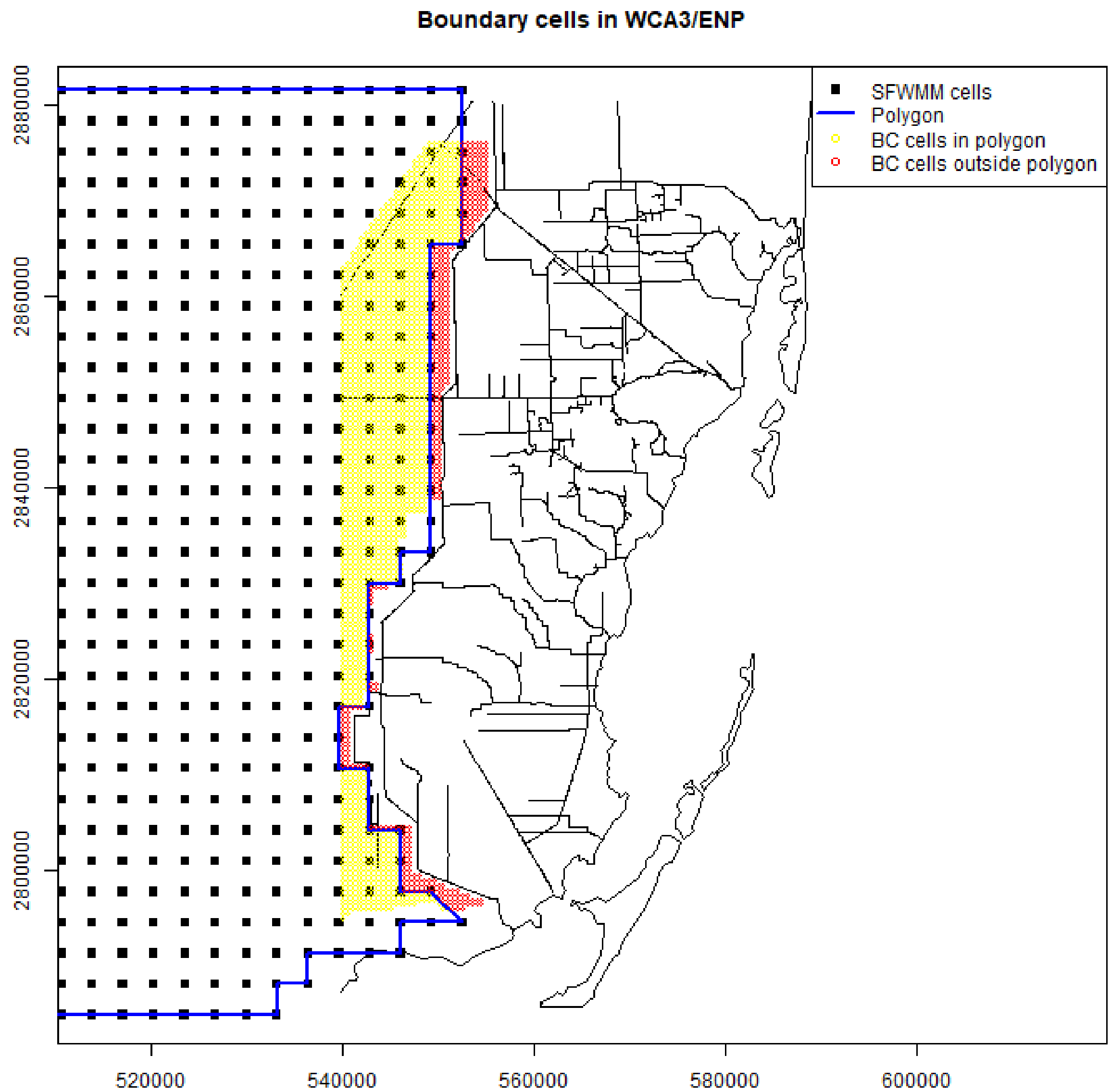


Figure 40. Boundary MODFLOW cells in WCA3/ENP. Stage data for MODFLOW cells in yellow were interpolated from data at SFWMM cells representing WCA3/ENP (SFWMM cells inside or touching the polygon shown by the blue line). MODFLOW cells in red were assigned the interpolated value for the closest MODFLOW cell in yellow. X and Y coordinates in meters, UTM17N, NAD83.

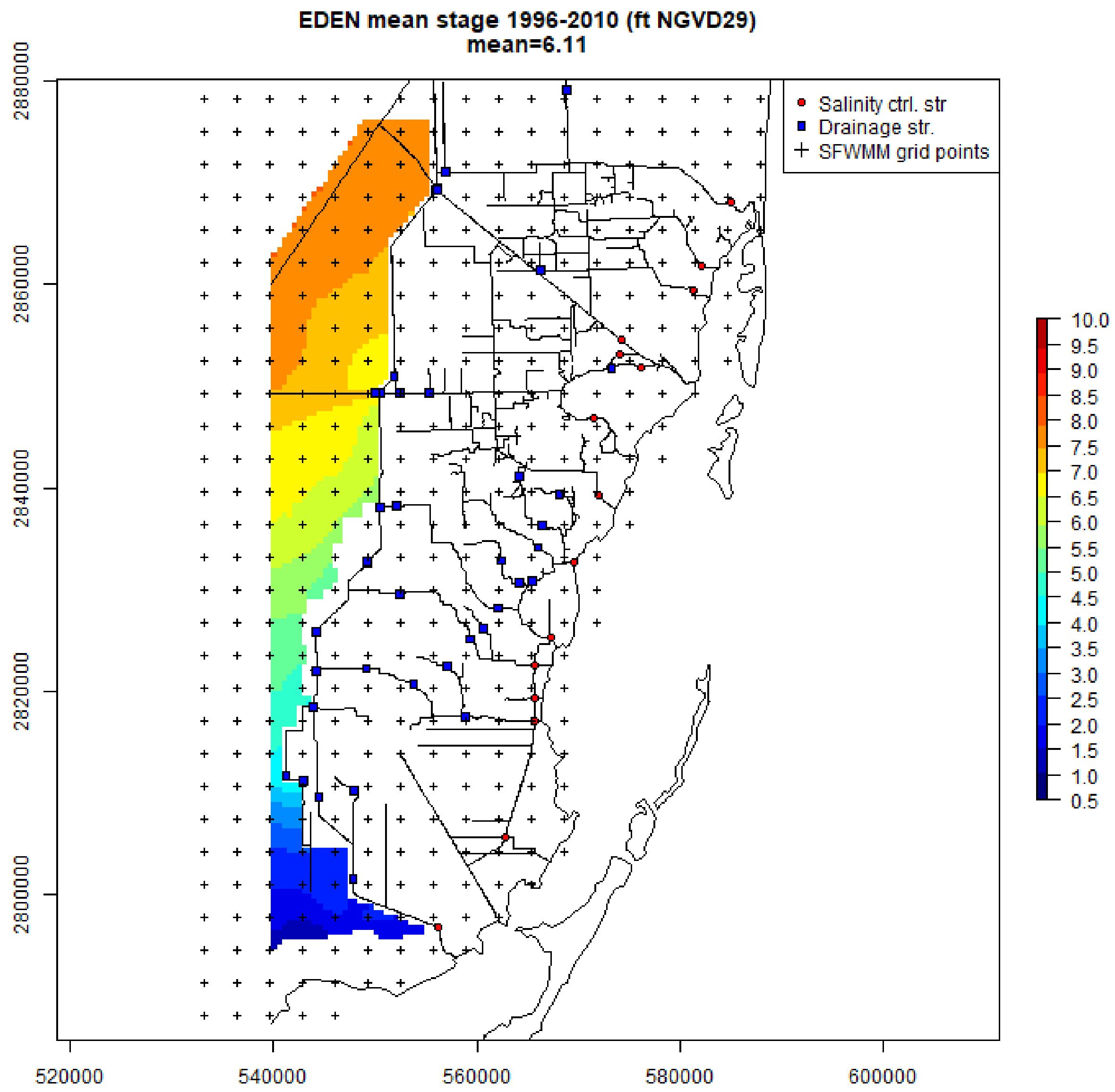


Figure 41. Average historical stages for 1996-2010 based on data from the EDEN network. X and Y coordinates in meters, UTM17N, NAD83.

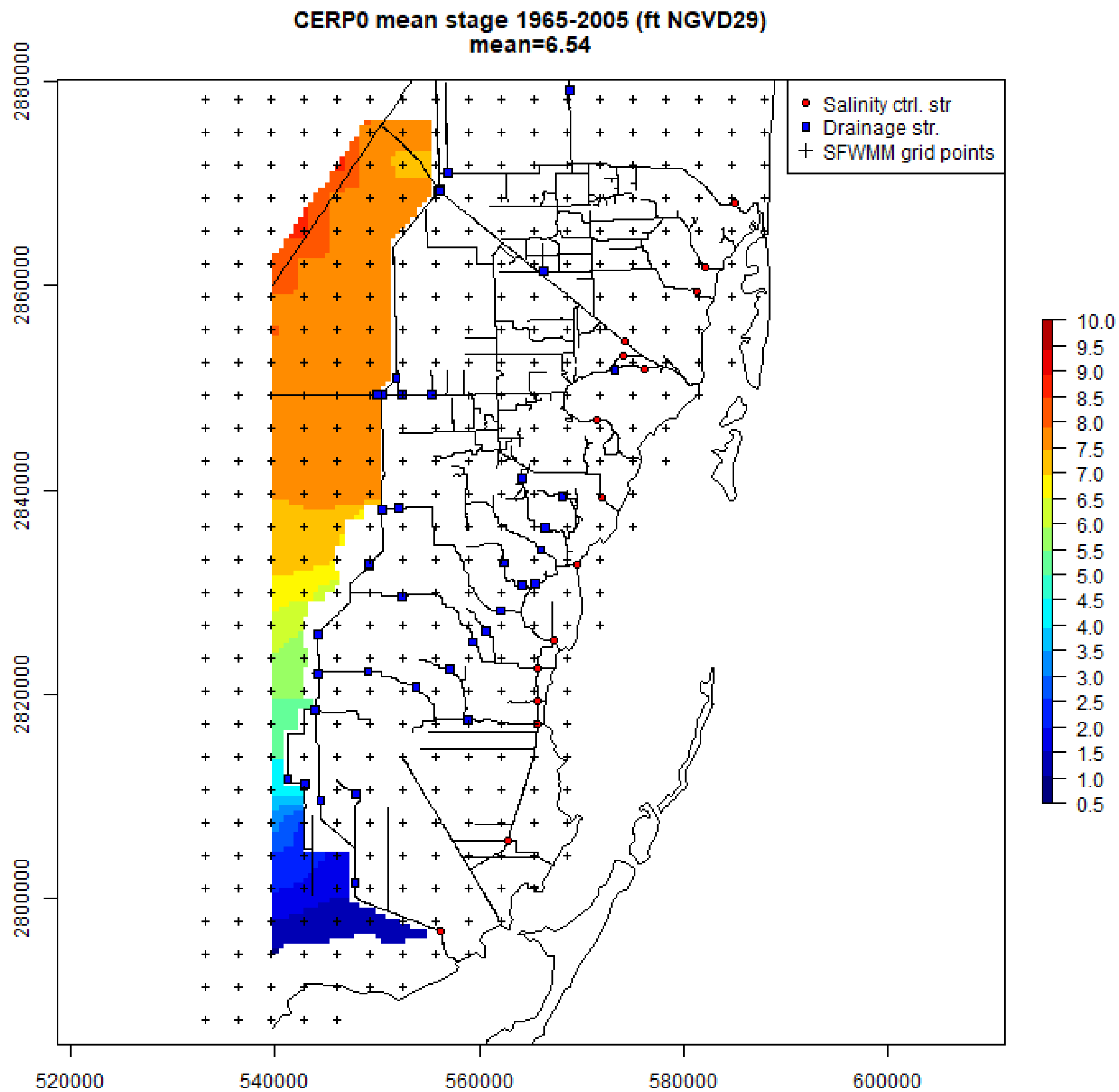


Figure 42. Average simulated stages for the CERP0 scenario (1965-2005). X and Y coordinates in meters, UTM17N, NAD83.

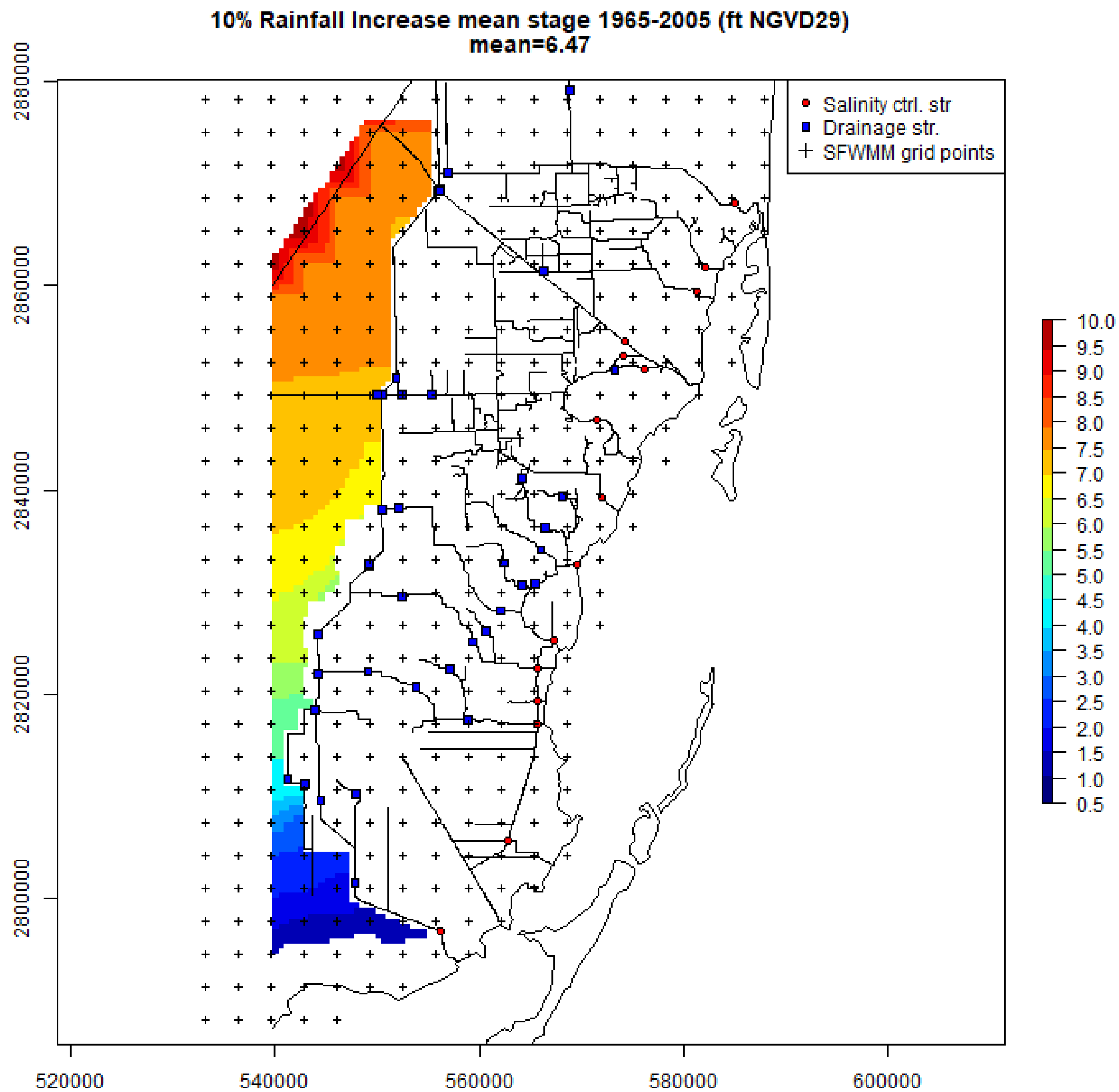


Figure 43. Average simulated stages for the 10% increased rainfall scenario (1965-2005). X and Y coordinates in meters, UTM17N, NAD83.

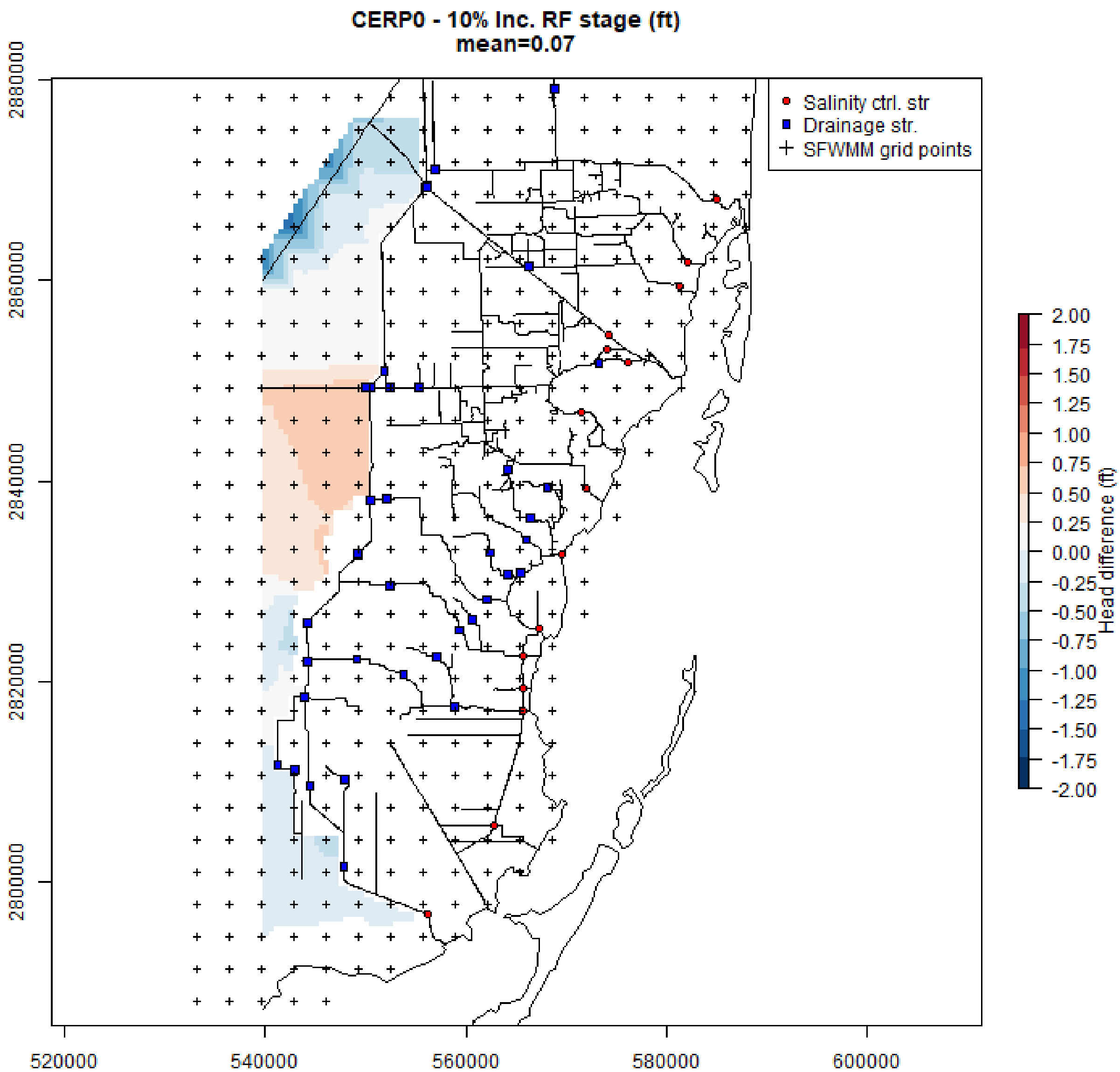


Figure 44. Differences in average simulated stages between the CERP0 scenario and 10% increased rainfall scenario (1965-2005). X and Y coordinates in meters, UTM17N, NAD83.

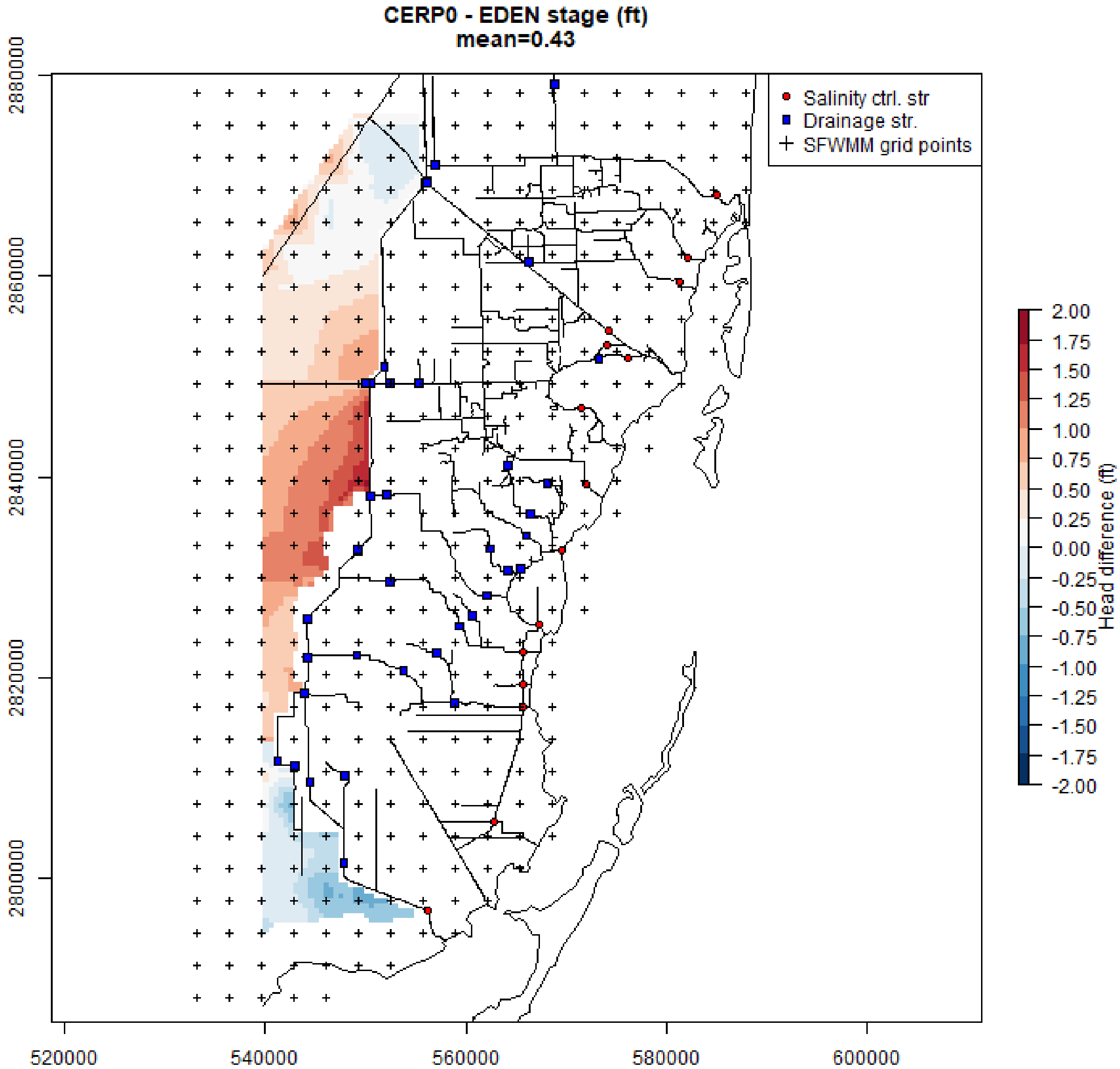


Figure 45. Differences between simulated stages in the CERP0 scenario (1965-2005) and the EDEN dataset (1996-2010). X and Y coordinates in meters, UTM17N, NAD83.

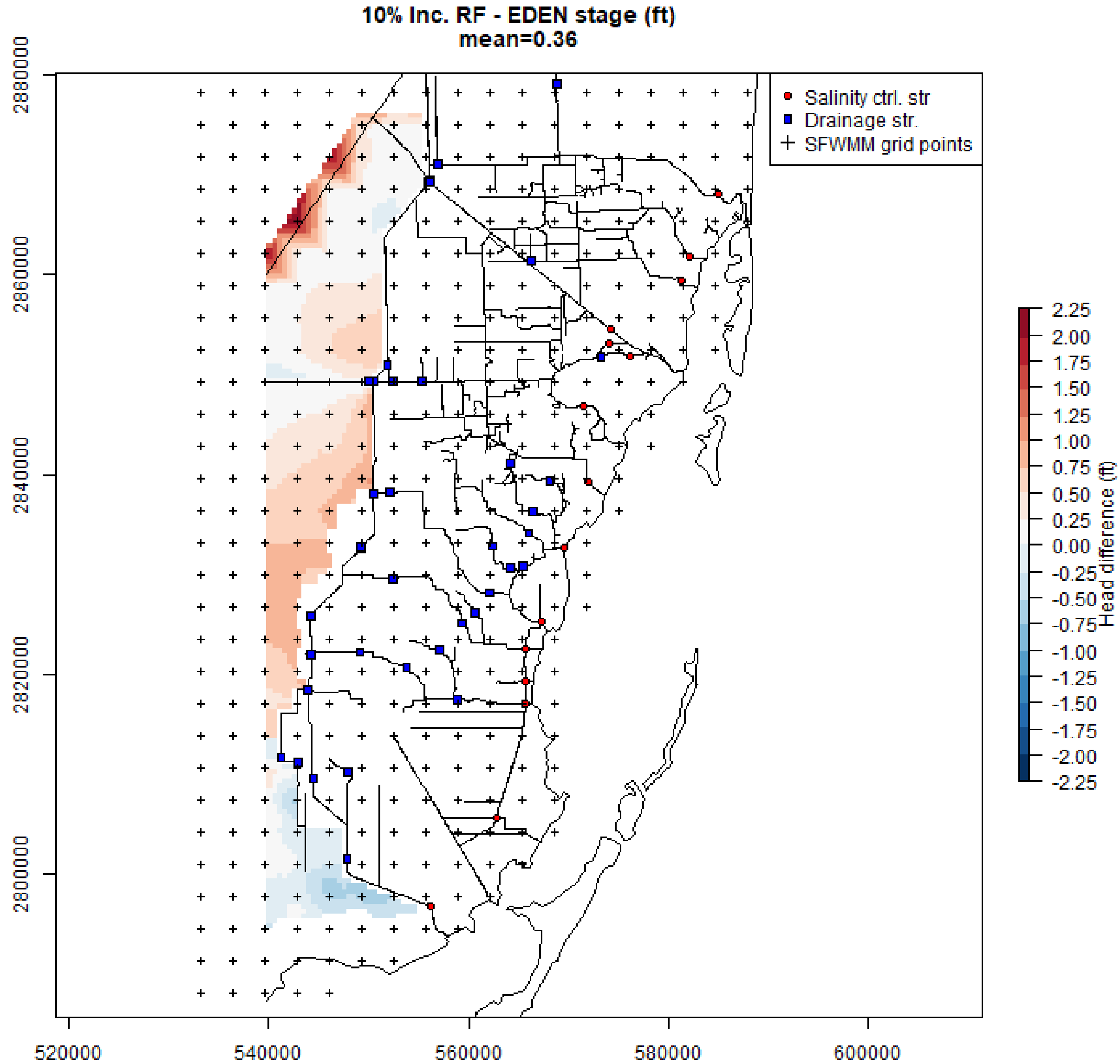


Figure 46. Differences between simulated stages in the 10% increased rainfall scenario (1965-2005) and the EDEN dataset (1996-2010). X and Y coordinates in meters, UTM17N, NAD83.

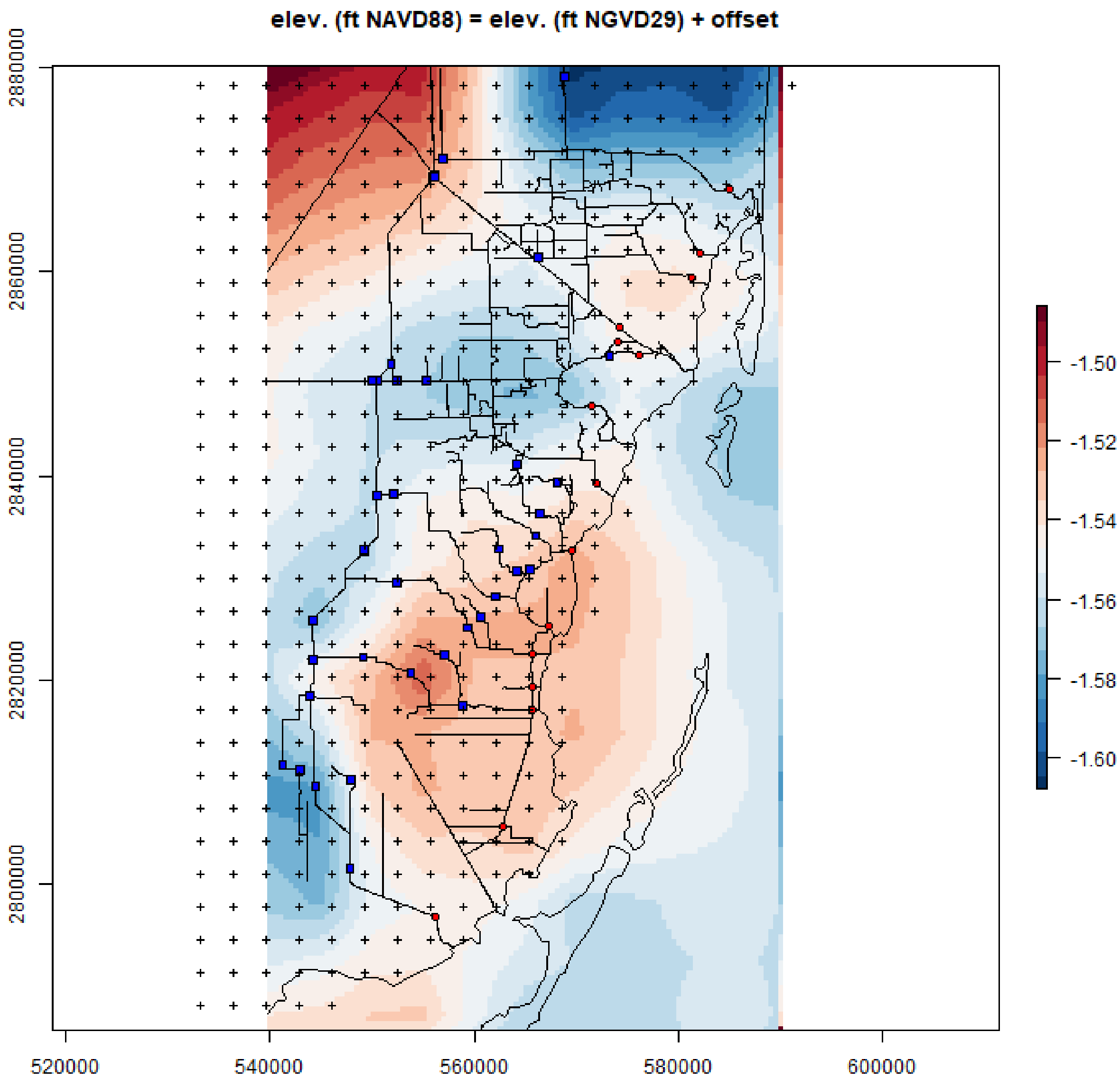


Figure 47. Datum shift offsets for the MODFLOW model domain (elevation in ft NAVD88 = elevation in ft NGVD29 + offset). X and Y coordinates in meters, UTM17N, NAD83.

Future freshwater/saltwater source

The saltwater intrusion package (SWI2) in the Miami-Dade MODFLOW model requires input of an array called isource, which defines the density of sources and sinks in each model grid cell. In addition, the isource array is used by a boundary condition pre-processing script in conjunction with the variable ibound (Appendix D. Description of boundary condition file (ibound)) to define whether each model grid cell is computationally active or is assigned a GHB or drain boundary condition.

In this model, only two density zones are simulated: zone 1 is the freshwater zone, while zone 2 is the seawater zone. "If isource > 0, sources and sinks have the same fluid density as the zone given by isource. If such a source is not present in the cell, then sources and sinks have the same fluid density as the active zone at the top of the aquifer. When isource = 0, sources and sinks have the same fluid density as the active zone at the top of the aquifer. When isource < 0, source have the same fluid density as the zone with a number equal to |isource|, while sinks have the same fluid density as the active zone at the top of the aquifer. This option is used when simulating the ocean bottom where infiltrating water is salt while exfiltrating water is of the same type as water at the top of the aquifer." (Bakker et al., 2013).

In the Miami-Dade MODFLOW model, land areas are assigned an isource of 0, ocean areas are given an isource of -2, while the Turkey Point cooling canals are given an isource of 2 to reflect that water in the cooling canals is currently as dense as seawater (Figure 48). Analysis of the calibration and scenario runs performed by the USGS shows that ocean areas, which are given an isource of -2, are defined based on the mean sea level during the last year of the simulation period. Therefore, a similar approach was followed in developing isource arrays for the two future sea level rise scenarios to be modeled as part of this project. That is, ocean areas are defined based on the predicted mean sea level at 2069.

Figure 49 shows the model topography and extent of flooding for the IPCC AR5 RCP8.5 SLR scenario based on a 2069 MSL of 0.47 ft NAVD88. A total of 1,290 additional cells are below the 2069 MSL beyond the 1,246 cells that are flooded based on the 1983-2001 National Tidal Datum Epoch of 1983—2001. In defining the ocean cells (Figure 50) for this scenario, a total of 44 isolated cells below sea level and cells near the Lower East Coast protective levee were excluded. Figure 51 shows the model topography and extent of flooding for the USACE High SLR scenario based on a 2069 MSL of 1.79 ft NAVD88. A total of 2,231 additional cells are below the 2069 MSL beyond the 1,246 cells that are flooded based on the 1983-2001 National Tidal Datum Epoch of 1983—2001. In defining the ocean cells (Figure 52) for this scenario, a total of 42 isolated cells below sea level and cells near the Lower East Coast protective levee were excluded as well. Eight (8) additional cells below the 2069 MSL that are located inside the Turkey Point power plant cooling canals, also keep the original isource value of +2. The assumption is that the power plant and its cooling canals will be protected by levees in the future.

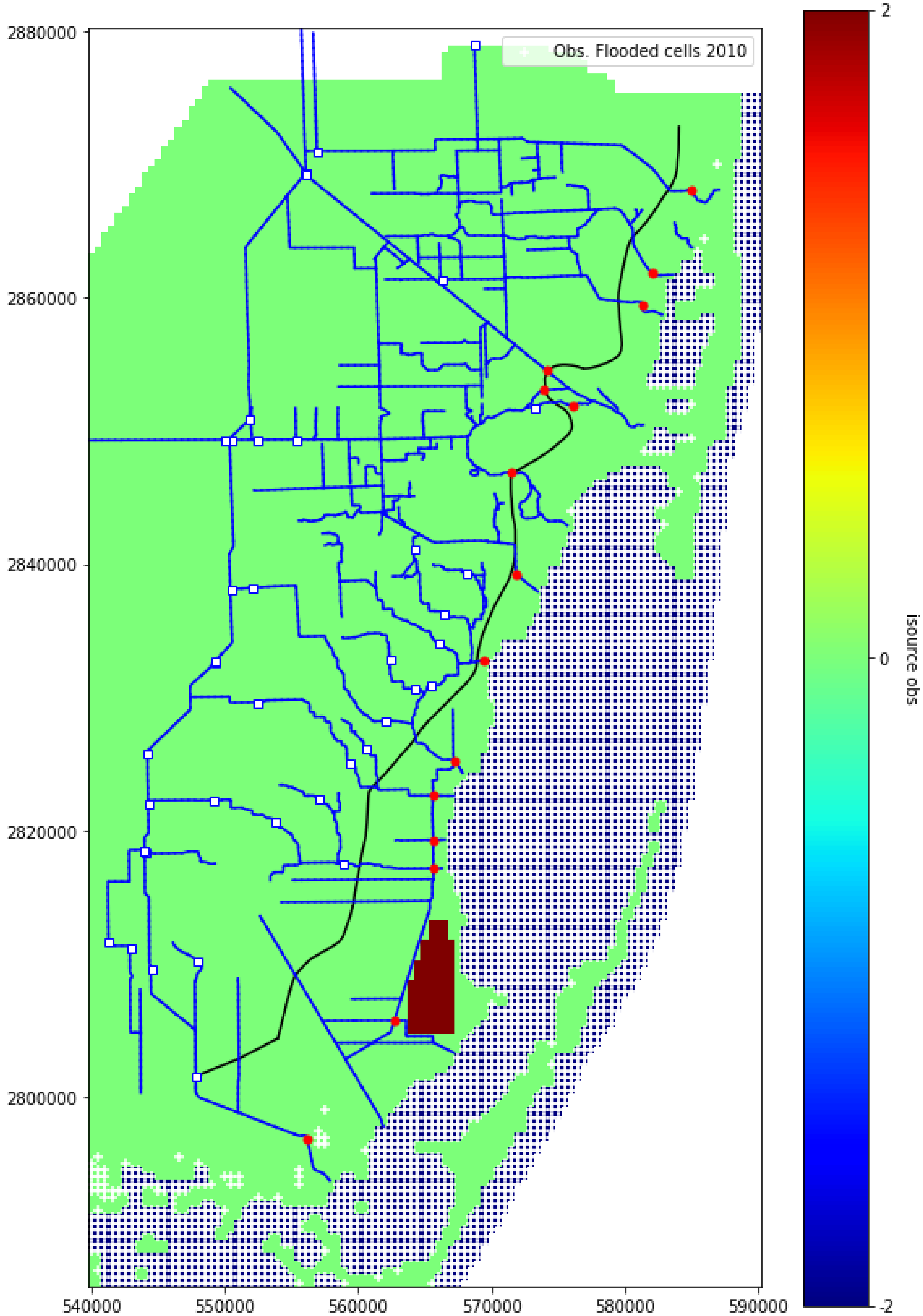


Figure 48. Freshwater/saltwater source (isource) for the 1996-2010 calibration run. Cells marked with white '+' are below the 2010 historical mean sea level at Virginia Key and their isource value is (for the most part) equal to -2 (blue). X and Y coordinates in meters, UTM17N, NAD83.

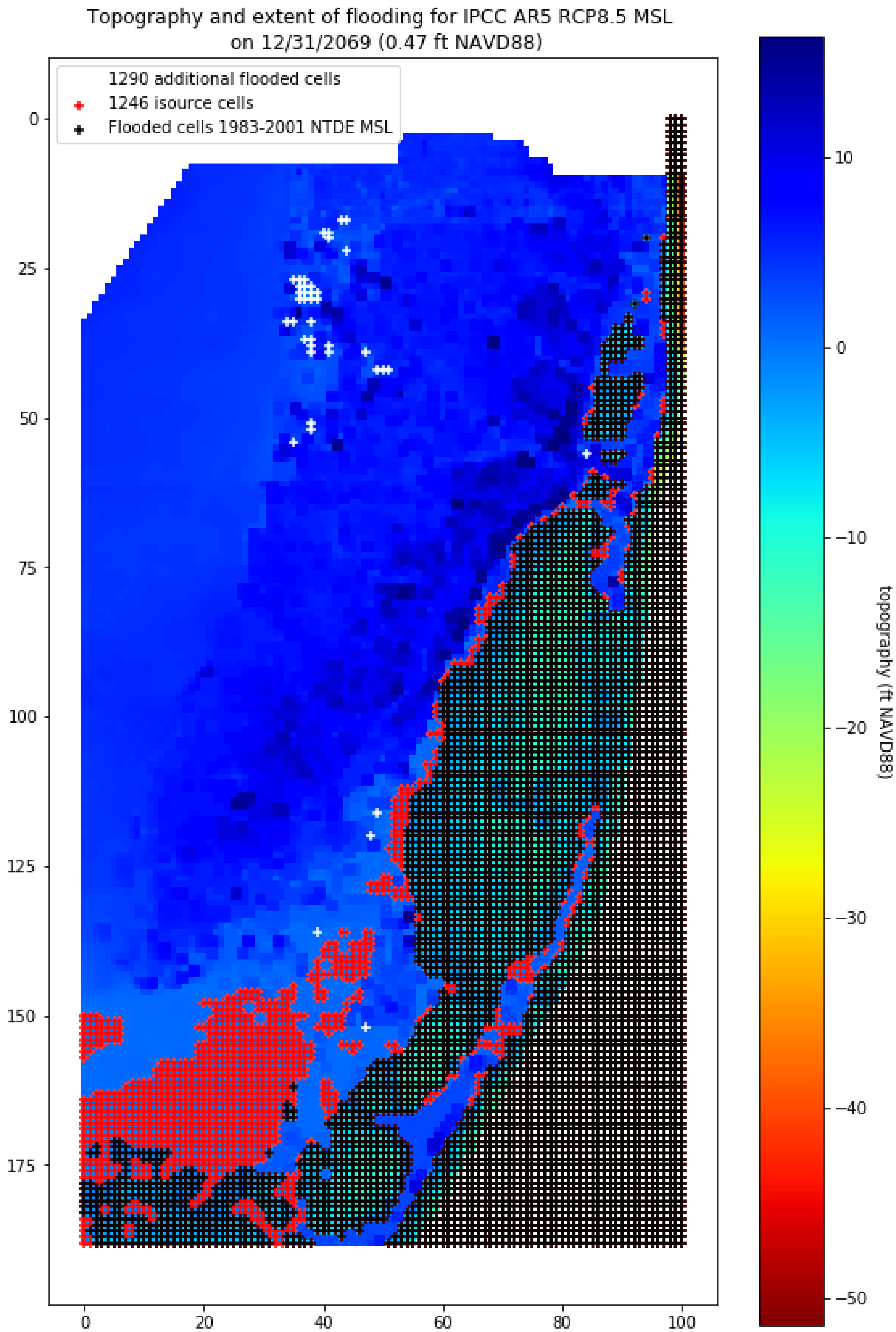


Figure 49. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 2069 at Virginia Key for the IPCC AR5 RCP8.5 SLR scenario (0.47 ft NAVD88). A total of 1290 additional model grid cells would be below MSL compared to the cells flooded based on 1983-2001 NTDE MSL. X and Y coordinates in meters, UTM17N, NAD83.

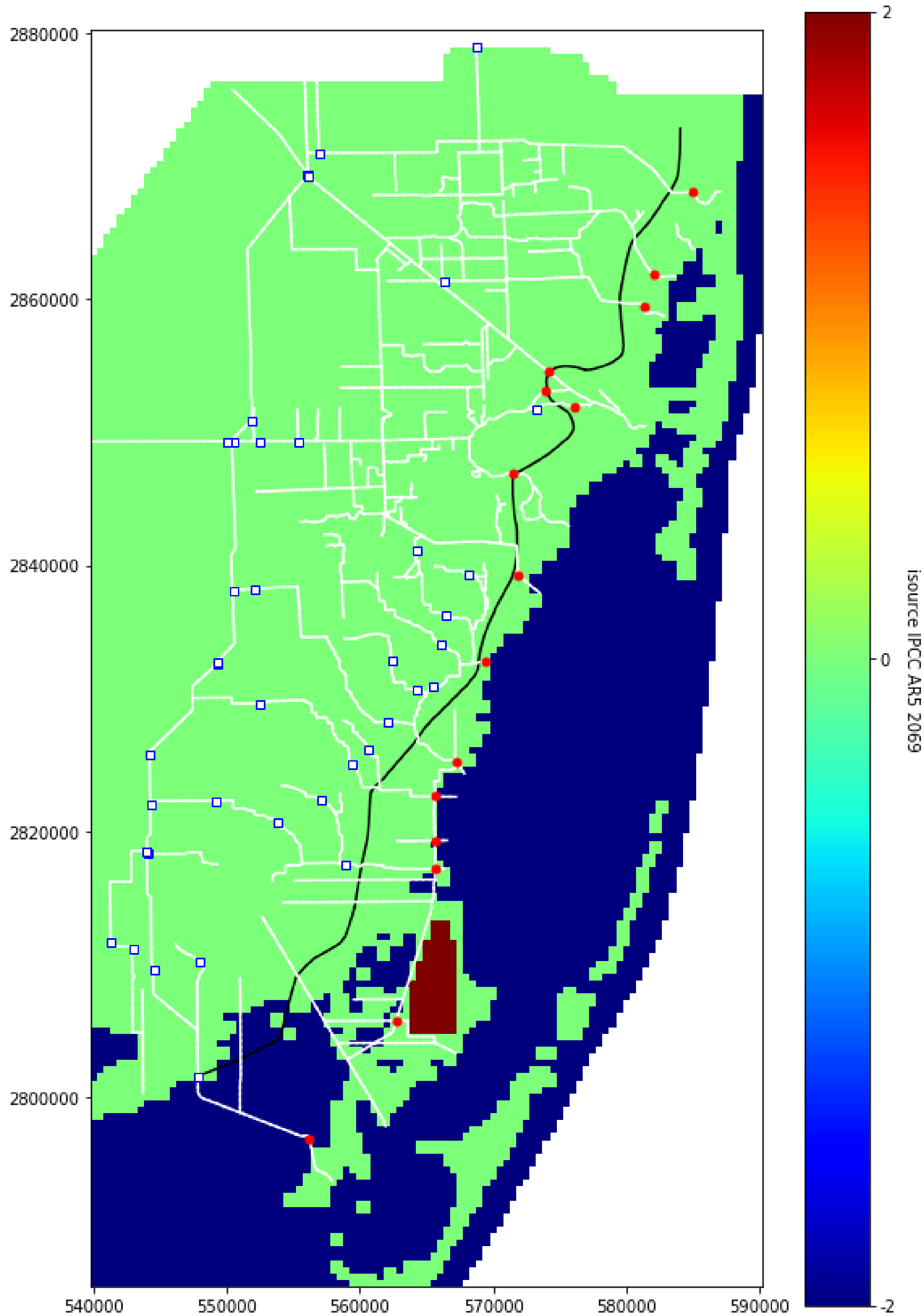


Figure 50. Freshwater/saltwater source (isource) based on the mean sea level on 2069 at Virginia Key for the IPCC AR5 RCP8.5 SLR scenario (0.47 ft NAVD88). X and Y coordinates in meters, UTM17N, NAD83.

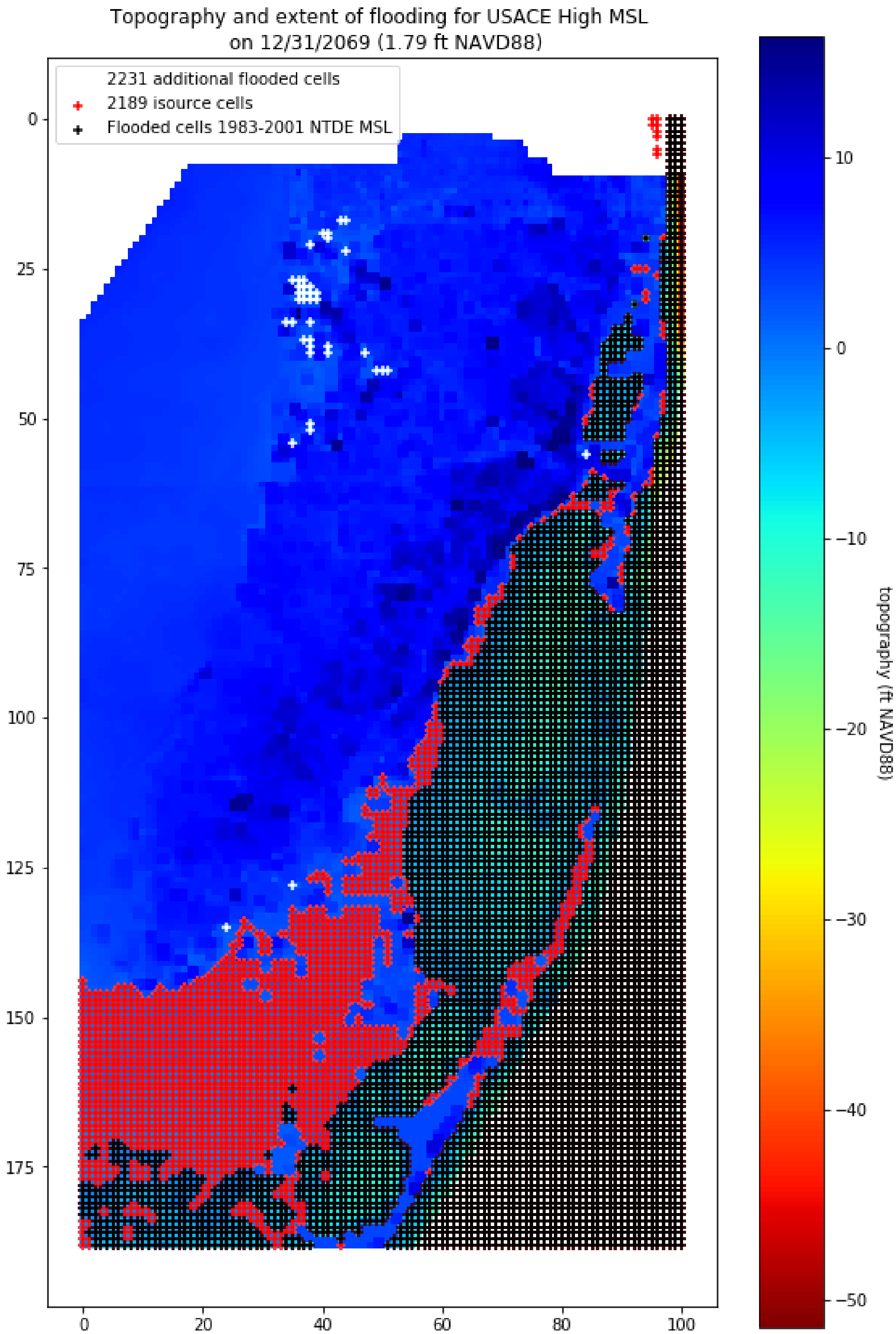


Figure 51. Topography and extent of flooding (cells with '+' black and white markers) for mean sea level on 2069 at Virginia Key for the USACE High SLR scenario (1.79 ft NAVD88). A total of 2231 additional model grid cells would be below MSL. X and Y coordinates in meters, UTM17N, NAD83.

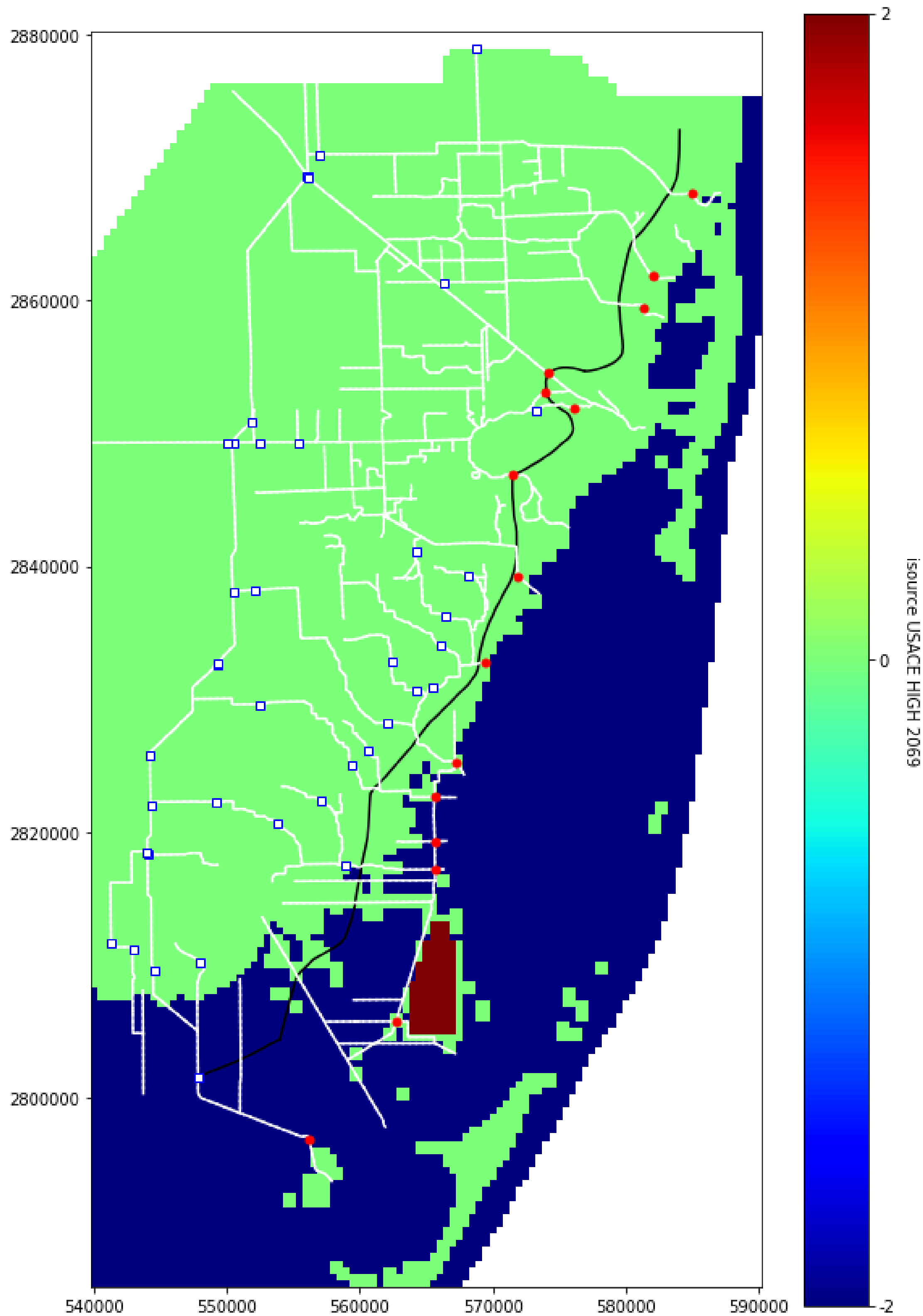


Figure 52. Freshwater/saltwater source (isource) based on the mean sea level on 2069 at Virginia Key for the USACE High SLR scenario (1.79 ft NAVD88). X and Y coordinates in meters, UTM17N, NAD83.

Initial elevation of the freshwater/saltwater interface

The saltwater intrusion package in the Miami-Dade MODFLOW model requires input of an array called izeta, which defines the initial elevation of the freshwater/saltwater interface. The izeta array can be defined for each model layer, in which case "the zeta surface is placed at the top of the model layer when a value is entered that is above the top of the model layer and it is placed at the bottom of the model layer when a value is entered that is below the bottom of the model layer. For the case of a surface that is present at only one point in the vertical everywhere, the same grid of zeta values may be entered for every model layer and the SWI2 package will determine in which cells, the elevation of the zeta surface falls between the top and bottom of each layer." (Bakker et al., 2013).

This second approach was used in the calibration/verification model simulation by the USGS. Figure 53 shows the starting elevation of the freshwater/saltwater interface used in the model calibration/verification. It is based on the position of the interface at the base of the Biscayne Aquifer, defined as the location with a chloride concentration of 100 mg/L in 1995 (Sonenshein, 1997). Model grid cells to the west of the location of the interface at the bottom of the aquifer were assigned an izeta value equal to the bottom of model layer 3 (i.e. bottom of the aquifer) whereas most ocean grid cells were assigned an izeta value equal to the bathymetry. It is unclear how the initial elevation of the interface was interpolated for model cells in between these two regions.

An izeta surface needs to be developed for the future modeling scenarios with sea level rise, which start in the year 2055. Ideally, the source of the initial elevation of the freshwater/saltwater interface should come from observations or from a long-term simulation up to the year 2055 with sea levels rising along the selected sea level rise curve. The future scenario run based on the IPCC AR5 RCP8.5 SLR curve has an initial mean sea level of 0.06 ft NAVD88, which is close to the final mean sea level (0.05 ft NAVD88) in the USGS Scenarios 2 and 3 (Hughes and White, 2016). Given that we will be using the increased well pumpage file from Scenario 3 in our future scenario model runs, the final zeta surfaces for model layers 1-3 from the USGS Scenario 3 were used as initial zeta surfaces for our IPCC AR5 RCP8.5 SLR scenario run.

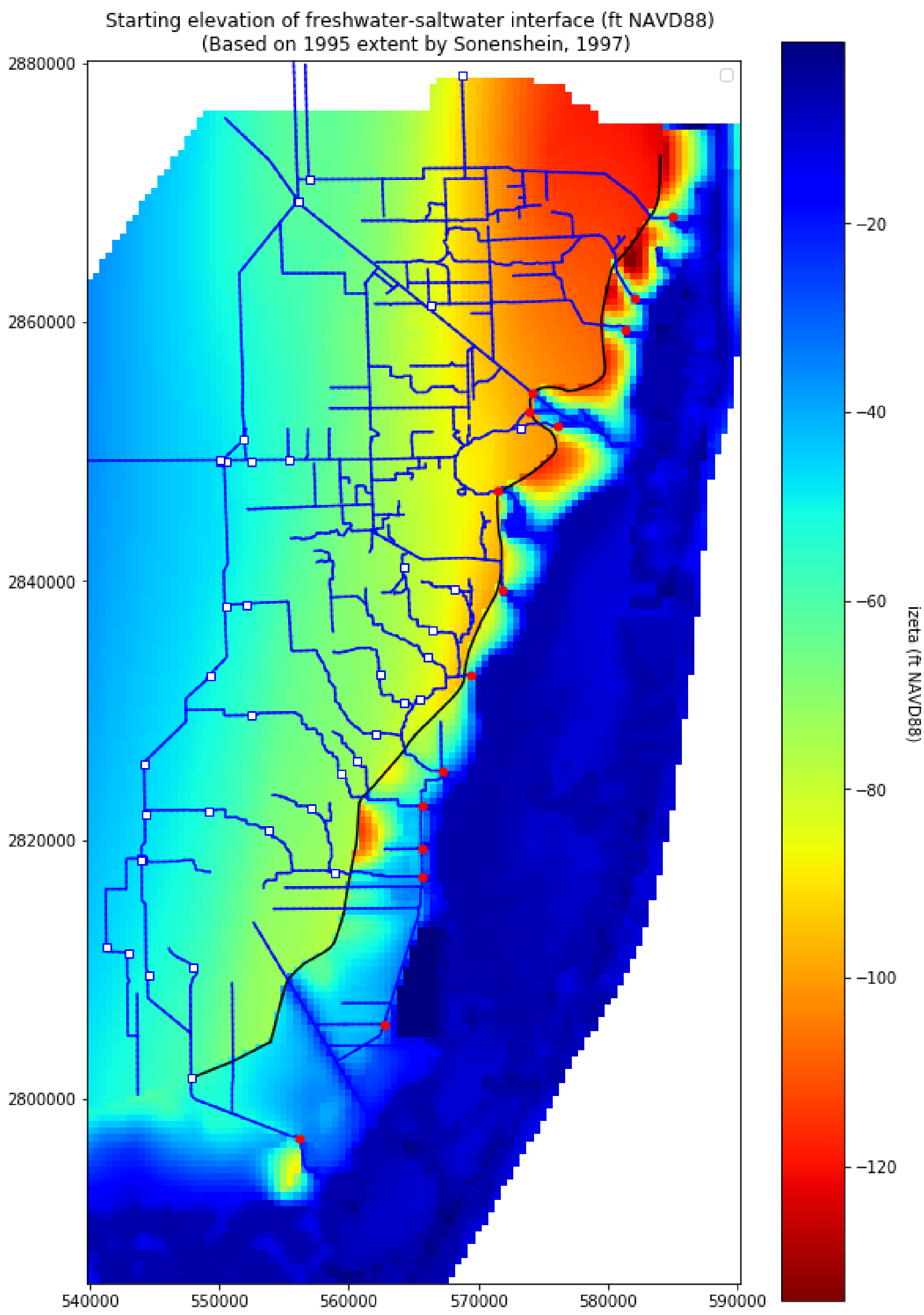


Figure 53. Starting elevation of the freshwater/saltwater interface in 1996. X and Y coordinates in meters, UTM17N, NAD83.

Main future scenario and sensitivity runs

As part of this project, we performed two main future scenario runs and three additional sensitivity runs using the calibrated Miami-Dade MODFLOW model developed by the USGS. The future scenario and sensitivity runs were run for the period 2055-2069 with the intent of using the first five years of the simulation as a spin-up period and dropping them from the analysis. The model input and output files for these future runs often use the same timestamps as in the calibration/verification period (1996-2010); however, they represent input and simulated conditions between 2055-2069.

Main modeling assumptions

The following are common assumptions in all five (5) future scenario and sensitivity runs:

- 2030 land use and directly-connected impervious areas (DCIA), 2018 permitted quarry lakes, calibrated crop coefficients
- 2010 septic return flow from the USGS scenarios
- The western boundary condition consists of water levels in Water Conservation Area 3 (WCA3) and Eastern Everglades National Park (ENP) from CERP0 SFWMM run obtained from J. Barnes (average for Julian day at each cell is repeated every year)
- The 1-D surface water network, structures, effective gate openings, and specified pump discharges remain the same as in the USGS 1996-2010 calibration/verification of the model (Figure 88).

The two main scenario runs (runs 1 and 2 on Table 5) are identical except that they use two different tidal boundary conditions which represent tidal predictions plus two different sea level rise curves (IPCC AR5 RCP8.5 Median curve, and USACE High curve, respectively). Runs 3-5 are variations of the first two runs.

All runs, with the exception of run 3, use 2030-2040 wellfield pumpage from USGS Scenario 1 for Miami Dade Water and Sewer Department (MDWASD) wells (372.58 MGD), and 2010 wellfield pumpage for other wells (52.65 MGD) for a total wellfield pumpage of 425.23 MGD (Table 6). Pumpage at a particular wellfield (Figure 89) is distributed equally among all wells and the 2030-2040 daily pumpage timeseries is repeated during every year of the scenario runs. All pumpage is removed from the bottom layer of the model (layer 3), which is the primary production zone for the Biscayne Aquifer in this area.

Figure 54 shows the average wellfield pumpage by model grid cell in the 1996-2010 calibration/verification run, which adds up to an average annual total pumpage of 385.27 MGD. Figure 55 shows the average 2030-2040 wellfield pumpage for the future scenario runs, which totals 425.23 MGD. Differences in pumpage between the future scenario runs and the calibration/verification run are shown in Figure 56, where the addition of the South Miami Heights wellfield is evident as well as the removal of the Leisure, Naranja, Elevated Tank, Everglades Labor and Newton wellfields. Increased pumpage at the

Southwest, Northwest, and West wellfields are also evident. In addition, some decreases are observed in the Miami Springs-Hialeah-Preston wellfield and the Alexander Orr wellfield. See Hughes and White (2016) for more details on the source of future wellfield pumpage.

Run 3 is a worse-case scenario for flooding due to its use of a high SLR curve and no pumpage. The main future scenario runs (runs 1 and 2) use a rainfall timeseries from a bias-corrected LOCA model run with increased rainfall when compared to historical conditions, and assumes a 5% increase in RET resulting from increased future temperatures (Obeysekera et al., 2014). Runs 4 and 5 are the same as 1 and 2, but using historical rainfall and reference ET (RET). More details on these assumptions and a description of the MODFLOW input files modified for these runs can be found in Appendix E. MODFLOW input file modifications for scenario simulations.

Table 5. Assumptions for two main scenario runs (1 and 2) and the three additional scenario sensitivity runs (3-5).

Run short-name	**(1) LOW SLR**	**(2) HIGH SLR**	**(3) HIGH SLR + NO PUMPAGE**	**(4) LOW SLR + HIST RAIN/RET**	**(5) HIGH SLR + HIST RAIN/RET**
Run description	**Low SLR scenario (IPCC median)**	**High SLR scenario (USACE High)**	**High SLR scenario with no pumpage**	**Low SLR scenario with historical rainfall/RET**	**High SLR scenario with historical rainfall/RET**
Rainfall and recharge					
1996-2010 NEXRAD rainfall with 1.05 correction factor				X	X
Bias-corrected LOCA rainfall for 2055-2069 (no correction factor applied)	X	X	X		
Reference evapotranspiration (RET)					
1996-2010 RET from the USGS				X	X
1996-2010 RET from the USGS with 1.05 adjustment factor due to future temperature increase	X	X	X		
PWS pumpage					
No pumpage			X		
Future Pumpage as in USGS Scen. 1 for 2030-2040	X	X		X	X
Tidal boundary condition					
Predicted sea levels for 2055-2069 + SLR from IPCC AR5 RCP8.5 median curve	X			X	
Predicted sea levels for 2055-2069 + SLR from USACE High curve		X	X		X

Table 6. Wellfield pumpage from USGS Scenario 1 run in million gallons per day (MGD).

Wellfield	Wellfield pumpage rate (MGD)
Hialeah	3.1
Preston	37.2
Miami Springs	29.7
Northwest	85.4
Alexander Orr	40
Snapper Creek	21.9
Southwest	137.28
West	15
South Miami Heights	3
TOTAL for MDWASD wells	**372.58**
Other wells	**52.65**
TOTAL pumpage	**425.23**

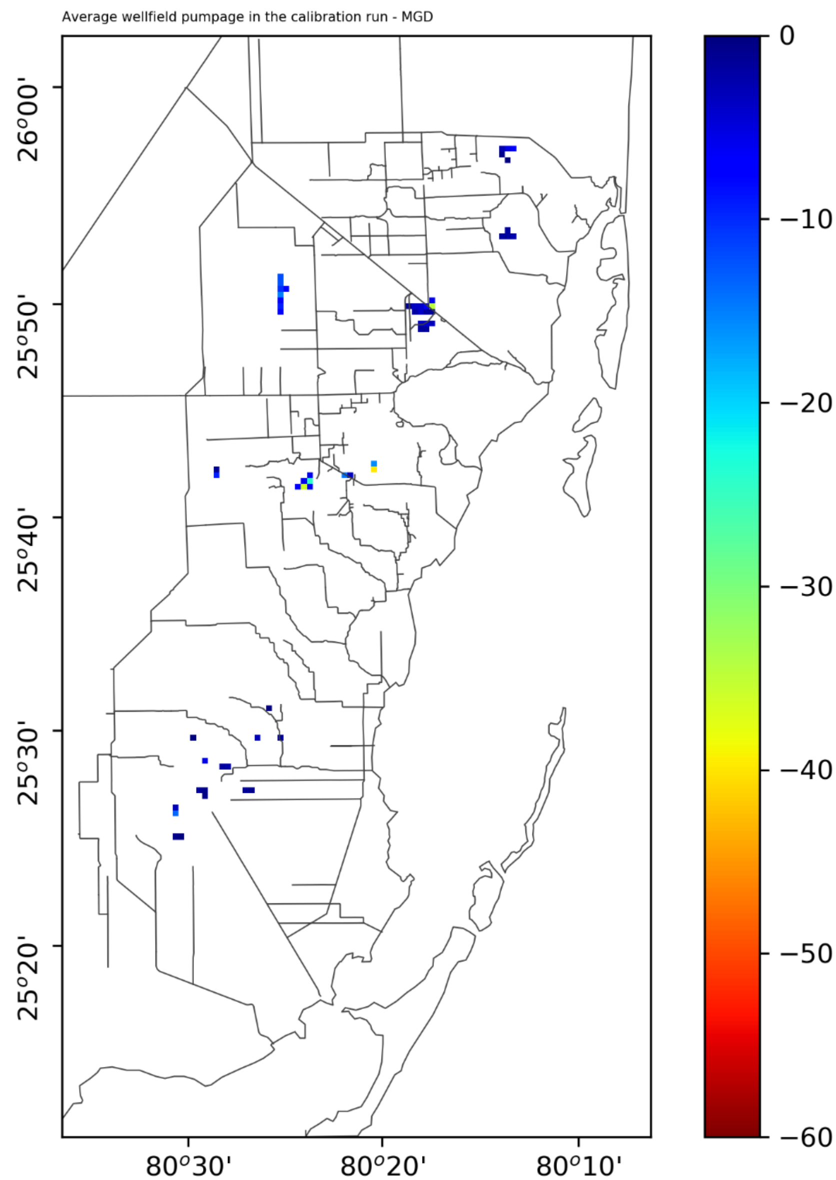


Figure 54. Average wellfield pumpage during the calibration/verification period (1996-2010) in MGD. Higher pumpage values are indicated by dark red colors (i.e. higher negative recharge).

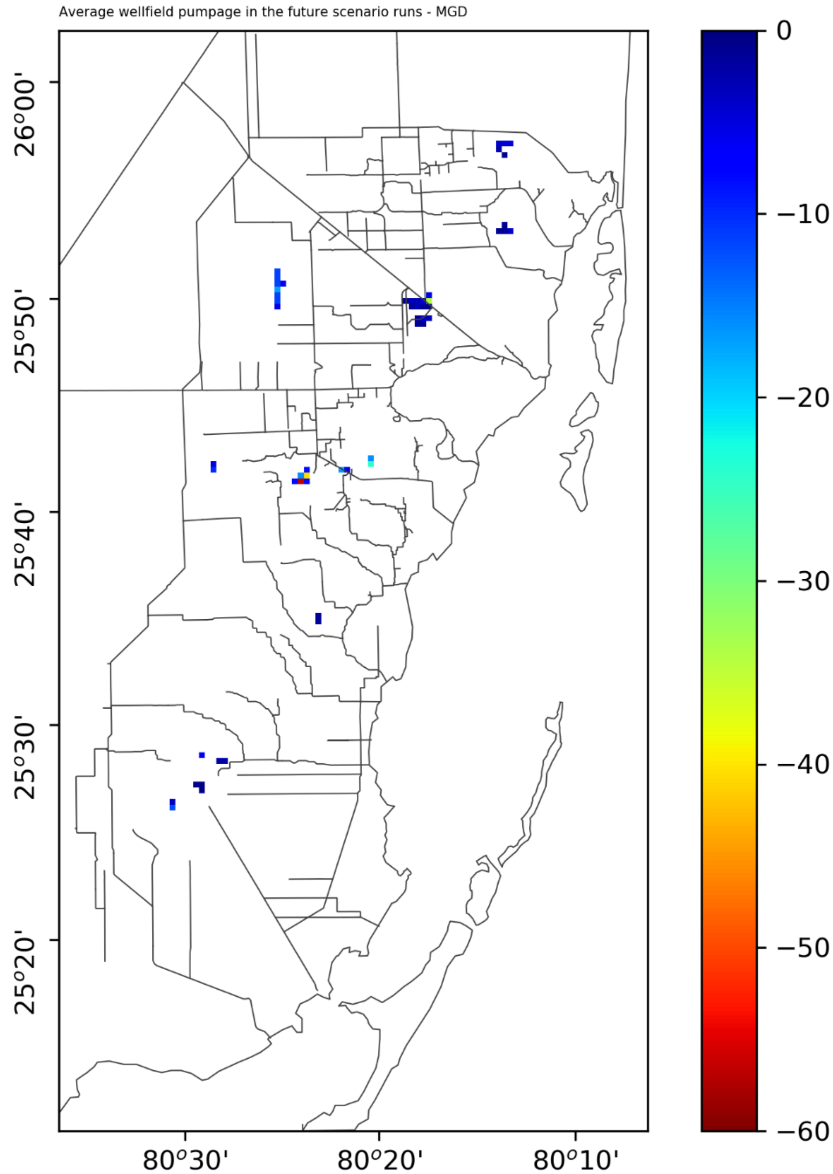


Figure 55. Average wellfield pumpage in the future scenario and sensitivity runs in MGD. Higher pumpage values are indicated by dark red colors (i.e. higher negative recharge).

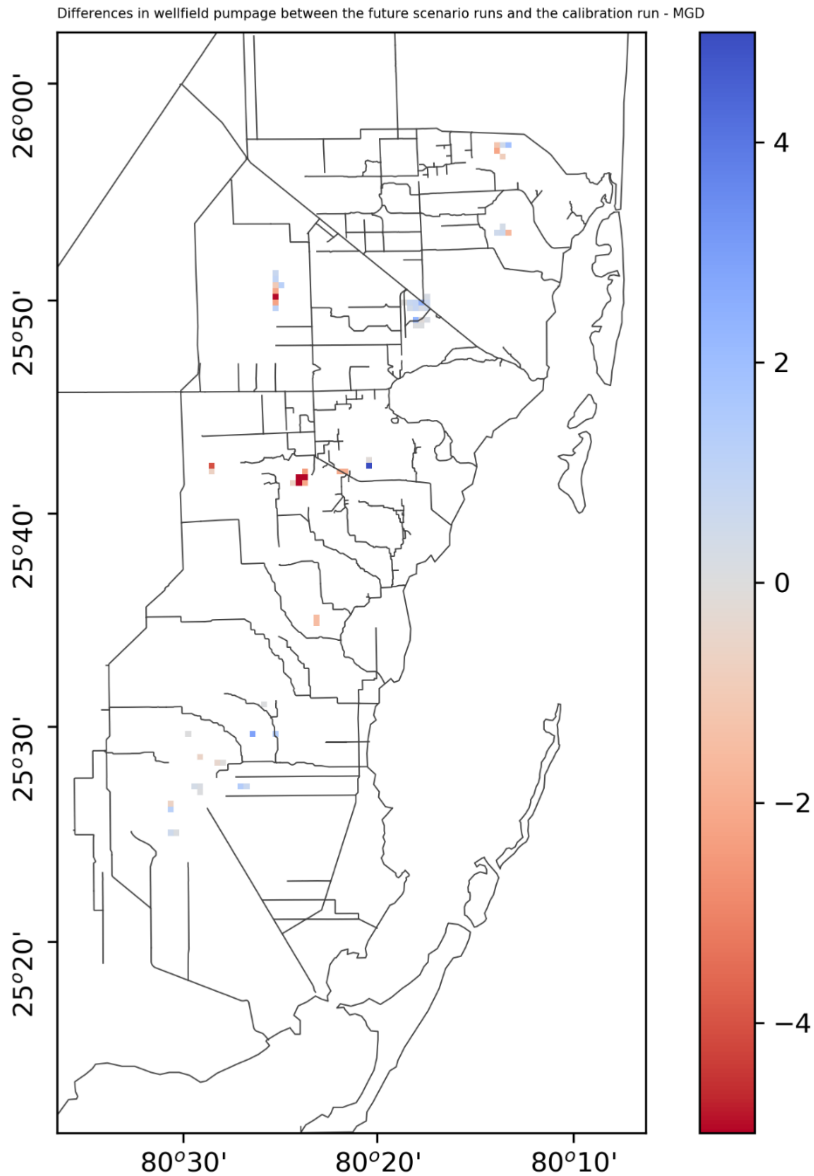


Figure 56. Differences in average wellfield pumpage in the future scenario and sensitivity runs minus the calibration run in MGD. Higher pumpage values in the future scenario runs are indicated by red colors (i.e. higher negative recharge). Differences range from -21 to +16 MGD at individual model cells, but -5 to +5 MGD range chosen for display purposes.

Initial condition runs

In order to provide a reasonable set of initial conditions for modeling these scenarios, a couple of long-term simulations for the period 1996-2054 were performed. In particular the initial location of the saltwater/freshwater interface in 2055 is critical and difficult to derive analytical methods. The simulations were broken into three periods (1996-2025, 2026-2040, and 2041-2054) so that the isource variable, which defines the density (saltwater vs. freshwater) of sources and sinks in each model layer in the SWI2 package, could vary throughout this relatively long period. The two long-term simulations were based on a repetition of the stresses (rainfall, RET, irrigation, wellfield pumpage, structure operations) during the 1996-2010 calibration/verification period; however, the eastern boundary condition at Virginia Key was based on future tidal predictions plus sea level rise along one of the two SLR curves of interest (IPCC AR5 RCP8.5 median or USACE High SLR curves).

These two long-term model runs provide the initial conditions required by the Urban Miami-Dade MODFLOW model, which consist of initial heads for each of the three groundwater layers, stages on the surface water reaches, and the location of the saltwater/freshwater interface (izeta surface). The scenario runs will be run for the period 2055-2069 with the intent of using the first five years of the simulation as a spin-up period, which will be dropped from the analysis. This should minimize the influence of errors in the initial condition on the simulated groundwater levels for the 2060-2069 period of interest. The model set up and results for these long-term runs are described in the next sections. In some cases, results from the main modeling runs for the period 2055-2069 are presented as well.

Input file modifications

Figure 57 summarizes the input file structure of the model and files that had to be updated. The folder named “model” has two green subfolders that were updated. These subfolders split into 6 yellow subfolders and 3 pink subfolders. The red square represents the updated “nam” file, which contains the list of directories where the model is directed to read the remaining input files. Cyan squares represent files for specific processes (e.g., ghb = general head boundaries) that contain model details such as discretization, surface and saltwater package information or path information for files that change on a daily basis through the simulation period. Finally, ochre/brown squares represent updated files with the data. Full lines arrows on the figure represent where the model is directed to, and dotted lines join folders with their subfolders and files.

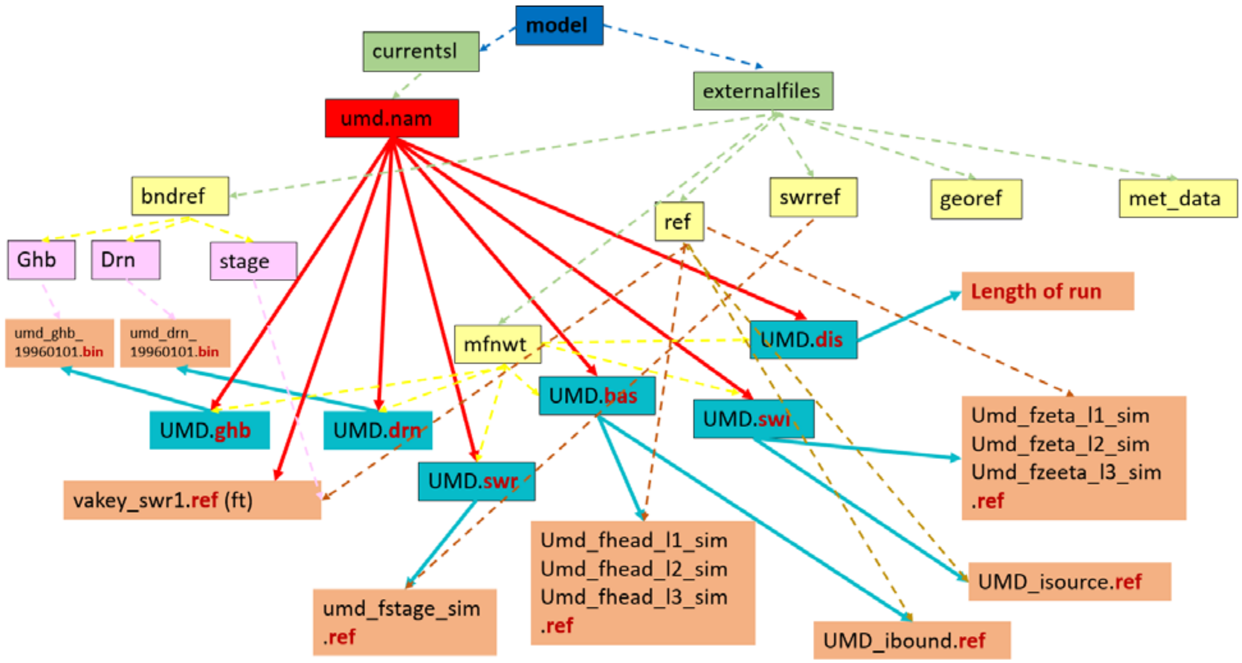


Figure 57. Model structure of folders and files updated for the first three simulation period runs. Folders in green, yellow, and pink. Files in red, cyan, and ochre/brown.

Freshwater/saltwater source (isource variable)

We computed the isource variable for the year 2025 using average Virginia Key daily tide predictions plus sea level rise for the year 2025 compared to the land surface elevation to determine if a particular location was flooded with sea water. Similar calculations were made for 2040 and 2054.

Figure 58 shows the isource maps for the low sea level rise scenario run 1996-2025 (a), 2026-2040 (b), 2041-2054 (c) and 2055-2069 (d), and Figure 59 for the high sea level scenario run 1996-2025 (a), 2026-2040 (b), 2041-2054 (c) and 2055-2069 (d). Green indicates fresh water, blue is salt water, and brown is elevated salinity water in the Turkey Point Nuclear Plant cooling canals. Isource for each run is different, especially in the southern area of the map where seawater overtops the land surface. Greater differences in the isource occur in the high sea level rise scenario.

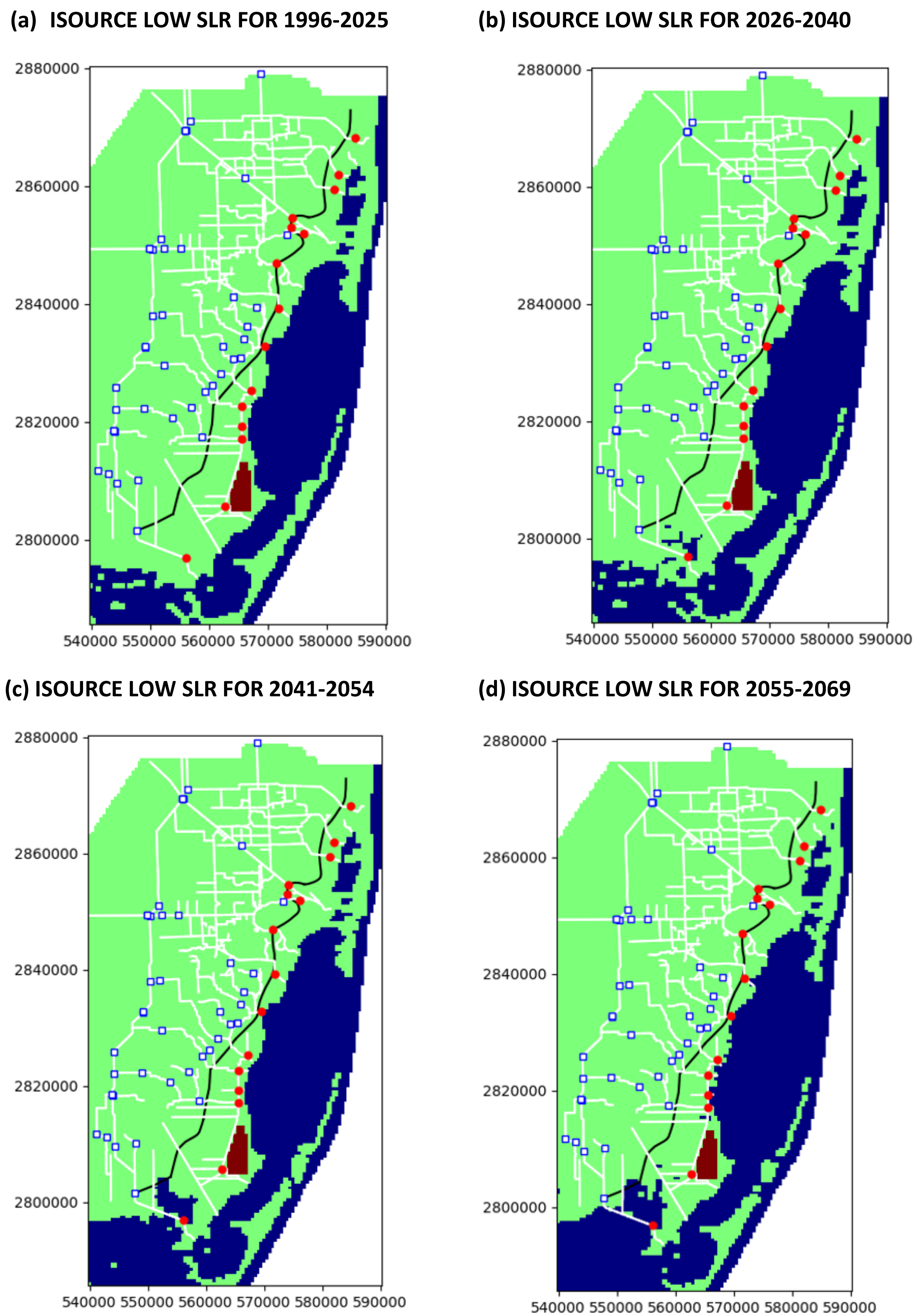


Figure 58. Isource for Low SLR (a) run (1996-2025), (b) run (2026-2040), (c) run (2041-2054) and (d) run (2055-2069). Green indicates fresh water, blue is seawater, and brown is elevated salinity water in the Turkey Point Nuclear Plant cooling canals. Water control structures are shown as blue squares and salinity control structures are shown as red dots.

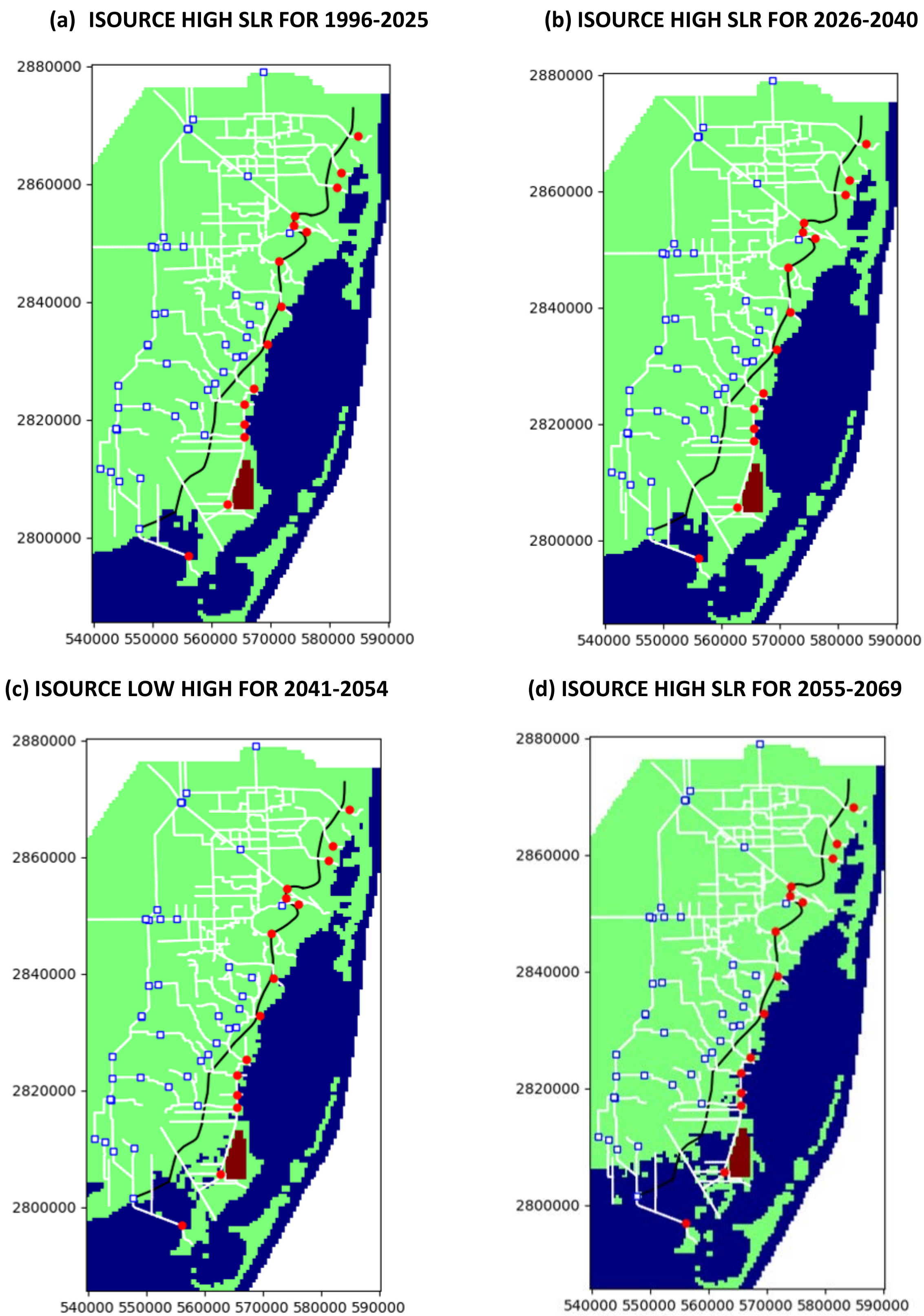


Figure 59. Isource for High SLR (a) 1996-2025 run, (b) 2026-2040 run, (c) 2041-2054 run, and (d) 2055-2069 run. Green indicates fresh water, blue is seawater, and brown is elevated salinity water in the Turkey Point Nuclear Plant cooling canals. Water control structures are shown as blue squares and salinity control structures are shown as red dots.

Step-wise model runs

Initial conditions that were updated and passed to each subsequent run include the simulated final heads of layers 1 through 3, the simulated final zeta surfaces of layer 1 through 3, simulated final canal stages, and isource. The first run of the low SLR scenario was 30 years long, starting with the time step 12/31/1995 and ended with time step 1/1/2026. With the python script "processumdfinalstageheadzeta.py", we created files of the head, stage, and zeta from the last day (12/3/2025) of the run. Updated files were named the same as previous files, except that we added low_26_40, which is the name of the run for which the files were used as initial conditions. We updated directories for the model to read input files from and the scenarios.dis file which determines the length of the run.

The second run was 15 years long. Before running the model, we created DRN and GHB binary files with the Python script UMD_Scenario_BND. We changed only the Virginia Key stage file and isource input files to create new DRN and GHB files. We used updated DRN, GHB, heads, zeta, stage, and isource files to re-run the model starting with time step 12/31/2025 and ending with time step 1/1/2041. After the run, we again updated isource, GHB, DRN, head, zeta, and stage files, which we used as the initial condition for the third run. The third run was 14 years long starting on 12/31/2040 and ending on 1/1/2055. We used the same approach we used for the high SLR scenario. Table 7 shows values used to create isource for each run for low and high SLR scenarios and the run length for each run. The isource was created by averaging Virginia Key predicted tide plus SLR of the last year of the run. From the year 2005 to the year 2054 for the low scenario, the average Virginia Key stage increased by 0.589 foot, and for the high scenario by 1.19 foot. The second part of the table indicates the number of time steps for each run. The longest run 1996-2025 has 10958 time steps, and the shortest run 2041-2054 has 5113 time steps.

Table 7. Average values of Virginia Key stage for creating isource and length of the model in file scenarios.dis

SCENARIO	RUN	ISOURCE		SCENARIOS.DIS Number of time steps (days)
		Year	Yearly average of Virginia Key (feet NAVD88)	
LOW	1996-2025	original	-0.76 (2010 average-original)	10958
	2026-2040	2025	-0.54	5479
	2041-2054	2040	-0.27	5113
	2055-2069	2054	0.049	5479
HIGH	1996-2025	original	-0.30 (original highsl)	10958
	2026-2040	2025	-0.30	5479
	2041-2054	2040	0.24	5113
	2055-2069	2054	0.89	5479

Map results of step-wise model runs

Wet season heads in Layer 1 are plotted below for each of the three simulations. The days selected for comparison are evenly spaced every 15 years, so the heads can be compared when the repeated tidal (plus SLR), rainfall, and other time series inputs cause groundwater levels to be on the same trend in their cycles. Figure 60 shows simulated wet season heads of Layer 1 in feet NAVD88 for the low SLR scenario on 12/31/2024 (a), on 12/31/2039 (b) and on 12/31/2054 (c). Figure 61 shows depth to water maps for the same heads as in Figure 60. Figure 62 shows simulated wet season heads of layer one in feet NAVD88 for the high SLR scenario on 12/31/2024 (a), on 12/31/2039 (b) and on 12/31/2054 (c). Figure 63 shows depth to water maps for the same heads as in Figure 62.

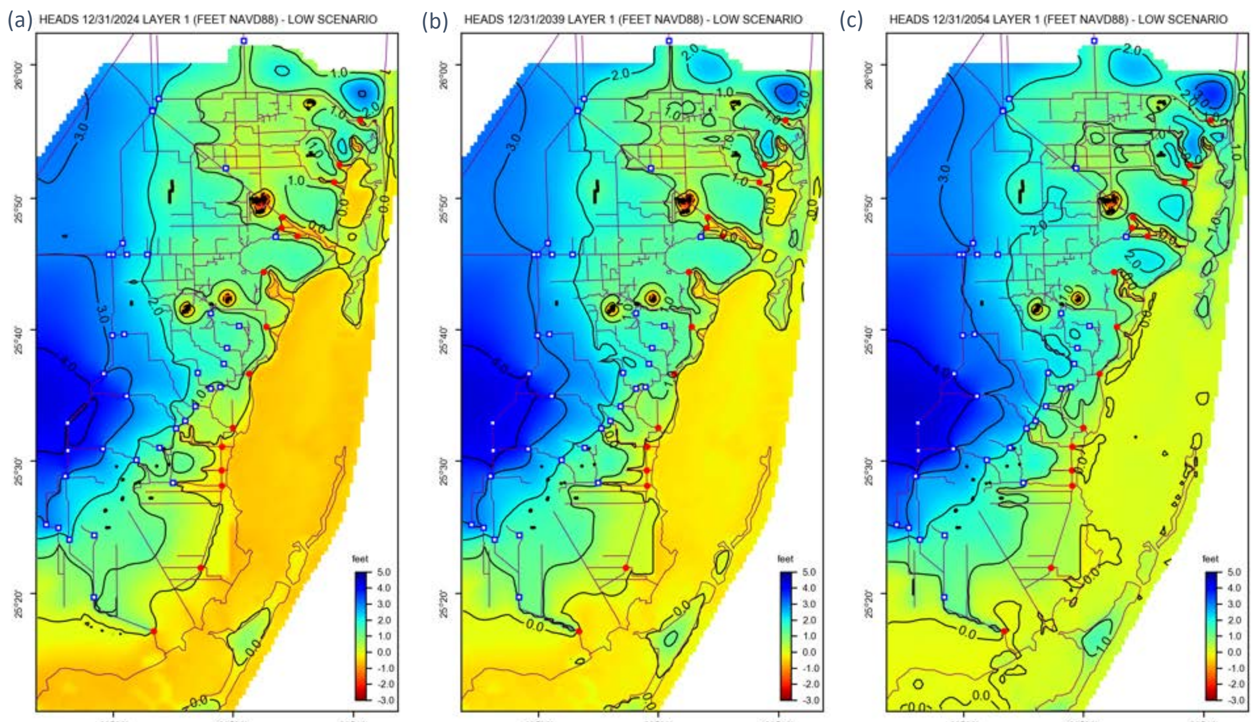


Figure 60 . Simulated heads of Low SLR scenario on (a) 12/31/2024, (b) 12/31/2039 and (c) 12/31/2054.

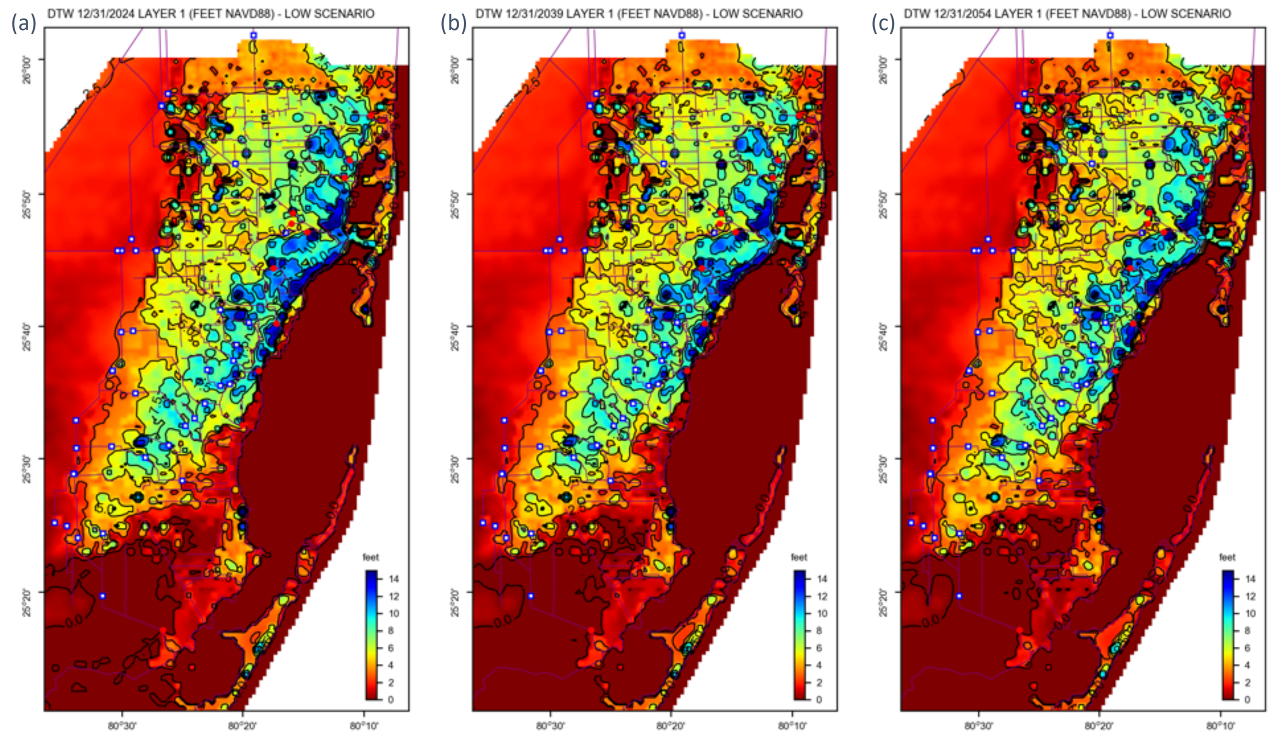


Figure 61. Simulated depth to water of Low SLR scenario on (a) 12/31/2024, (b) 12/31/2039 and (c) 12/31/2054.

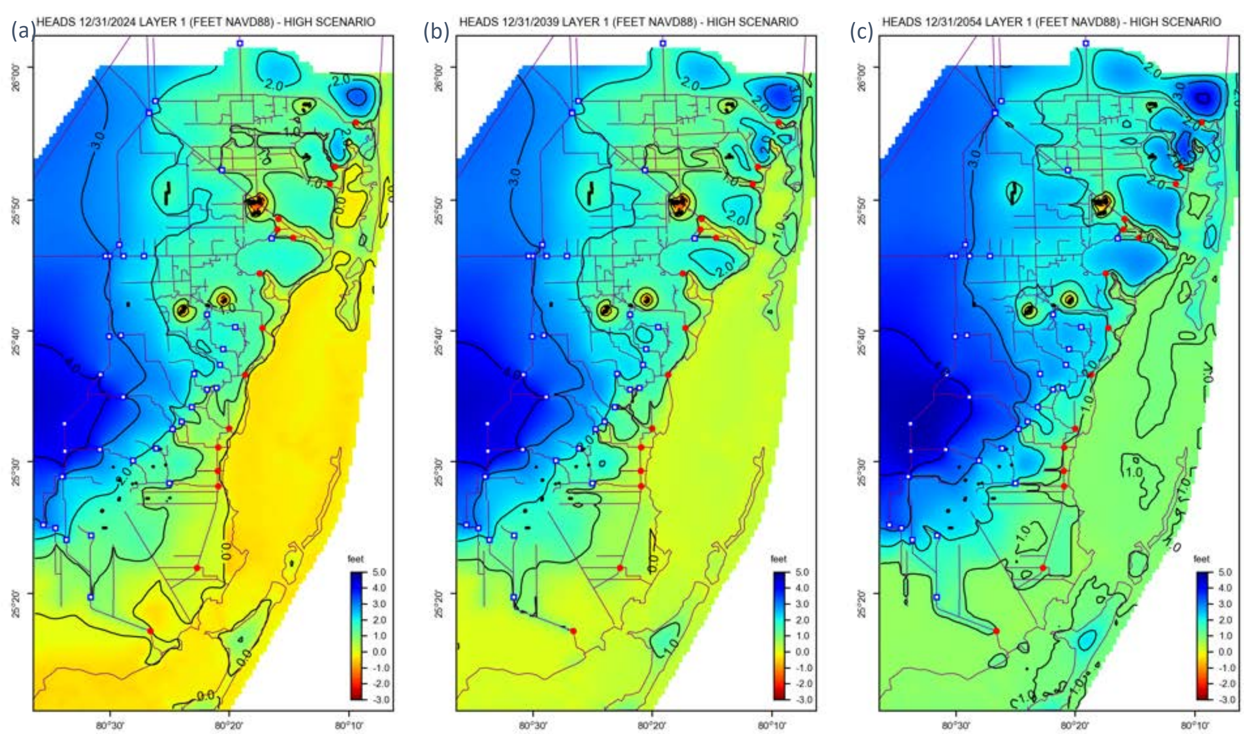


Figure 62. Simulated heads of High SLR scenario on (a) 12/31/2024, (b) 12/31/2039 and (c) 12/31/2054.

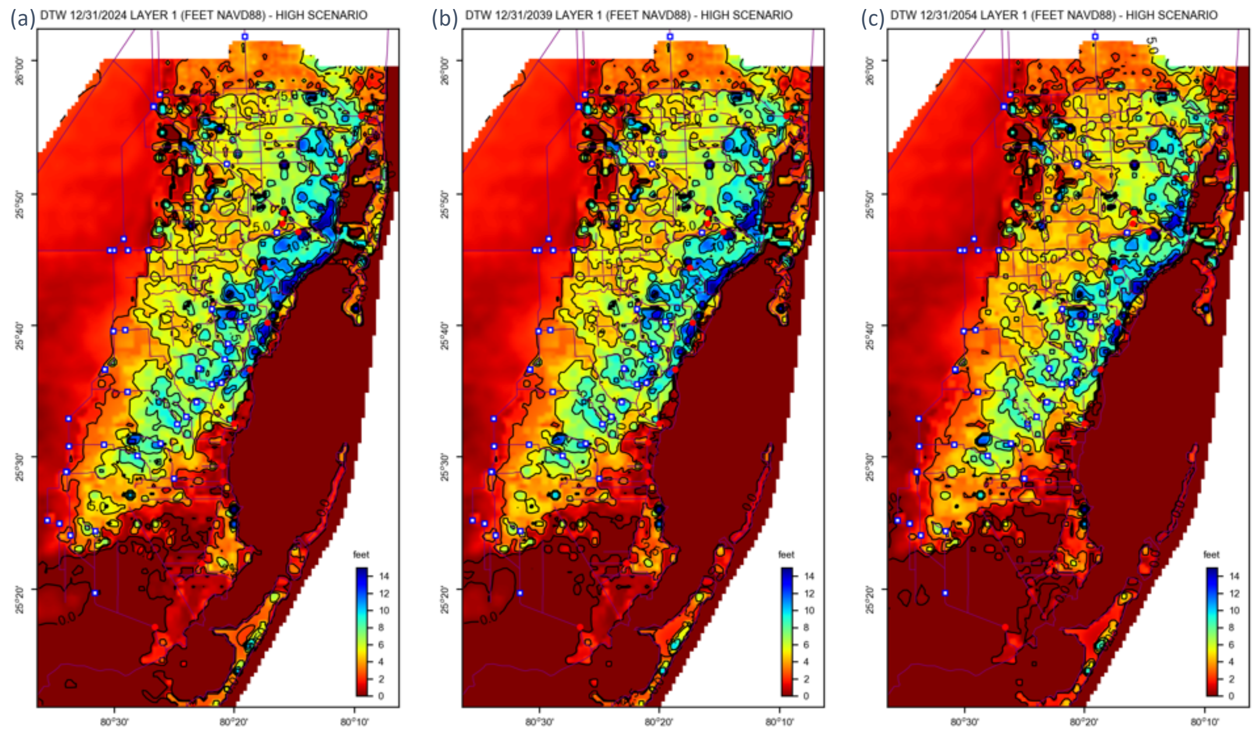


Figure 63. Simulated depth to water of High SLR scenario on (a) 12/31/2024, (b) 12/31/2039 and (c) 12/31/2054

The changes in the heads at different times for the low SLR scenario are shown in Figure 64; the changes are shown between (a) 12/31/2039 and 12/31/2024, (b) 12/31/2054 and 12/31/2039 and (c) 12/31/2054 and 12/31/2024. Similarly, the changes in the heads for the high SLR scenario are shown in Figure 65. The heads are increasing everywhere on the map except in small portions of the south area of the map, where head elevation has decreased.

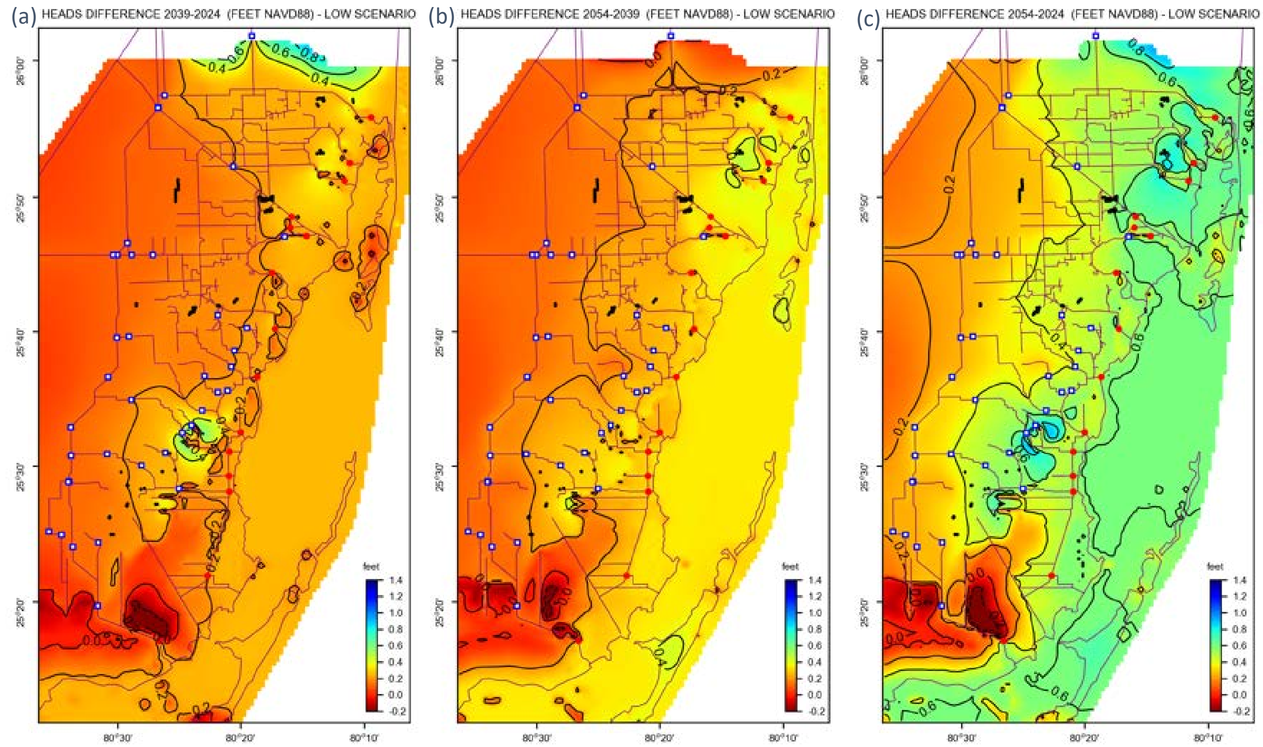


Figure 64. Simulated difference in heads for low SLR scenario between (a) 12/31/2039 and 12/31/2024, (b) 12/31/2054 and 12/31/2039 and (c) 12/31/2054 and 12/31/2024.

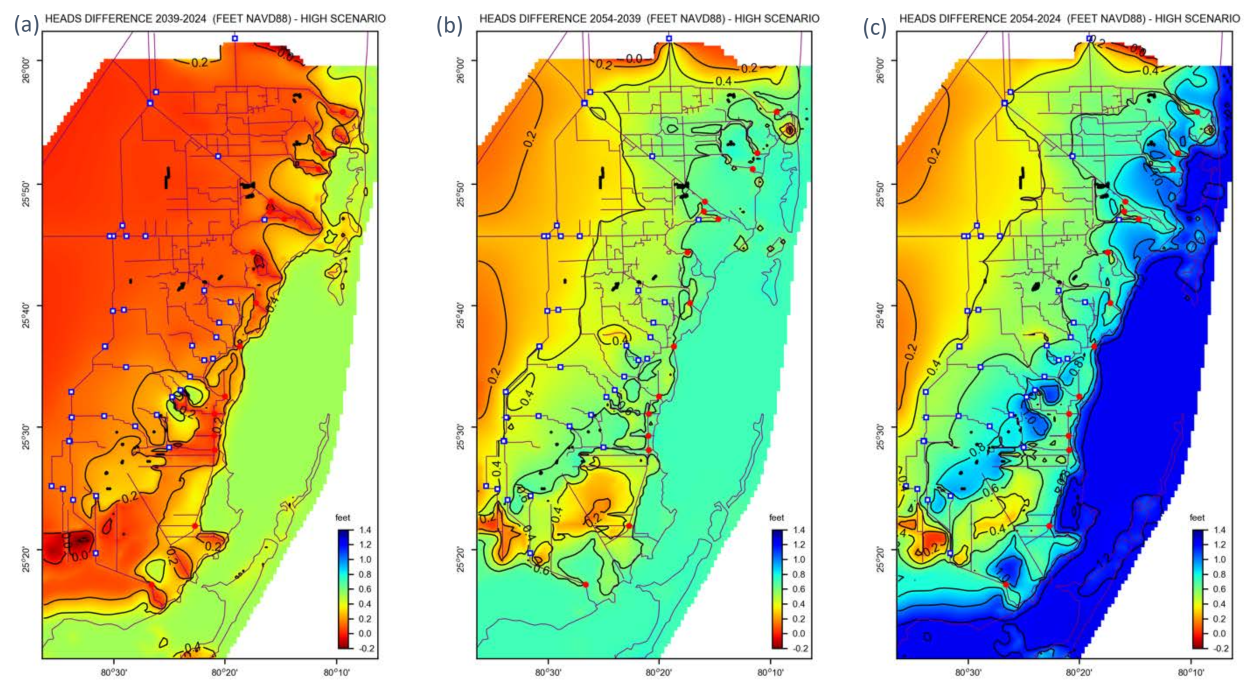


Figure 65. Simulated difference in heads for high SLR scenario between (a) 12/31/2039 and 12/31/2024, (b) 12/31/2054 and 12/31/2039 and (c) 12/31/2054 and 12/31/2024.

The greatest difference at any model grid cell in the low SLR scenario between 12/31/2024 and 12/31/2069 with updated rainfall data is 4.85 feet, and between 12/31/2024 and 12/31/2069 with historical rainfall is 4.06 feet. The greatest difference in the high SLR scenario between 12/31/2024 and 12/31/2069 with updated rainfall data is 3.96 feet, between 12/31/2024 and 12/31/2069 with historical rainfall is 3.95 feet, and between 12/31/2024 and 12/31/2069 with no pumpage is 9.03 feet.

Cross-section results of step-wise and final model runs

We plotted cross-sections of head changes through the years by choosing model Column 51 (north-to-south section at 564750 UTM meters East) and model Row 95 (west-to-east section at 2832750 UTM meters North) as shown in Figure 66.

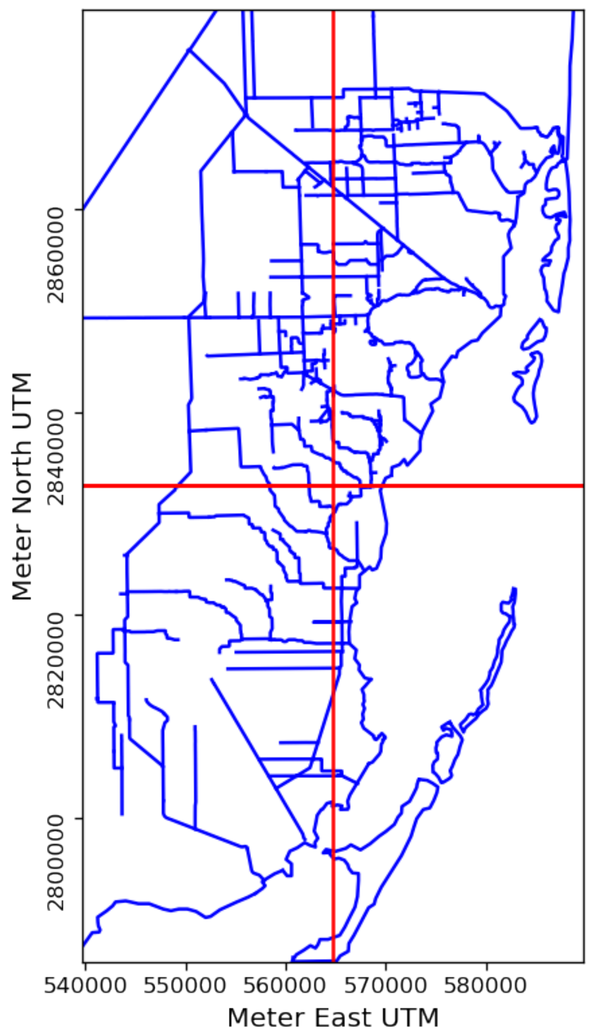


Figure 66. Cross-section location map. Blue color represents hydrography and red lines represent cross-sections.

The cross-sections for the low SLR scenario are shown in Figure 67 from (a) west-to-east and (b) north-to-south. As above, the days selected for comparison are evenly spaced every 15 years, so the heads are at the same time in their trend cycle. The green line represents topography. The red line represents day 12/31/2024, and it has the lowest head elevation of all lines. As the simulations progress in time, the head elevation increases. The blue line represents day 12/31/2039, and the yellow line represents day 12/31/2054. The purple color represents day 12/31/2069 with the updated rainfall data, the maroon color represents day 12/31/2069 with the historical rainfall. The cross-section for the high SLR scenario is shown in Figure 68 from (a) west-to-east and (b) north-to-south and, in addition to the times and scenarios depicted in Figure 67, includes a gray line representing day 12/31/2069 with no water supply wells operating. This is the most conservative scenario in terms of high groundwater levels that is considered in this report.

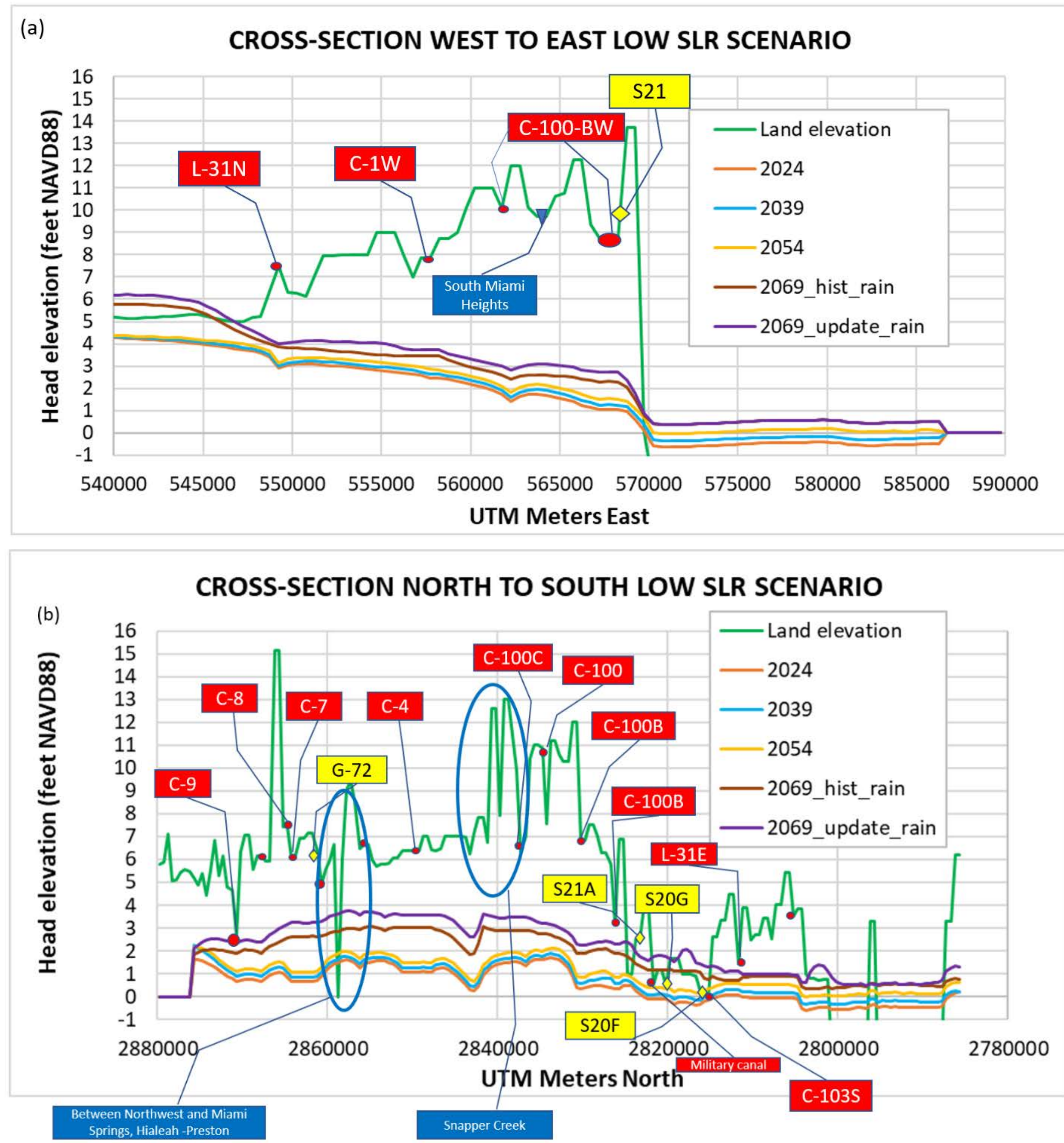
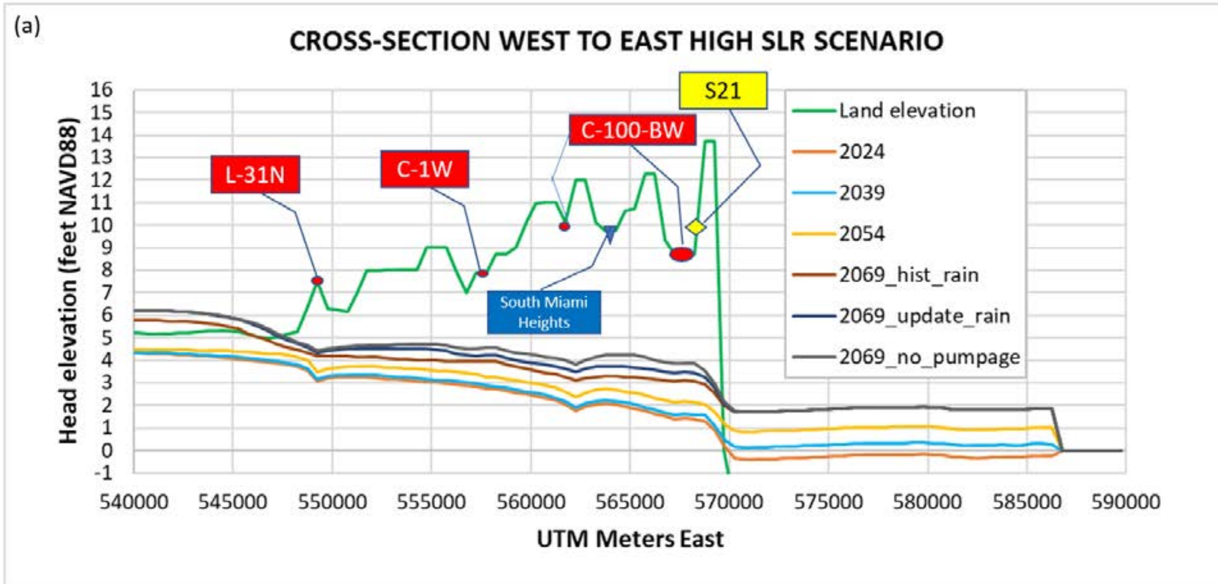


Figure 67. Cross-section for low SLR scenario and sensitivity runs (a) West to East and (b) North to South.



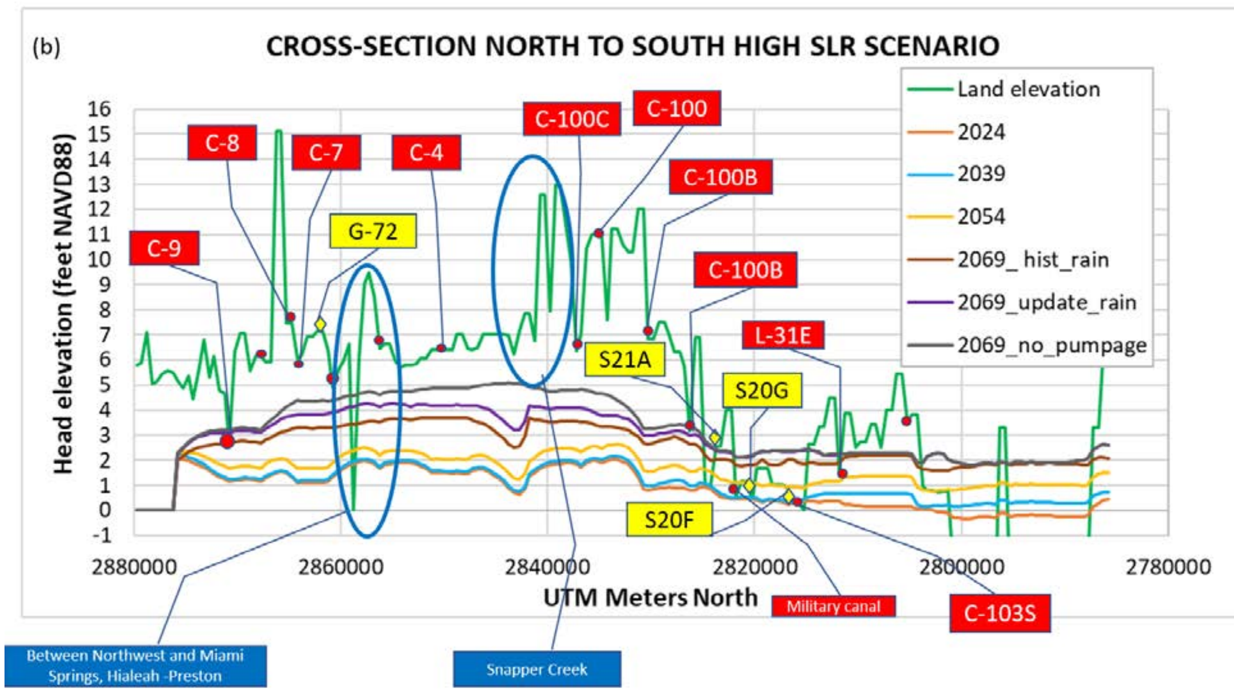


Figure 68. Cross-section for high SLR scenario and sensitivity runs (a) West to East and (b) North to South.

Saltwater intrusion zeta surfaces

The Saltwater Intrusion Package (SWI) utilizes a sharp-interface approximation to simulate the position of the subsurface interface between seawater and freshwater. The position of the interface is given by the steepest region of the Zeta surface, which separates seawater and freshwater. In areas where an aquifer layer is filled with all seawater or all freshwater, the Zeta surface has the same elevation as the top or bottom of the layer respectively. Landward motion of the steep part of the Zeta elevation surface reflects seawater intrusion. Zeta surfaces can be used to convert the equivalent freshwater head model results into actual heads by accounting for the density difference between saltwater and freshwater. This was not done in this report because the difference is expected to be small at the water table and the equivalent freshwater head is higher and therefore gives conservatively high estimates of the water table's impact on flooding. In general, the zeta surface maps show some landward progression of the saltwater interface. Figure 69, Figure 70, Figure 71, Figure 72, Figure 73, and Figure 74 show the Zeta surfaces for the 3 sequential time periods for each of the 3 model layers, under the low and high SLR scenarios. In general, the zeta surface maps show some landward progression of the saltwater interface.

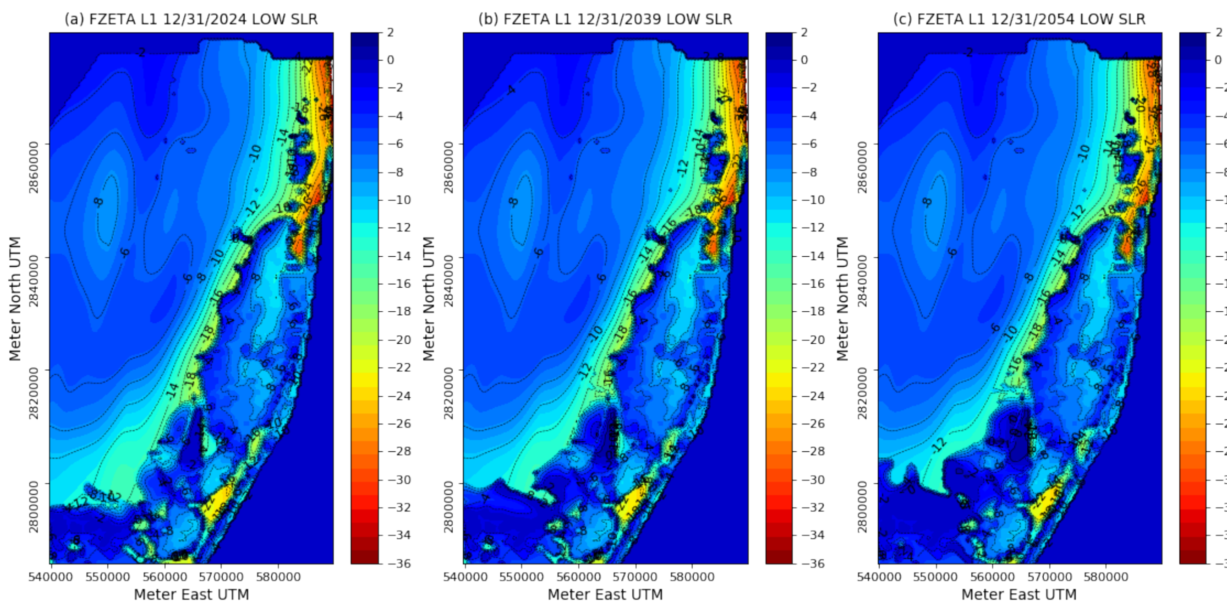


Figure 69. Zeta surfaces of layer 1 low SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

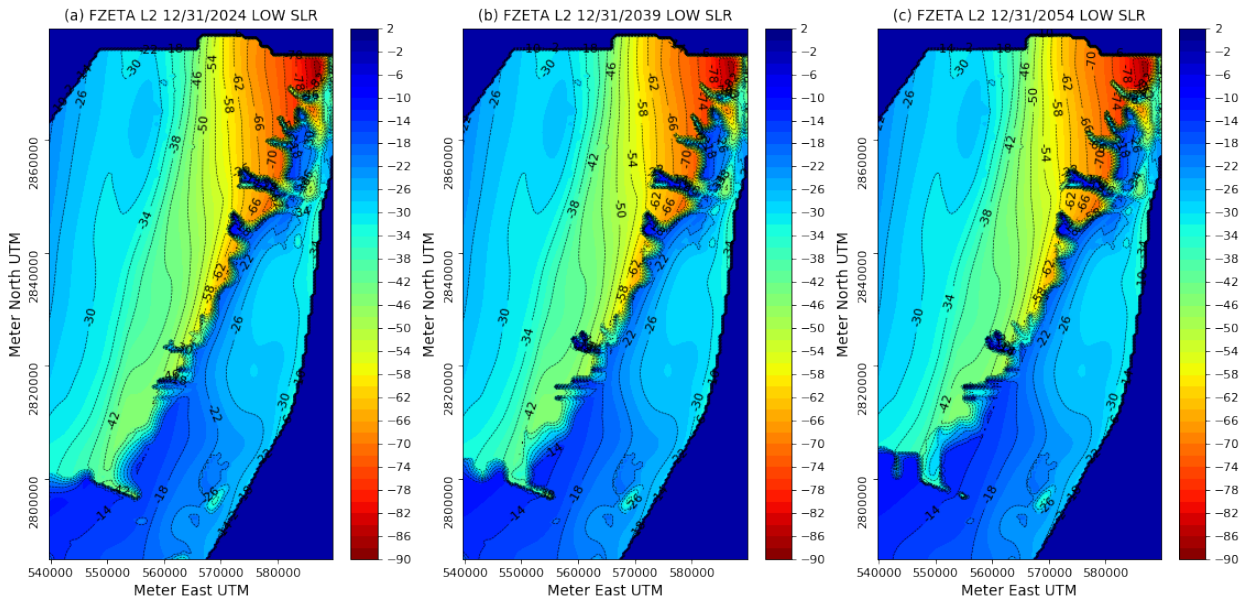


Figure 70. Zeta surfaces of layer 2 low SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

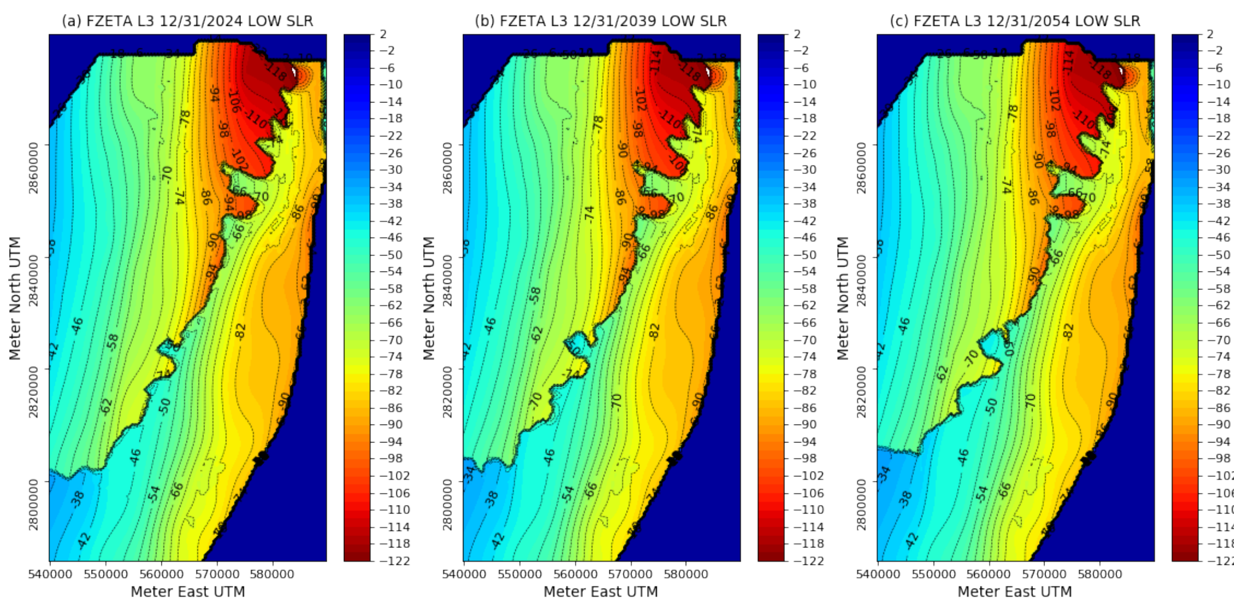


Figure 71. Zeta surfaces of layer 3 low SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

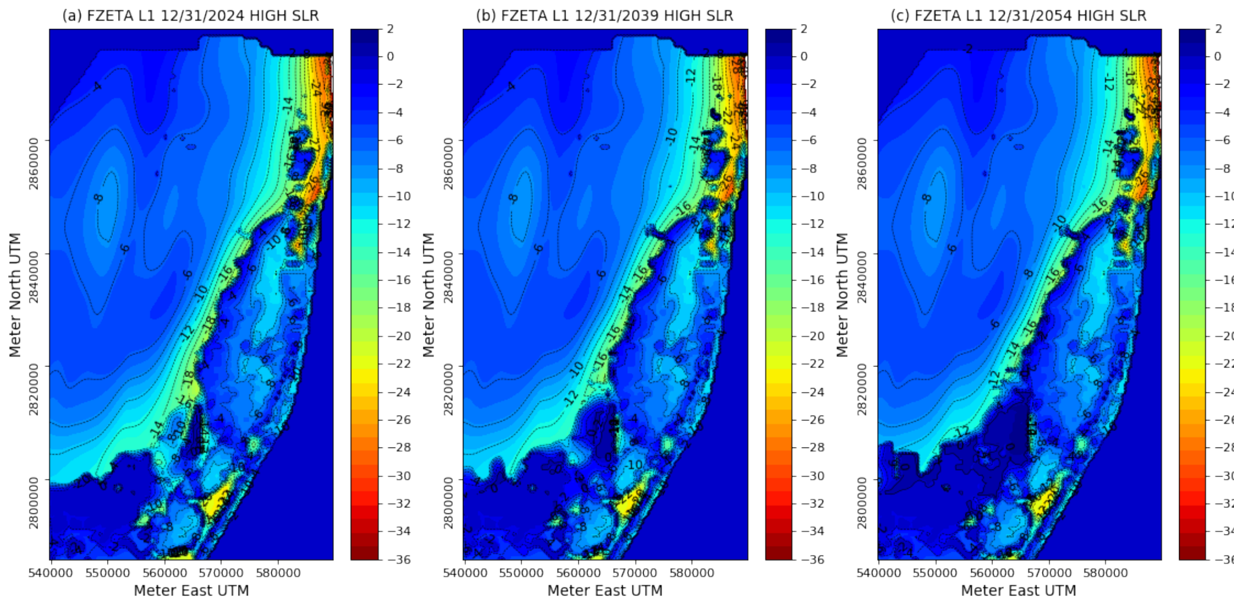


Figure 72. Zeta surfaces of layer 1 high SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

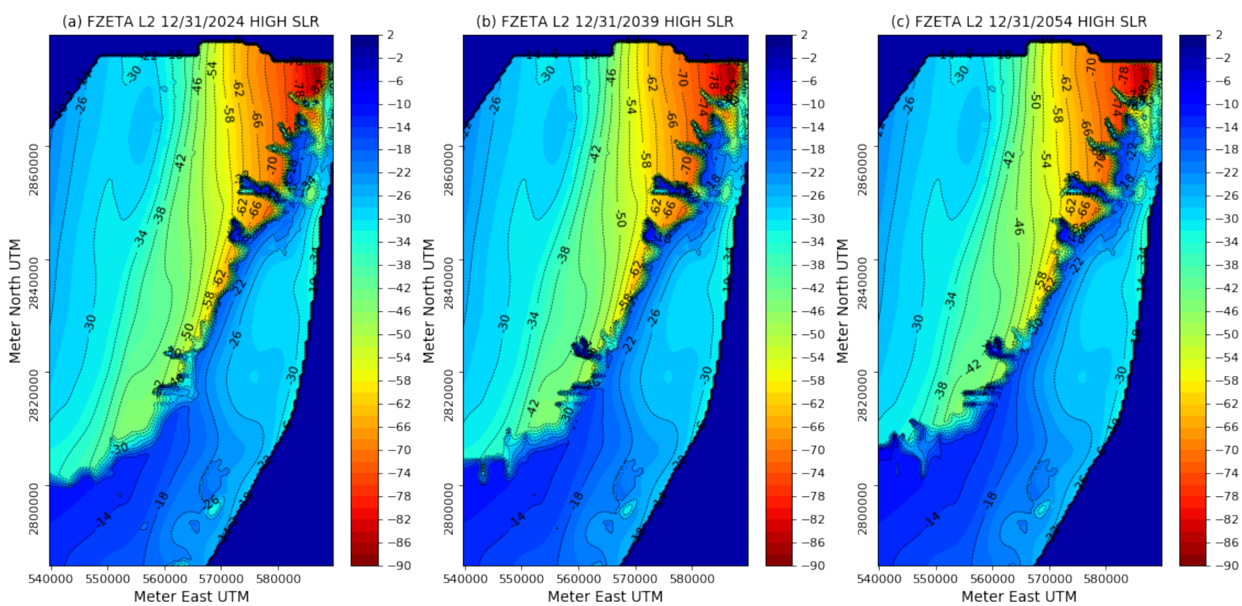


Figure 73. Zeta surfaces of layer 2 high SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

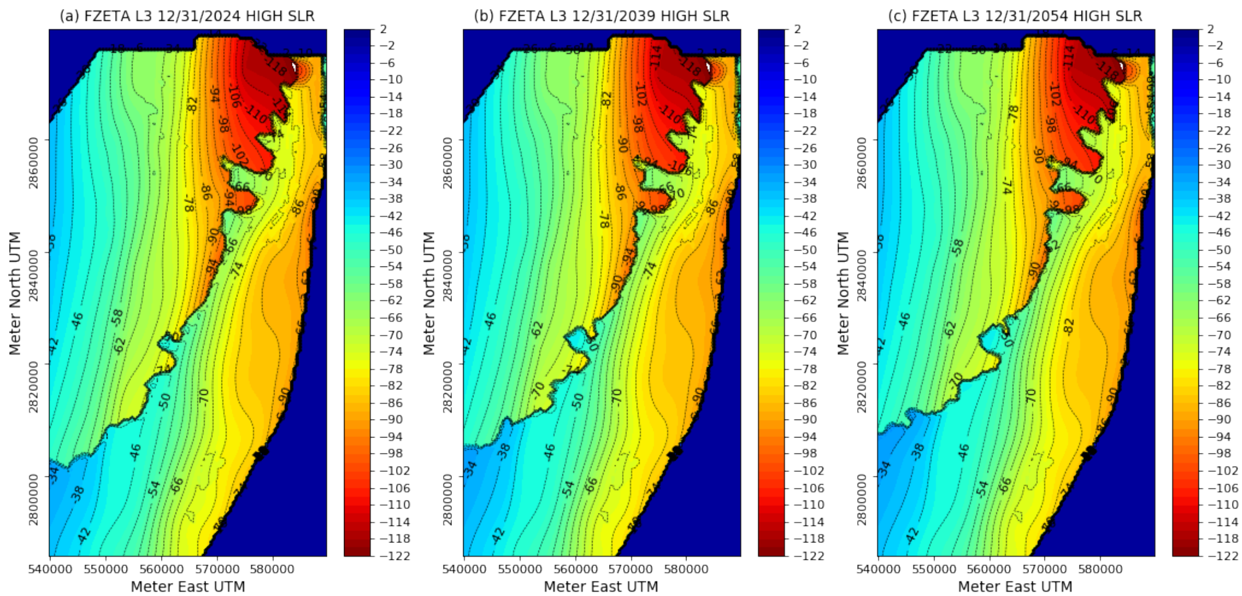


Figure 74. Zeta surfaces of layer 3 high SLR scenario for (a) 12/31/2024, (b) 12/31/2039, (c) 12/31/2054.

Saltwater intrusion zeta surface cross-sections

It can be helpful to view the Zeta surfaces in cross section. The sections show the extent of saltwater intrusion, with saltwater to the south and east of the near-vertical dashed lines near the centers of the sections.

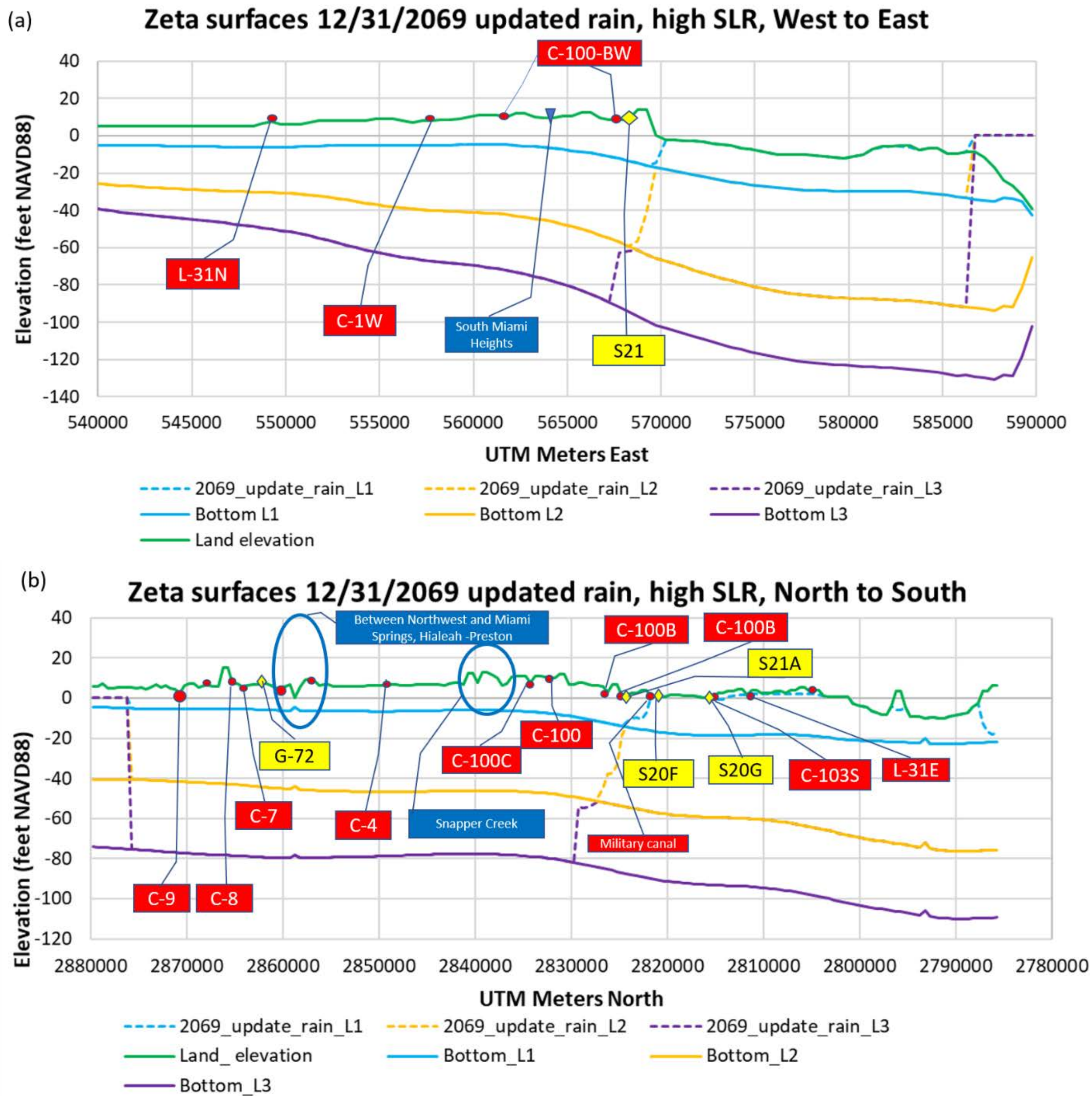


Figure 75. Cross sections showing Land elevation, bottom elevations of Layers 1, 2, and 3, and 2069 Zeta surfaces of Layers 1, 2, and 3 for high SLR scenario with updated rain. (a) West to East. (b) North to South. Location of cross-sections shown in Figure 66.

Results from main future scenario and sensitivity runs

Results from the main five model runs are summarized hereafter in terms of three major variables: (1) wet season average heads in the top layer of the model, (2) wet season average depth to the groundwater table, and (3) the spatial location of the freshwater/saltwater interface at the bottom of the three model layers at the end of the last dry season (May 31st) in the simulation. These results are presented as absolutes as well as differences from the calibration/verification run. Differences between the sensitivity runs and the two main scenario runs are also presented. Additional results are presented in the Initial conditions section.

The calibration/verification run encompasses a 15-year period from 1996-2010 where the first year was a warm-up period to reduce the influence of initial conditions, the period 1997-2004 was the model calibration period, and the period 2005-2010 was the model verification period. Results for the two main future scenario runs and the three sensitivity runs are presented for the 10-year period from 2060-2069. Wet season averages are over 2,760 simulation days in the calibration/verification run, and over 1,840 days in the future scenario and sensitivity runs.

Figure 76-Figure 77 and Table 8 show the simulated wet season average heads on the model's top layer on the left panels and the corresponding wet season average depth of the water table on the right panels for the calibration run and the two main future scenario runs with low and high sea level rise. Figure 78 and Table 8 show differences in the wet season average heads with respect to the calibration for each of these runs. Figure 79 and Table 9 show differences in the wet season average heads in the sensitivity runs (runs 3-5) compared to the base future scenario runs (runs 1 and 2).

The average wet season head map (Figure 76a) for the calibration period shows higher water levels in the Everglades as expected with heads up to 6.43 ft NAVD88 in the area, and a gradient towards the east-southeast reaching the wet season mean sea level of -0.70 ft NAVD88 downstream of the salinity control structures. The lowest simulated stages are at the cones of depression near the wellfields with the lowest simulated wet season average head of about -4.23 ft NAVD88. The areas where the water table is deeper is on the coastal ridge as expected, with lower depths to water table near the coast and the Southern Glades area and other areas of the Everglades that are ponded on average during the wet season.

The average wet season head map for the low SLR scenario (Figure 76c) shows the highest water levels in the Everglades of up to 7.07 ft NAVD88 decreasing towards the east-southeast and reaching the wet season mean sea level of 0.41 ft NAVD88 downstream of the salinity control structures. As shown in Figure 78a, heads are increased throughout the Everglades, especially in Northeast Shark River Slough on the northeastern side of Everglades National Park compared to the calibration run. This is due to the increased heads simulated for this area in the CERP0 run of the South Florida Water Management Model (SFWMM), which are being used as the western boundary condition in the future scenario runs. An exception is the Southern Glades region of ENP where heads are lower in the future scenario runs, consistent with CERP0 simulation. It is notable that the CERP0 simulation assumes historical predicted tides as boundary conditions and does not reflect the expected increases in sea level rise in the future. The model pre-processor takes the maximum of the local topography, Virginia Key stage, and CERP0 stages (or historical

EDEN stages in the case of the calibration run) in this area, converts them to equivalent freshwater heads if appropriate and uses this head in developing GHB or drain boundary conditions for each model grid cell.

As shown in Figure 82, it is evident how the EDEN timeseries (red trace) was controlling for the majority of the calibration run for this sample cell (row 160, column 5) due it being much higher than the Virginia Key historical water levels (green trace). However, in the low SLR run, both the CERP0 water levels (black trace) and the Virginia Key predicted tide (blue trace for IPCC AR5) are relatively close to each other and quite often lower than historical EDEN stages. Therefore, the heads used in defining GHB and drain boundary conditions are often lower in the low SLR scenario than in the original calibration run.

A *decrease in heads with respect to the calibration run* is also simulated in areas northeast of the C-111 canal near the S-197 structure in the low SLR scenario run. After further investigation, it was found that simulated heads in this region were greatly overestimated in the calibration run (Figure 83), whereas the low SLR scenario run often used GHBs and drain boundary conditions in this area due to most of it being inundated at the mean sea level for the last year of the simulation. For more information, see section Future freshwater/saltwater source regarding the model's isource variable. Therefore, caution is required when interpreting head *changes* in this area. Changes in simulated heads can also be observed near wellfields (Figure 89). Heads in the vicinity of the Southwest wellfield decrease due to increased pumpage, while heads in the vicinity of the Alexander Orr and Miami Springs-Hialeah-Preston wellfields increase due to decreased pumpage (Figure 56). Figure 67 shows the evolution of simulated heads for the initial condition runs and the low SLR scenario and sensitivity runs as east-west and nort-south cross-sections of simulated heads on the last day of each simulation.

The spatial location of the freshwater/saltwater interface at the bottom of the three model layers on May 31st of the last year of simulation are presented in Figure 80 for each future scenario and sensitivity run using the calibration run as a reference (base) run. There is a caveat that the LOCA run used in the future scenario and sensitivity runs was chosen as the 95th percentile of all future model runs; therefore, it is bound to underestimate the inland migration of the saltwater front if actual future rainfall were to decrease especially in the dry season. In addition, the two main future scenario runs and the no pumpage sensitivity run have a different rainfall sequence and RET than the calibration run. However, as observed in Figure 81b and d, the differences in the location of the interface are small between the two sensitivity runs with the same historical rainfall and RET as in the calibration run, and the corresponding future scenario runs using the corresponding SLR curve.

From Figure 80a, one can see how under the low SLR scenario, the salinity control structures are often able to hold the saltwater intrusion front to the east at the *bottom* of the top model layer (layer 1). However, saltwater starts intruding into this top layer near salinity control structures S-20G and S-20F in eastern portions of the C-103 and C-103N basins and into the Model Lands area (Figure 87 and Figure 88). The Aerojet canal and the C-111 canal seem to also be able to control the migration of the saltwater intrusion front at the bottom of this top layer. Significant inland migration of the saltwater intrusion line at the bottom of the aquifer (bottom of layer 3) is simulated in the Southern Glades, C-111 Basin, Model Lands, and eastern portions of the C-102, C-103 and C-103N basins. Some migration of the front at the bottom of the aquifer is also observed near canals in other areas of the model, especially near the Miami

Springs-Hialeah-Preston wellfields. Keeping those canals at higher stages might keep the saltwater front at bay in these areas at the expense of flood control capacity.

The average wet season head map for the high SLR scenario (Figure 76d) shows the highest water levels in the Everglades of up to 7.07 ft NAVD88 decreasing towards the east-southeast and reaching the wet season mean sea level of 1.57 ft NAVD88 downstream of the salinity control structures. As shown in Figure 78b, heads are increased throughout the Everglades, especially in Northeast Shark River Slough on the northeastern side of Everglades National Park compared to the calibration run. This is due to the increased heads simulated for this area in the CERP0 run of the South Florida Water Management Model (SFWMM), which are being used as the western boundary condition in the future scenario runs. As explained above, heads in the southern Glades region southwest of the C-111 canal are also likely underestimated in the high SLR scenario even when stages at Virginia Key under USACE High SLR scenario are much higher than those simulated by CERP0 and most historical EDEN stages (Figure 82). The simulated head *increase* in areas northeast of the C-111 canal near the S-197 structure *with respect to the calibration run* is likely underestimated for the same reasons discussed above for the low SLR scenario. Changes in simulated heads with respect to the calibration run can also be observed near particular wellfields (Figure 89) in the same direction as in the low SLR scenario. Figure 68 shows the evolution of simulated heads for the initial condition runs and the high SLR scenario and sensitivity runs as east-west and nort-south cross-sections of simulated heads on the last day of each simulation.

Figure 79a shows the difference in wet season average heads in the high SLR scenario minus the low SLR scenario. As expected, differences in head are zero on the Everglades, where both scenarios use the same CERP0 boundary condition, increasing to about 1.26 ft near the coast, reflecting the different tidal boundary conditions used in the two SLR scenarios.

From Figure 80b, one can see how under the high SLR scenario, the salinity control structures are generally able to hold the saltwater intrusion front to the east at the *bottom* of the top model layer (layer 1). However, saltwater starts intruding into this top layer near salinity control structures S-21A, S-20G and S-20F in the eastern portions of the C-102, C-103, C-103N basins and into the Model Lands area (Figure 87 and Figure 88). From Figure 81a, it is evident that the saltwater front migrates even further inland than in the low SLR run. Contrary to the low SLR scenario, the Aerojet canal and the C-111 canal do not seem to be able to control the migration of the saltwater intrusion front at the bottom of this top model layer in the high SLR scenario. Significant inland migration of the saltwater intrusion line at the bottom of the aquifer (bottom of layer 3) is simulated in the Southern Glades, C-111 Basin, Model Lands, and eastern portions of the C-102, C-103 and C-103N basins. The simulated migration of the saltwater intrusion front at the bottom of the aquifer in the high SLR scenario is very similar to that of the low SLR scenario run for areas north of the S-123 structure. Figure 75 shows cross sections showing the location of the freshwater/saltwater interface in the three model layers at the end of the simulation period for the high SLR scenario.

Comparison of Figure 76c versus Figure 77c, and Figure 76e versus Figure 77e for the low and high SLR scenarios and corresponding historical rainfall and RET sensitivity runs, respectively, show very small differences. This can be confirmed from Figure 79b and d, and Table 8. Average wet season heads are lower throughout the mainland in both the low SLR and high SLR sensitivity runs with historical rainfall and RET. This means that the increase in rainfall imposed on the base low and high SLR future scenario runs is able to counteract the 5% imposed increase in RET for a net increase in available water in the system compared to historical conditions in the calibration/verification run. Heads do increase in the Miami Beach-Key Biscayne area as a result of the historical rainfall; however, we caution about interpretations in this region due to both the LOCA grid and the SFWMM grids not including this area and some extrapolations being performed in developing a rainfall timeseries for this area.

As mentioned previously, the changes in the location of the saltwater/freshwater interface between the runs with historical rainfall and RET and the base future scenario runs (Figure 81b and d) are negligible. There is a caveat that the LOCA run used in the future scenario and sensitivity runs was chosen as the 95th percentile of all future model runs; therefore, it is bound to underestimate the inland migration of the saltwater front if actual future rainfall were to decrease especially in the dry season.

Results for the worst-case sensitivity run (run 3) with high SLR and no pumpage can be seen in Figure 77a and the differences with respect to the calibration and high SLR run with future pumpage are shown in Figure 78c and Figure 79e, respectively. Large increases in heads are observed near the now non-existent cones of depression near wells with differences up to 6.0 ft. Increased heads of 0.5 ft or more compared to the high SLR scenario run persist quite a large distance from the wells. Surprisingly, Figure 81e shows a very small effect of wellfield pumpage on the location of the freshwater/saltwater interface at the end of the last modeled last season. As mentioned earlier, decreased rainfall especially during the dry season, could have major impacts on the location of the interface, so these figures should be interpreted with caution and not used in future planning or policy decisions.

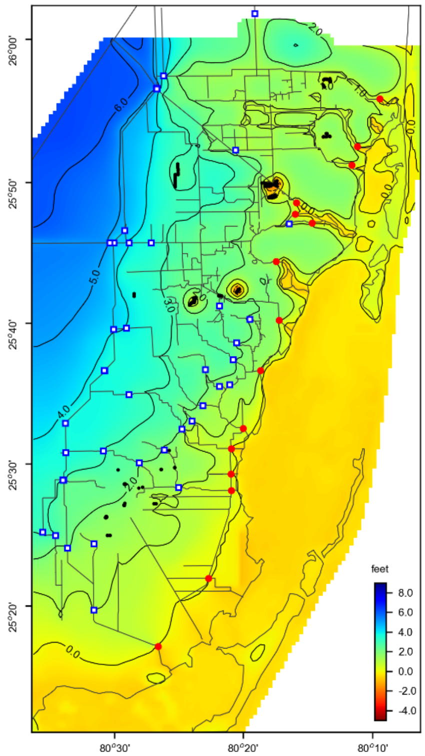
Table 8. Spatial range and spatial average of wet season average heads in the calibration and future scenario runs.

Run short-name	Range of wet season average head (ft NAVD88)	Average of wet season average head (ft NAVD88)	Range of difference in wet season average heads with respect to calibration (ft)	Average difference in wet season average heads with respect to calibration (ft)
CALIBRATION	-4.23 to +6.43	+1.67	-	-
(1) LOW SLR	-2.58 to +7.07	+2.50	-0.71 to +3.01	+0.69
(2) HIGH SLR	-1.83 to +7.07	+3.19	-0.20 to +3.57	+1.27
(3) HIGH SLR + NO PUMPAGE	+0.77 to +7.07	+3.34	-0.16 to +8.50	+1.38
(4) LOW SLR + HIST RAIN/RET	-2.86 to +7.05	+2.43	-1.14 to +2.61	+0.63
(5) HIGH SLR + HIST RAIN/RET	-2.11 to +7.06	+3.14	-0.59 to +3.21	+1.22

Table 9. Spatial range of differences in wet season average heads between future scenario and sensitivity runs.

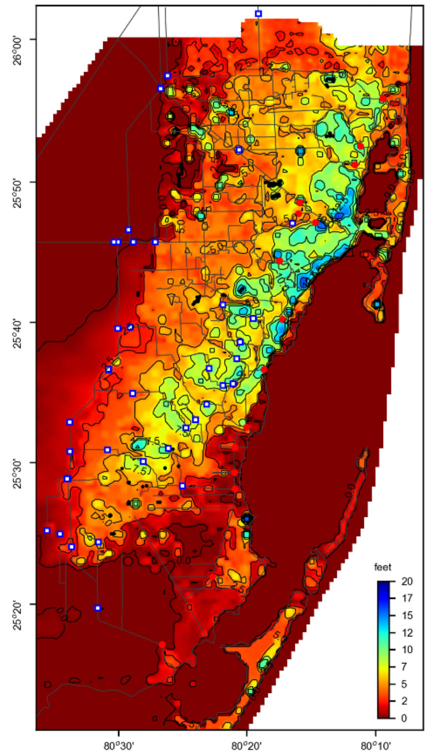
Runs compared	Range of difference in wet season average heads (ft)	Average difference in wet season average heads (ft)
HIGH SLR – LOW SLR	-0.13 to +1.26	+0.58
(LOW SLR + HIST RAIN/RET) – LOW SLR	-0.43 to +0.58	-0.06
(HIGH SLR + HIST RAIN/RET) – (LOW SLR + HIST RAIN/RET)	-0.06 to +1.22	+0.59
(HIGH SLR + HIST RAIN/RET) – (HIGH SLR)	-0.40 to +0.51	-0.05
(HIGH SLR + NO PUMPAGE) – HIGH SLR	-0.12 to +6.10	+0.12

Wet season average heads (ft NAVD88)
CALIBRATION (1996-2010)



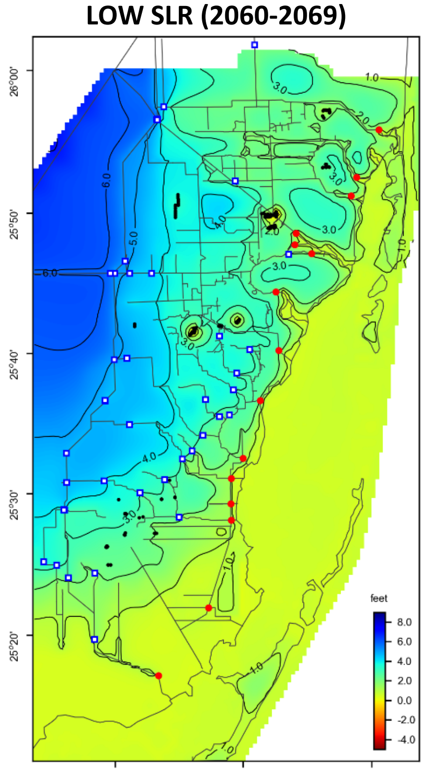
(a)

Wet season average depth to water table (ft)
CALIBRATION (1996-2010)



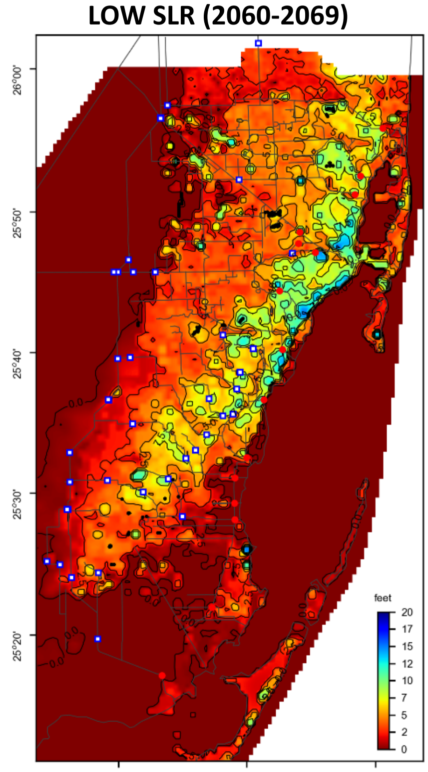
(b)

Wet season average heads (ft NAVD88)
LOW SLR (2060-2069)



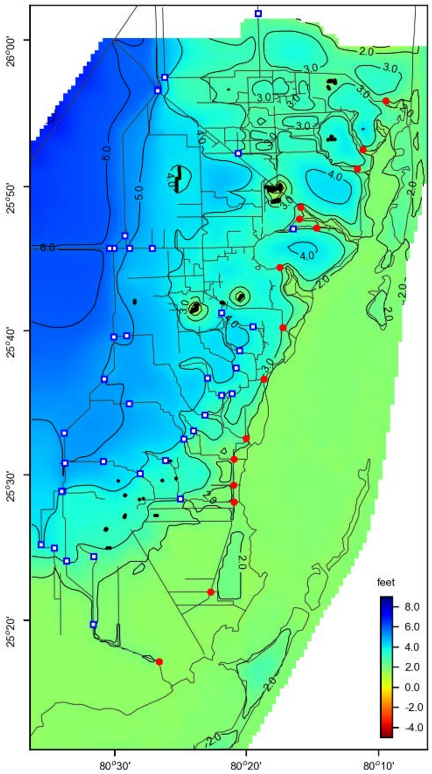
(c)

Wet season average depth to water table (ft)
LOW SLR (2060-2069)



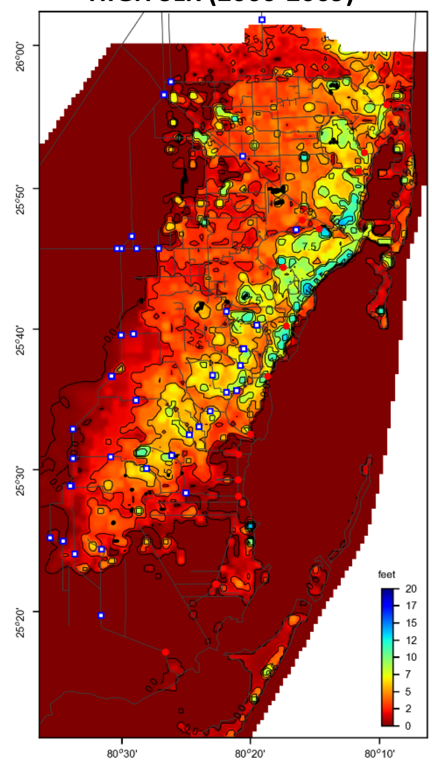
(d)

Wet season average heads (ft NAVD88)
HIGH SLR (2060-2069)



(e)

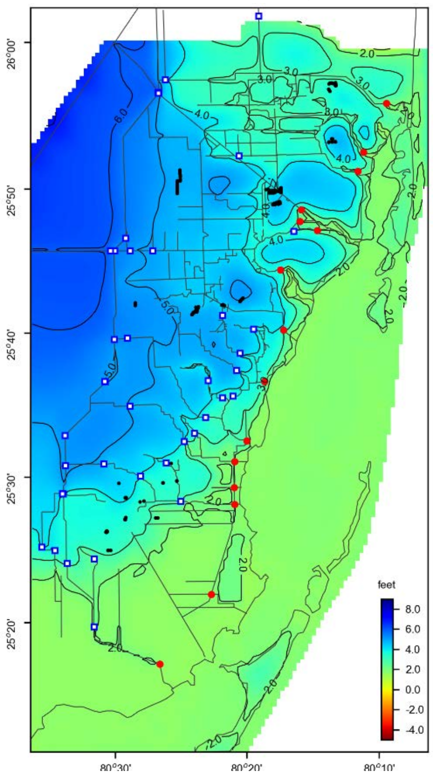
Wet season average depth to water table (ft)
HIGH SLR (2060-2069)



(f)

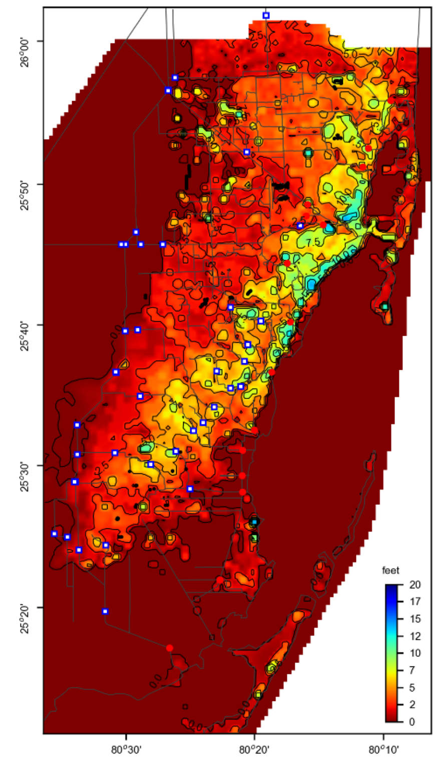
Figure 76. Left panel: Average wet season heads (ft NAVD88) for (a) Calibration run (1996-2010), (c) Low SLR run (2060-2069), (e) High SLR run (2060-2069). Right panel (b), (d), (f): Average wet season depth to water table (ft) for the same runs.

Wet season average heads (ft NAVD88)
HIGH SLR + NO PUMPAGE (2060-2069)



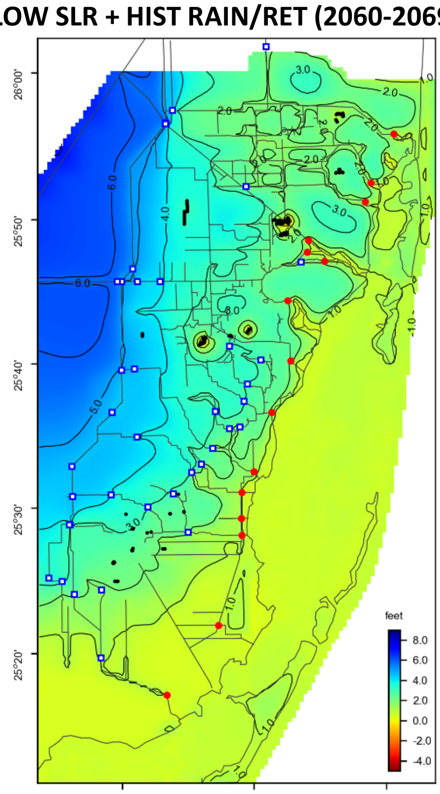
(a)

Wet season average depth to water table (ft)
HIGH SLR + NO PUMPAGE (2060-2069)



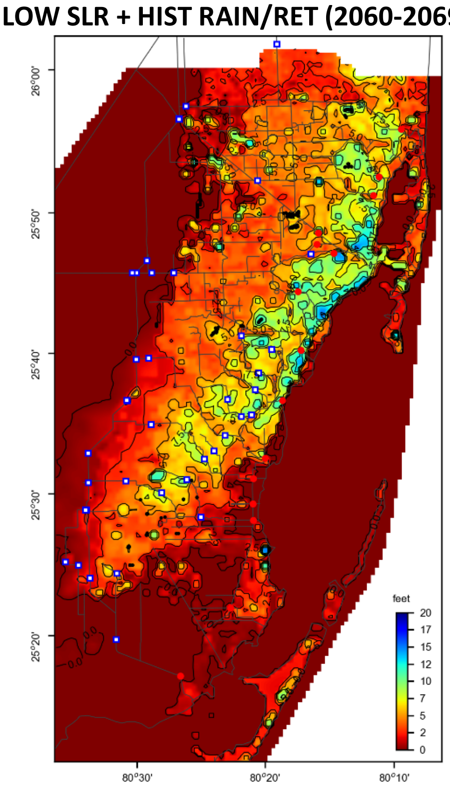
(b)

Wet season average heads (ft NAVD88)
LOW SLR + HIST RAIN/RET (2060-2069)



(c)

Wet season average depth to water table (ft)
LOW SLR + HIST RAIN/RET (2060-2069)



(d)

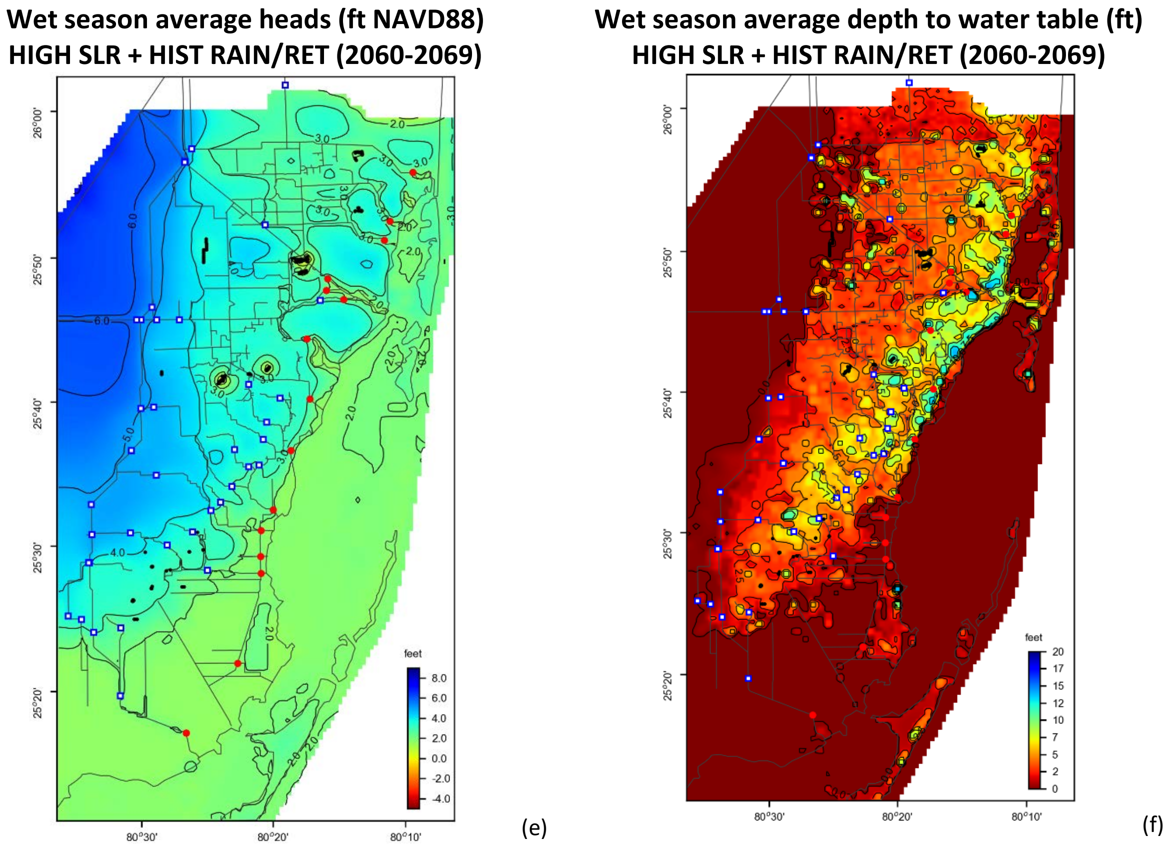
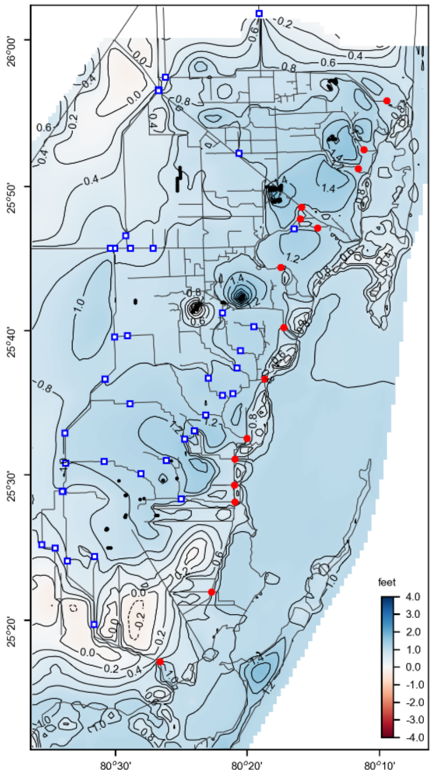


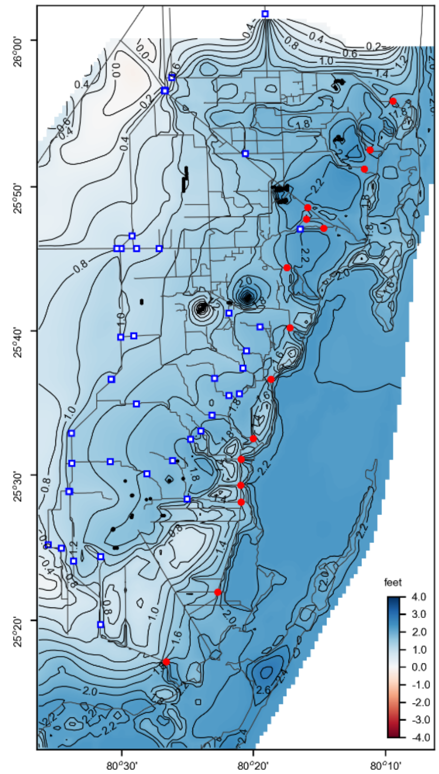
Figure 77. Left panel: Average wet season heads (ft NAVD88) for (a) High SLR + no pumpage run (2060-2069), (b) Low SLR + historical rainfall and RET run (2060-2069), (e) High SLR + historical rainfall and RET run (2060-2069). Right panel (b), (d), (f): Average wet season depth to water table (ft) for the same runs.

Difference in Wet season average heads (ft)
LOW SLR – CALIBRATION



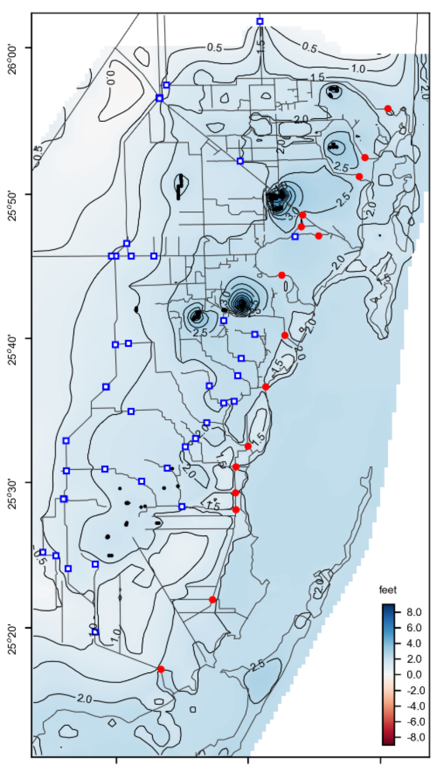
(a)

Difference in Wet season average heads (ft)
HIGH SLR – CALIBRATION



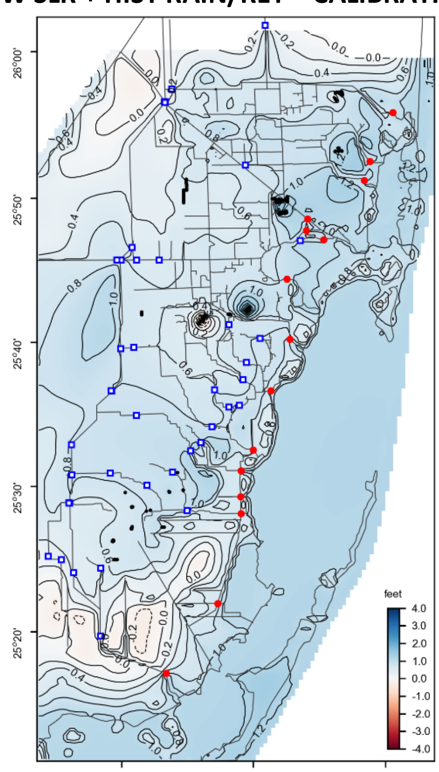
(b)

Difference in Wet season average heads (ft)
HIGH SLR + NO PUMPAGE – CALIBRATION



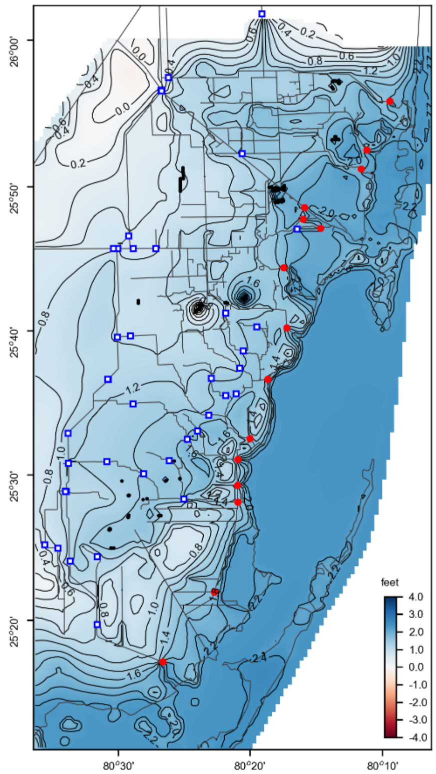
(c)

Difference in Wet season average heads (ft)
LOW SLR + HIST RAIN/RET – CALIBRATION



(d)

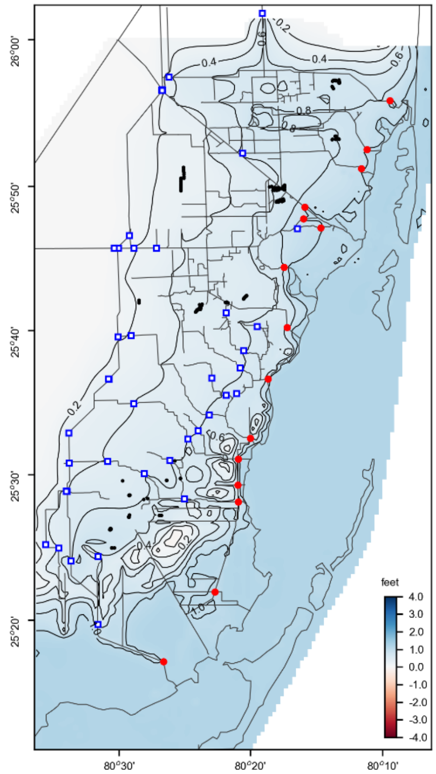
Difference in Wet season average heads (ft)
HIGH SLR + HIST RAIN/RET – CALIBRATION



(e)

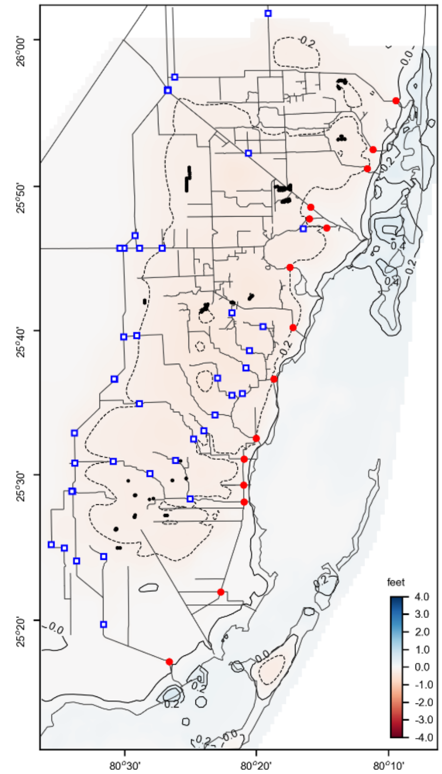
Figure 78. Difference in average wet season heads (ft) between each of the following runs (2060-2069) and the calibration run (1996-2010): (a) Low SLR run, (b) High SLR run, (c) High SLR + no pumpage run, (d) Low SLR + historical rainfall and RET run, (e) High SLR + historical rainfall and RET run. Note different scale in panel (c). Cool colors reflect higher heads in the scenario run than in the calibration run.

Difference in Wet season average heads (ft)
HIGH SLR – LOW SLR



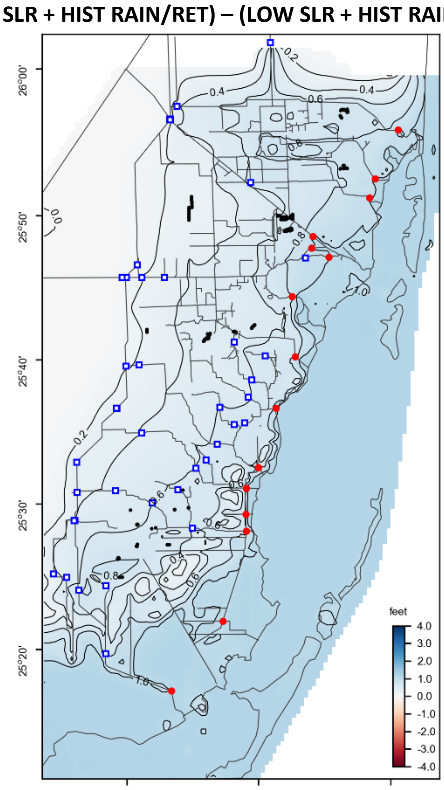
(a)

Difference in Wet season average heads (ft)
(LOW SLR + HIST RAIN/RET) – LOW SLR



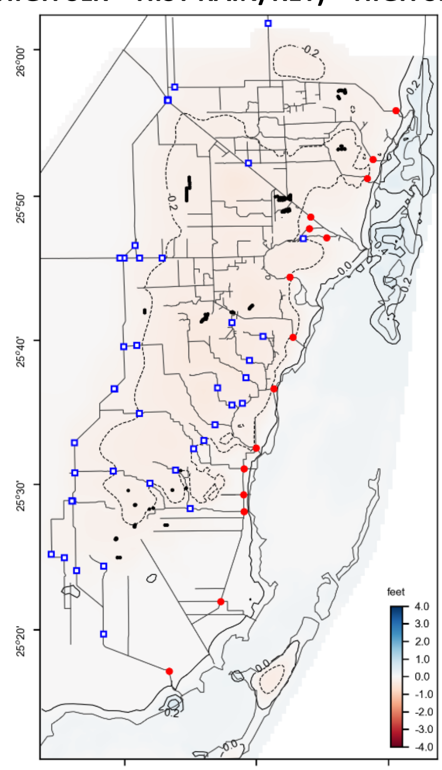
(b)

Difference in Wet season average heads (ft)
(HIST SLR + HIST RAIN/RET) – (LOW SLR + HIST RAIN/RET)



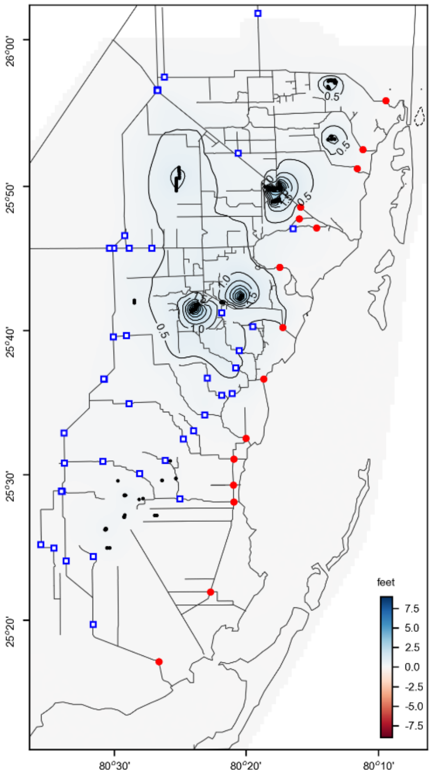
(c)

Difference in Wet season average heads (ft)
(HIGH SLR + HIST RAIN/RET) – HIGH SLR



(d)

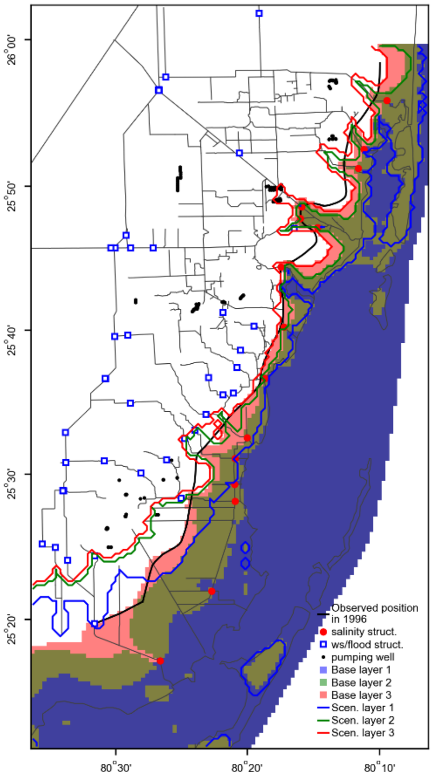
Difference in Wet season average heads (ft)
(HIGH SLR + NO PUMPAGE) – HIGH SLR



(e)

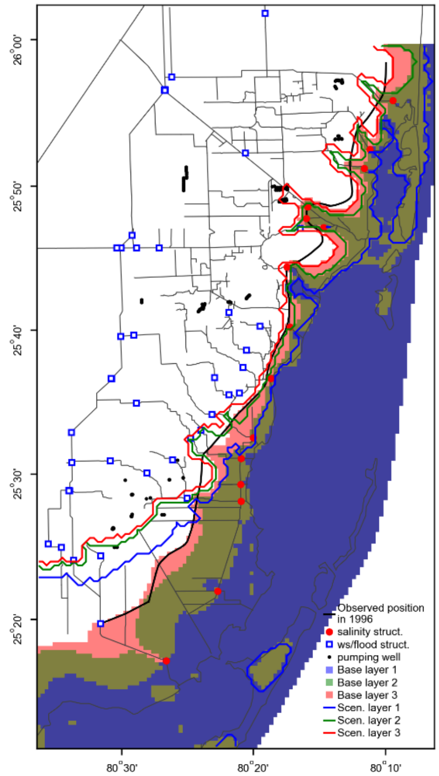
Figure 79. Difference in average wet season heads (ft) between the following runs (2060-2069): (a) High SLR run minus Low SLR run, (b) Low SLR + historical rainfall and RET minus Low SLR run, (c) High SLR + historical rainfall and RET minus Low SLR + historical rainfall and RET, (d) High SLR + historical rainfall and RET minus High SLR run, and (e) High SLR + no pumpage minus High SLR. Note different scale in panel (e). Cool colors reflect higher heads in the first run than in the second run.

**Difference in freshwater/saltwater interface
LOW SLR vs. CALIBRATION**



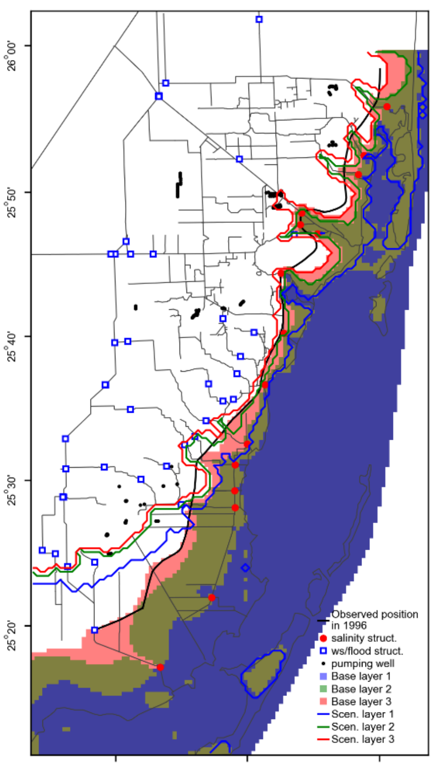
(a)

**Difference in freshwater/saltwater interface
HIGH SLR vs. CALIBRATION**



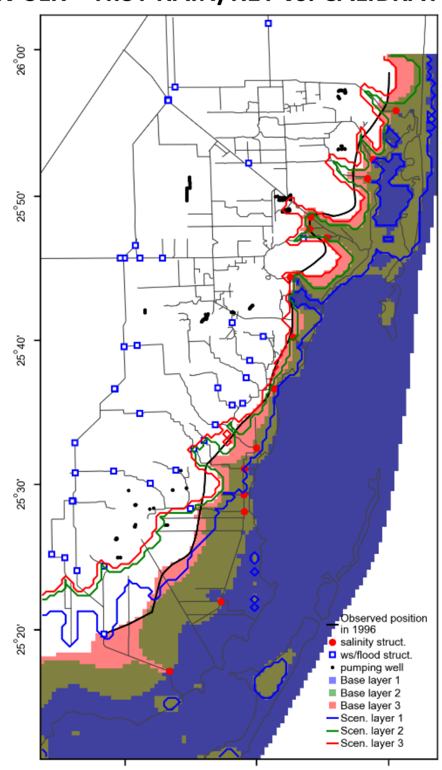
(b)

**Difference in freshwater/saltwater interface
HIGH SLR + NO PUMPAGE vs. CALIBRATION**



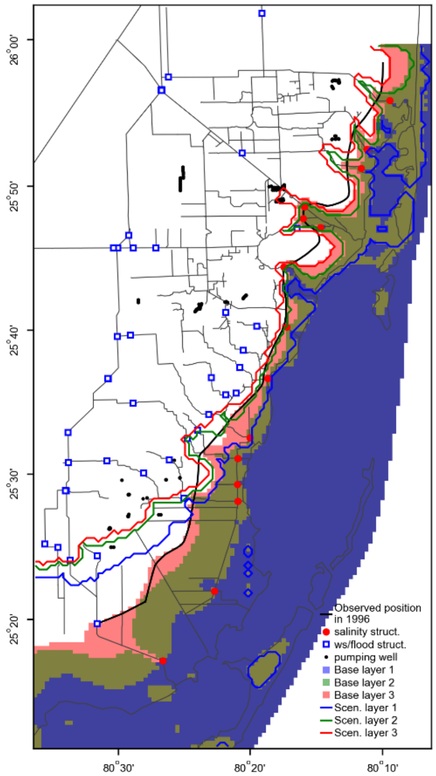
(c)

**Difference in freshwater/saltwater interface
LOW SLR + HIST RAIN/RET vs. CALIBRATION**



(d)

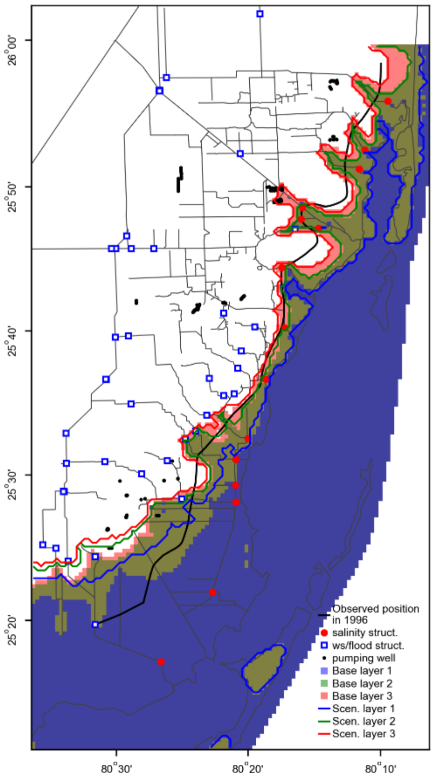
Difference in freshwater/saltwater interface
HIGH SLR + HIST RAIN/RET vs. CALIBRATION



(e)

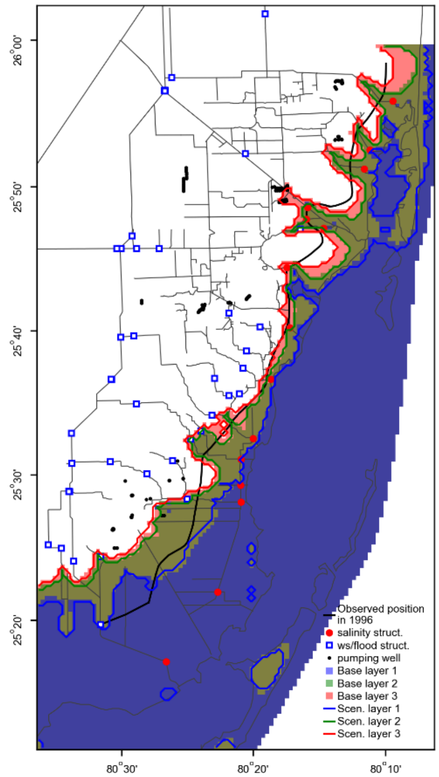
Figure 80. Simulated change in the position of the freshwater-saltwater interface from the calibration run (1996-2010, labelled as 'Base') in the following runs (each labelled as 'Scen' in their own plot): (a) Low SLR run, (b) High SLR run, (c) High SLR + no pumpage run, (d) Low SLR + historical rainfall and RET run, (e) High SLR + historical rainfall and RET run, at the end of the dry season (May 31st). Note that the runs in panels (a), (b) and (c) have a different rainfall sequence and RET than the calibration run. The observed position in 1996 of the interface at the bottom of the Biscayne aquifer (Sonenshein, 1997), corresponding to the bottom of layer 3 in the model, is shown in black.

Difference in freshwater/saltwater interface
HIGH SLR vs. LOW SLR



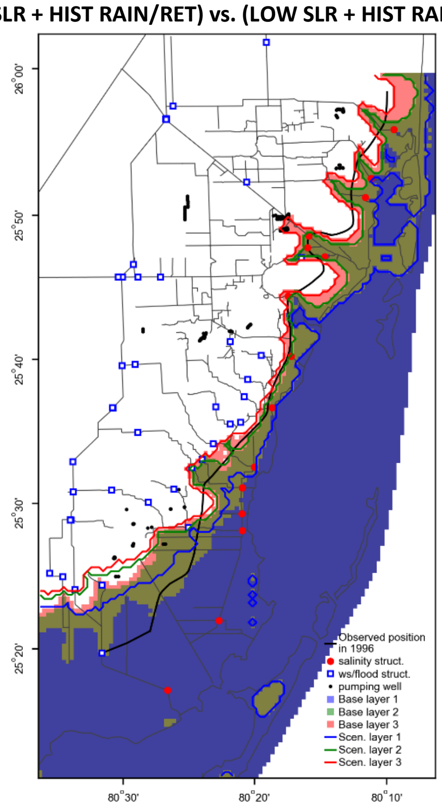
(a)

Difference in freshwater/saltwater interface
(LOW SLR + HIST RAIN/RET) vs. LOW SLR



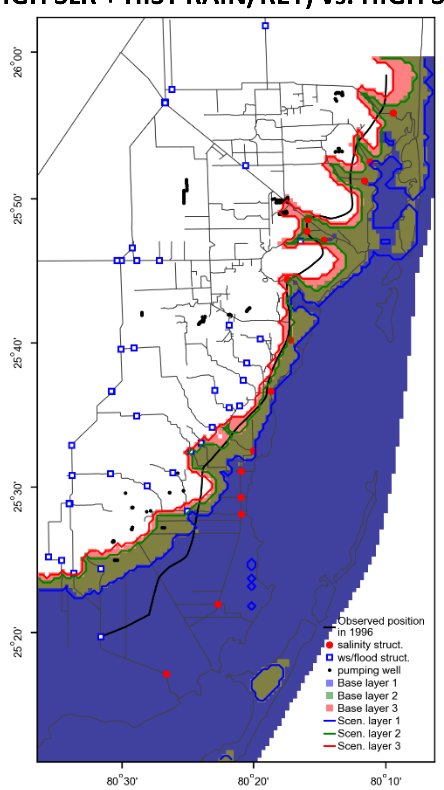
(b)

Difference in freshwater/saltwater interface
(HIST SLR + HIST RAIN/RET) vs. (LOW SLR + HIST RAIN/RET)



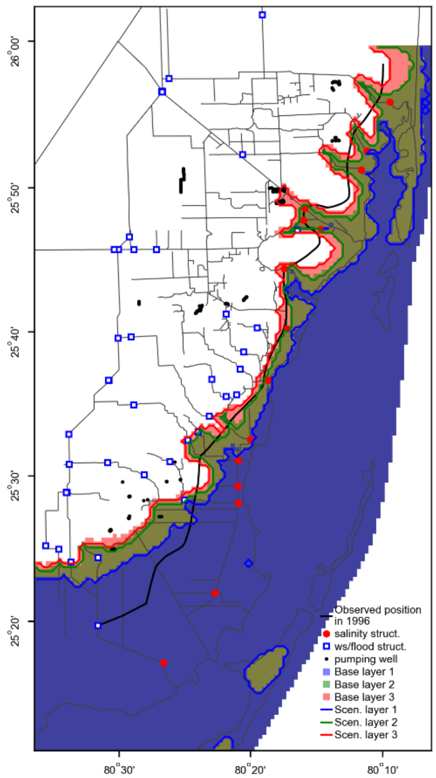
(c)

Difference in freshwater/saltwater interface
(HIGH SLR + HIST RAIN/RET) vs. HIGH SLR



(d)

Difference in freshwater/saltwater interface
(HIGH SLR + NO PUMPAGE) vs. HIGH SLR



(e)

Figure 81. Simulated change in the position of the freshwater-saltwater interface for: (a) High SLR run versus Low SLR run, (b) Low SLR + historical rainfall and RET versus Low SLR run, (c) High SLR + historical rainfall and RET versus Low SLR + historical rainfall and RET, (d) High SLR + historical rainfall and RET versus High SLR run, and (e) High SLR + no pumpage versus High SLR, at the end of the dry season (May 31st). The first run is labelled 'Scen.' in each respective plot, the 2nd run is labelled 'Base.' The observed position in 1996 of the interface at the bottom of the Biscayne aquifer (Sonenshein, 1997), corresponding to the bottom of layer 3 in the model, is shown in black.

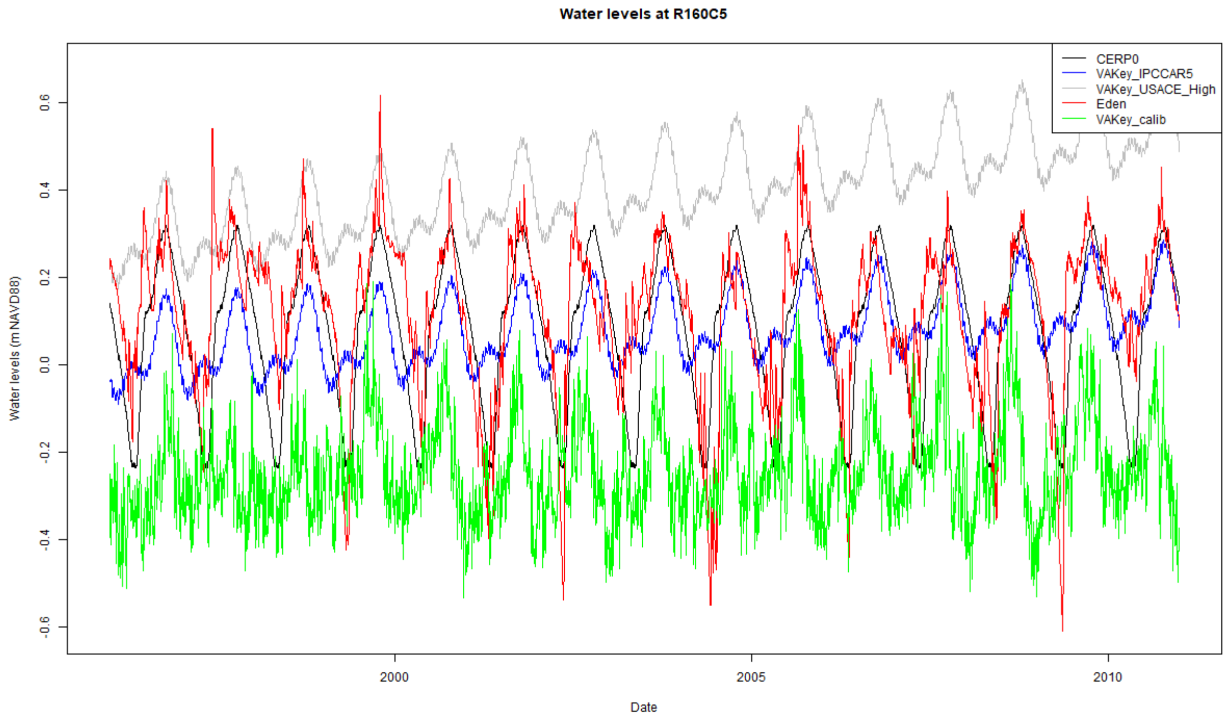


Figure 82. Comparison of various water levels used in setting boundary conditions in the southern Glades for a particular model grid cell (row 160, column 5). Red line represents the historical EDEN water levels, the black line represents the annually-repeating CERP0 timeseries for this cell, the green line represents historical tidal water levels at Virginia Key. The blue and grey lines represent the projected tidal timeseries with sea level rise for the low SLR (IPCC AR5) and high SLR (USACE High) scenarios, respectively. Units are m NAVD88.

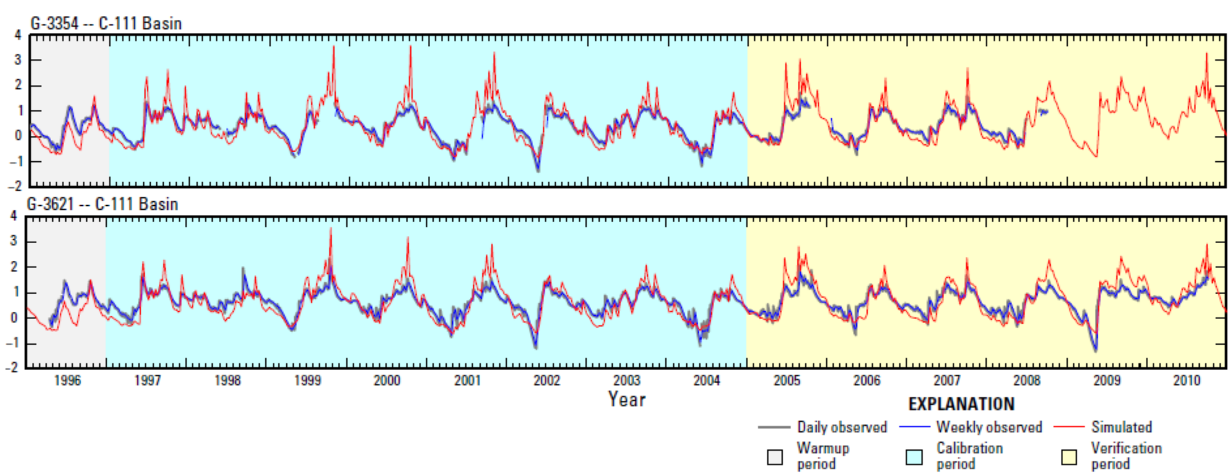


Figure 83. Observed and simulated water levels in the calibration/verification run for wells in the C-111 Basin. From Figure 5-5 Hughes and White (2016), with permission. Vertical units are ft NAVD88.

Model limitations and recommendations

The following modeling limitations and recommendations may improve future modeling:

1. There is a caveat that the LOCA run used in the future scenario and sensitivity runs was chosen as the 95th percentile of all future model runs; therefore, it is bound to underestimate the inland migration of the saltwater front if actual future rainfall were to decrease (especially in the dry season). It is notable that 70% of the LOCA model runs evaluated predict a decrease in wet season rainfall, while 30% predict an increase. This is consistent with previous studies by Obeysekera et al. (2014). It is also consistent with findings by Kirtman and others (FIU Rainfall Workshop, May 16, 2019) who evaluated the US Bureau's BCSD statistically-downscaled climate data product and found that most models projected a drying of south Florida in the future.
2. There are various limitations in using the CERP0 SFWMM run to provide the western boundary conditions in the future scenario and sensitivity runs performed as part of this project. First, CERP0 assumes no sea level rise, which results in the simulated stages in the Southern Glades being too low. Second, CERP0 uses a historical rainfall timeseries, which is different from the LOCA rainfall being used in the future scenario and sensitivity runs. For this reason, we used Julian-day-average water levels repeated every year of the simulation as boundary conditions in our run. This smooths out peaks and valleys. It would be advisable to partner with the South Florida Water Management District (SFWMD) in the future to make various runs of CERP0 using various future sea level rise and rainfall projections. The Unified Sea Level Rise (SLR) Projections developed by the Southeast Florida Regional Climate Change Compact (2015), some of which are used in this project, should be used as boundary conditions in future CERP0 SFWMM scenario runs. Rainfall from LOCA or other downscaled model products (after going through a model culling exercise based on retrospective run performance) could be used to define scenarios bracketing future projected changes in rainfall. The recent rainfall workshop at FIU (sponsored by the SFWMD) on May 16, 2019, aims to provide a strategy for the development of a unified set of rainfall scenarios for the state.
3. The Virginia Key timeseries used in this project are only future tidal predictions shifted along one of two sea level rise curves and do not include meteorological effects. In the future, it would be advisable to incorporate meteorological effects in the oceanic boundary timeseries, which could be provided from hydrodynamic models or be synthetically derived.
4. The 1-D surface water network, structures, effective gate openings, and specified pump discharges remain the same as in the USGS 1996-2010 calibration/verification of the model. Other than directing the directly-connected impervious area fraction (DCIA) of rainfall from the grid cells to the canal system (as a pre-processed timeseries), the only other interaction between the canals and the model grid is through canal bed leakance into and from the groundwater. At the moment there is no 2-D surface water modeling capability in this model and no overbank or structure flows are allowed from ponded areas to the 1-D surface water network. Therefore, as the groundwater levels go up, water levels in the canals are expected to go up since the structure gates and tidal tailwater conditions are constraining flows out of the system (see Figure 90 for an example). Although conservative, this may be constraining the simulated future changes in heads

throughout the county since adaptation measures may be implemented in the future to increase the canal system capacity. For example, structure gates may be operated differently than historically, structures may be retrofitted to increase their capacities, forward pumps may be installed at the salinity control structures, and canals and structures may be protected by impermeable levees or dikes. In addition, the northern boundary condition is based on historical stages, which constrains head increases in the northern portion of the model.

In the future, canals may also have to be operated at higher levels in order to keep saltwater intrusion at bay especially in the dry season, which might affect groundwater heads and flood control capabilities of the system. The surface water package (SWR1) used in this model could, in theory, be set up to maintain canal stages at certain user-specified levels, instead of using historical effective gate opening information. However, this feature has not been tested and is likely to require much smaller timesteps than the current daily model timestep for stability, which would result in much longer run times. In addition, SWR1 could also be used with the unsaturated zone package UZF1 to send groundwater discharge to the land surface and excess infiltration to the canal reaches. Modeling these potential future adaptation and policy changes to the water management system is beyond the scope of this modeling effort.

References

Bakker, Mark, Schaars, Frans, Hughes, J.D., Langevin, C.D., and Dausman, A.M. 2013. Documentation of the seawater intrusion (SWI2) package for MODFLOW: U.S. Geological Survey Techniques and Methods, book 6, chap. A46, 47 p., http://pubs.usgs.gov/tm/6a46/.

Cannon A. J., Stephen R. S., Murdock T. Q. 2015. Bias correction of GCM precipitation by quantile mapping: How well do methods preserve changes in quantiles and extremes? Journal of Climate 28:6938-6959.

Codiga, D. L., 2011. Unified Tidal Analysis and Prediction Using the UTide MATLAB Functions. Technical Report 2011-01. Graduate School of Oceanography, University of Rhode Island, Narragansett, RI. 59pp. ftp://www.po.gso.uri.edu/ pub/downloads/codiga/pubs/2011Codiga-UTide-Report.pdf

Hwang, S. and Graham, W. D. 2013. Development and comparative evaluation of a stochastic analog method to downscale daily GCM precipitation, Hydrol. Earth Syst. Sci., 17, 4481–4502, doi:10.5194/hess-17-4481-2013.

Hughes, J.D., and White, J.T., 2016, Hydrologic conditions in urban Miami-Dade County, Florida, and the effect of groundwater pumpage and increased sea level on canal leakage and regional groundwater flow (ver. 1.2, July 2016): U.S. Geological Survey Scientific Investigations Report 2014–5162, 175 p., http://dx.doi.org/10.3133/sir20145162.

Hughes, J.D., Langevin, C.D., Chartier, K.L., and White, J.T., 2012, Documentation of the Surface-Water Routing (SWR1) Process for modeling surface-water flow with the U.S. Geological Survey Modular Ground-Water Model (MODFLOW–2005): U.S. Geological Survey Techniques and Methods, book 6, chap. A40 (Version 1.0), 113 p. https://doi.org/10.3133/tm6A40.

Irizarry M., Obeysekera J., Dessalegne T. 2016. Determination of Future Intensity-Duration-Frequency Curves for Level of Service Planning Projects. Task 2 - Deliverable 2.1 to SFWMD - Conduct an extreme rainfall analysis in climate model outputs to determine temporal changes in IDF curves.

Keith and Schnars, 2004, South Miami-Dade watershed study and plan—Sub-task 1.5: Consultant report prepared for the South Florida Regional Planning Council, Miami-Dade County, and the South Florida Water Management District.

Maurer E. P., Brekke L., Pruitt T., Duffy P. B. 2007. Fine-resolution climate projections enhance regional climate change impact studies. Eos Trans. AGU, 88(47), 504.

Obeysekera, J., Barnes, J., and Nungesser, M. 2014. Climate sensitivity runs and regional hydrologic modeling for predicting the response of the Greater Florida Everglades Ecosystem to climate change. Environmental Management. DOI 10.1007/s00267-014-0315-x.

Obeysekera J., Irizarry M., Park J., Barnes J., Dessalegne, T. 2010. Climate Change and its Implications for Water Resources Management in South Florida.

Panofsky H. W., Brier G. W. 1968. Some Applications of Statistics to Meteorology. The Pennsylvania State University Press: Philadelphia.

Pierce D. W., Cayan D. R., Thrasher B. L. 2014. Statistical downscaling using Localized Constructed Analogs (LOCA). Journal of Hydrometeorology 15:2558–2585.

Reclamation. 2013. Downscaled CMIP3 and CMIP5 Climate and Hydrology Projections: Release of Downscaled CMIP5 Climate Projections, Comparison with preceding Information, and Summary of User Needs. U.S. Department of the Interior, Bureau of Reclamation, Technical Services Center, Denver, Colorado. 47pp.

Sonenshein, R.S. 1997. Delineation and extent of saltwater intrusion in the Biscayne aquifer, Eastern Dade County, Florida, 1995: U.S. Geological Survey Water Resources Investigations Report 96-4285, 1 sheet.

Southeast Florida Regional Climate Change Compact Sea Level Rise Work Group (Compact). October 2015. Unified Sea Level Rise Projection for Southeast Florida. A document prepared for the Southeast Florida Regional Climate Change Compact Steering Committee. 35 p. At http://www.southeastfloridaclimatecompact.org/wp-content/uploads/2015/10/2015-Compact-Unified-Sea-Level-Rise-Projection.pdf

U.S. Geological Survey. 2012. Everglades Depth Estimation Network (EDEN) database: U.S. Geological Survey, accessed June 1, 2012, at http://sofia.usgs.gov/eden/models/groundelevmod.php.

Appendix A. MATLAB/Octave code for future tidal prediction

The code was developed under contract with the SFWMD. MATLAB/Octave code was developed to facilitate tide predictions at tide stations of interest for future periods based on UTIDE output adjusted by parameterized quadratic (or linear) sea level rise projection curves. The main script is proj_allstas.m and it calls projecttides_new.m. Both are included later in this appendix. To adjust all the data by a single SLR value (i.e. MSL constant in time) as described above, use adjtype=1 in call to proj_allstas.m. To adjust all the data along a SLR curve use adjtype=2. The header to the main function proj_allstas.m is:

```
function
proj_allstas(outflag,csvname,fbase,nyrs,scen,RegRate,RegAccel,adjtype)
```

This function is used to project tides into the future for ALL stations of interest. It will typically be run using tidal predictions as input for NOAA stations and using raw water levels as input for ENP and SFWMD tailwater stations. It calls UTIDE codes ut_solv.m and ut_reconstr.m.

The input arguments to proj_allstas are:

outflag: 1 to output plots, 0 not to
csvname: name of CSV file with hourly tidal pred. and water level data for all stations (input times must be in GMT timezone but output from code will be in EST timezone). The code assumes data in ft NGVD29. If not, check conversion factors in code. Format: Year, Mo, Day, Hour, station 1 data, station 2, etc.
fbase: if(adjtype==1) future base MIDDLE year to project tides
(i.e. this is the year at which one single SLR value
will be computed to adjust MSL for ALL data)
Note: Must be >=1992
if(adjtype==2) future base STARTING year to project tides. Note: Must be >=1992
nyrs: if(adjtype==1) Number of years around fbase to project tides
(i.e. tides will be predicted for fbase+/-(nyrs/2))
if(adjtype==2) Number of years AFTER fbase to project tides
(i.e. tides will be predicted for fbase+nyrs)
scen: SLR scenario for future tide projections from SE FL Reg. Compact Climate Change (Oct. 2015):
0: User-defined SLR linear trend and acceleration
Pre-defined scenarios from SE FL Reg. Compact Climate Change (Oct. 2015):
1: USACE Low/NOAA Intermediate-Low
2: IPCC AR5 median
3: USACE High
4: NOAA High
RegRate (a): + regional SLR trend rate, linear part (mm/yr)
set to [] to use global rate instead (1.7 mm/yr)
RegAccel (b): + regional accel trend rate (mm/yr^2)
set to [] to use the global acceleration given by scen
Note: If Regional SLR rate and acceleration are empty, then corresponding global values are used. This could mix and match regional and global rate and acceleration. Proceed with caution!!!

adjtype: 1: To shift entire timeseries by a certain SLR value (i.e. MSL constant in time)
2: To shift the timeseries ALONG a SLR curve (i.e. MSL varies with time)

The code proj_allstas.m reads a csv file with station metadata with the filename hardcoded as station_metadata.csv. This file is used for filtering data based on some criteria. It has one line per station and the columns are as follows:

1st column: station: station name
2nd column: minyr: minimum year, i.e. only data for years beyond the minyr will be considered
3rd column: minvald: minimum value--i.e. values less than that will be set to missing. Useful to eliminate low outliers from analysis.
4th column: byear: base year of data--i.e. 1992 for NOAA data which is the middle of 1983-2001 tidal epoch or -999 to allow code to determine it based on the mid-point of the raw water level data.

The general steps followed by this code are summarized in Figure 84 and briefly described below:

- Read station_metadata.csv.
- Read raw water level data and/or tidal prediction data for ALL stations of interest from csvname file.
- Based on fbase and nyrs it determines timestamps for future projection times of interest (on an hourly timestep).
- It then loops through each station in csvname file and does the following.
- It calls UTIDE function ut_solv.m with default settings but with no SLR trend, using OLS instead of the default IRLS and linearized confidence interval method with white noise floor assumption instead of Monte Carlo. For details see Codiga (2011).
- It calls UTIDE function ut_reconstr for projection times of interest to properly account for effects of lunar/nodal corrections (on daily tidal range and diurnal inequalities). However, the effect of lunar nodal cycle (LNC) on mean sea level (sinusoidal with 18.61-year frequency) is neglected.
- At stations with byear not equal to -999 (e.g. NOAA tidal stations), it calls project_tides_new.m to move the projected tide levels from ut_reconstr up on a SLR curve. The SLR curve is defined based on scen and the values of RegRate and RegAccel which are hardcoded in proj_allstas.m. See the description of project_tides_new.m below for more information.
- At non-NOAA stations, prior to project_tides_new.m being called, the mean fit by UTIDE based on all the available raw water level data at the station is adjusted to the mean for only the last 19 years of data (to average out effects of LNC). If less than 19 years of data are available, the adjustment=0. In general, this adjustment is minimal compared to subsequent SLR adjustment.
- Output is hourly projected tides with sea level rise adjustment. Time is local time (EST) and units are ft NGVD29. These are written to proj_tide_all.csv which can be imported into Excel and saved as DSS file using the DSS EXCEL Add-in and then averaged to daily within HEC-DSSVue.

The header to the function projecttides_new.m, which is called by proj_allstas.m, is:

```
function pred =
projecttides_new(outflag,station,csvf,fac,byear,fbase,scen,RegRate,Reg
Accel,adjtype)
```

The input arguments to projecttides_new are:
outflag: 1 to output plots, 0 not to
station: name of station to process
csvf: name of CSV file with current hourly or daily tidal data to be projected into the future
Format should be year,mo,day,[hr],tidal_value
fac: conversion factor from meters to match units in csv file (csvf)
byear: base year for historical raw water level data from which tidal predictions were made (e.g. 1992=middle of 1983-2001 tidal epoch for NOAA stations; must be >=1992)
fbase: future base year to project tides
scen: SLR scenario for future tide projections:
0: User-defined SLR linear trend and acceleration
Pre-defined scenarios from SE FL Reg. Compact Climate Change (Oct., 2015):
1: USACE Low/NOAA Intermediate-Low
2: IPCC AR5 RCP8.5 median
3: USACE High
4: NOAA High
See Table 10 below for more details.
RegRate (a): + regional SLR trend rate, linear part (mm/yr)
set to [] to use global rate instead (1.7 mm/yr)
RegAccel (b): + regional accel trend rate (mm/yr^2)
set to [] to use the global acceleration given by scen
Note: If Regional SLR rate and acceleration are empty, then corresponding global values are used. This could mix and match regional and global rate and acceleration. Proceed with caution!!!
adjtype: 1: To shift entire timeseries by a certain SLR value (i.e. MSL constant in time)
2: To shift the timeseries ALONG a SLR curve (i.e. MSL varies with time)

projecttides_new.m uses the following quadratic equation to adjust predicted tides with sea level rise from a period centered at byear to a future period centered at fbase:

$$\Delta h_{SLR}(byear, fbase) = b[(fbase - 1992)^2 - (byear - 1992)^2] + a[(fbase - 1992) - (byear - 1992)]$$

Equation 8

Table 10. Coefficients for pre-defined quadratic SLR scenarios from SE FL Reg. Compact Climate Change (Oct., 2015). These equations give global sea level trends. For local trends could use a=0.0022 m/yr based on Key West data.

SLR Scenario	**SLR Rate a (m/yr)**	**SLR acceleration b (m/yr^2)**	$\boldsymbol{\Delta h_{SLR}(1992, 2100)}$ **(m)**
1:USACE Low/NOAA Intermediate-Low	0.0017	$2.71262*10^{-5}$	0.5
2: IPCC AR5 RCP8.5 median	0.0017	$4.684499*10^{-5}$	0.73
3: USACE High	0.0017	$1.13*10^{-4}$	1.5
4: NOAA High	0.0017	$1.56*10^{-4}$	2.0

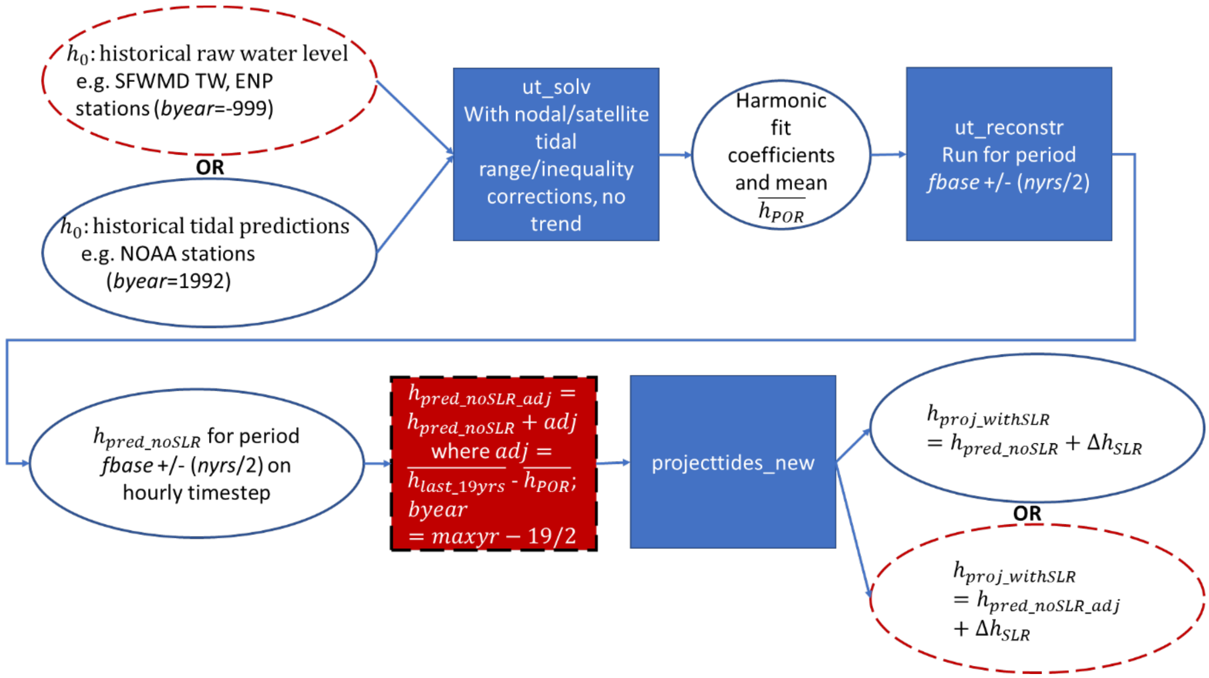


Figure 84. Step-by-step procedure for projecting tides into the future based on quadratic sea level rise curves. Boxes and ovals in red and dashed lines are only performed for stations with byear initially defined as -999 in station_metadata.csv.

Scripts to develop tidal timeseries for future scenarios considering quadratic sea level rise curve:

proj_allstas.m:

```
function proj_allstas(outflag,csvname,fbase,nyrs,scen,RegRate,RegAccel,adjtype)
  %function to project tides into the future for ALL stations
  %Will be run with tidal predictions as input for NOAA stations
  %and with raw water levels as input for ENP and SFWMD TW stations
  %Utide must be run for period centered around fbase to correctly
  %account for effects of lunar/nodal corrections (on daily tidal range
  %and diurnal inequalities). Effect of lunar nodal cycle (LNC) on
  %mean sea level (sinusoidal with 18.61 year frequency) is neglected.
  %However, at non-NOAA stations, the mean fit by Utide based on all the
  %available raw data at the station is adjusted to the mean for only the
  %last 19 years of data (to average out effects of LNC). If less than 19
  %years of data are available, adjustment=0. In general, this adjustment
  %is minimal compared to subsequent SLR adjustment.
  %There are two options to project tides into the future. In the first option,
  %the entire timeseries is raised by a certain SLR projection (a single value).
  %In the second option, the timeseries is adjusted in time by shifting it
  %ALONG a SLR curve.

  %outflag: 1 to output plots, 0 not to
  %csvname: name of CSV file with hourly tidal pred. and water level data
  %         for all stations (times must be in GMT timezone)
  %Assumes data in ft NGVD29. If not, check conversion factors in code
  %      Format: Year, Mo, Day, Hour, station 1 data, station 2, etc.
  %fbase: if(adjtype==1) future base MIDDLE year to project tides
  %                      (i.e. this is the year at which one single SLR value
  %                      will be computed to adjust MSL for ALL data)
  %                      Note: Must be >=1992
  %       if(adjtype==2) future base STARTING year to project tides
  %                      Note: Must be >=1992
  %nyrs: if(adjtype==1) Number of years around fbase to project tides
  %                     (i.e. tides will be predicted for fbase+/-(nyrs/2))
  %      if(adjtype==2) Number of years AFTER fbase to project tides
  %                     (i.e. tides will be predicted for fbase+nyrs)
  %scen: SLR scenario for future tide projections:
  %      from SE FL Reg. Compact Climate Change (Oct 2015):
  %      1: USACE Low/NOAA Intermediate-Low
  %      2: IPCC AR5 median
  %      3: USACE High
  %      4: NOAA High
  %RegRate: + regional SLR trend rate (mm/yr)
  %         set to [] to use global rate instead (1.7 mm/yr)
  %RegAccel: + regional accel trend rate (mm/yr^2)
  %         set to [] to use the global accel. given by scen
  %If Regional SLR rate and accel. are empty, then corresponding global
  %values are used.
  %Note: This could mix and match regional and global rate and accel.
  %Proceed with caution!!!
  %adjtype: 1: To shift entire timeseries by a certain SLR value
  %            (i.e. MSL constant in time)
  %         2: To shift the timeseries ALONG a SLR curve
  %            (i.e. MSL varies with time)

  %outflag=1; csvname='stations_35.csv'; fbase=1992; nyrs=109; scen=3;
  %adjtype=2;RegRate=[];RegAccel=[];

  %outflag=1; csvname='stations_35.csv'; fbase=2016; nyrs=52; scen=3;
  %adjtype=1;RegRate=4.2;RegAccel=0;
```

```
pkg load statistics
disp(adjtype)

%Read starting year and minimum value for filtering
%from a csv file (one line per station, 1st column is the
% station name; 2nd column is minimum year--
%i.e. only data for years beyond the minyr will be considered;
%3rd column is minimum value--i.e. values less than that will be set
%to missing; 4th column is base year of data--i.e. 1992 for NOAA data
%which is the middle of 1983-2001 tidal epoch or -999 to determine it
%based on the mid-point of the raw water level data)
[station,minyr,minvald,byear]=textread('station_metadata.csv',...
    '%s %f %f%f','headerlines',1,'delimiter',',');

%Read water level data
wl = csvread(csvname,1,0); %miriza-add c0=0 for octave
nstas = size(wl,2)-4;

%Prediction dates tt (GMT) and local dates tt2
if (adjtype==1)
  fb=round(fbase-(nyrs-1)/2);
  fe=round(fbase+(nyrs-1)/2);
  nyrs2=fe-fb+1;
  fmid=mean([fb,fe]);
else
  fb=fbase;
  fe=fbase+nyrs-1;
end
sd = datenum(['01-Jan-' num2str(fb) ' 05:00:00']); %add SS for octave
iv = 1/24; %hourly predictions
ed = datenum(['01-Jan-' num2str(fe+1) ' 04:00:00']); %add SS for octave
tt=(sd:iv:ed)'; %(simplified to make octave faster)
tt2=tt-5/24.;
stt2=datevec(tt2);

%Preallocate memory for final predictions
allstas = zeros(size(tt2,1),size(wl,2));

%Loop through all stations
for i = 1:nstas
  ista=i+4;
  a = wl;
  fprintf(1,'i=%s,ista=%s\n',num2str(i),num2str(ista))
  drawnow('expose');
  %drawnow('update');
  %j=find(a(:,ista) <= minvald(i));
  a(a(:,ista) <= minvald(i),ista)=NaN;
  a=a(:,[1:4,ista]);
  %j = find(a(:,1) >= minyr(i));
  a=a(a(:,1) >= minyr(i),1:5);
  %plot(a(:,5));
  Month=a(:,2);
  Year = a(:,1);
  Day=a(:,3);
  Hour=a(:,4);
 t = datenum(Year,Month,Day,Hour,0,0);
  TW = a(:,5);
  %plot(t,TW);
  more off;
  %Call ut_solve
  fprintf(1,'starting ut_solve for i=%s\n',num2str(i));
  %drawnow('update');
```

```
drawnow('expose');
coef = ut_solv(t,TW,[],26.0,'auto','OLS','White','LinCI','NoDiagn',...
    'NoTrend');
%drawnow('update');
drawnow('expose');

%Call ut_reconstr
fprintf(1,'starting ut_reconstr for i=%s\n',num2str(i));
%drawnow('update');
drawnow('expose');
pp = ut_reconstr(tt,coef);
%drawnow('update');
drawnow('expose');
pred = [stt2(:,1:4) pp];

figure;
%plot(tt2,pp);
plot(t-5/24., TW, '-g', tt2, pp, '-y');
datetick('x','YYYY');
legend('Raw data','Pred. tide w/o SLR','Location','northwest')
xlabel ('Time');
ylabel ('ft NGVD29');
limits=axis();
line([datenum(fbase,1,1) datenum(fbase,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fbase,1,1),limits(3),num2str(fbase),...
  'horizontalalignment','center','color','r');
line([datenum(fb,1,1) datenum(fb,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fb,1,1),limits(3),num2str(fb),...
  'horizontalalignment','center','color','r');
line([datenum(fe+1,1,1) datenum(fe+1,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fe+1,1,1),limits(3),num2str(fe+1),...
  'horizontalalignment','center','color','r');
if (byear(i) ~=999)
  line([datenum(byear(i),1,1) datenum(byear(i),1,1)],[limits(3) limits(4) ],...
    'color','k');
  text(datenum(byear(i),1,1),limits(3),num2str(byear(i)),...
    'horizontalalignment','center','color','b');
end
title (['Water level and tide at station ' char(station(i)) '(i=' ...
    num2str(i) ')']);
if(outflag==1)
  print(1,['pred_tide_' char(station(i)) '.png'],'-dpng');
end
close;

csvf = ['pred_tide_' char(station(i)) '.csv'];
csvwrite(csvf,pred);

%adjust data for SLR
fprintf(1,'starting projecttides_new for i=%s\n',num2str(i));
fac=39.3701/12;
%For stations with defined byear (e.g. NOAA),
%simply call projecttides_new to add in SLR
if (byear(i) ~= -999)
  predf=projecttides_new(outflag,char(station(i)),csvf,...
      fac,byear(i),fbase,scen,RegRate,RegAccel,adjtype);
else %For other stations
  %Step 1: Subtract the mean from the Utide fit
  pp2 = pp - coef.mean;
  %Step 2: Get the mean of the raw data for the most recent 19 year
```

```
%period to define the MSL for a tidal epoch (so effects of lunar
%nodal cycle, LNC are averaged out). Assumption here is that if there
%are data gaps, they are uniformly distributed throughout the dataset,
%which may or may not be the case. Add that mean to result of Step 1.
%Note: Stations with less than 19 years of POR will still be
%processed but means may reflect influence of LNC!
%Convert raw times to EST first
t = t - 5/24.;
maxt = max(t);
tmp = datevec(maxt);
mint = datenum(tmp(1)-19,tmp(2),tmp(3),tmp(4),tmp(5),tmp(6));
midt = mean([mint maxt]);
midyr = midt/365.25;
jj=find(t >= mint & t <=maxt);
%If starting date of data is more than a year after mint,
%recompute mint, midt and midyr
%This will happen if station has less than 19 years of POR
%(may want to exclude these!)
if ((t(jj(1)) - mint) > 365)
    mint = t(jj(1));
    midt = mean([mint maxt]);
    midyr = midt/365.25;
end
%epoch_mean=mean(TW(jj),'omitnan');
epoch_mean=nanmean(TW(jj));
pp3 = pp2 + epoch_mean;
predadj = [stt2(:,1:4) pp3];
adj=epoch_mean-coef.mean;

csvfadj = ['pred_tide_' char(station(i)) '_adj.csv'];
csvwrite(csvfadj,predadj);

figure;
plot(tt2, pp, '-y', tt2, pp3, '-b');
datetick('x','YYYY');
legend('Pred. tide w/o SLR',...
    'Pred. tide w/o SLR (adj. mean)','Location','northwest')
xlabel ('Time');
ylabel ('ft NGVD29');
limits=axis();
line([datenum(fbase,1,1) datenum(fbase,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fbase,1,1),limits(3),num2str(fbase),...
  'horizontalalignment','center','color','r');
line([datenum(fb,1,1) datenum(fb,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fb,1,1),limits(3),num2str(fb),...
  'horizontalalignment','center','color','r');
line([datenum(fe+1,1,1) datenum(fe+1,1,1)],[limits(3) limits(4) ],...
  'color','k');
text(datenum(fe+1,1,1),limits(3),num2str(fe+1),...
  'horizontalalignment','center','color','r');
line([datenum(floor(midyr),1,1)+(((midyr)-floor(midyr))*365) ...
  datenum(floor(midyr),1,1)+(((midyr)-floor(midyr))*365)],...
  [limits(3) limits(4) ],'color','k');
text(datenum(floor(midyr),1,1)+(((midyr)-floor(midyr))*365),limits(3),...
  num2str(round(midyr*100)/100),'horizontalalignment','center','color','b');
line([limits(1) limits(2)],[mean(pp) mean(pp) ],...
  'color','k');
line([limits(1) limits(2)],[mean(pp3) mean(pp3) ],...
  'color','k');
text(limits(1),mean(pp),num2str(round(mean(pp)*100)/100),... %round(mean(pp),2)
  'color','k');
```

```
      text(limits(1),mean(pp3),num2str(round(mean(pp3)*100)/100),...
%round(mean(pp3),2)
        'color','k');
      title (['Pred. tide at ' char(station(i)) '(adj=' ...
        num2str(round(adj*100)/100) ' ft)']) %round(adj,2)
      if(outflag==1)
        print(1,['pred_tide_' char(station(i)) '_adj.png'],'-dpng');
      end
      close;

      %Step 3: Call projecttides_new to add SLR
      predf=projecttides_new(outflag,char(station(i)),csvfadj,...
          fac,midyr,fbase,scen,RegRate,RegAccel,adjtype);
    end

    %Concatenate results to create a single csv file at the end
    if (i==1)
      allstas(:,1:5)=predf;
    else
      allstas(:,(i+4))=predf(:,end);
    end

  end

 %Finally write out all predictions to a single csv file
 ofile=fopen('proj_tide_all.csv','w');
fmt=strcat(repmat('%f,',1,size(allstas,2)-1),'%f\n');
fprintf(ofile,fmt,allstas');
 fclose(ofile);

end

function SetDefaultValue(position, argName, defaultValue)
% Initialise a missing or empty value in the caller function.
%
% SETDEFAULTVALUE(POSITION, ARGNAME, DEFAULTVALUE) checks to see if the
% argument named ARGNAME in position POSITION of the caller function is
% missing or empty, and if so, assigns it the value DEFAULTVALUE.
%
% Example:
% function x = TheCaller(x)
% SetDefaultValue(1, 'x', 10);
% end
% TheCaller()    % 10
% TheCaller([])  % 10
% TheCaller(99)  % 99
%
% $Author: Richie Cotton $  $Date: 2010/03/23 $

if evalin('caller', 'nargin') < position || ...
      isempty(evalin('caller', argName))
   assignin('caller', argName, defaultValue);
end
end
```

Top 5 rows and 8 columns of stations_35.csv:

Year	Month	Day	Hour	BK	TC	8722371	8722381
1965	1	1	0	-902	-902	0.752	0.936

1965	1	1	1	-902	-902	0.832	1.039
1965	1	1	2	-902	-902	0.813	1.02
1965	1	1	3	-902	-902	0.702	0.884

Top 5 lines of station_metadata.csv:

station	minyr	minvald	byear
BK	1900	-900	-999
TC	1900	-900	-999
8722371	1900	-900	1992
8722381	1900	-900	1992

projecttides_new.m:

```
function pred = projecttides_new(outflag,station,csvf,fac,...
    byear,fbase,scen,RegRate,RegAccel,adjtype)

  %outflag: 1 to output plots, 0 not to
  %station: name of station to process
  %csvf: name of CSV file with current hourly or daily
  %      tidal data to be projected into the future
  %      Format should be year,mo,day,[hr],tidal_value
  %fac: conversion factor from meters to match units in csv file (csvf)
  %byear: base year for historical raw water level data from which
  %       tidal predictions were made (e.g. 1992=middle of 1983-2001
  %       tidal epoch for NOAA stations; must be >=1992)
  %       (can have a fractional part)
  %fbase: future base year to project tides
  %scen: SLR scenario for future tide projections:
  %      from SE FL Reg. Compact Climate Change (Oct 2015):
  %      1: USACE Low/NOAA Intermediate-Low
  %      2: IPCC AR5 RCP8.5 median
  %      3: USACE High
  %      4: NOAA High
  %RegRate: + regional SLR trend rate (mm/yr)
  %         set to [] to use global rate instead (1.7 mm/yr)
  %RegAccel: + regional accel trend rate (mm/yr^2)
  %         set to [] to use the global accel. given by scen
  %If Regional SLR rate and accel. are empty, then corresponding global
  %values are used.
  %Note: This could mix and match regional and global rate and accel.
  %Proceed with caution!!!
  %adjtype: 1: To shift entire timeseries by a certain SLR value (single value)
  %         2: To shift the timeseries ALONG a SLR curve

  %outflag=0; station='Key West'; RegRate=1.7; fac=39.3701/12;adjtype=1;

  pkg load financial
  %Get tidal data (with MSL corresponding to byear)
  %[year,mo,day,hr,tdata]=textread(csvf,'%f,%f,%f,%f,%f',...
  %    'headerlines',1);
  M=csvread(csvf);
  year=M(:,1);
  mo=M(:,2);
  day=M(:,3);
  if (size(M,2)==4)
```

```
  freq='daily';
  hr=0;
  tdata=M(:,4);
else
  freq='hourly';
  hr=M(:,4);
  tdata=M(:,5);
end
tcurr=datenum(year,mo,day,hr,0,0);
stcurr=datevec(tcurr);
nyears=(max(year)-min(year)+1);

%Prediction dates tt (GMT) and local dates tt2
if (adjtype==1)
  fb=round(fbase-(nyears-1)/2);
  fe=round(fbase+(nyears-1)/2);
  nyrs2=fe-fb+1;
  fmid=mean([fb,fe]);
else
  fb=fbase;
  fe=fbase+nyears-1;
end

%Determine future dates for predictions
sd=datenum(fb,1,1);
if (strcmp(freq,'daily'))
    iv=1;
    ed=sd+(size(year,1)-1);
else
    iv=1/24;
    ed=sd+(size(year,1)-1)/24;
end
tfut=(sd:iv:ed)';
stfut=datevec(tfut);

%Derive acceleration rate b from year 2100 SLR projections from 1992 MSL
%in meters
gmsl=[0.5 0.73 1.5 2.0]; %Global MSL (m) for each scenario
scennames=char('USACE Low','IPCC AR5 RCP8.5 Med.','USACE High',...
    'NOAA High');
ag=1.7/1000; %global SLR rate (trend, 1.7/1000 m/year=1.7 mm/year)

%If Regional SLR rate and accel. not defined, then use global
%values. Note: This could mix and match regional and global rate
%and accel. Proceed with caution!
if (isempty(RegRate))
  a=ag;
else
  a=RegRate/1000;
end
if (isempty(RegAccel))
    b=(gmsl-ag*(2100-1992))/(2100-1992)^2; %global SLR acceleration
                                           %always derived based on ag
else
    b=RegAccel/1000;
end

if (adjtype==1)
  %Derive SLR from byear (must be >=1992) up to (fbase+1)
  %here t is integer year
  t=floor(byear):(fbase+1);
else
  %Derive SLR from sd to ed (here t is fractional year)
```

```
    dv=stfut;
    dv(:,2:3)=1;
    dv(:,4:end)=0;
    doy=tfut-datenum(dv);
   yrs=stfut(:,1);
    yrsday=yeardays(yrs);
    t=(yrs + (doy ./ yrsday))';
  end
  %slr=(b.'*(((t-1992).^2)-((byear-1992).^2))+ a*((t-1992)-(byear-1992))).';
  %Use this version for compatibility with MATLAB R2016a
  %Note!!! Make sure this works for various scenarios and year fractions
  slr=(b.'*(((t-1992).^2)-((byear-1992).^2))+ ...
      repmat(a*((t-1992)-(byear-1992)),size(b,2),1)).';
  slrt=t;

  %Unit conversion
  slr=slr*fac;
  u=[transpose(t) slr];

  if (adjtype==1)
    %Determine the SLR adjustment for particular fbase year
    if (isempty(RegAccel))
      adj=slr(slrt==fbase,scen);
    else
      adj=slr(slrt==fbase);
    end
  else
    if (isempty(RegAccel))
      adj=slr(:,scen);
    else
      adj=slr;
    end
  end

  %Adjust data to future MSL
  tidefut=tdata+adj;

  figure;
  plot(tcurr,tdata,'b-',tfut,tidefut,'r-');
  if (isempty(RegRate) && isempty(RegAccel))
    legend('Pred. tide w/o SLR',['Proj. tide w/SLR: ' ...
        scennames(scen,:)],'Location','northwest');
  else
    legend('Pred. tide w/o SLR','Proj. tide w/SLR',...
    'Location','northwest');
  end
  datetick('x','YYYY');
  %axis([min([tcurr;tfut]) max([tcurr;tfut]) min([tdata;tidefut]) ...
  %   max([tdata;tidefut]) ]);
  grid();
  xlabel('Time (Year)');
  ylabel('ft NGVD29');
  if (isempty(RegAccel))
    title ({['Pred. tide with and w/o SLR at ' station '(adj=' ...
      num2str(round(max(adj)*100)/100) ' ft)'],['a=' num2str(a*1000)... %round(adj,2)
      'mm/yr,b=' num2str(round(b(scen)*1000*1000)/1000) 'mm/yr^2']});
%round(b(scen)*1000,3)
  else
    title ({['Pred. tide with and w/o SLR at ' station '(adj=' ...
      num2str(round(max(adj)*100)/100) ' ft)'],['a=' num2str(a*1000)... %round(adj,2)
      'mm/yr,b=' num2str(b*1000) 'mm/yr^2']});
  end
  limits=axis();
```

```
  line([datenum(fbase,1,1) datenum(fbase,1,1)],[limits(3) limits(4) ],...
    'color','k');
  text(datenum(fbase,1,1),limits(3),num2str(fbase),...
    'horizontalalignment','center','color','r');
  line([datenum(fb,1,1) datenum(fb,1,1)],[limits(3) limits(4) ],...
    'color','k');
  text(datenum(fb,1,1),limits(3),num2str(fb),...
    'horizontalalignment','center','color','r');
  line([datenum(fe+1,1,1) datenum(fe+1,1,1)],[limits(3) limits(4) ],...
    'color','k');
  text(datenum(fe+1,1,1),limits(3),num2str(fe+1),...
    'horizontalalignment','center','color','r');
  line([datenum(floor(byear),1,1)+(((byear)-floor(byear))*365) ...
    datenum(floor(byear),1,1)+(((byear)-floor(byear))*365)],...
    [limits(3) limits(4) ],'color','k');
  text(datenum(floor(byear),1,1)+(((byear)-floor(byear))*365),limits(3),...
    num2str(round(byear*100)/100),'horizontalalignment','center','color','b');

  if (adjtype==1)
    line([limits(1) limits(2)],[mean(tdata) mean(tdata) ],...
        'color','k');
    line([limits(1) limits(2)],[mean(tidefut) mean(tidefut) ],...
        'color','k');
    text(limits(1),mean(tdata),num2str(round(mean(tdata)*100)/100),...
%round(mean(tdata),2)
        'color','k');
    text(limits(1),mean(tidefut),num2str(round(mean(tidefut)*100)/100),...
%round(mean(tidefut),2)
        'color','k');
  else
    line([limits(1) limits(2)],[mean(tdata) mean(tdata) ],...
        'color','k');
    line([limits(1) limits(2)],[mean(tdata)+max(adj) mean(tdata)+max(adj) ],...
        'color','k');
    text(limits(1),mean(tdata),num2str(round(mean(tdata)*100)/100),...
%round(mean(tdata),2)
        'color','k');
    text(limits(1),mean(tdata)+max(adj),num2str(round((mean(tdata)+max(adj))*100)/100)
,... %round(mean(tidefut),2)
        'color','k');
  end
  if(outflag==1)
    print(1,['Proj_tide_' station '.png'],'-dpng');
  end
  close;

  figure;
  plot(slrt,slr);
  grid();
  xlabel('Time (Year)');
  ylabel('ft');
  if (isempty(RegRate) && isempty(RegAccel))
    title({['Projected SLR at ' station ' since ' num2str(round(byear))], ...
        ['global rate a=' num2str(a*1000) 'mm/yr and global accel. b='...
        num2str(round(b(scen)*1000*1000)/1000) 'mm/yr^2']}); %round(b(scen)*1000,3)
    %legend(scennames,'location','northwest');
    legend(strcat(cellstr(scennames),' b=',...
           arrayfun("num2str",round(1000*1000*b)/1000,"UniformOutput",false)',...
           'mm/yr^2'),'location','northwest');
  elseif (isempty(RegRate))
    title({['Projected SLR at ' station ' since ' num2str(round(byear))], ...
        ['global rate a=' num2str(a*1000)...
        'mm/yr and reg. accel. b=' num2str(b*1000) 'mm/yr^2']});
```

```
  elseif (isempty(RegAccel))
    title({['Projected SLR at ' station ' since ' num2str(round(byear))], ...
        ['reg. rate a=' num2str(a*1000)...
        'mm/yr and global accel. b='...
        num2str(round(b(scen)*1000*1000)/1000) 'mm/yr^2']}); %round(b(scen)*1000,3)
    %legend(scennames,'location','northwest');
    legend(strcat(cellstr(scennames),' b=',...
           arrayfun("num2str",round(1000*1000*b)/1000,"UniformOutput",false)',...
           'mm/yr^2'),'location','northwest');
  else
    title({['Projected SLR at ' station ' since ' num2str(round(byear))], ...
        ['reg. rate a=' num2str(a*1000)...
        'mm/yr and reg. accel. b=' num2str(b*1000) 'mm/yr^2']});
  end
  if(outflag==1)
    print(1,['Proj_SLR_' station '.png'],'-dpng');
  end
  close;

  %Write out future predicted tide
  if (strcmp(freq,'daily'))
    pred = [year mo day tidefut];
  else
    pred = [year mo day hr tidefut];
  end
  csvwrite(['proj_tide_' station '.csv'],pred);


end

function SetDefaultValue(position, argName, defaultValue)
% Initialise a missing or empty value in the caller function.
%
% SETDEFAULTVALUE(POSITION, ARGNAME, DEFAULTVALUE) checks to see if the
% argument named ARGNAME in position POSITION of the caller function is
% missing or empty, and if so, assigns it the value DEFAULTVALUE.
%
% Example:
% function x = TheCaller(x)
% SetDefaultValue(1, 'x', 10);
% end
% TheCaller()    % 10
% TheCaller([])  % 10
% TheCaller(99)  % 99
%
% $Author: Richie Cotton $  $Date: 2010/03/23 $

if evalin('caller', 'nargin') < position || ...
      isempty(evalin('caller', argName))
   assignin('caller', argName, defaultValue);
end
end
```

Appendix B. R code for rainfall bias correction

```
##############################################################################

adjprecip_gridwmm_eval <- function(){

##############################################################################
#Note: Must re-do LOCA run 25 (i=25) manually by reading data using var.get.nc and
then creating the brick
#For some reason it does not work to create the brick directly from the netCDF file in
the loop.
#Therefore, one must run this script manually up to the foreach loop, and then
manually re-do
LOCA run 25 (i=25) as described above. Then run the rest of the script manually as
well.
LOCA CRS: crs="+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"
##############################################################################

library(reshape)
library(RNetCDF)
library(raster)
library(rgdal)
#library(fields)
#library(RColorBrewer)
library(pals)
library(foreach)
library(parallel)
library(doParallel)
#library(tcltk)
#library(doSNOW)
#library(gdalUtils)
library(rasterVis)
library(lattice)
library(magic)

#Main variables
vn="pr"
vnl="Precip"
#season=5:10 #wet season
season=1:12 #entire year
LOCA_dir="Z:/miriza/Work/R/LOCA_dataset/Data"
NEXRAD_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/USGS_MODFLOW_NEXRAD"
SFWMM_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/SFWMD"

setwd("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_BC/LOCA_vs_SFWMM_entirey
r")

#Future base period
startyr2=2055
endyr2=2069
nyrs2=endyr2-startyr2+1
allyrs2=startyr2:endyr2
ndays2=as.integer(difftime(strptime(paste("01.01.",endyr2+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyr2,sep=""), format =
"%d.%m.%Y"),units="days"))
dates2=seq(as.Date("2055/1/1"), as.Date("2069/12/31"),"days")
yrs2=as.numeric(format(dates2,'%Y'))
mos2=as.numeric(format(dates2,'%m'))
days2=as.numeric(format(dates2,'%d'))

#Historical date range in M-D MODFLOW NEXRAD rainfall dataset
```

```
startyrh=1996
endyrh=2010
nyrsh=endyrh-startyrh+1
allyrsh=startyrh:endyrh
ndaysh=as.integer(difftime(strptime(paste("01.01.",endyrh+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrh,sep=""), format =
"%d.%m.%Y"),units="days"))
datesnh=seq(as.Date("1996/1/1"), as.Date("2010/12/31"),"days")
yrsnh=as.numeric(format(datesnh,'%Y'))
mosnh=as.numeric(format(datesnh,'%m'))


#Historical date range for Bias-correction (BC)
startyrh2=1991
endyrh2=2005
nyrsh2=endyrh2-startyrh2+1
allyrsh2=startyrh2:endyrh2
ndaysh2=as.integer(difftime(strptime(paste("01.01.",endyrh2+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrh2,sep=""), format =
"%d.%m.%Y"),units="days"))
datesnh2=seq(as.Date("1991/1/1"), as.Date("2005/12/31"),"days")
yrsnh2=as.numeric(format(datesnh2,'%Y'))
mosnh2=as.numeric(format(datesnh2,'%m'))
daysnh2=as.numeric(format(datesnh2,'%d'))


#LOCA date range
#Historical period
startyrlh=1950
endyrlh=2005
nyrslh=endyrlh-startyrlh+1
allyrslh=startyrlh:endyrlh
ndayslh=as.integer(difftime(strptime(paste("01.01.",endyrlh+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrlh,sep=""), format =
"%d.%m.%Y"),units="days"))
dateslh=seq(as.Date("1950/1/1"), as.Date("2005/12/31"),"days")
yrslh=as.numeric(format(dateslh,'%Y'))
moslh=as.numeric(format(dateslh,'%m'))
#Future period
startyrlf=2006
endyrlf=2099
nyrslf=endyrlf-startyrlf+1
allyrslf=startyrlf:endyrlf
ndayslf=as.integer(difftime(strptime(paste("01.01.",endyrlf+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrlf,sep=""), format =
"%d.%m.%Y"),units="days"))
dateslf=seq(as.Date("2006/1/1"), as.Date("2099/12/31"),"days")
yrslf=as.numeric(format(dateslf,'%Y'))
moslf=as.numeric(format(dateslf,'%m'))

#Read in raster with 1996-2010 NEXRAD rainfall data on the M-D MODFLOW grid
#Proj4js.defs["EPSG:26917"] = "+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs"
n=brick(paste(NEXRAD_dir,"/nexrad_rainfall.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
#Load offset mask
offsm=raster(paste(NEXRAD_dir,"/UMD_offshore.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
```

```
#ib=raster(paste(NEXRAD_dir,"/UMD_ibound.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
#offsm[ib==1]=1
offsm[offsm==2]=0
#Adjust values
n=mask(n,offsm,maskvalue=0,updatevalue=NA)
print(paste("after mask-->n\n",sep=""))
#Subset data for months of interest
n=subset(n,which(mosnh%in%season))
print(paste("after subset-->n\n",sep=""))
#Get mean
nm=calc(n,mean)
print(paste("after temporal mean-->nm\n",sep=""))
minnm=minValue(nm)
maxnm=maxValue(nm)
#Get extent
extn=bbox(n)
#Go 1 SFWMM cell (2 mi = 3218.7 m) outside the NEXRAD extent
extn2=extn
extn2[,1]=extn2[,1]-3218.7
extn2[,2]=extn2[,2]+3218.7

#SFWMM date range
datessh=seq(as.Date("1914/1/1"), as.Date("2016/12/31"),"days")
yrssh=as.numeric(format(datessh,'%Y'))
mossh=as.numeric(format(datessh,'%m'))

#Read in SFWMM netCDF file
#Proj4js.defs["ESRI:102258"] = "+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=m +no_defs";
ncf=open.nc(paste(SFWMM_dir,"/rain_v4.7_1914_2016_sfwmd.nc",sep=""))
#SFWMM coordinates are in ft NAD1983 HARN StatePlane FL East FIPS0901 (but ESRI:102258
is in m)
cds=var.get.nc(ncf,"coords")
cds=cds
#Change SFWMM cell centroid coordinates to match projection of M-D MODFLOW grid
d <- data.frame(x=cds[1,], y=cds[2,])
coordinates(d) <- c("x", "y")
proj4string(d) <- CRS("+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=ft +no_defs")
CRS.new <- CRS("+init=epsg:26917") #m
d.n <- spTransform(d, CRS.new) #m
d.ns=crop(d.n,extn2)
#plot(d.ns,pch=1)
cellsinmodel=which(d.n@coords[,1]>=extn2[1,1] & d.n@coords[,1]<=extn2[1,2] &
d.n@coords[,2]>=extn2[2,1] & d.n@coords[,2]<=extn2[2,2])
roco=var.get.nc(ncf,"roco")
rocosinmodel=roco[,cellsinmodel]
#range of rows 3-32
#range of cols 52-68
wmmrain=var.get.nc(ncf,"rainfall")[,,(which(yrssh%in%allyrsh2 & mossh%in%season))]
#This extract data for 1991-2005 and season of interest
#Start with an NA array to accomodate the subset of cells of interest
wmmraininmodel=array(dim=c(diff(range(rocosinmodel[1,]))+1,diff(range(rocosinmodel[2,]
))+1,dim(wmmrain)[3]))
for (c in 1:dim(rocosinmodel)[2]) {
  wmmraininmodel[(rocosinmodel[1,c]-min(rocosinmodel[1,])+1),(rocosinmodel[2,c]-
min(rocosinmodel[2,])+1),]=
                  wmmrain[rocosinmodel[2,c],rocosinmodel[1,c],]
}
#Create a brick from the SFWMM rainfall data
#Range defined below are for outer boundaries of cells
```

```
wmmb=brick(arev(wmmraininmodel,1),xmn=min(cds[1,cellsinmodel])-
5280,xmx=max(cds[1,cellsinmodel])+5280,
           ymn=min(cds[2,cellsinmodel])-5280,ymx=max(cds[2,cellsinmodel])+5280,
           crs="+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81 +k=0.9999411764705882
+x_0=200000 +y_0=0 +ellps=GRS80 +units=ft +no_defs")
#Project SFWMM data to MODFLOW model grid
wmmb=projectRaster(wmmb,n,method='ngb')
print(paste("after projection-->wmmb\n",sep=""))
nds2=nlayers(wmmb)/nyrsh2 #number of days in the season
#Get mean
wmmbm=calc(wmmb,mean)
print(paste("after temporal mean-->wmmbm\n",sep=""))
#Adjust values
wmmbm=mask(wmmbm,offsm,maskvalue=0,updatevalue=NA)
print(paste("after mask-->wmmbm\n",sep=""))
minwmmm=minValue(wmmbm)
maxwmmm=maxValue(wmmbm)

#Read LOCA projections
projs=read.table(paste(LOCA_dir,"/loca_projections.txt",sep=""),stringsAsFactors=FALSE
)
nprojs=nrow(projs)
fns=paste(vn,"_",projs[,1],sep="")
modelp=apply(projs,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelpbase=apply(projs,1,function(x) strsplit(x,"_")[[1]][1])
modelprip=apply(projs,1,function(x) strsplit(x,"_")[[1]][2])
modelprcp=apply(projs,1,function(x) strsplit(x,"_")[[1]][3])
modelpbases=unique(modelpbase)
minpm=vector(length=nprojs,mode="double")
maxpm=vector(length=nprojs,mode="double")
minpmbc=vector(length=nprojs,mode="double")
maxpmbc=vector(length=nprojs,mode="double")
meanpm=vector(length=nprojs,mode="double")
meanpmbc=vector(length=nprojs,mode="double")

hist=read.table(paste(LOCA_dir,"/loca_historical.txt",sep=""),stringsAsFactors=FALSE)
fnsh=paste(vn,"_",hist[,1],sep="")
modelh=apply(hist,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelhbase=apply(hist,1,function(x) strsplit(x,"_")[[1]][1])
modelhrip=apply(hist,1,function(x) strsplit(x,"_")[[1]][2])
minhm=vector(length=nprojs,mode="double")
maxhm=vector(length=nprojs,mode="double")
meanhm=vector(length=nprojs,mode="double")


#Define number of cores
ncore=3
#Create log file
st=Sys.time()
#logfil=paste("Log_",gsub("[ :]","_",st),".txt",sep="")
#cat(paste("Log file:",st,"\n"),file=logfil)

#Register processors
outfil=paste("Out_",gsub("[ :]","_",st),".txt",sep="")
cl <- makePSOCKcluster(ncore,outfile=outfil)
registerDoParallel(cl)
#registerDoSNOW(cl)

# Extract subset of data of interest for projections
foreach
(i=1:nrow(projs),.packages=c("raster","foreach","RNetCDF","rgdal","parallel","doParall
el")) %dopar% {
```

```
logfil=paste("Log_",gsub("[ :]","_",st),"_",i,".txt",sep="")
cat(paste("Log file:",st,"\n"),file=logfil)
cat(paste("i=",i,"\n",sep=""),file=logfil,append=TRUE)

#Get LOCA data for projections
#Get variable attributes from netCDF file
ncfile<-open.nc(paste(LOCA_dir,"/",fns[i],"_2006-2100.nc",sep=""))
units=att.get.nc(ncfile,fns[i],"units")
cat(paste(units,"\n",sep=""),file=logfil,append=TRUE)
if (units == "kg m-2 s-1") {
  conv=141.7323*24  #mm/s to in/day
} else if (units == "mm") {
  conv=1/25.4  #mm to in (per day of course)
} else {
  stop("Different type of units")
}

#scal=try(att.inq.nc(ncfile,fns[i],"scale_factor"),silent=TRUE)
#if (class(scal) == "try-error") {
#  scal=1
#} else {
#  scal=att.get.nc(ncfile,fns[i],"scale_factor")
#}
scal=1
cat(paste(scal,"\n",sep=""),file=logfil,append=TRUE)
close.nc(ncfile)

#get data as a brick
b=brick(paste(LOCA_dir,"/",fns[i],"_2006-2100.nc",sep=""))
cat(paste("after brick b\n",sep=""),file=logfil,append=TRUE)

#Fix longitudes to go from -180 to +180 so raster projection can proceed correctly
extent(b)=c(xmin(b)-360,xmax(b)-360,ymin(b),ymax(b))

#Extract subset of LOCA data for future years of interst
bproj=subset(b,which(yrslf%in%allyrs2 & moslf%in%season))
cat(paste("after subset-->bproj\n",sep=""),file=logfil,append=TRUE)

#Project LOCA data to MODFLOW model grid
bproj=projectRaster(bproj,n,method='ngb')
cat(paste("after projection-->bproj\n",sep=""),file=logfil,append=TRUE)

#Adjust values
#bproj=mask(bproj,offsm,maskvalue=0,updatevalue=NA)
#cat(paste("after mask-->bproj\n",sep=""),file=logfil,append=TRUE)
#bproj=calc(bproj,fun=function(x) x*conv*scal)
#cat(paste("after calculation-->bproj\n",sep=""),file=logfil,append=TRUE)

rm(b)

#Get mean
projm=calc(bproj,mean)
cat(paste("after temporal mean-->projm\n",sep=""),file=logfil,append=TRUE)
#Then project the mean
#projm=projectRaster(projm,n,method='ngb')
cat(paste("after projection-->projm\n",sep=""),file=logfil,append=TRUE)
#Mask values
#projm=mask(projm,offsm,maskvalue=0,updatevalue=NA)
#cat(paste("after mask-->projm\n",sep=""),file=logfil,append=TRUE)
#Adjust values
projm=calc(projm,fun=function(x) x*conv*scal)
cat(paste("after calculation-->projm\n",sep=""),file=logfil,append=TRUE)
```

```
  maxpm[i]=maxValue(projm)
  minpm[i]=minValue(projm)

  # Merge with historical data if available
  idh=which(modelhbase %in% modelpbase[i])
  if (length(idh) == 0) {
    cat(paste("No historical data found for model:",modelp[i],"
(i=",i,")\n",sep=""),file=logfil,append=TRUE)
  } else if (length(idh) > 1) {
    stop(paste("Multiple historical data found for model:",modelp[i],"
(i=",i,")",sep=""))
  } else {
    cat(paste("Historical data found for model:",modelp[i],"
(i=",i,")\n",sep=""),file=logfil,append=TRUE)
    ncfile2<-open.nc(paste(LOCA_dir,"/",fnsh[idh],"_1950-2005.nc",sep=""))

    units2=att.get.nc(ncfile2,fnsh[idh],"units")
    cat(paste(units2,"\n",sep=""),file=logfil,append=TRUE)
    if (units2 == "kg m-2 s-1") {
      conv2=141.7323*24  #mm/s to in/day
    } else if (units2 == "mm") {
      conv2=1/25.4  #mm to in (per day of course)
    } else {
      stop("Different type of units")
    }

    #scale2=try(att.inq.nc(ncfile2,fnsh[idh],"scale_factor"),silent=TRUE)
    #if (class(scale2) == "try-error") {
    #  scale2=1
    #} else {
    #  scale2=att.get.nc(ncfile2,fnsh[idh],"scale_factor")
    #}
    scale2=1
    cat(paste(scale2,"\n",sep=""),file=logfil,append=TRUE)
    close.nc(ncfile2)

    #get data as a brick
    b=brick(paste(LOCA_dir,"/",fnsh[idh],"_1950-2005.nc",sep=""))
    cat(paste("after brick->b\n",sep=""),file=logfil,append=TRUE)

    #Fix longitudes to go from -180 to +180 so raster projection can proceed correctly
    extent(b)=c(xmin(b)-360,xmax(b)-360,ymin(b),ymax(b))

    #Extract subset of LOCA data for historical years of interest
    bhist=subset(b,which(yrslh%in%allyrsh2 & moslh%in%season))
    cat(paste("after subset-->bhist\n",sep=""),file=logfil,append=TRUE)

    #Project LOCA data to MODFLOW model grid
    #bhist=projectRaster(bhist,n,method='ngb')
    #cat(paste("after projection-->bhist\n",sep=""),file=logfil,append=TRUE)

    #Adjust values
    #bhist=mask(bhist,offsm,maskvalue=0,updatevalue=NA)
    #cat(paste("after mask-->bhist\n",sep=""),file=logfil,append=TRUE)
    #bhist=calc(bhist,fun=function(x) x*conv2*scale2)
    #cat(paste("after calculation-->bhist\n",sep=""),file=logfil,append=TRUE)

    rm(b)

    #Get mean
    histm=calc(bhist,mean)
    cat(paste("after temporal mean-->histm\n",sep=""),file=logfil,append=TRUE)
    #Then project the mean
```

```
    histm=projectRaster(histm,n,method='ngb')
    cat(paste("after projection-->histm\n",sep=""),file=logfil,append=TRUE)
    #Mask values
    histm=mask(histm,offsm,maskvalue=0,updatevalue=NA)
    cat(paste("after mask-->histm\n",sep=""),file=logfil,append=TRUE)
    #Adjust values
    histm=calc(histm,fun=function(x) x*conv2*scale2)
    cat(paste("after calculation-->histm\n",sep=""),file=logfil,append=TRUE)

    maxhm[i]=maxValue(histm)
    minhm[i]=minValue(histm)

    #Get bias-corrected mean
    projmbc=(projm/histm)*wmmbm
    maxpmbc[i]=maxValue(projmbc)
    minpmbc[i]=minValue(projmbc)

    #Save data

save(bhist,histm,bproj,projm,n,nm,wmmbm,projmbc,file=paste(fns[i],"_rasters.RData",sep
=""))
    rm(bhist,histm,bproj,projm,projmbc)

  }
}

#Clean up the cluster
stopImplicitCluster()

#Make levelplots of annual means
#Get overall ranges
minz=min(c(minpm,minhm,minpmbc,minwmmm))*nds2
maxz=max(c(maxpm,maxhm,maxpmbc,maxwmmm))*nds2

minz=floor(minz)
maxz=ceiling(maxz)

#Create levelplot of nm
#png("NEXRAD_mean_rainfall.png")
#col.regions=brewer.gnbu(100)
#print(levelplot(nm*nds2,margin=FALSE,at=seq(minz,maxz,1),
#          main=paste("NEXRAD rainfall (",round(nds2*cellStats(nm,mean),2),"
in/yr)\n",startyrh,"-",endyrh,sep=""),
#          xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
#dev.off()
#rm(n,nm)

#Create levelplot of wmmbm
png("SFWMM_mean_rainfall.png")
#col.regions=brewer.gnbu(100)
print(levelplot(wmmbm*nds2,margin=FALSE,at=seq(minz,maxz,1),
          main=paste("SFWMM rainfall (",round(nds2*cellStats(wmmbm,mean),2),"
in/yr)\n",startyrh2,"-",endyrh2,sep=""),
          xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
dev.off()

#Create levelplots for the projections
for (i in 1:nrow(projs)) {
  print(paste("i=",i))
  load(paste(fns[i],"_rasters.RData",sep=""),verbose=TRUE)
  #Create levelplot of histm
  meanhm[i]=round(nds2*cellStats(histm,mean),2)
  png(paste("LOCA_mean_rainfall_hist_",fns[i],".png",sep=""))
```

```
  print(levelplot(histm*nds2,margin=FALSE,at=seq(minz,maxz,1),
            main=paste("LOCA rainfall (",meanhm[i]," in/yr)\n",
            startyrh2,"-",endyrh2," (",fns[i],")",sep=""),
            xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
  dev.off()
  #Create levelplot of projm
  meanpm[i]=round(nds2*cellStats(projm,mean),2)
  png(paste("LOCA_mean_rainfall_proj_",fns[i],".png",sep=""))
  print(levelplot(projm*nds2,margin=FALSE,at=seq(minz,maxz,1),
            main=paste("LOCA rainfall (",meanpm[i]," in/yr)\n",
            startyr2,"-",endyr2," (",fns[i],")",sep=""),
            xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
  dev.off()
  #Create levelplot of projmbc
  meanpmbc[i]=round(nds2*cellStats(projmbc,mean),2)
  png(paste("LOCA_mean_rainfall_projbc_",fns[i],".png",sep=""))
  print(levelplot(projmbc*nds2,margin=FALSE,at=seq(minz,maxz,1),
            main=paste("B.C. LOCA rainfall (",meanpmbc[i]," in/yr)\n",
            startyr2,"-",endyr2," (",fns[i],")",sep=""),
            xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83",cex=0.5))
  dev.off()

  png(paste("LOCA_projbc_to_SFWMM_ratio_",fns[i],".png",sep=""))
  print(levelplot(projmbc/wmmbm,col.regions=brewer.rdbu(21),margin=FALSE,
            at=seq(0.70,1.30,0.025),
            main=paste("B.C.",fns[i]," (",startyr2,"-",endyr2,")\n",
            "to SFWMM rainfall (",startyrh,"-",endyrh,")",sep=""),
            xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83",cex=0.5))
  dev.off()

  png(paste("LOCA_projbc-SFWMM_",fns[i],".png",sep=""))
  print(levelplot(nds2*(projmbc-wmmbm),col.regions=brewer.rdbu(29),margin=FALSE,
            at=seq(-16,16,1),
            main=paste("B.C.",fns[i]," (",startyr2,"-",endyr2,")\n",
            "- SFWMM rainfall (",startyrh,"-",endyrh,") (",
           round(nds2*cellStats((projmbc-wmmbm),mean),2)," in/yr)",sep=""),
            xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83",cex=0.5))
  dev.off()

}

statis=as.data.frame(cbind(meanhm,meanpm,meanpmbc))
rownames(statis)=fns
colnames(statis)=c('meanhm','meanpm','meanpmbc')

save(statis,file="mean_rainfall_stats.RData")
write.csv(statis,file="mean_rainfall_stats.csv")

png("Bias_correction_check.png")
plot(statis[,1]/(cellStats(wmmbm,mean)*nds2),statis[,2]/statis[,3],xlab="mean(hist)/me
an(WMM)",ylab="mean(proj)/mean(projbc)",main="Check of bias-correction")
grid()
dev.off()


}
```

```
#############################################################################

adjprecip_gridwmm_actual <- function(){

#############################################################################
#Note: Must re-do LOCA run 25 (i=25) manually by reading data using var.get.nc and
then creating the brick
#For some reason it does not work to create the brick directly from the netCDF file in
the loop.
#Therefore, one must run this script manually up to the foreach loop, and then
manually re-do
LOCA run 25 (i=25) as described above. Then run the rest of the script manually as
well.
LOCA CRS: crs="+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"
#############################################################################

library(reshape)
library(RNetCDF)
library(raster)
library(rgdal)
#library(fields)
#library(RColorBrewer)
library(pals)
library(foreach)
library(parallel)
library(doParallel)
#library(tcltk)
#library(doSNOW)
#library(gdalUtils)
library(rasterVis)
library(lattice)
library(magic)

#Main variables
vn="pr"
vnl="Precip"
#season=5:10 #wet season
season=1:12 #entire year
LOCA_dir="Z:/miriza/Work/R/LOCA_dataset/Data"
NEXRAD_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/USGS_MODFLOW_NEXRAD"
SFWMM_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/SFWMD"

setwd("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_BC/LOCA_vs_SFWMM_entirey
r_barrier_islands")

#Future base period
startyr2=2055
endyr2=2069
nyrs2=endyr2-startyr2+1
allyrs2=startyr2:endyr2
ndays2=as.integer(difftime(strptime(paste("01.01.",endyr2+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyr2,sep=""), format =
"%d.%m.%Y"),units="days"))
dates2=seq(as.Date("2055/1/1"), as.Date("2069/12/31"),"days")
yrs2=as.numeric(format(dates2,'%Y'))
mos2=as.numeric(format(dates2,'%m'))
days2=as.numeric(format(dates2,'%d'))

#Historical date range in M-D MODFLOW NEXRAD rainfall dataset
startyrh=1996
endyrh=2010
nyrsh=endyrh-startyrh+1
```

```
allyrsh=startyrh:endyrh
ndaysh=as.integer(difftime(strptime(paste("01.01.",endyrh+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrh,sep=""), format =
"%d.%m.%Y"),units="days"))
datesnh=seq(as.Date("1996/1/1"), as.Date("2010/12/31"),"days")
yrsnh=as.numeric(format(datesnh,'%Y'))
mosnh=as.numeric(format(datesnh,'%m'))


#Historical date range for Bias-correction (BC)
startyrh2=1991
endyrh2=2005
nyrsh2=endyrh2-startyrh2+1
allyrsh2=startyrh2:endyrh2
ndaysh2=as.integer(difftime(strptime(paste("01.01.",endyrh2+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrh2,sep=""), format =
"%d.%m.%Y"),units="days"))
datesnh2=seq(as.Date("1991/1/1"), as.Date("2005/12/31"),"days")
yrsnh2=as.numeric(format(datesnh2,'%Y'))
mosnh2=as.numeric(format(datesnh2,'%m'))
daysnh2=as.numeric(format(datesnh2,'%d'))


#LOCA date range
#Historical period
startyrlh=1950
endyrlh=2005
nyrslh=endyrlh-startyrlh+1
allyrslh=startyrlh:endyrlh
ndayslh=as.integer(difftime(strptime(paste("01.01.",endyrlh+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrlh,sep=""), format =
"%d.%m.%Y"),units="days"))
dateslh=seq(as.Date("1950/1/1"), as.Date("2005/12/31"),"days")
yrslh=as.numeric(format(dateslh,'%Y'))
moslh=as.numeric(format(dateslh,'%m'))
#Future period
startyrlf=2006
endyrlf=2099
nyrslf=endyrlf-startyrlf+1
allyrslf=startyrlf:endyrlf
ndayslf=as.integer(difftime(strptime(paste("01.01.",endyrlf+1,sep=""), format =
"%d.%m.%Y"),
                strptime(paste("01.01.",startyrlf,sep=""), format =
"%d.%m.%Y"),units="days"))
dateslf=seq(as.Date("2006/1/1"), as.Date("2099/12/31"),"days")
yrslf=as.numeric(format(dateslf,'%Y'))
moslf=as.numeric(format(dateslf,'%m'))

#Read in raster with 1996-2010 NEXRAD rainfall data on the M-D MODFLOW grid
#Proj4js.defs["EPSG:26917"] = "+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs"
n=brick(paste(NEXRAD_dir,"/nexrad_rainfall.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
#Load offset mask
offsm=raster(paste(NEXRAD_dir,"/UMD_offshore.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
ib=raster(paste(NEXRAD_dir,"/UMD_ibound.nc",sep=""),crs="+proj=utm +zone=17
+ellps=GRS80 +datum=NAD83 +units=m +no_defs")
offsm[ib==1]=1
offsm[offsm==2]=0
```

```
#Adjust values
n=mask(n,offsm,maskvalue=0,updatevalue=NA)
print(paste("after mask-->n\n",sep=""))
#Subset data for months of interest
n=subset(n,which(mosnh%in%season))
print(paste("after subset-->n\n",sep=""))
#Get mean
nm=calc(n,mean)
print(paste("after temporal mean-->nm\n",sep=""))
minnm=minValue(nm)
maxnm=maxValue(nm)
#Get extent
extn=bbox(n)
#Go 1 SFWMM cell (2 mi = 3218.7 m) outside the NEXRAD extent
extn2=extn
extn2[,1]=extn2[,1]-3218.7
extn2[,2]=extn2[,2]+3218.7

#SFWMM date range
datessh=seq(as.Date("1914/1/1"), as.Date("2016/12/31"),"days")
yrssh=as.numeric(format(datessh,'%Y'))
mossh=as.numeric(format(datessh,'%m'))

#Read in SFWMM netCDF file
#Proj4js.defs["ESRI:102258"] = "+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=m +no_defs";
ncf=open.nc(paste(SFWMM_dir,"/rain_v4.7_1914_2016_sfwmd.nc",sep=""))
#SFWMM coordinates are in ft NAD1983 HARN StatePlane FL East FIPS0901 (but ESRI:102258
is in m)
cds=var.get.nc(ncf,"coords")
cds=cds
#Change SFWMM cell centroid coordinates to match projection of M-D MODFLOW grid
d <- data.frame(x=cds[1,], y=cds[2,])
coordinates(d) <- c("x", "y")
proj4string(d) <- CRS("+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=ft +no_defs")
CRS.new <- CRS("+init=epsg:26917") #m
d.n <- spTransform(d, CRS.new) #m
d.ns=crop(d.n,extn2)
#plot(d.ns,pch=1)
cellsinmodel=which(d.n@coords[,1]>=extn2[1,1] & d.n@coords[,1]<=extn2[1,2] &
d.n@coords[,2]>=extn2[2,1] & d.n@coords[,2]<=extn2[2,2])
roco=var.get.nc(ncf,"roco")
rocosinmodel=roco[,cellsinmodel]
#range of rows 3-32
#range of cols 52-68
wmmrain=var.get.nc(ncf,"rainfall")[,,(which(yrssh%in%allyrsh2 & mossh%in%season))]
#This extract data for 1991-2005 and season of interest
#Start with an NA array to accomodate the subset of cells of interest
wmmraininmodel=array(dim=c(diff(range(rocosinmodel[1,]))+1,diff(range(rocosinmodel[2,]
))+1,dim(wmmrain)[3]))
for (c in 1:dim(rocosinmodel)[2]) {
  wmmraininmodel[(rocosinmodel[1,c]-min(rocosinmodel[1,])+1),(rocosinmodel[2,c]-
min(rocosinmodel[2,])+1),]=
                  wmmrain[rocosinmodel[2,c],rocosinmodel[1,c],]
}
#Create a brick from the SFWMM rainfall data
#Range defined below are for outer boundaries of cells
wmmb=brick(arev(wmmraininmodel,1),xmn=min(cds[1,cellsinmodel])-
5280,xmx=max(cds[1,cellsinmodel])+5280,
           ymn=min(cds[2,cellsinmodel])-5280,ymx=max(cds[2,cellsinmodel])+5280,
           crs="+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81 +k=0.9999411764705882
+x_0=200000 +y_0=0 +ellps=GRS80 +units=ft +no_defs")
```

```
#Project SFWMM data to MODFLOW model grid
wmmb=projectRaster(wmmb,n,method='ngb')
print(paste("after projection-->wmmb\n",sep=""))
nds2=nlayers(wmmb)/nyrsh2 #number of days in the season
#Apply mask
wmmb=mask(wmmb,offsm,maskvalues=0,updatevalues=NA)
#Fill in values with offsm=1
#Calculate distance and direction from all NA pixels to the nearest non-NA pixel
dist=distance(subset(wmmb,1))
direct=direction(subset(wmmb,1),from=FALSE)
#Retrieve coordinates of NA pixels
#NA raster
rna=is.na(wmmb)
#Store coordinates
na.x=init(rna,'x')
na.y=init(rna,'y')
#Calculate coordinates to nearest non-NA pixel
co.x = na.x + dist * sin(direct)
co.y = na.y + dist * cos(direct)
co = cbind(co.x[], co.y[])
# extract values of nearest non-NA cell with coordinates co
NAVals <- raster::extract(wmmb, co, method='simple')
r.NAVals <- rna # initiate new raster
r.NAVals[] <- NAVals # store values in raster
# cover nearest non-NA value at NA locations of original raster
wmmb.filled=cover(x=wmmb,y=r.NAVals)
#Mask values
wmmb=mask(wmmb.filled,offsm,maskvalue=0,updatevalue=NA)
print(paste("after mask-->wmmbm\n",sep=""))
#Get mean
wmmbm=calc(wmmb,mean)
print(paste("after temporal mean-->wmmbm\n",sep=""))
plot(wmmbm)
minwmmm=minValue(wmmbm)
maxwmmm=maxValue(wmmbm)

#Read LOCA projections
projs=read.table(paste(LOCA_dir,"/loca_projections.txt",sep=""),stringsAsFactors=FALSE
)
nprojs=nrow(projs)
fns=paste(vn,"_",projs[,1],sep="")
modelp=apply(projs,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelpbase=apply(projs,1,function(x) strsplit(x,"_")[[1]][1])
modelprip=apply(projs,1,function(x) strsplit(x,"_")[[1]][2])
modelprcp=apply(projs,1,function(x) strsplit(x,"_")[[1]][3])
modelpbases=unique(modelpbase)
minpm=vector(length=nprojs,mode="double")
maxpm=vector(length=nprojs,mode="double")
minpmbc=vector(length=nprojs,mode="double")
maxpmbc=vector(length=nprojs,mode="double")
meanpm=vector(length=nprojs,mode="double")
meanpmbc=vector(length=nprojs,mode="double")

hist=read.table(paste(LOCA_dir,"/loca_historical.txt",sep=""),stringsAsFactors=FALSE)
fnsh=paste(vn,"_",hist[,1],sep="")
modelh=apply(hist,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelhbase=apply(hist,1,function(x) strsplit(x,"_")[[1]][1])
modelhrip=apply(hist,1,function(x) strsplit(x,"_")[[1]][2])
minhm=vector(length=nprojs,mode="double")
maxhm=vector(length=nprojs,mode="double")
meanhm=vector(length=nprojs,mode="double")
```

```
st=Sys.time()

# Extract subset of data of interest for projections
#Chosing run i=56 (MRI-CGCM3_r1i1p1_rcp85) for daily bias-correction
i = 56

  logfil=paste("Log_",gsub("[ :]","_",st),"_",i,".txt",sep="")
  cat(paste("Log file:",st,"\n"),file=logfil)
  cat(paste("i=",i,"\n",sep=""),file=logfil,append=TRUE)

  #Get LOCA data for projections
  #Get variable attributes from netCDF file
  ncfile<-open.nc(paste(LOCA_dir,"/",fns[i],"_2006-2100.nc",sep=""))
  units=att.get.nc(ncfile,fns[i],"units")
  cat(paste(units,"\n",sep=""),file=logfil,append=TRUE)
  if (units == "kg m-2 s-1") {
    conv=141.7323*24  #mm/s to in/day
  } else if (units == "mm") {
    conv=1/25.4  #mm to in (per day of course)
  } else {
    stop("Different type of units")
  }

  #scal=try(att.inq.nc(ncfile,fns[i],"scale_factor"),silent=TRUE)
  #if (class(scal) == "try-error") {
  #  scal=1
  #} else {
  #  scal=att.get.nc(ncfile,fns[i],"scale_factor")
  #}
  scal=1
  cat(paste(scal,"\n",sep=""),file=logfil,append=TRUE)
  close.nc(ncfile)

  #get data as a brick
  b=brick(paste(LOCA_dir,"/",fns[i],"_2006-2100.nc",sep=""))
  cat(paste("after brick b\n",sep=""),file=logfil,append=TRUE)

  #Fix longitudes to go from -180 to +180 so raster projection can proceed correctly
  extent(b)=c(xmin(b)-360,xmax(b)-360,ymin(b),ymax(b))

  #Extract subset of LOCA data for future years of interst
  bproj=subset(b,which(yrslf%in%allyrs2 & moslf%in%season))
  cat(paste("after subset-->bproj\n",sep=""),file=logfil,append=TRUE)

  #Project LOCA data to MODFLOW model grid
  bproj=projectRaster(bproj,n,method='ngb')
  cat(paste("after projection-->bproj\n",sep=""),file=logfil,append=TRUE)

  #Adjust values
  bproj=calc(bproj,fun=function(x) x*conv*scal)
  cat(paste("after calculation-->bproj\n",sep=""),file=logfil,append=TRUE)
  #Apply mask
  bproj=mask(bproj,offsm,maskvalues=0,updatevalues=NA)
  #Fill in values with offsm=1
  #Calculate distance and direction from all NA pixels to the nearest non-NA pixel
  dist=distance(subset(bproj,1))
  direct=direction(subset(bproj,1),from=FALSE)
  #Retrieve coordinates of NA pixels
  #NA raster
  rna=is.na(bproj)
  #Store coordinates
  na.x=init(rna,'x')
  na.y=init(rna,'y')
```

```
  #Calculate coordinates to nearest non-NA pixel
  co.x = na.x + dist * sin(direct)
  co.y = na.y + dist * cos(direct)
  co = cbind(co.x[], co.y[])
  # extract values of nearest non-NA cell with coordinates co
  NAVals <- raster::extract(bproj, co, method='simple')
  r.NAVals <- rna # initiate new raster
  r.NAVals[] <- NAVals # store values in raster
  # cover nearest non-NA value at NA locations of original raster
  bproj.filled=cover(x=bproj,y=r.NAVals)
  #Mask values
  bproj=mask(bproj.filled,offsm,maskvalue=0,updatevalue=NA)
  print(paste("after mask-->bprojm\n",sep=""))
  #Get mean
  projm=calc(bproj,mean)

  rm(b)

  maxpm[i]=maxValue(projm)
  minpm[i]=minValue(projm)

  # Merge with historical data if available
  idh=which(modelhbase %in% modelpbase[i])
  if (length(idh) == 0) {
    cat(paste("No historical data found for model:",modelp[i],"
(i=",i,")\n",sep=""),file=logfil,append=TRUE)
  } else if (length(idh) > 1) {
    stop(paste("Multiple historical data found for model:",modelp[i],"
(i=",i,")",sep=""))
  } else {
    cat(paste("Historical data found for model:",modelp[i],"
(i=",i,")\n",sep=""),file=logfil,append=TRUE)
    ncfile2<-open.nc(paste(LOCA_dir,"/",fnsh[idh],"_1950-2005.nc",sep=""))

    units2=att.get.nc(ncfile2,fnsh[idh],"units")
    cat(paste(units2,"\n",sep=""),file=logfil,append=TRUE)
    if (units2 == "kg m-2 s-1") {
      conv2=141.7323*24  #mm/s to in/day
    } else if (units2 == "mm") {
      conv2=1/25.4  #mm to in (per day of course)
    } else {
      stop("Different type of units")
    }

    #scale2=try(att.inq.nc(ncfile2,fnsh[idh],"scale_factor"),silent=TRUE)
    #if (class(scale2) == "try-error") {
    #  scale2=1
    #} else {
    #  scale2=att.get.nc(ncfile2,fnsh[idh],"scale_factor")
    #}
    scale2=1
    cat(paste(scale2,"\n",sep=""),file=logfil,append=TRUE)
    close.nc(ncfile2)

    #get data as a brick
    b=brick(paste(LOCA_dir,"/",fnsh[idh],"_1950-2005.nc",sep=""))
    cat(paste("after brick->b\n",sep=""),file=logfil,append=TRUE)

    #Fix longitudes to go from -180 to +180 so raster projection can proceed correctly
    extent(b)=c(xmin(b)-360,xmax(b)-360,ymin(b),ymax(b))

    #Extract subset of LOCA data for historical years of interest
    bhist=subset(b,which(yrslh%in%allyrsh2 & moslh%in%season))
```

```
    cat(paste("after subset-->bhist\n",sep=""),file=logfil,append=TRUE)

    #Project LOCA data to MODFLOW model grid
    bhist=projectRaster(bhist,n,method='ngb')
    cat(paste("after projection-->bhist\n",sep=""),file=logfil,append=TRUE)

    #Adjust values
    bhist=calc(bhist,fun=function(x) x*conv2*scale2)
    cat(paste("after calculation-->bhist\n",sep=""),file=logfil,append=TRUE)
    #Apply mask
    bhist=mask(bhist,offsm,maskvalues=0,updatevalues=NA)
    #Fill in values with offsm=1
    #Calculate distance and direction from all NA pixels to the nearest non-NA pixel
    dist=distance(subset(bhist,1))
    direct=direction(subset(bhist,1),from=FALSE)
    #Retrieve coordinates of NA pixels
    #NA raster
    rna=is.na(bhist)
    #Store coordinates
    na.x=init(rna,'x')
    na.y=init(rna,'y')
    #Calculate coordinates to nearest non-NA pixel
    co.x = na.x + dist * sin(direct)
    co.y = na.y + dist * cos(direct)
    co = cbind(co.x[], co.y[])
    # extract values of nearest non-NA cell with coordinates co
    NAVals <- raster::extract(bhist, co, method='simple')
    r.NAVals <- rna # initiate new raster
    r.NAVals[] <- NAVals # store values in raster
    # cover nearest non-NA value at NA locations of original raster
    bhist.filled=cover(x=bhist,y=r.NAVals)
    #Mask values
    bhist=mask(bhist.filled,offsm,maskvalue=0,updatevalue=NA)
    print(paste("after mask-->bhistm\n",sep=""))
    #Get mean
    histm=calc(bhist,mean)

    rm(b)

    maxhm[i]=maxValue(histm)
    minhm[i]=minValue(histm)

    #Get bias-corrected mean
    projmbc=(projm/histm)*wmmbm
    maxpmbc[i]=maxValue(projmbc)
    minpmbc[i]=minValue(projmbc)

    #Save data

#save(bhist,histm,bproj,projm,n,nm,wmmbm,projmbc,file=paste(fns[i],"_rasters.RData",se
p=""))
    #rm(bhist,histm,bproj,projm,projmbc)

  }


#Get list of on-shore (active cells)
onsh=which(values(offsm!=0))
ngages=length(onsh)

#Only 1 run of interest
nruns=1
allprojs=projs[i,1]
```

```
#Initialize arrays
PU1=array(dim=c(ndaysh2,(ngages+3),nruns))
PU2=array(dim=c(ndays2,(ngages+3),nruns))
PUmoyr1=array(dim=c(nyrsh2,12,ngages,nruns))
PUyr1=array(dim=c(nyrsh2,ngages,nruns))
PUmo1=array(dim=c(12,ngages,nruns))
PUmoyr2=array(dim=c(nyrs2,12,ngages,nruns))
PUyr2=array(dim=c(nyrs2,ngages,nruns))
PUmo2=array(dim=c(12,ngages,nruns))

PH1=array(dim=c(ndaysh2,(ngages+3)))
PHmoyr1=array(dim=c(nyrsh2,12,ngages))
PHyr1=array(dim=c(nyrsh2,ngages))
PHmo1=array(dim=c(12,ngages))

#Populate the arrays
PU1[,(1:3),]=cbind(yrsnh2,mosnh2,daysnh2)
PU2[,(1:3),]=cbind(yrs2,mos2,days2)
PH1[,(1:3)]=cbind(yrsnh2,mosnh2,daysnh2)

PU1[,(4:(ngages+3)),]=aperm(getValues(bhist)[onsh,],c(2,1))
PU2[,(4:(ngages+3)),]=aperm(getValues(bproj)[onsh,],c(2,1))
PH1[,(4:(ngages+3))]=aperm(getValues(wmmb)[onsh,],c(2,1))

#Overall seasonal cycle boxplot
bproj_gridave=cbind(yrs2,mos2,days2,cellStats(bproj,mean))
bhist_gridave=cbind(yrsnh2,mosnh2,daysnh2,cellStats(bhist,mean))
wmmb_gridave=cbind(yrsnh2,mosnh2,daysnh2,cellStats(wmmb,mean))

bproj_gridavemoyr=tapply(bproj_gridave[,4],
               list(bproj_gridave[,1],bproj_gridave[,2]),sum,na.rm=TRUE)

bhist_gridavemoyr=tapply(bhist_gridave[,4],
               list(bhist_gridave[,1],bhist_gridave[,2]),sum,na.rm=TRUE)

wmmb_gridavemoyr=tapply(wmmb_gridave[,4],
               list(wmmb_gridave[,1],wmmb_gridave[,2]),sum,na.rm=TRUE)

bproj_gridavemo=tapply(bproj_gridave[,4],
               list(bproj_gridave[,2]),sum,na.rm=TRUE)/nyrs2

bhist_gridavemo=tapply(bhist_gridave[,4],
               list(bhist_gridave[,2]),sum,na.rm=TRUE)/nyrsh2

wmmb_gridavemo=tapply(wmmb_gridave[,4],
               list(wmmb_gridave[,2]),sum,na.rm=TRUE)/nyrsh2

bproj_gridavemod=by(data=bproj_gridave[,4],INDICES=bproj_gridave[,2],FUN=identity)

bhist_gridavemod=by(data=bhist_gridave[,4],INDICES=bhist_gridave[,2],FUN=identity)

wmmb_gridavemod=by(data=wmmb_gridave[,4],INDICES=wmmb_gridave[,2],FUN=identity)

png(paste("Allmodels_moyrboxplot_gridave_curr.png",sep=""))
boxplot((bhist_gridavemoyr),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1
,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(min(bhist_gridavemoyr,bproj_gridavemoyr,wmmb_gridavemoyr),
                max(bhist_gridavemoyr,bproj_gridavemoyr,wmmb_gridavemoyr)))
boxplot((bhist_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",box
wex=0.2,at=(1:12)-.3)
```

```
boxplot(wmmb_gridavemoyr,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
lines((1:12),bhist_gridavemo,lwd=2,col="red")
lines((1:12),wmmb_gridavemo,lwd=2,col="blue")
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Sim.:",startyrh2,"-",endyrh2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light blue"),
        lty=c(NA,NA),lwd=c(NA,NA),border=c("red","blue"),cex=0.6)
dev.off()

png(paste("Allmodels_modboxplot_gridave_curr.png",sep=""))
boxplot((bhist_gridavemod),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,
1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(0,
max(max(sapply(bhist_gridavemod,max,simplify="vector")),max(sapply(bproj_gridavemod,ma
x,simplify="vector")),
                    max(sapply(wmmb_gridavemod,max,simplify="vector")))))
boxplot((bhist_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",boxw
ex=0.2,at=(1:12)-.3)
boxplot(wmmb_gridavemod,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Sim.:",startyrh2,"-",endyrh2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light blue"),
        lty=c(NA,NA),lwd=c(NA,NA),border=c("red","blue"),cex=0.6)
dev.off()


png(paste("Allmodels_moyrboxplot_gridave_currfut.png",sep=""))
boxplot((bhist_gridavemoyr),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1
,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(min(bhist_gridavemoyr,bproj_gridavemoyr,wmmb_gridavemoyr),
                max(bhist_gridavemoyr,bproj_gridavemoyr,wmmb_gridavemoyr)))
boxplot((bhist_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",box
wex=0.2,at=(1:12)-.3)
boxplot((bproj_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="dark green",boxwex=0.2,at=(1:12))
boxplot(wmmb_gridavemoyr,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
lines((1:12),bhist_gridavemo,lwd=2,col="red")
lines((1:12),bproj_gridavemo,lwd=2,col="dark green")
lines((1:12),wmmb_gridavemo,lwd=2,col="blue")
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light green","light
blue"),
        lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","dark green","blue"),cex=0.6)
dev.off()

png(paste("Allmodels_modboxplot_gridave_currfut.png",sep=""))
boxplot((bhist_gridavemod),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,
1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
```

```
         ylim=c(0,
max(max(sapply(bhist_gridavemod,max,simplify="vector")),max(sapply(bproj_gridavemod,ma
x,simplify="vector")),
                    max(sapply(wmmb_gridavemod,max,simplify="vector")))))
boxplot((bhist_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",boxw
ex=0.2,at=(1:12)-.3)
boxplot((bproj_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="dark green",boxwex=0.2,at=(1:12))
boxplot(wmmb_gridavemod,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light green","light
blue"),
        lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","dark green","blue"),cex=0.6)
dev.off()

#For each active cell
for (i in 1:ngages) {
  print(paste("i = ",i,sep=""))

  PUi=as.matrix(PU1[,(i+3),])
  ffsU=apply(X=PUi,MARGIN=2,FUN=function(x) ecdf(x)(x))

  PUi2=as.matrix(PU2[,(i+3),])
  ffsU2=apply(X=PUi2,MARGIN=2,FUN=function(x) ecdf(x)(x))

  PHi=PH1[,(i+3)]
  ffsH=ecdf(PHi)(PHi)

  PHmoyr1[,,i]=tapply(PH1[,(i+3)],
               list(PH1[,1],PH1[,2]),sum,na.rm=TRUE)
  PHyr1[,i]=tapply(PH1[,(i+3)],list(PH1[,1]),sum,na.rm=TRUE)
  PHmo1[,i]=tapply(PH1[,(i+3)],list(PH1[,2]),sum,na.rm=TRUE)/nyrsh2

  #Only plot every 100 cells
  if (i%%100==0) {
    print(paste("toplot"))

    png(paste("Allmodels_CDFs_gage",i,"_",i,"_current.png",sep=""))
    matplot(PUi,ffsU,xlim=c(0,max(max(PUi,PHi))),cex=0.1,main=paste("CDFs for gage
",i," (",i,")",sep=""),
            xlab="Precip. (in)",ylab="Prob. exc.")
    points(PHi,ffsH,cex=0.1)
    legend("bottomright",legend=c(paste("Sim. (colors):",startyrh2,"-
",endyrh2),paste("Hist.:",startyrh2,"-",endyrh2)),
           pch=c(5,1),col=c("red","black"))
    grid()
    dev.off()
  }

  for (m in 1:nruns) {
    print(paste("m = ",m,sep=""))

    PUmoyr1[,,i,m]=tapply(PU1[,(i+3),m],
                   list(PU1[,1,m],
                   PU1[,2,m]),sum,na.rm=TRUE)
    PUyr1[,i,m]=tapply(PU1[,(i+3),m],
                list(PU1[,1,m]),sum,na.rm=TRUE)
    PUmo1[,i,m]=tapply(PU1[,(i+3),m],
                list(PU1[,2,m]),sum,na.rm=TRUE)/nyrsh2
```

```
    PUmoyr2[,,i,m]=tapply(PU2[,(i+3),m],
                   list(PU2[,1,m],
                   PU2[,2,m]),sum,na.rm=TRUE)
    PUyr2[,i,m]=tapply(PU2[,(i+3),m],
                list(PU2[,1,m]),sum,na.rm=TRUE)
    PUmo2[,i,m]=tapply(PU2[,(i+3),m],
                list(PU2[,2,m]),sum,na.rm=TRUE)/nyrs2

    #Only plot every 100 cells
    if (i%%100==0) {
      print(paste("toplot"))


png(paste("Model_",m,"_QQplot_CDF_gage",i,"_",i,"_currfut.png",sep=""),height=960,poin
tsize=20)
      nf=layout((c(1,2,3)),heights=c(5,5,5))

qqplot(PHi,PUi[,m],xlim=c(0,max(max(PHi,PUi[,m]))),ylim=c(0,max(max(PHi,PUi[,m]))),
             main=c(paste("QQplot for gage ",i," (",i,")",sep=""),
             paste("m=",m," (",allprojs[m],"), ",startyrh2,"-",endyrh2,sep="")),
             xlab=paste("Hist. Precip. (in):",startyrh2,"-",endyrh2),ylab=paste("Sim.
Precip. (in):",startyrh2,"-",endyrh2))
      lines(c(0,max(max(PHi,PUi[,m]))),c(0,max(max(PHi,PUi[,m]))),col="red")

legend("topleft",legend=c("QQ","1:1"),pch=c(1,NA),col=c("black","red"),lty=c(NA,1))
      grid()

qqplot(PUi[,m],PUi2[,m],xlim=c(0,max(max(PUi[,m],PUi2[,m]))),ylim=c(0,max(max(PUi[,m],
PUi2[,m]))),
             main=c(paste("QQplot for gage ",i," (",i,")",sep=""),
             paste("m=",m," (",allprojs[m],"), ",startyrh2,"-",endyrh2,sep="")),
             xlab=paste("Sim. Precip. (in):",startyrh2,"-",endyrh2),ylab=paste("Sim.
Precip. (in):",startyr2,"-",endyr2))
      lines(c(0,max(max(PUi[,m],PUi2[,m]))),c(0,max(max(PUi[,m],PUi2[,m]))),col="red")

legend("topleft",legend=c("QQ","1:1"),pch=c(1,NA),col=c("black","red"),lty=c(NA,1))
      grid()

plot(PUi[,m],ffsU[,m],xlim=c(0,max(PUi[,m],PUi2[,m],PHi)),cex=0.2,main="CDF",xlab="Pre
cip. (in)",
           ylab="Prob. exc.",col="red")
      points(PUi2[,m],ffsU2[,m],cex=0.2,col="green")
      points(PHi,ffsH,cex=0.2)
      legend("bottomright",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2),
             paste("Hist.:",startyrh2,"-",endyrh2)),
             pch=1,col=c("red","green","black"))
      grid()
      dev.off()
    }
  }#end m

  #Only plot every 100 cells
  if (i%%100==0) {
    print(paste("toplot"))
    #Create boxplots for seasonal cycle and inter-annual variability
    png(paste("Allmodels_moboxplot_gage",i,"_",i,"_currfut.png",sep=""))

boxplot(t(PUmo1[,i,]),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,1,1,a
lpha=1),
            main=c(paste("Seasonal cycle of Precip. for gage ",i," (",i,")",sep="")),
            xlab="Month",ylab="Precip. (in)",
```

```
ylim=c(min(PUmo1[,i,],PUmo2[,i,],PHmo1[,i]),max(PUmo1[,i,],PUmo2[,i,],PHmo1[,i])))

boxplot(t(PUmo1[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex=0.2
5,at=(1:12)-.15)
    boxplot(t(PUmo2[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.25,at=(1:12)+.15)
    lines((1:12)-0.15,PHmo1[,i],lwd=2)
    grid()
    abline(v=1:12,lty=3,col="grey")
    legend("topleft",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2),
           paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green",NA),
           lty=c(NA,NA,1),lwd=c(NA,NA,2),border=c("black","black",NA),cex=0.6)
    dev.off()

    png(paste("Allmodels_moyrboxplot_gage",i,"_",i,"_currfut.png",sep=""))

boxplot((PUmoyr1[,,i,]),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,1,1
,alpha=1),
            main=c(paste("Seasonal cycle of Precip. for gage ",i," (",i,")",sep="")),
            xlab="Month",ylab="Precip. (in)",

ylim=c(min(PUmoyr1[,,i,],PUmoyr2[,,i,],PHmoyr1[,,i]),max(PUmoyr1[,,i,],PUmoyr2[,,i,],P
Hmoyr1[,,i])))

boxplot((PUmoyr1[,,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex=0
.2,at=(1:12)-.3)
    boxplot((PUmoyr2[,,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.2,at=(1:12))

boxplot(PHmoyr1[,,i],xaxt="n",yaxt="n",add=TRUE,boxfill="black",border="black",boxwex=
0.2,at=(1:12)+.3)
    grid()
    abline(v=1:12,lty=3,col="grey")
    legend("topleft",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2),
          paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green","black"),
           lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","light
green","black"),cex=0.6)
    dev.off()


png(paste("Allmodels_yrboxplot_gage",i,"_",i,"_currfut.png",sep=""),width=960,pointsiz
e=20)

boxplot(t(PUyr1[,i,]),names=allyrsh2,xlim=c(0.5,nyrsh2+0.5),boxfill=rgb(1,1,1,alpha=1)
,border=rgb(1,1,1,alpha=1),
            main=c(paste("Inter-annual var. of Precip. for gage ",i,"
(",i,")",sep="")),
            xlab="Year",ylab="Precip. (in)",

ylim=c(min(PUyr1[,i,],PUyr2[,i,],PHyr1[,i]),max(PUyr1[,i,],PUyr2[,i,],PHyr1[,i])))

boxplot(t(PUyr1[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex=0.2
5,at=(1:nyrsh2)-0.15)
    boxplot(t(PUyr2[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.25,at=(1:nyrsh2)+0.15)
    lines((1:nyrsh2)-0.15,PHyr1[,i],lwd=2)
    grid()
    abline(v=allyrsh2,lty=3,col="grey")
    legend("topleft",legend=c(paste("Sim.:",startyrh2,"-
",endyrh2),paste("Sim.:",startyr2,"-",endyr2,"shifted"),
```

```
           paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green",NA),
           lty=c(NA,NA,1),lwd=c(NA,NA,2),border=c("black","black",NA),cex=0.6)
    dev.off()
  }

}#end i

#Save unadjusted data (pre-BC)
save(PU1,PUmoyr1,PUyr1,PUmo1,PU2,PUmoyr2,PUyr2,PUmo2,PH1,PHmoyr1,PHyr1,PHmo1,file="Pre
BC_Precip.RData")

#Do AQDM or MQDM Bias Correction here
#Initialize BC matrices to uncorrected ones
PUBC1=PU1
PUBC2=PU2
PUBCmoyr1=PUmoyr1
PUBCyr1=PUyr1
PUBCmo1=PUmo1
PUBCmoyr2=PUmoyr2
PUBCyr2=PUyr2
PUBCmo2=PUmo2

if (file.exists("Inf_cells.txt")) {
  file.remove("Inf_cells.txt")
}

infc=0
l=list()

for (i in 1:ngages) {
  err1=0
  print(paste("i = ",i,sep=""))


  #This gives values for the month
  Ubymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU1[,(i+3),x],
                                              INDICES=PU1[,2,x],FUN=identity))

  U2bymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU2[,(i+3),x],
                                              INDICES=PU2[,2,x],FUN=identity))

  H1bymo=by(data=PH1[,(i+3)],INDICES=PH1[,2],FUN=identity)


  #This gives sorted values for the month
  #sortedUbymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU1[,(i+3),x],
  #                                             INDICES=PU1[,2,x],FUN=sort))

  #This gives CDF value (non-exceedance prob.) for a particular value with a month's
CDF
  #ffsUbymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU1[,(i+3),x],
  #                                             INDICES=PU1[,2,x],FUN=function(x)
ecdf(x)(x)))

  #This can be used to get quantile of interest
  #ecdfUbymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU1[,(i+3),x],
  #                                             INDICES=PU1[,2,x],FUN=function(x)
ecdf(x)))

  #ecdfU2bymo=lapply(seq_len(nruns),FUN=function(x) by(data=PU2[,(i+3),x],
  #                                             INDICES=PU2[,2,x],FUN=function(x)
ecdf(x)))
```

```
  #ecdfH1bymo=by(data=PH1[,(i+3)],INDICES=PH1[,2],FUN=function(x) ecdf(x))

  for (m in 1:nruns) {
    print(paste("m = ",m,sep=""))

    #MQDM
    PUBC1[,(i+3),m]=mapply(function(mo,xmc) {
                                if (xmc==0 ||
(quantile(H1bymo[mo][[1]],ecdf(Ubymo[[m]][mo][[1]])(xmc)))<=0.0001) {
                                  0
                                } else {

xmc*(quantile(H1bymo[mo][[1]],ecdf(Ubymo[[m]][mo][[1]])(xmc)))/

(quantile(Ubymo[[m]][mo][[1]],ecdf(Ubymo[[m]][mo][[1]])(xmc)))
                                }},
                            mo=PU1[,2,m],xmc=PU1[,(i+3),m])

    if (length(which(PUBC1[,(i+3),m]==Inf))) {
      cat(paste("PUBC1: i",i,"\n"),file="Inf_cells.txt",append=TRUE)
      arr=which(PUBC1[,(i+3),m]==Inf)
      PUBC1[arr,(i+3),m]=0
      err1=0 #used to be set to 1, but now set to 0 since Inf values changed to 0 (see
comment at bottom)
    }
    #AQDM
    #PUBC1[,(i+3),m]=mapply(function(mo,xmc) {
    #
xmc+(quantile(H1bymo[mo][[1]],ecdf(Ubymo[[m]][mo][[1]])(xmc)))-
    #
(quantile(Ubymo[[m]][mo][[1]],ecdf(Ubymo[[m]][mo][[1]])(xmc)))
    #                             },
    #                        mo=PU1[,2,m],xmc=PU1[,(i+3),m])

    #PUBC1[PUBC1[,(i+3),m]<0,(i+3),m]=0

    #MQDM
    PUBC2[,(i+3),m]=mapply(function(mo,xmp) {
                                if (xmp==0 ||
(quantile(H1bymo[mo][[1]],ecdf(U2bymo[[m]][mo][[1]])(xmp)))<=0.0001) {
                                  0
                                } else {

xmp*(quantile(H1bymo[mo][[1]],ecdf(U2bymo[[m]][mo][[1]])(xmp)))/

(quantile(Ubymo[[m]][mo][[1]],ecdf(U2bymo[[m]][mo][[1]])(xmp)))
                                }},
                            mo=PU2[,2,m],xmp=PU2[,(i+3),m])

    if (length(which(PUBC2[,(i+3),m]==Inf))) {
      infc = infc+1
      cat(paste("PUBC2: i",i,"\n"),file="Inf_cells.txt",append=TRUE)
      arr=which(PUBC2[,(i+3),m]==Inf)
      mos=PUBC2[arr,2,1]
      myl=list("cell"=i,"errarray"=arr,"xmp"=PU2[arr,(i+3),m],"mos"=mos,

"qq1"=(quantile(H1bymo[mos][[1]],ecdf(U2bymo[[m]][mos][[1]])(PU2[arr,(i+3),m]))),

"qq2"=(quantile(Ubymo[[m]][mos][[1]],ecdf(U2bymo[[m]][mos][[1]])(PU2[arr,(i+3),m]))))
      l[[infc]]=myl
      PUBC2[arr,(i+3),m]=0
      err1=0 #used to be set to 1, but now set to 0 since Inf values changed to 0 (see
comment at bottom)
```

```
    }
    #AQDM
    #PUBC2[,(i+3),m]=mapply(function(mo,xmp) {
    #
xmp+(quantile(H1bymo[mo][[1]],ecdf(U2bymo[[m]][mo][[1]])(xmp)))-
    #
(quantile(Ubymo[[m]][mo][[1]],ecdf(U2bymo[[m]][mo][[1]])(xmp)))
    #                            },
    #                        mo=PU2[,2,m],xmp=PU2[,(i+3),m])

    #PUBC2[PUBC2[,(i+3),m]<0,(i+3),m]=0

    ffsUBC1=ecdf(PUBC1[,(i+3),m])(PUBC1[,(i+3),m])
    ffsUBC2=ecdf(PUBC2[,(i+3),m])(PUBC2[,(i+3),m])
    ffsH=ecdf(PH1[,(i+3)])(PH1[,(i+3)])

    PUBCmoyr1[,,i,m]=tapply(PUBC1[,(i+3),m],
                   list(PUBC1[,1,m],
                   PUBC1[,2,m]),sum,na.rm=TRUE)
    PUBCyr1[,i,m]=tapply(PUBC1[,(i+3),m],
                list(PUBC1[,1,m]),sum,na.rm=TRUE)
    PUBCmo1[,i,m]=tapply(PUBC1[,(i+3),m],
                list(PUBC1[,2,m]),sum,na.rm=TRUE)/nyrsh2

    PUBCmoyr2[,,i,m]=tapply(PUBC2[,(i+3),m],
                   list(PUBC2[,1,m],
                   PUBC2[,2,m]),sum,na.rm=TRUE)
    PUBCyr2[,i,m]=tapply(PUBC2[,(i+3),m],
                list(PUBC2[,1,m]),sum,na.rm=TRUE)
    PUBCmo2[,i,m]=tapply(PUBC2[,(i+3),m],
                list(PUBC2[,2,m]),sum,na.rm=TRUE)/nyrs2

    #Now generate corrected plots
    #Only plot every 100 cells
    if (i%%100==0 & err1==0) {
      print(paste("toplot"))


png(paste("Model_",m,"_QQplot_CDF_gage",i,"_",i,"_currfut_BC.png",sep=""),height=960,p
ointsize=20)
      nf=layout((c(1,2,3)),heights=c(5,5,5))

qqplot(PH1[,(i+3)],PUBC1[,(i+3),m],xlim=c(0,max(max(PH1[,(i+3)],PUBC1[,(i+3),m]))),yli
m=c(0,max(max(PH1[,(i+3)],PUBC1[,(i+3),m]))),
             main=c(paste("Post-BC QQplot for gage ",i," (",i,")",sep=""),
             paste("m=",m," (",allprojs[m],"), ",startyrh2,"-",endyrh2,sep="")),
             xlab=paste("Hist. Precip. (in):",startyrh2,"-",endyrh2),ylab=paste("BC
Sim. Precip. (in):",startyrh2,"-",endyrh2))

lines(c(0,max(max(PH1[,(i+3)],PUBC1[,(i+3),m]))),c(0,max(max(PH1[,(i+3)],PUBC1[,(i+3),
m]))),col="red")

legend("topleft",legend=c("QQ","1:1"),pch=c(1,NA),col=c("black","red"),lty=c(NA,1))
      grid()

qqplot(PUBC1[,(i+3),m],PUBC2[,(i+3),m],xlim=c(0,max(max(PUBC1[,(i+3),m],PUBC2[,(i+3),m
]))),
             ylim=c(0,max(max(PUBC1[,(i+3),m],PUBC2[,(i+3),m]))),
             main=c(paste("Post B-C QQplot for gage ",i," (",i,")",sep=""),
             paste("m=",m," (",allprojs[m],"), ",startyrh2,"-",endyrh2,sep="")),
             xlab=paste("BC Sim. Precip. (in):",startyrh2,"-",endyrh2),ylab=paste("BC
Sim. Precip. (in):",startyr2,"-",endyr2))
```

```
lines(c(0,max(max(PUBC1[,(i+3),m],PUBC2[,(i+3),m]))),c(0,max(max(PUBC1[,(i+3),m],PUBC2
[,(i+3),m]))),col="red")

legend("topleft",legend=c("QQ","1:1"),pch=c(1,NA),col=c("black","red"),lty=c(NA,1))
      grid()

plot(PUBC1[,(i+3),m],ffsUBC1,xlim=c(0,max(PUBC1[,(i+3),m],PUBC2[,(i+3),m],PH1[,(i+3)])
),cex=0.2,
           main="Post-BC CDF",xlab="Precip. (in)",ylab="Prob. exc.",col="red")
      points(PUBC2[,(i+3),m],ffsUBC2,cex=0.2,col="green")
      points(PH1[,(i+3)],ffsH,cex=0.2)
      legend("bottomright",legend=c(paste("BC Sim.:",startyrh2,"-",endyrh2),paste("BC
Sim.:",startyr2,"-",endyr2),
             paste("Hist.:",startyrh2,"-",endyrh2)),
             pch=1,col=c("red","green","black"))
      grid()
      dev.off()
    }

  }#end m

  #Only plot every 100 cells
  if (i%%100==0 & err1==0) {
    print(paste("toplot"))

    #Create boxplots for seasonal cycle and inter-annual variability
    png(paste("Allmodels_moboxplot_gage",i,"_",i,"_currfut_BC.png",sep=""))

boxplot(t(PUBCmo1[,i,]),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,1,1
,alpha=1),
            main=c(paste("Post-BC Seas. cycle of Precip.-gage ",i," (",i,")",sep="")),
            xlab="Month",ylab="Precip. (in)",

ylim=c(min(PUBCmo1[,i,],PUBCmo2[,i,],PHmo1[,i]),max(PUBCmo1[,i,],PUBCmo2[,i,],PHmo1[,i
])))

boxplot(t(PUBCmo1[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex=0
.25,at=(1:12)-.15)
    boxplot(t(PUBCmo2[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.25,at=(1:12)+.15)
    lines((1:12)-0.15,PHmo1[,i],lwd=2)
    grid()
    abline(v=1:12,lty=3,col="grey")
    legend("topleft",legend=c(paste("BC Sim.:",startyrh2,"-",endyrh2),paste("BC
Sim.:",startyr2,"-",endyr2),
           paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green",NA),
           lty=c(NA,NA,1),lwd=c(NA,NA,2),border=c("black","black",NA),cex=0.6)
    dev.off()

    png(paste("Allmodels_moyrboxplot_gage",i,"_",i,"_currfut_BC.png",sep=""))

boxplot((PUBCmoyr1[,,i,]),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb(1,1
,1,alpha=1),
            main=c(paste("Seasonal cycle of Precip. for gage ",i," (",i,")",sep="")),
            xlab="Month",ylab="Precip. (in)",

ylim=c(min(PUBCmoyr1[,,i,],PUBCmoyr2[,,i,],PHmoyr1[,,i]),max(PUBCmoyr1[,,i,],PUBCmoyr2
[,,i,],PHmoyr1[,,i])))

boxplot((PUBCmoyr1[,,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex
=0.2,at=(1:12)-.3)
```

```
    boxplot((PUBCmoyr2[,,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.2,at=(1:12))

boxplot(PHmoyr1[,,i],xaxt="n",yaxt="n",add=TRUE,boxfill="black",border="black",boxwex=
0.2,at=(1:12)+.3)
    grid()
    abline(v=1:12,lty=3,col="grey")
    legend("topleft",legend=c(paste("BC Sim.:",startyrh2,"-",endyrh2),paste("BC
Sim.:",startyr2,"-",endyr2),
          paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green","black"),
           lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","light
green","black"),cex=0.6)
    dev.off()

png(paste("Allmodels_yrboxplot_gage",i,"_",i,"_currfut_BC.png",sep=""),width=960,point
size=20)

boxplot(t(PUBCyr1[,i,]),names=allyrsh2,xlim=c(0.5,nyrsh2+0.5),boxfill=rgb(1,1,1,alpha=
1),border=rgb(1,1,1,alpha=1),
            main=c(paste("Post-BC Inter-annual var. of Precip.-gage ",i,"
(",i,")",sep="")),
            xlab="Year",ylab="Precip. (in)",

ylim=c(min(PUBCyr1[,i,],PUBCyr2[,i,],PHyr1[,i]),max(PUBCyr1[,i,],PUBCyr2[,i,],PHyr1[,i
])))

boxplot(t(PUBCyr1[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="red",border="red",boxwex=0
.25,at=(1:nyrsh2)-0.15)
    boxplot(t(PUBCyr2[,i,]),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="light green",boxwex=0.25,at=(1:nyrsh2)+0.15)
    lines((1:nyrsh2)-0.15,PHyr1[,i],lwd=2)
    grid()
    abline(v=allyrsh2,lty=3,col="grey")
    legend("topleft",legend=c(paste("BC Sim.:",startyrh2,"-",endyrh2),paste("BC
Sim.:",startyr2,"-",endyr2,"shifted"),
           paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("red","light green",NA),
           lty=c(NA,NA,1),lwd=c(NA,NA,2),border=c("black","black",NA),cex=0.6)
    dev.off()
  }

}#end i

#Save list of cells with issues
save(l,file="List_of_list_inf_cells.RData")

#The list shows that the numerator (Xoc) is almost 0 (0.0001-0.007 in/d) when the
denominator (Xmc) is zero.
#Therefore, for all intents and purposes, one can make the Inf values equal to 0. Re-
run the code above setting Inf to 0.

#Save adjusted (post-BC) file
save(PUBC1,PUBCmoyr1,PUBCyr1,PUBCmo1,PUBC2,PUBCmoyr2,PUBCyr2,PUBCmo2,PH1,PHmoyr1,PHyr1
,PHmo1,
     file="PostBC_MQDM_Precip.RData")

#Populate a new brick with the new post-BC data and save to ncfile to read in python
and create the binary files
onshrc=rowColFromCell(bhist,onsh)
bhbc=array(dim=c(nrow(bhist),ncol(bhist),nlayers(bhist)))
for (g in 1:ngages) {
  print(paste("g=",g))
  bhbc[onshrc[g,1],onshrc[g,2],]=PUBC1[,(g+3),1]
}
```

```
bhistbc=brick(bhbc,crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs")
extent(bhistbc)=extent(wmmbm)
writeRaster(bhistbc,paste(fns[56],"_histBC_Precip.nc",sep=""),format="CDF",varname="pr
",varunit="in/day",
            longname="Bias-corrected historical
precipitation",xname="x",yname="y",zname="t",zunit=paste("days since",startyrh2),
            NAflag=-999)

bpbc=array(dim=c(nrow(bproj),ncol(bproj),nlayers(bproj)))
#bpbc=array(dim=c(nrow(bhist),ncol(bhist),nlayers(bproj)))
for (g in 1:ngages) {
  print(paste("g=",g))
  bpbc[onshrc[g,1],onshrc[g,2],]=PUBC2[,(g+3),1]
}
bprojbc=brick(bpbc,crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs")
extent(bprojbc)=extent(wmmbm)
writeRaster(bprojbc,paste(fns[56],"_projBC_Precip.nc",sep=""),format="CDF",varname="pr
",varunit="in/day",
            longname="Bias-corrected projected
precipitation",xname="x",yname="y",zname="t",zunit=paste("days since",startyr2),
            NAflag=-999)

save(bhistbc,bprojbc,file="PostBC_MQDM_Precip_bricks.RData")

minz=32
maxz=83
#Create levelplots for the bias-corrected model projection
print(paste("i=",i))
#Create levelplot of histmbc
histmbc=calc(bhistbc,mean)
meanhmbc=round(nds2*cellStats(histmbc,mean),2)
png(paste("LOCA_mean_rainfall_hist_",fns[56],"_dailyBC.png",sep=""))
print(levelplot(histmbc*nds2,margin=FALSE,at=seq(minz,maxz,1),
          main=paste("Daily B.C. LOCA rainfall (",meanhmbc," in/yr)\n",
          startyrh2,"-",endyrh2," (",fns[56],")",sep=""),
          xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
dev.off()

#Create levelplot of projmdbc
projmdbc=calc(bprojbc,mean)
meanpmdbc=round(nds2*cellStats(projmdbc,mean),2)
png(paste("LOCA_mean_rainfall_proj_",fns[56],"_dailyBC.png",sep=""))
print(levelplot(projmdbc*nds2,margin=FALSE,at=seq(minz,maxz,1),
          main=paste("Daily B.C. LOCA rainfall (",meanpmdbc," in/yr)\n",
          startyr2,"-",endyr2," (",fns[56],")",sep=""),
          xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83"))
dev.off()

png(paste("LOCA_projbc_to_SFWMM_ratio_",fns[56],"_dailyBC.png",sep=""))
print(levelplot(projmdbc/wmmbm,col.regions=brewer.rdbu(21),margin=FALSE,
         at=seq(0.70,1.30,0.025),
         main=paste("Daily B.C.",fns[56]," (",startyr2,"-",endyr2,")\n",
         "to SFWMM rainfall (",startyrh,"-",endyrh,")",sep=""),
         xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83",cex=0.5))
dev.off()

png(paste("LOCA_projbc-SFWMM_",fns[56],"_dailyBC.png",sep=""))
print(levelplot(nds2*(projmdbc-wmmbm),col.regions=brewer.rdbu(29),margin=FALSE,
          at=seq(-16,16,1),
          main=paste("Daily B.C.",fns[56]," (",startyr2,"-",endyr2,")\n",
          "- SFWMM rainfall (",startyrh,"-",endyrh,") (",
```

```
          round(nds2*cellStats((projmdbc-wmmbm),mean),2)," in/yr)",sep=""),
          xlab="X (m) UTM17N, NAD83",ylab="Y (m) UTM17N, NAD83",cex=0.5))
dev.off()

#Overall seasonal cycle boxplot
bprojdbc_gridave=cbind(yrs2,mos2,days2,cellStats(bprojbc,mean))
bhistdbc_gridave=cbind(yrsnh2,mosnh2,daysnh2,cellStats(bhistbc,mean))
wmmb_gridave=cbind(yrsnh2,mosnh2,daysnh2,cellStats(wmmb,mean))

bprojdbc_gridavemoyr=tapply(bprojdbc_gridave[,4],
               list(bprojdbc_gridave[,1],bprojdbc_gridave[,2]),sum,na.rm=TRUE)

bhistdbc_gridavemoyr=tapply(bhistdbc_gridave[,4],
               list(bhistdbc_gridave[,1],bhistdbc_gridave[,2]),sum,na.rm=TRUE)

wmmb_gridavemoyr=tapply(wmmb_gridave[,4],
               list(wmmb_gridave[,1],wmmb_gridave[,2]),sum,na.rm=TRUE)

bprojdbc_gridavemo=tapply(bprojdbc_gridave[,4],
               list(bprojdbc_gridave[,2]),sum,na.rm=TRUE)/nyrs2

bhistdbc_gridavemo=tapply(bhistdbc_gridave[,4],
               list(bhistdbc_gridave[,2]),sum,na.rm=TRUE)/nyrsh2

wmmb_gridavemo=tapply(wmmb_gridave[,4],
               list(wmmb_gridave[,2]),sum,na.rm=TRUE)/nyrsh2

bprojdbc_gridavemod=by(data=bprojdbc_gridave[,4],INDICES=bprojdbc_gridave[,2],FUN=iden
tity)

bhistdbc_gridavemod=by(data=bhistdbc_gridave[,4],INDICES=bhistdbc_gridave[,2],FUN=iden
tity)

wmmb_gridavemod=by(data=wmmb_gridave[,4],INDICES=wmmb_gridave[,2],FUN=identity)

png(paste("Allmodels_moyrboxplot_gridave_curr_dailyBC.png",sep=""))
boxplot((bhistdbc_gridavemoyr),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rg
b(1,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(min(bhistdbc_gridavemoyr,bprojdbc_gridavemoyr,wmmb_gridavemoyr),
                max(bhistdbc_gridavemoyr,bprojdbc_gridavemoyr,wmmb_gridavemoyr)))
boxplot((bhistdbc_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",
boxwex=0.2,at=(1:12)-.3)
boxplot(wmmb_gridavemoyr,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
lines((1:12),bhistdbc_gridavemo,lwd=2,col="red")
lines((1:12),wmmb_gridavemo,lwd=2,col="blue")
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Daily B.C. Sim.:",startyrh2,"-",endyrh2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light blue"),
        lty=c(NA,NA),lwd=c(NA,NA),border=c("red","blue"),cex=0.6)
dev.off()

png(paste("Allmodels_modboxplot_gridave_curr_dailyBC.png",sep=""))
boxplot((bhistdbc_gridavemod),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb
(1,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(0,
max(max(sapply(bhistdbc_gridavemod,max,simplify="vector")),max(sapply(bprojdbc_gridave
mod,max,simplify="vector")),
```

```
                    max(sapply(wmmb_gridavemod,max,simplify="vector")))))
boxplot((bhistdbc_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",b
oxwex=0.2,at=(1:12)-.3)
boxplot(wmmb_gridavemod,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Daily Sim.:",startyrh2,"-",endyrh2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light blue"),
        lty=c(NA,NA),lwd=c(NA,NA),border=c("red","blue"),cex=0.6)
dev.off()

png(paste("Allmodels_moyrboxplot_gridave_currfut_dailyBC.png",sep=""))
boxplot((bhistdbc_gridavemoyr),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rg
b(1,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(min(bhistdbc_gridavemoyr,bprojdbc_gridavemoyr,wmmb_gridavemoyr),
                max(bhistdbc_gridavemoyr,bprojdbc_gridavemoyr,wmmb_gridavemoyr)))
boxplot((bhistdbc_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",
boxwex=0.2,at=(1:12)-.3)
boxplot((bprojdbc_gridavemoyr),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="dark green",boxwex=0.2,at=(1:12))
boxplot(wmmb_gridavemoyr,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
lines((1:12),bhistdbc_gridavemo,lwd=2,col="red")
lines((1:12),bprojdbc_gridavemo,lwd=2,col="dark green")
lines((1:12),wmmb_gridavemo,lwd=2,col="blue")
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Daily B.C. Sim.:",startyrh2,"-",endyrh2),paste("Daily
B.C. Sim.:",startyr2,"-",endyr2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light green","light
blue"),
        lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","dark green","blue"),cex=0.6)
dev.off()

png(paste("Allmodels_modboxplot_gridave_currfut_dailyBC.png",sep=""))
boxplot((bhistdbc_gridavemod),xlim=c(0.5,12+0.5),boxfill=rgb(1,1,1,alpha=1),border=rgb
(1,1,1,alpha=1),
         main=c(paste("Seasonal cycle of Precip. for entire domain",sep="")),
         xlab="Month",ylab="Precip. (in)",
         ylim=c(0,
max(max(sapply(bhistdbc_gridavemod,max,simplify="vector")),max(sapply(bprojdbc_gridave
mod,max,simplify="vector")),
                    max(sapply(wmmb_gridavemod,max,simplify="vector")))))
boxplot((bhistdbc_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="pink",border="red",b
oxwex=0.2,at=(1:12)-.3)
boxplot((bprojdbc_gridavemod),xaxt="n",yaxt="n",add=TRUE,boxfill="light
green",border="dark green",boxwex=0.2,at=(1:12))
boxplot(wmmb_gridavemod,xaxt="n",yaxt="n",add=TRUE,boxfill="light
blue",border="blue",boxwex=0.2,at=(1:12)+.3)
grid()
abline(v=1:12,lty=3,col="grey")
legend("topleft",legend=c(paste("Daily B.C. Sim.:",startyrh2,"-",endyrh2),paste("Daily
B.C. Sim.:",startyr2,"-",endyr2),
      paste("Hist.:",startyrh2,"-",endyrh2)),fill=c("pink","light green","light
blue"),
        lty=c(NA,NA,NA),lwd=c(NA,NA,NA),border=c("red","dark green","blue"),cex=0.6)
dev.off()
}
```

Appendix C. R code for calculating average Everglades water levels by julian day

```
################################################################################

interp_sfwmmstages <- function(){

################################################################################

library(reshape)
library(RNetCDF)
library(raster)
library(rgdal)
library(fields)
#library(RColorBrewer)
library(pals)
library(foreach)
library(parallel)
library(doParallel)
#library(tcltk)
#library(doSNOW)
#library(gdalUtils)
library(rasterVis)
library(lattice)
library(magic)
library(akima)
library(concaveman)

#Main variables
SFWMM_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Water_levels/SFWMD"
setwd(SFWMM_dir)

#Load shapefiles
hydrog=readOGR("Z:/miriza/Work/FIU/FL_Building_Code/Data/GIS/swr_geography/umd_swr_hyd
rography.shp")
basemap=readOGR("Z:/miriza/Work/FIU/FL_Building_Code/Data/GIS/basemap/basemap.shp")
saltstr=readOGR("Z:/miriza/Work/FIU/FL_Building_Code/Data/GIS/structures/salinitycontr
olstructures.shp")
drstr=readOGR("Z:/miriza/Work/FIU/FL_Building_Code/Data/GIS/structures/drainagefloodco
ntrolstructures.shp")

#Read in SFWMM netCDF file
#Proj4js.defs["ESRI:102258"] = "+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=m +no_defs";
ncf=open.nc("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/SFWMD/rain_v4.7_1914_20
16_sfwmd.nc")
#SFWMM coordinates are in ft NAD1983 HARN StatePlane FL East FIPS0901 (but ESRI:102258
is in m)
cds=var.get.nc(ncf,"coords")
cds=cds
#Change SFWMM cell centroid coordinates to match projection of M-D MODFLOW grid
d <- data.frame(x=cds[1,], y=cds[2,])
coordinates(d) <- c("x", "y")
proj4string(d) <- CRS("+proj=tmerc +lat_0=24.33333333333333 +lon_0=-81
+k=0.9999411764705882 +x_0=200000 +y_0=0 +ellps=GRS80 +units=ft +no_defs")
CRS.new <- CRS("+init=epsg:26917") #m
d.n <- spTransform(d, CRS.new) #m
roco=t(var.get.nc(ncf,"roco"))
rocoshift=roco
```

```
rocoshift[,2]=roco[,2]-29
xys=d.n@coords
xys=xys[(rocoshift[,2]>=1),]
rocoshift=rocoshift[(rocoshift[,2]>=1),]
ROWCOL=rocoshift[,1]*100+rocoshift[,2]
close.nc(ncf)

#Find SFWMM rocos for WCA-3B/ENP eastern-most cells
wca3benp_east_rocos=rbind(c(32,27),c(31,27),c(30,27),c(29,27),c(28,27),c(27,27),c(26,2
6),c(25,26),c(24,26),c(23,26),c(22,26),c(21,26),c(20,26),

c(19,26),c(18,26),c(17,26),c(16,25),c(15,24),c(14,24),c(13,24),c(12,24),c(11,23),c(10,
24),c(9,24),c(8,25),
                          c(7,25),c(6,26),c(5,27),c(4,25),c(3,22),c(2,21))
wca3benp_east_ROWCOL=wca3benp_east_rocos[,1]*100+wca3benp_east_rocos[,2]
indx=which(ROWCOL%in%wca3benp_east_ROWCOL)
x3b=xys[indx,1]
y3b=xys[indx,2]

#Find SFWMM cells west and up to WCA-3B/ENP eastern-most cells
#Index will be 1 if 2x2 cell will be kept for interpolation
indkeep=vector(mode="numeric",length=dim(rocoshift)[1])
for (r in 1:dim(wca3benp_east_rocos)[1]) {
  indkeep[which(rocoshift[,1]==wca3benp_east_rocos[r,1] &
rocoshift[,2]<=wca3benp_east_rocos[r,2])]=1
}
xxs=xys[indkeep==1,1]
yys=xys[indkeep==1,2]
concpoly=concaveman(as.matrix(cbind(xxs,yys)),concavity=1.2)

#Read in raster with EDEN cells on the M-D MODFLOW grid
#Proj4js.defs["EPSG:26917"] = "+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m
+no_defs"
n=raster("Z:/miriza/Work/FIU/FL_Building_Code/Data/Water_levels/MD_MODFLOW/eden_cells.
nc",crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")
#Get extent
extn=bbox(n)
#Centroids of cells on to which to interpolate the SFWMM data
x0 = seq(extn[1,1]+250,extn[1,2]-250,500)
y0 = seq(extn[2,2]-250,extn[2,1]+250,-500)
p <- rasterToPoints(n, fun=function(x){x == 1})
#Coordinates of cells on which to interpolate the SFWMM data to
#x0 = p[,1]
#y0 = p[,2]
#Go 1 SFWMM cell (2 mi = 3218.7 m) outside the NEXRAD extent
#extn2=extn
#extn2[,1]=extn2[,1]-3218.7
#extn2[,2]=extn2[,2]+3218.7
n2=as.array(n)[,,1]

#Write out coordinates to get NGVD29 to NAVD88 datum conversion using Corpscon 6.0.1
#Coordinates are in EPSG:26917 UTM Zone 17N NAD83 m
grcoords=expand.grid(x0,y0)
names(grcoords)=c("x","y")
write.table(grcoords,file="MODFLOW_grid_coords.csv",sep=",",col.names=TRUE,row.names=T
RUE)
grrocos=expand.grid(seq(1,ncol(n),1),seq(1,nrow(n),1))
```

```
#Read in raster with ibound
ibm=raster("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/USGS_MODFLOW_NEXRAD/UMD_
ibound.nc",crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")
ibm[ibm!=5]=0
ibm2=as.array(ibm)[,,1]

#Get eden cells with ibm=5
iii=which(n2==1 & ibm2==5,arr.ind=TRUE)

#Find MODFLOW grid coordinates within concave hull polygon of SFWMM grid centroids
inpoly=point.in.polygon(grcoords[,1],grcoords[,2],concpoly[,1],concpoly[,2])
indinpoly=which(inpoly==1)
#Outside of polygon
indoutpoly=which(inpoly==0)
#Outside of polygon and also ib=5 (cells that require stages to be defined for BCs)
ibmv=as.vector(ibm)
indoutpolyib5=which(inpoly==0 & ibmv==5)
#Inside of polygon and also ib=5
indinpolyib5=which(inpoly==1 & ibmv==5)

#For each MODFLOW cell in indoutpolyib5 get closest MODFLOW cell with data
p=cbind(grcoords[indinpolyib5,1],grcoords[indinpolyib5,2])
q=cbind(grcoords[indoutpolyib5,1],grcoords[indoutpolyib5,2])
r=rdist(p,q)
idclosest=apply(r,2,which.min)

#Check that correct cells are being selected
png("Inside_outside_WCA3ENP_polygon_cells.png",width=720,height=720)
#plot(n)
plot(xys[which(indkeep==1),1],xys[which(indkeep==1),2],pch=15,xlim=c(extn[1,1],extn[1,
2]),ylim=c(extn[2,1],extn[2,2]),asp=1,
     main='Boundary cells in WCA3/ENP',xlab='',ylab='')
lines(basemap)
lines(hydrog)
points(x0[iii[,2]],y0[iii[,1]],col='yellow',cex=0.5,pch=1)
lines(concpoly[,1],concpoly[,2],col='blue',lwd=2.0)
#points(grcoords[indinpoly,1],grcoords[indinpoly,2],col='black',cex=0.2)
points(grcoords[indoutpolyib5,1],grcoords[indoutpolyib5,2],col='red',cex=0.5,pch=1)
legend('topright',legend=c('SFWMM cells','Polygon','BC cells in polygon','BC cells
outside polygon'),pch=c(15,NA,1,1),
        col=c('black','blue','yellow','red'),lwd=c(NA,2.0,NA,NA))
dev.off()

#Read in EDEN stage data (m NAVD88) for 1996-2010
eden_stage=brick("Z:/miriza/Work/FIU/FL_Building_Code/Data/Water_levels/MD_MODFLOW/ede
n_stage.nc",crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")
eden_mean=calc(eden_stage,mean)
eden_mean=mask(eden_mean,n,maskvalue=0,updatevalue=NA)
eden_mean=mask(eden_mean,ibm,maskvalue=0,updatevalue=NA)

#Read in topo (ft NGVD29) for CERP0 run
cerp0_ael=read.csv("cerp0_ael.roco",header=FALSE)
#Read in daily_stg_minus_lsel.bin for CERP0 run
cerp0_dsml = read.table("cerp0_dsml.txt",sep="",header=TRUE)
names(cerp0_dsml)[4:(dim(cerp0_dsml)[2])]=simplify2array(lapply(strsplit(names(cerp0_d
sml)[4:(dim(cerp0_dsml)[2])],"[.]"),FUN=function(x)
as.numeric(x[2])*100+as.numeric(x[3])))
cerp0_stage = cerp0_dsml
```

```
x=vector(length=(dim(cerp0_dsml)[2]-3),mode='numeric')
y=vector(length=(dim(cerp0_dsml)[2]-3),mode='numeric')
#Add topo to daily_stg_minus_lsel.bin
for (i in 4:(dim(cerp0_dsml)[2])) {
  cell_topo=cerp0_ael[which(cerp0_ael[,1]==as.numeric(names(cerp0_dsml)[i])),2]
  print(paste("i=",i,"cell topo=",cell_topo))
  cerp0_stage[,i]=cerp0_dsml[,i]+cell_topo
  x[i-3]=xys[which(ROWCOL==as.numeric(names(cerp0_dsml)[i])),1]
  y[i-3]=xys[which(ROWCOL==as.numeric(names(cerp0_dsml)[i])),2]
}

#Check coordinates of SFWMM cells read in
plot(n)
points(x,y,col='red',cex=0.5)

cerp0_intstage = array(data=-999,dim=c(length(y0),length(x0),dim(cerp0_dsml)[1]))

#Interpolate daily stage values for CERP0 run
for (d in 1:(dim(cerp0_stage)[1])) {
  print(paste("d=",d))
  intstage=interp(x,y,as.numeric(cerp0_stage[d,4:(dim(cerp0_stage)[2])]),x0,y0)
  nas=which(!(is.na(intstage$z)),arr.ind=TRUE)
  #points(x0[nas[,1]],y0[nas[,2]],cex=0.1)
  for (ii in 1:dim(iii)[1]) {
    cerp0_intstage[iii[ii,1],iii[ii,2],d]=(t(intstage$z))[iii[ii,1],iii[ii,2]]

#print(paste("iii[ii,1]=",iii[ii,1],"iii[ii,2]=",iii[ii,2],"intstage=",cerp0_intstage[
iii[ii,1],iii[ii,2],d]))
  }
}


#Replace values outside polygon
for (io5 in 1:length(indoutpolyib5)) {
  cerp0_intstage[grrocos[indoutpolyib5[io5],2],grrocos[indoutpolyib5[io5],1],]=

cerp0_intstage[grrocos[indinpolyib5[idclosest[io5]],2],grrocos[indinpolyib5[idclosest[
io5]],1],]

print(paste(io5,indoutpolyib5[io5],grrocos[indoutpolyib5[io5],2],grrocos[indoutpolyib5
[io5],1],idclosest[io5],grrocos[idclosest[io5],2],
        grrocos[idclosest[io5],1]))
}


cerp0=brick(cerp0_intstage,xmn=extn[1,1],xmx=extn[1,2],ymn=extn[2,1],ymx=extn[2,2],crs
="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")
cerp0_mean=calc(cerp0,mean)
cerp0_mean=mask(cerp0_mean,n,maskvalue=0,updatevalue=NA)
cerp0_mean=mask(cerp0_mean,ibm,maskvalue=0,updatevalue=NA)

#Read in topo (ft NGVD29) for IncRF run
incrf_ael=read.csv("incRF_ael.roco",header=FALSE)
#Read in daily_stg_minus_lsel.bin for incrf run
incrf_dsml = read.table("incRF_dsml.txt",sep="",header=TRUE)
names(incrf_dsml)[4:(dim(incrf_dsml)[2])]=simplify2array(lapply(strsplit(names(incrf_d
sml)[4:(dim(incrf_dsml)[2])],"[.]"),FUN=function(x)
as.numeric(x[2])*100+as.numeric(x[3])))
```

```
incrf_stage = incrf_dsml
x=vector(length=(dim(incrf_dsml)[2]-3),mode='numeric')
y=vector(length=(dim(incrf_dsml)[2]-3),mode='numeric')
#Add topo to daily_stg_minus_lsel.bin
for (i in 4:(dim(incrf_dsml)[2])) {
  cell_topo=incrf_ael[which(incrf_ael[,1]==as.numeric(names(incrf_dsml)[i])),2]
  print(paste("i=",i,"cell topo=",cell_topo))
  incrf_stage[,i]=incrf_dsml[,i]+cell_topo
  x[i-3]=xys[which(ROWCOL==as.numeric(names(incrf_dsml)[i])),1]
  y[i-3]=xys[which(ROWCOL==as.numeric(names(incrf_dsml)[i])),2]
}

plot(n)
points(x,y,col='red',cex=0.5)

incrf_intstage = array(data=-999,dim=c(length(y0),length(x0),dim(incrf_dsml)[1]))

#Interpolate daily stage values for incrf run
for (d in 1:(dim(incrf_stage)[1])) {
  print(paste("d=",d))
  intstage=interp(x,y,as.numeric(incrf_stage[d,4:(dim(incrf_stage)[2])]),x0,y0)
  nas=which(!(is.na(intstage$z)),arr.ind=TRUE)
  #points(x0[nas[,1]],y0[nas[,2]],cex=0.1)
  for (ii in 1:dim(iii)[1]) {
    incrf_intstage[iii[ii,1],iii[ii,2],d]=(t(intstage$z))[iii[ii,1],iii[ii,2]]

#print(paste("iii[ii,1]=",iii[ii,1],"iii[ii,2]=",iii[ii,2],"intstage=",incrf_intstage[
iii[ii,1],iii[ii,2],d]))
  }
}

#Replace values outside polygon
for (io5 in 1:length(indoutpolyib5)) {
  incrf_intstage[grrocos[indoutpolyib5[io5],2],grrocos[indoutpolyib5[io5],1],]=

incrf_intstage[grrocos[indinpolyib5[idclosest[io5]],2],grrocos[indinpolyib5[idclosest[
io5]],1],]

print(paste(io5,indoutpolyib5[io5],grrocos[indoutpolyib5[io5],2],grrocos[indoutpolyib5
[io5],1],idclosest[io5],grrocos[idclosest[io5],2],
        grrocos[idclosest[io5],1]))
}

incrf=brick(incrf_intstage,xmn=extn[1,1],xmx=extn[1,2],ymn=extn[2,1],ymx=extn[2,2],crs
="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")
incrf_mean=calc(incrf,mean)
incrf_mean=mask(incrf_mean,n,maskvalue=0,updatevalue=NA)
incrf_mean=mask(incrf_mean,ibm,maskvalue=0,updatevalue=NA)

#Create plots
minz=floor(2*min(min(getValues(cerp0_mean),na.rm=TRUE),min(getValues(incrf_mean),na.rm
=TRUE)))/2
maxz=ceiling(2*max(max(getValues(cerp0_mean),na.rm=TRUE),max(getValues(incrf_mean),na.
rm=TRUE)))/2
brks1=seq(minz,maxz,0.5)

png("CERP0_mean_stage_6505.png",width=720,height=720)
```

```
plot(cerp0_mean,main=c("CERP0 mean stage 1965-2005 (ft
NGVD29)",paste("mean=",round(cellStats(cerp0_mean,mean),2),sep="")),
     breaks=brks1,col=jet(length(brks1)))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

cellStats(cerp0_mean,mean) #6.515426

png("IncRF_mean_stage_6505.png",width=720,height=720)
plot(incrf_mean,main=c("10% Rainfall Increase mean stage 1965-2005 (ft
NGVD29)",paste("mean=",round(cellStats(incrf_mean,mean),2),sep="")),
     breaks=brks1,col=jet(length(brks1)))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

cellStats(incrf_mean,mean) #6.422174

diff_mean=cerp0_mean-incrf_mean

png("CERP0-IncRF_mean_stage_6505.png",width=720,height=720)
brks=seq(-2,2,0.25)
plot(diff_mean,main=c("CERP0 - 10% Inc. RF stage
(ft)",paste("mean=",round(cellStats(diff_mean,mean),2),sep="")),
     col=rev(brewer.rdbu(length(brks))),breaks=brks,
     legend.args=list(text='Head difference (ft)', side=4, font=1,
line=2.75,cex.lab=0.5))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

cellStats(diff_mean,mean) #0.09325231

#Get julian day averages
jd=(strptime(paste(cerp0_dsml$Mo,cerp0_dsml$Da,cerp0_dsml$Year,sep="/"),format="%m/%d/
%Y"))$yday+1
jd[jd==366]=365

cerp0_jdmean=stackApply(cerp0,indices=jd,fun=mean)
incrf_jdmean=stackApply(incrf,indices=jd,fun=mean)
```

```
#Read in datum conversion offset created from Corpscon6 based on
MODFLOW_grid_coords.csv
#Then elevation in ft NAVD88 = elevation in ft NGVD29 + datum_offset (negative)
datum_offset=read.csv("MODFLOW_cell_datum_shift_vertcon05.csv",header=TRUE)
datum_offsetarray=array(datum_offset[,2],dim=c(ncol(n),nrow(n)))
datum_offsetraster=raster(t(datum_offsetarray),xmn=extn[1,1],xmx=extn[1,2],ymn=extn[2,
1],ymx=extn[2,2],crs="+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs")

png("datum_offset_vertcon05.png",width=720,height=720)
plot(datum_offsetraster,col=rev(brewer.rdbu(20)),main="elev. (ft NAVD88) = elev. (ft
NGVD29) + offset")
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
dev.off()

#Convert data from ft NGVD29 to m NAVD88
cerp0_jdmean_mnavd88=overlay(cerp0_jdmean,datum_offsetraster,fun=function(r1,r2)
{return((r1+r2)/3.28)})
incrf_jdmean_mnavd88=overlay(incrf_jdmean,datum_offsetraster,fun=function(r1,r2)
{return((r1+r2)/3.28)})

#Reclassify values <-300 to -999 again
cerp0_jdmean_mnavd88=reclassify(cerp0_jdmean_mnavd88,cbind(-Inf,-300,-999))
incrf_jdmean_mnavd88=reclassify(incrf_jdmean_mnavd88,cbind(-Inf,-300,-999))

#Repeat values for Julian day every year for 1965-2005
cerp0_daily_mnavd88=subset(cerp0_jdmean_mnavd88,jd)
incrf_daily_mnavd88=subset(incrf_jdmean_mnavd88,jd)

#Save as netCDF and RData files
writeRaster(cerp0_daily_mnavd88,"CERP0_daily_stage_mNAVD88_6505_julrep.nc",format="CDF
",varname="stage",varunit="m",
            longname="Stage for CERP0 run (m
NAVD88)",xname="x",yname="y",zname="t",zunit=paste("days since",min(cerp0_dsml$Year)),
            NAflag=-999,overwrite=TRUE)

writeRaster(incrf_daily_mnavd88,"IncRF_daily_stage_mNAVD88_6505_julrep.nc",format="CDF
",varname="stage",varunit="m",
            longname="Stage for 10% Inc. RF run (m
NAVD88)",xname="x",yname="y",zname="t",zunit=paste("days since",min(incrf_dsml$Year)),
            NAflag=-999,overwrite=TRUE)

save(cerp0_daily_mnavd88,file="CERP0_daily_stage_mNAVD88_6505_julrep.RData")
save(incrf_daily_mnavd88,file="IncRF_daily_stage_mNAVD88_6505_julrep.RData")

#Repeat values for Julian day every year for 1996-2010
dates2=as.POSIXlt(seq(as.Date("1996/1/1"), as.Date("2010/12/31"),"days"))
jd2=dates2$yday + 1
jd2[jd2==366]=365

cerp0_daily_mnavd88_2=subset(cerp0_jdmean_mnavd88,jd2)
incrf_daily_mnavd88_2=subset(incrf_jdmean_mnavd88,jd2)

#Save as netCDF and RData files
```

```
writeRaster(cerp0_daily_mnavd88_2,"CERP0_daily_stage_mNAVD88_9610_julrep.nc",format="C
DF",varname="stage",varunit="m",
            longname="Stage for CERP0 run (m
NAVD88)",xname="x",yname="y",zname="t",zunit=paste("days
since",(min(dates2$year)+1900)),
            NAflag=-999,overwrite=TRUE)

writeRaster(incrf_daily_mnavd88_2,"IncRF_daily_stage_mNAVD88_9610_julrep.nc",format="C
DF",varname="stage",varunit="m",
            longname="Stage for 10% Inc. RF run (m
NAVD88)",xname="x",yname="y",zname="t",zunit=paste("days
since",(min(dates2$year)+1900)),
            NAflag=-999,overwrite=TRUE)

save(cerp0_daily_mnavd88_2,file="CERP0_daily_stage_mNAVD88_9610_julrep.RData")
save(incrf_daily_mnavd88_2,file="IncRF_daily_stage_mNAVD88_9610_julrep.RData")


#Convert EDEN mean from m NAVD88 to ft NGVD29 and plot
eden_mean_ftngvd29=overlay(eden_mean,datum_offsetraster,fun=function(r1,r2)
{return(3.28*r1-r2)})
png("EDEN_mean_stage_9610.png",width=720,height=720)
plot(eden_mean_ftngvd29,main=c("EDEN mean stage 1996-2010 (ft
NGVD29)",paste("mean=",round(cellStats(eden_mean_ftngvd29,mean),2),sep="")),
     breaks=brks1,col=jet(length(brks1)))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

png("CERP0-EDEN_mean_stage.png",width=720,height=720)
diff_meance=cerp0_mean-eden_mean_ftngvd29
brks=seq(-2,2,0.25)
plot(diff_meance,main=c("CERP0 - EDEN stage
(ft)",paste("mean=",round(cellStats(diff_meance,mean),2),sep="")),
     col=rev(brewer.rdbu(length(brks))),breaks=brks,
     legend.args=list(text='Head difference (ft)', side=4, font=1,
line=2.75,cex.lab=0.5))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

png("IncRF-EDEN_mean_stage.png",width=720,height=720)
diff_meanincrf=incrf_mean-eden_mean_ftngvd29
brks=seq(-2.25,2.25,0.25)
plot(diff_meanincrf,main=c("10% Inc. RF - EDEN stage
(ft)",paste("mean=",round(cellStats(diff_meanincrf,mean),2),sep="")),
     col=rev(brewer.rdbu(length(brks))),breaks=brks,
```

```
     legend.args=list(text='Head difference (ft)', side=4, font=1,
line=2.75,cex.lab=0.5))
lines(basemap)
lines(hydrog)
points(saltstr,pch=21,bg='red',col='black')
points(drstr,pch=22,bg='blue',col='black')
points(x,y,pch=3,col='black',cex=0.5)
legend('topright',legend=c('Salinity ctrl. str','Drainage str.','SFWMM grid
points'),pch=c(21,22,3),col='black',pt.bg=c('red','blue',NA))
dev.off()

#Plot some timeseries at some gages in S. Glades together with VA Key stages
vakey=read.csv("Z:/miriza/Work/FIU/FL_Building_Code/Data/Water_levels/VAKey_comparison
s.csv",stringsAsFactors=FALSE)
vakey[,1]=as.Date(vakey[,1],"%m/%d/%Y") #These are in ft NAVD88
ro=seq(160,171)
co=seq(5,12)
for (r in ro) {
  for (c in co) {
    cerp0_ts=cerp0_daily_mnavd88_2[r,c]
    eden_ts=eden_stage[r,c]

rng=c(min(cbind(cerp0_ts,eden_ts,vakey[,2:4]*0.3048)),max(cbind(cerp0_ts,eden_ts,vakey
[,2:4]*0.3048)))
    png(paste("Stage_timeseries_at_R",r,"C",c,".png",sep=""), width=1200,height=720)
    plot(vakey[,1],cerp0_ts[1,],ylim=rng,type='l',xlab="Date",ylab="Water levels (m
NAVD88)",
         main=paste("Water levels at R",r,"C",c,sep=""))
    lines(vakey[,1],vakey[,2]*0.3048,col='blue')
       lines(vakey[,1],vakey[,4]*0.3048,col='grey')
    lines(vakey[,1],eden_ts[1,],col='red')
    lines(vakey[,1],vakey[,3]*0.3048,col='green')

legend('topright',legend=c('CERP0',names(vakey)[2],names(vakey)[4],'Eden',names(vakey)
[3]),
              col=c('black','blue','grey','red','green'),lty=1)
    dev.off()
  }
}

}
```

Appendix D. Description of boundary condition file (ibound)

The Miami-Dade MODFLOW model pre-processing utilities use a file called umd_ibound.ref, which defines nine (9) different zones of boundary conditions based on an id assigned to each cell with values from 0 to 8 (Figure 85). In particular, the Python pre-processing script UMD_Scenario_BND.py, which defines drain and general head boundary (GHB) boundary conditions for the model, handles each ibound zone differently as described below. As described in Hughes and White (2016) "coastal cells were defined to be coastal GHB or DRN cells on a daily basis using the average daily stage at Virginia Key. GHBs were specified for all coastal boundary cells having a surface elevation less than the stage at Virginia Key to allow for bidirectional water exchange based on the difference between the Biscayne aquifer and overlying coastal water bodies. Conversely, DRNs were specified for all coastal boundary cells having a surface elevation greater than or equal to the surface-water stage at Virginia Key to allow groundwater discharge at the surface in coastal areas." GHB and drain boundary conditions are usually only defined in the top layer of the model (layer 1) with the exception of cells on the northern and western boundaries for which GHBs are defined in all three model layers.

The handling of cells in each ibound zone often depends on another variable called isource, which defines whether the water on the cell is freshwater or seawater. If isource = 0, sources and sinks have the same fluid density as the active zone at the top of the aquifer. Zones with isource of 2, such as the cells in the Turkey Point cooling canals, have sources and sinks with density equal to that of seawater. Zones with isource of -2 have sources with the same fluid density as zone 2 (seawater), while sinks have the same fluid density as the zone at the top of the aquifer. This option is used when simulating the ocean bottom where infiltrating water is salt while exfiltrating water is of the same type as water at the top of the aquifer." (Bakker et al., 2013). In previous versions of the Miami-Dade MODFLOW model, zones with an isource of -2 are defined based on the mean sea level during the last year of the simulation period (Figure 48), and this is approach we follow.

Ibound zone definitions

Zone with ibound = 0 (red):

These are inactive model cells. No boundary conditions are defined in this zone.

Zone with ibound = 1 (blue):

These are the main computationally active model cells. No boundary conditions defined here.

Zone with ibound = 2 (green):

These are current ocean and coastal cells.

Cells in this zone are handled differently depending on the value of isource. If isource is equal to 0, then no boundary conditions are created for these cells and the cells are computationally active. Otherwise (i.e. when isource = -2, when the cell is under water during for the MSL of the last year of simulation), GHBs or drains are defined for the cells based on equivalent freshwater heads. The freshwater head is computed based on a saltwater head = max (cell topography, daily stage at Virginia Key tidal station).

Zone with ibound = 3 (purple):

These are cells immediately east of the Turkey Point cooling canals. GHBs or drains are defined for these cells based on equivalent freshwater heads. The freshwater head is computed based on a saltwater head = max (cell topography, daily stage at Virginia Key tidal station – 0.11 m), consistent with Hughes and White (2016).

Zone with ibound = 4 (orange):

These are cells on the Turkey Point cooling canals. GHBs or drains are defined for these cells based on equivalent freshwater heads. The freshwater head is computed based on a saltwater head = max (cell topography, daily stage at Virginia Key tidal station + 0.19 m), consistent with Hughes and White (2016).

Zone with ibound = 5 (yellow):

These are cells in the Water Conservation Areas (WCA) and Everglades National Park (ENP). GHBs or drains are defined for these cells based on a head computed based on the max (historical stage from EDEN network, cell topography, daily stage at Virginia Key station). The heads are converted to equivalent freshwater heads only when isource equals -2 for the cell (i.e. when the cell is under water during the MSL of the last year of simulation).

Zone with ibound = 6 (brown):

These are cells in the Southern Glades. GHBs or drains are defined for these cells based on a head computed based on the max (interpolated stage, cell topography, daily stage at Virginia Key tidal station). The interpolated stage for these cells is based on interpolation of stage data for zones 2, 3, 4, 5 and 8. The computed head is converted to freshwater heads only when isource equals -2 for the cell (i.e. when the cell is under water during the MSL of the last year of simulation).

Zone with ibound = 7 (pink):

These are cells north of the C-111 canal.

Cells in this zone are handled differently depending on the value of isource. If isource equals 0, then no boundary conditions are created for these cells and the cells are computationally active. Otherwise (i.e. when isource = -2, when the cell is under water during for the MSL of the last year of simulation), GHBs or drains are defined for the cells based on equivalent freshwater heads. The freshwater head is computed based on a saltwater head = max (interpolated stage, cell topography, daily stage at Virginia Key tidal station). The interpolated stage for these cells is based on interpolation of stage data for zones 2, 3, 4, 5 and 8.

Zone with ibound = 8 (gray):

GHBs are defined for the three model layers based on interpolated daily historical stages at surface water and groundwater sites: S30_H (SW), G-2034 (GW), G-1225 (GW), G-1473 (GW), and Virginia Key (SW).

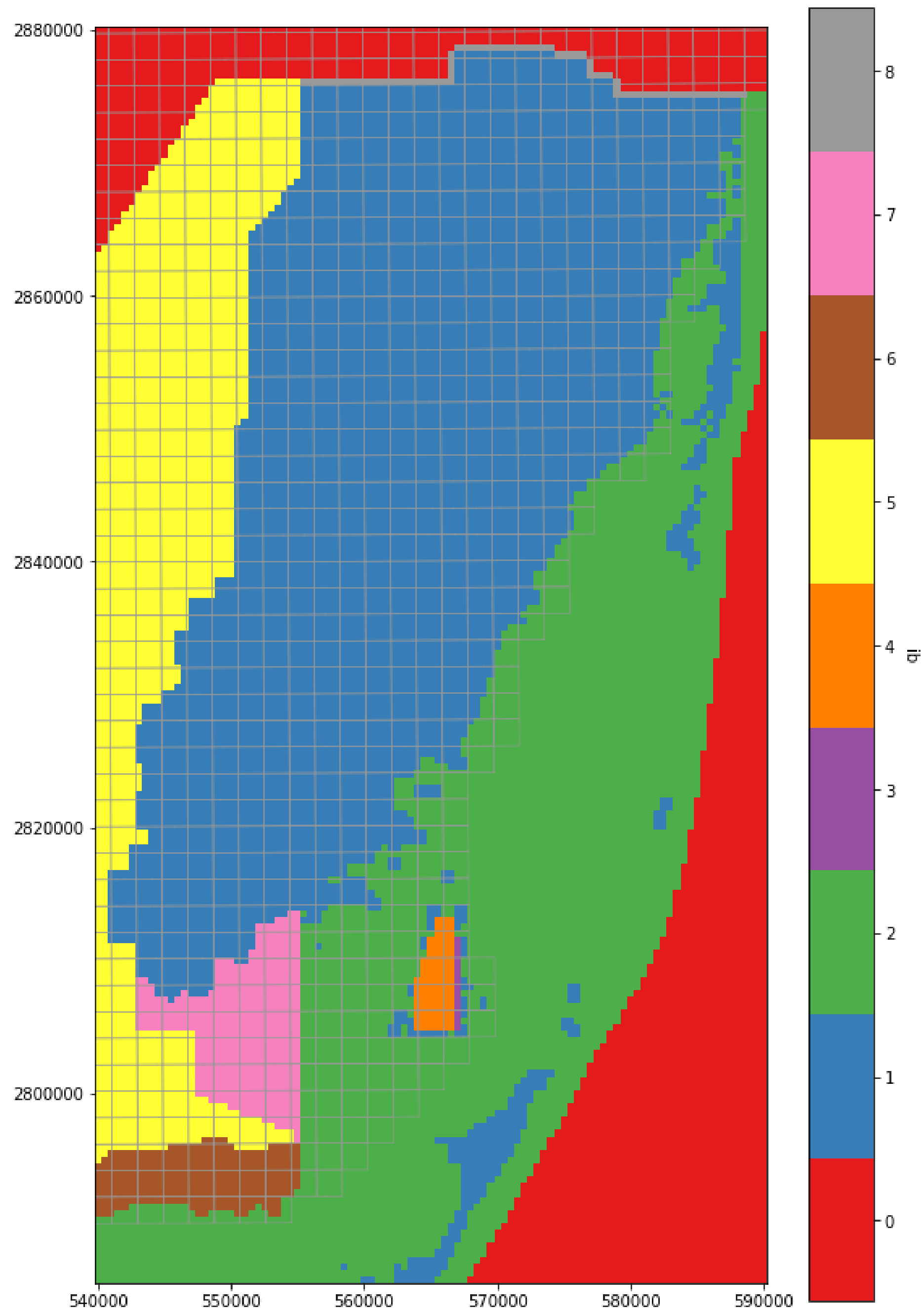


Figure 85. Boundary condition ids for Miami-Dade MODFLOW cells

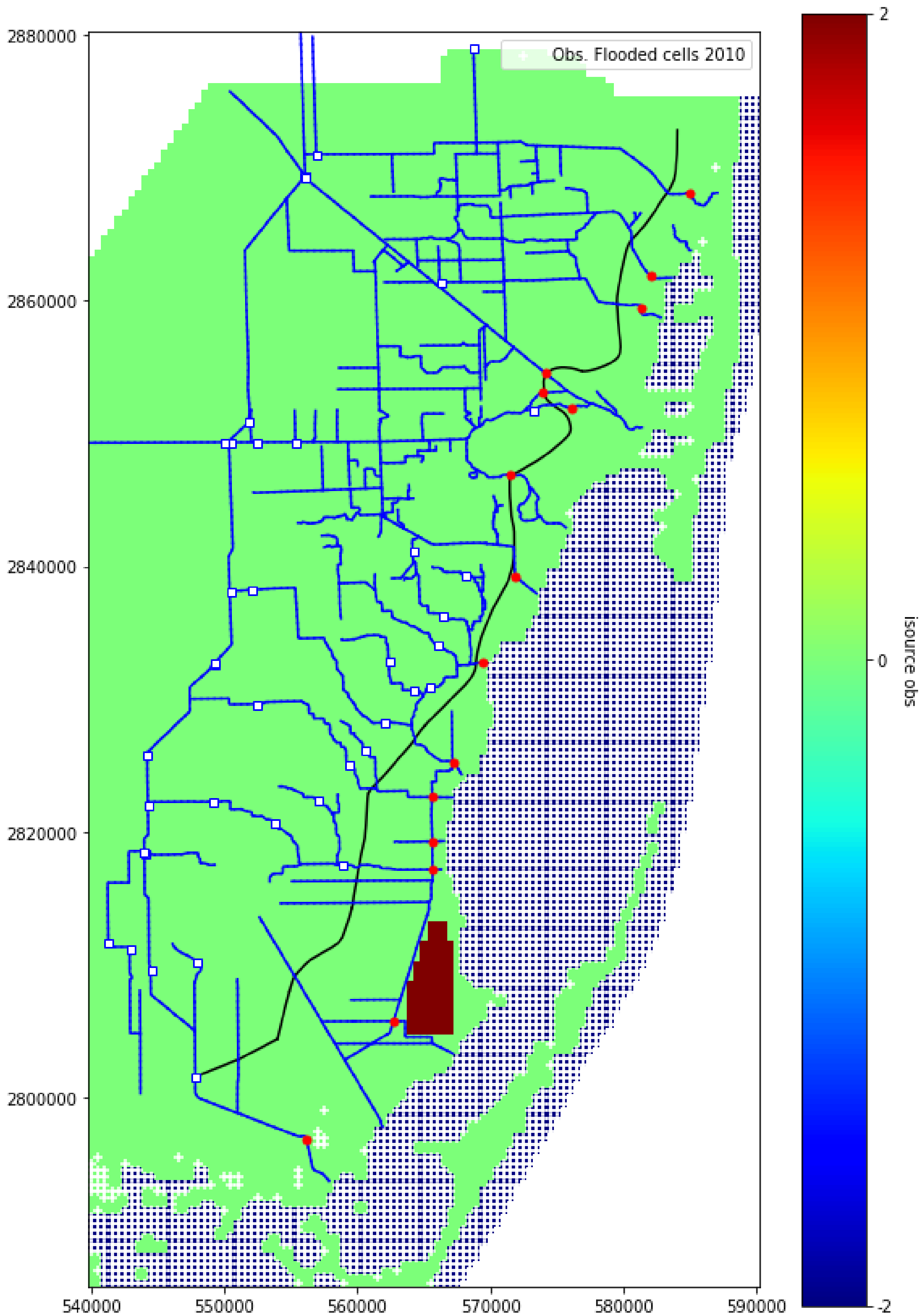


Figure 86. Freshwater/saltwater source (isource) for the 1996-2010 calibration run. Cells marked with white '+' are below the 2010 historical mean sea level at Virginia Key and their isource value is (for the most part) equal to -2 (blue).

Appendix E. MODFLOW input file modifications for scenario simulations

The following MODFLOW package input files are specified in the MODFLOW name file (.NAM) and were modified to simulate the scenarios as part of this project depending on how each model component is modeled (e.g. sea level rise scenario, land use, rainfall/RET).

<u>.BAS</u> – MODFLOW basic file = f (SLR scenario)

- ibound*.ref files – Defines active and inactive model areas. No changes in our simulations (same as used in calibration)
- ihead*.ref files = f (SLR scenario) - Initial head files are obtained from a long-term simulation ending in 2054 with tidal water levels along the same SLR curve as in the scenario being simulated.

<u>.DIS</u> – MODFLOW model discretization file – No changes in our simulations (same as used in calibration)

<u>.LPF</u> – MODFLOW layer property flow package file = f (LU/quarries) – All scenarios use the same .LPF file, which is based on 2018 permitted quarry coverage.

<u>.OC</u> – MODFLOW output control file – No changes in our simulations (same as used in calibration)

<u>.NWT</u> – MODFLOW Newton solver package file – No changes in our simulations (same as used in calibration)

<u>.GFB</u> – MODFLOW general flux boundary package file = f (rainfall, RET, LU). Most of these files are generated by the Python script MakeScenarioMET.py, which has been modified to handle different land use and rainfall assumptions.

- rech*.ref files = f (rainfall); 0.0254 multiplier (conversion from inches to m) used with LOCA bias-corrected future rainfall; 0.0257 (0.0254 * 1.05) used with NEXRAD historical rainfall where 1.05 is a bias-correction factor for NEXRAD.
- umd_nexrad_mult*.ref file = f (LU) – Contains fractions of cells that are pervious or impervious and not directly connected to the drainage network (1 – DCIA). All scenarios use the same file, which is based on 2030 land use.
- scenario_2010_rec*.bin files = f (rainfall, RET, LU) – Recreational irrigation computed based on 2030 land use (for low and medium density urban cells only) and rainfall/RET for the particular scenario run. They have '2010' as part of their name since the irrigation has been scaled to 2010 annual recreational irrigation withdrawals in the county. A 0.0254 (inches to m conversion) multiplier in the input file remains the same regardless of rainfall dataset used. Python script must be modified to use appropriate multipliers for rainfall and RET.
- scenario_2010_ag*.bin files = f (rainfall, RET, LU) – Agricultural irrigation computed based on 2030 land use and rainfall for the particular scenario run. They have '2010' as part of their name since the irrigation has been scaled to 2010 annual agricultural irrigation withdrawals in the county. A 0.0254 (inches to m conversion) multiplier in the input file remains the same regardless of rainfall dataset used. Python script must be modified to use appropriate multipliers for rainfall and RET.
- septic_return_2010.ref – No changes in our simulations (same as used in scenarios by the USGS).

.ETS – MODFLOW ET segments package file = f (RET, LU). Most of these files are generated by the Python script MakeScenarioMET.py, which has been modified to handle different land use and rainfall assumptions.

- ret*.ref files = f (RET); 0.00105 multiplier (0.001 conversion of mm to m * 1.05 open water multiplier) used when assuming RET stays the same as historically; 0.0011 multiplier (0.001 conversion from mm to m * 1.05 open water multiplier * 1.05 assumed future increase in RET) used when assuming that future RET will increase by 5% due to increases in temperature.
- umd_ets_petm_*_*2030.ref = f (LU) – All scenarios use the same. It is based on the 2030 land use. This file defines the proportion of the maximum ET rate for a given ET segment intersection.

.GHB – MODFLOW general head boundary condition file = f (SLR scenario, western boundary assumption). Most of these files are generated by the Python script UMD_Scenario_BND.py, which has been modified to handle different boundary condition assumptions.

- ghb*.bin files = f (SLR scenario, western boundary assumption)

.DRN – MODFLOW drain boundary condition file = f (SLR scenario, western boundary assumption). Most of these files are generated by the Python script UMD_Scenario_BND.py, which has been modified to handle different boundary condition assumptions.

- drn*.bin files = f (SLR scenario, western boundary assumption)

.WEL – MODFLOW well boundary condition file = f (wellfield pumpage assumption). Most scenario runs use the same wellfield pumpage file as Scenario run 1 for 2030-2040 from the USGS.

.SWR – MODFLOW surface water routing package file = f (SLR scenario, rainfall, RET, LU). Most datasets stay the same as in the model calibration except for the following:

- rech*.ref files = f (rainfall); 0.0254 multiplier (conversion from inches to m) used with LOCA bias-corrected future rainfall; 0.0257 (0.0254 * 1.05) used with NEXRAD historical rainfall where 1.05 is a bias-correction factor for NEXRAD.
- ret*.ref files = f (RET); 0.00105 multiplier (0.001 conversion of mm to m * 1.05 open water multiplier) used when assuming RET stays the same as historically; 0.0011 multiplier (0.001 conversion from mm to m * 1.05 open water multiplier * 1.05 assumed future increase in RET) used when assuming that future RET will increase by 5% due to increases in temperature.
- umd_istage*.ref file = f (SLR scenario) – Dataset 14a file with initial conditions for the surface water network obtained from a long-term simulation ending in 2054 with tidal water levels along the same SLR curve as in the scenario being simulated.
- swr_dcia_dataset*.ref file = f (LU, rainfall) – Defined as an external file in the .NAM file instead of in the .SWR file. It has references to a files with the fraction of directly-connected impervious area (DCIA) on each cell that is routed as direct runoff to each SWR reach based on a mapping of cells to closest SWR reach. It references the rech*.ref files which are a function of rainfall (f (rainfall)) and whose multiplier depends on rainfall dataset used as described above.

- VAKey*SWR1.ref file = f (SLR scenario) – Defined as an external file in the .NAM file instead of in the. SWR file. It has the tidal boundary condition applied downstream of salinity control structures (in ft NAVD88 with a 0.3048 conversion factor from ft to m specified at the top of the file).

.SWI – MODFLOW saltwater intrusion package file = f (SLR scenario, LU)

- umd_izeta*.ref file = f (SLR scenario) – Initial (vertical) location of the saltwater/freshwater interface obtained from a long-term simulation ending in 2054 with tidal water levels along the same SLR curve as in the scenario being simulated.
- umd_isource*.ref file = f (SLR scenario) – Defines density of water (saltwater vs. freshwater) for sinks and sources on every model grid cell. Consistent with the USGS scenarios, it is defined based on the mean sea level for the last year of simulation in our scenarios (2069).
- umd_Sy_L1_*.ref file = f (LU) – Defines the aquifer porosity which is based on 2018 permitted quarry coverage.

Table 11. Detailed descriptions of changes to MODFLOW input files for the future scenario and sensitivity runs.

Run short-name	**LOW SLR**	**HIGH SLR**	**HIGH SLR + NO PUMPAGE**	**LOW SLR + HIST RAIN/RET**	**HIGH SLR + HIST RAIN/RET**
Run description	Low SLR scenario (IPCC median)	High SLR scenario (USACE High)	High SLR scenario with no pumpage	Low SLR scenario with historical rainfall	High SLR scenario with historical rainfall
Rainfall					
1996-2010 NEXRAD rainfall with 1.05 correction factor				X	X
Bias-corrected LOCA rainfall for scenario pr_MRI-CGCM3_r1i1p1_rcp85 in 2055-2069 (no correction factor applied since bias-corrected to SFWMM rainfall dataset for 1991-2005)	X	X	X		
Reference evapotranspiration					
1996-2010 RET from the USGS				X	X
1996-2010 RET from the USGS with 1.05 adjustment factor due to future temperature increase	X	X	X		
Land use					
2030 land use and DCIA	X	X	X	X	X
2018 permitted quarry lakes	X	X	X	X	X
Calibrated crop coefficients	X	X	X	X	X
Groundwater properties					

Run short-name	LOW SLR	HIGH SLR	HIGH SLR + NO PUMPAGE	LOW SLR + HIST RAIN/RET	HIGH SLR + HIST RAIN/RET
Updated to reflect additional 2018 permitted quarry locations (update both lpf file and swi file)	X	X	X	X	X
Recharge					
Ag. Irrigation, rec. irrigation, rainfall and ET based on corresponding rainfall, LU and DCIA (umd_nexrad_mult) with adjustment factors derived from 2010 land use and water use data	X	X	X	X	X
Septic return for 2010	X	X	X	X	X
SWR package rainfall and RET					
Updated to reflect the same datasets used in recharge calculations	X	X	X	X	X
PWS pumpage					
No pumpage			X		
Future Pumpage as in USGS Scenario 1 for the period 2030-2040	X	X		X	X
Western boundary condition					
Water levels in WCA3 and Eastern ENP from CERP0 SFWMM run (average for Julian day at each cell is repeated every year)	X	X	X	X	X
Tidal boundary condition					
Predicted sea levels for 2055-2069 + SLR from IPCC AR5 RCP8.5 median curve	X			X	
Predicted sea levels for 2055-2069 + SLR from USACE High curve		X	X		X
All boundary conditions					
GHB and DRN boundary conditions based on corresponding western BC and sea levels and ibound array	X	X	X	X	X
Definition of source of water (fresh or saline) for every cell (isource array) based on 2069 MSL for corresponding SLR curve	X	X	X	X	X
Initial location of saltwater interface (izeta) based on long-term run up to 1954 along low SLR scenario	X			X	
Initial location of saltwater interface (izeta) based on long-term run up to 1954 along high SLR scenario		X	X		X

Run short-name	LOW SLR	HIGH SLR	HIGH SLR + NO PUMPAGE	LOW SLR + HIST RAIN/RET	HIGH SLR + HIST RAIN/RET
Structures and operations					
Same as 1996-2010 effective gate openings	X	X	X	X	X

Appendix F. Relevant figures from USGS Miami-Dade MODFLOW model documentation

The following figures were obtained from the USGS Miami-Dade MODFLOW model by Hughes and White (2016), with permission from the authors.

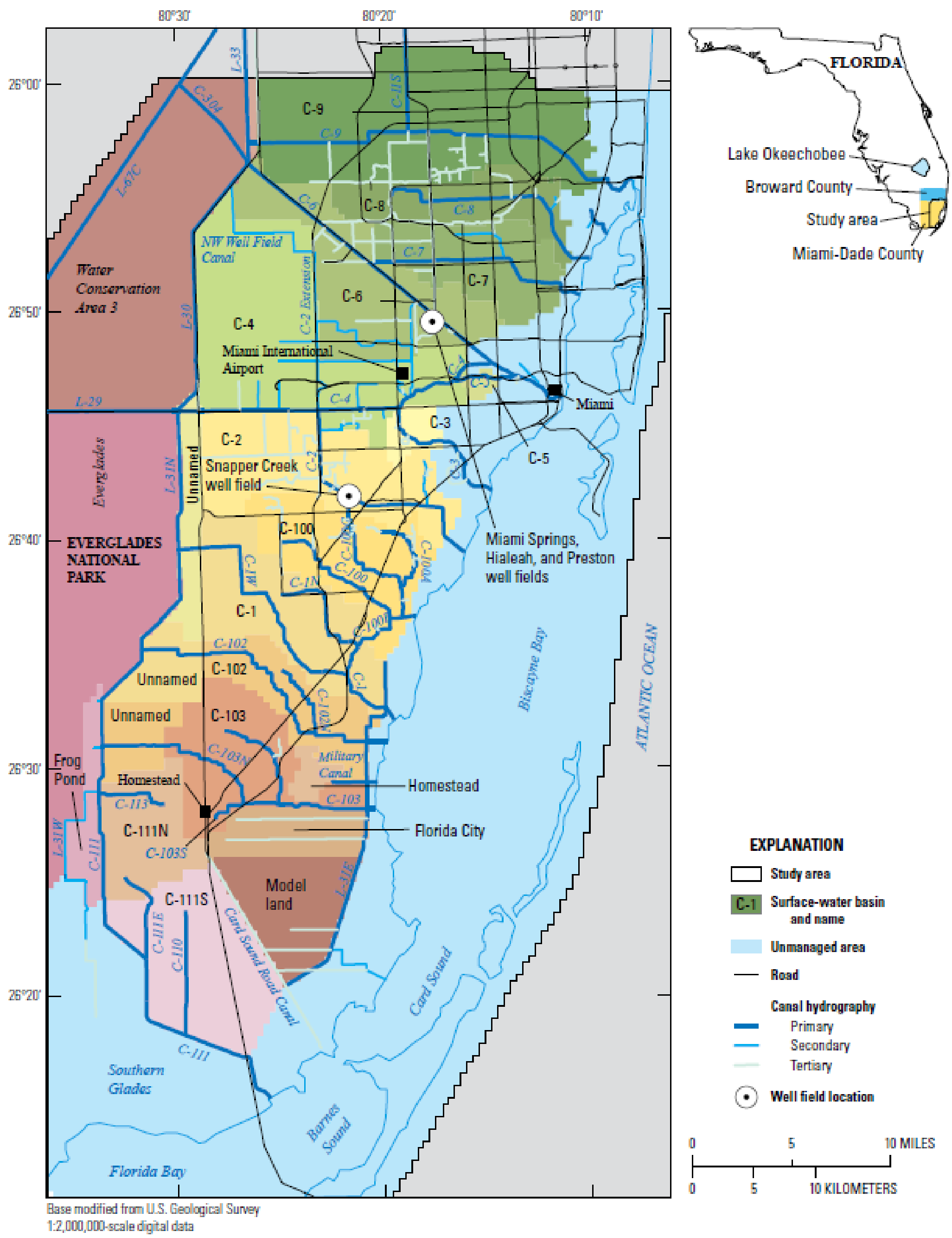


Figure 87. Study area in Southeastern Florida (Figure 1 of Hughes and White, 2016).

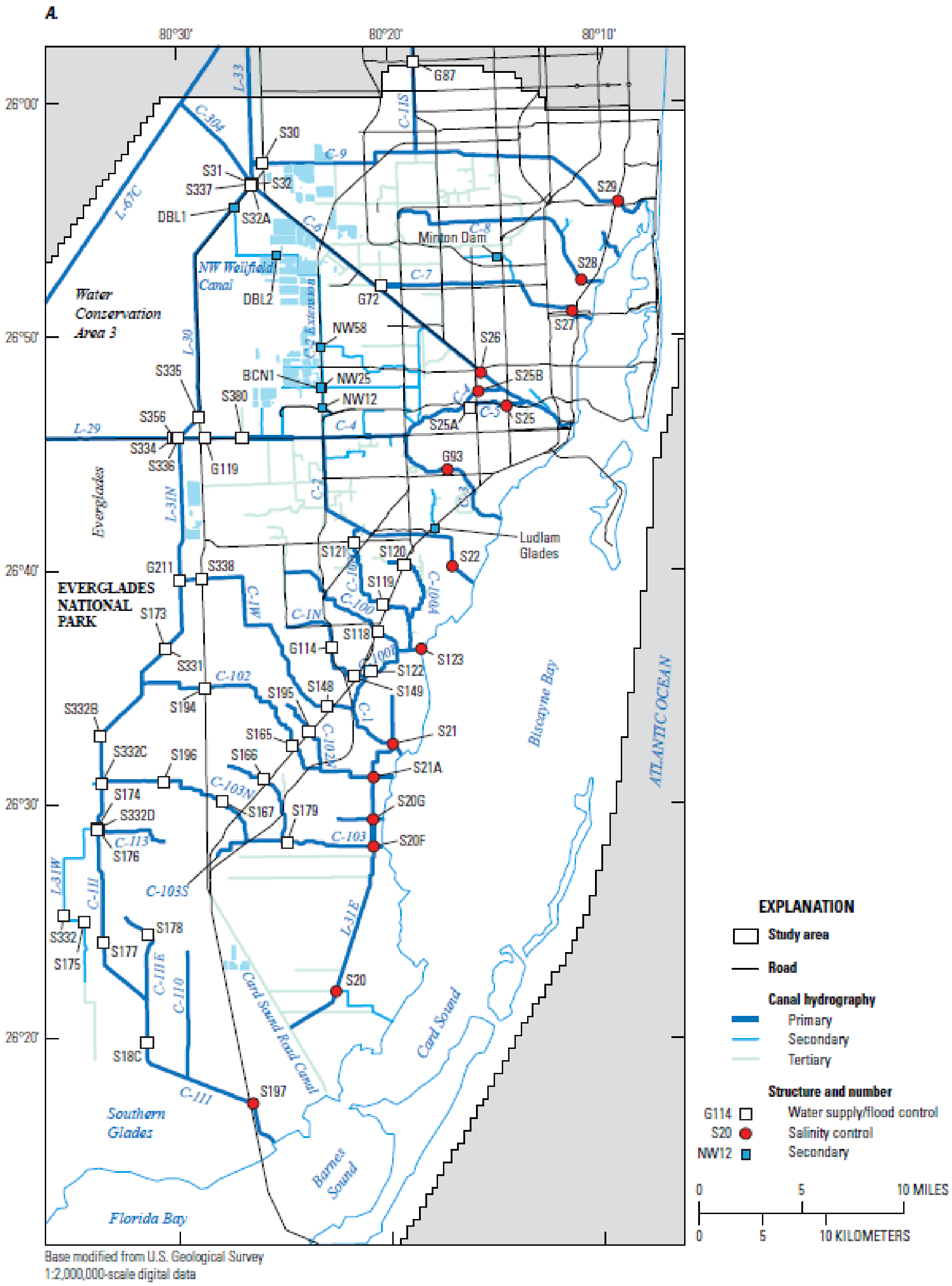


Figure 88. Location of primary and secondary surface-water control structures (Figure 9a of Hughes and White, 2016).

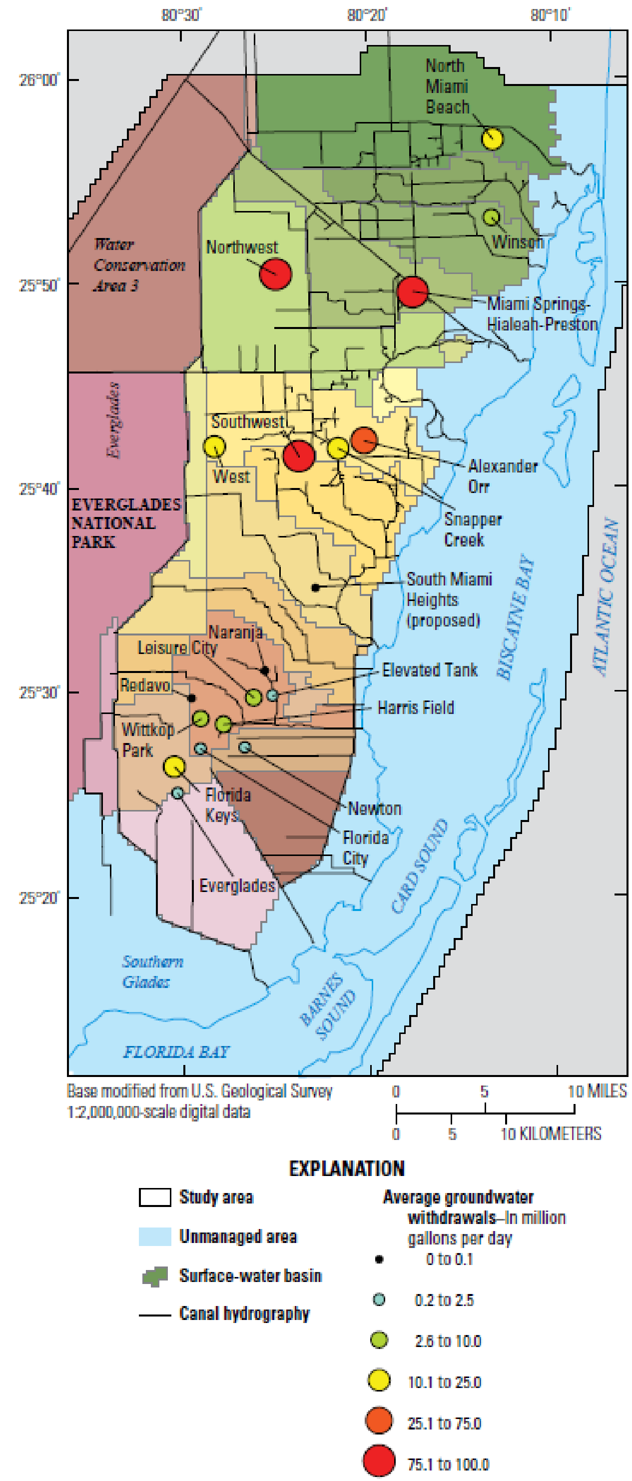


Figure 89. Average municipal water use in the study area for the period 1996-2010 (Figure 18 of Hughes and White, 2016).

Appendix G. Simulated canal stages for the High SLR + no pumpage sensitivity run

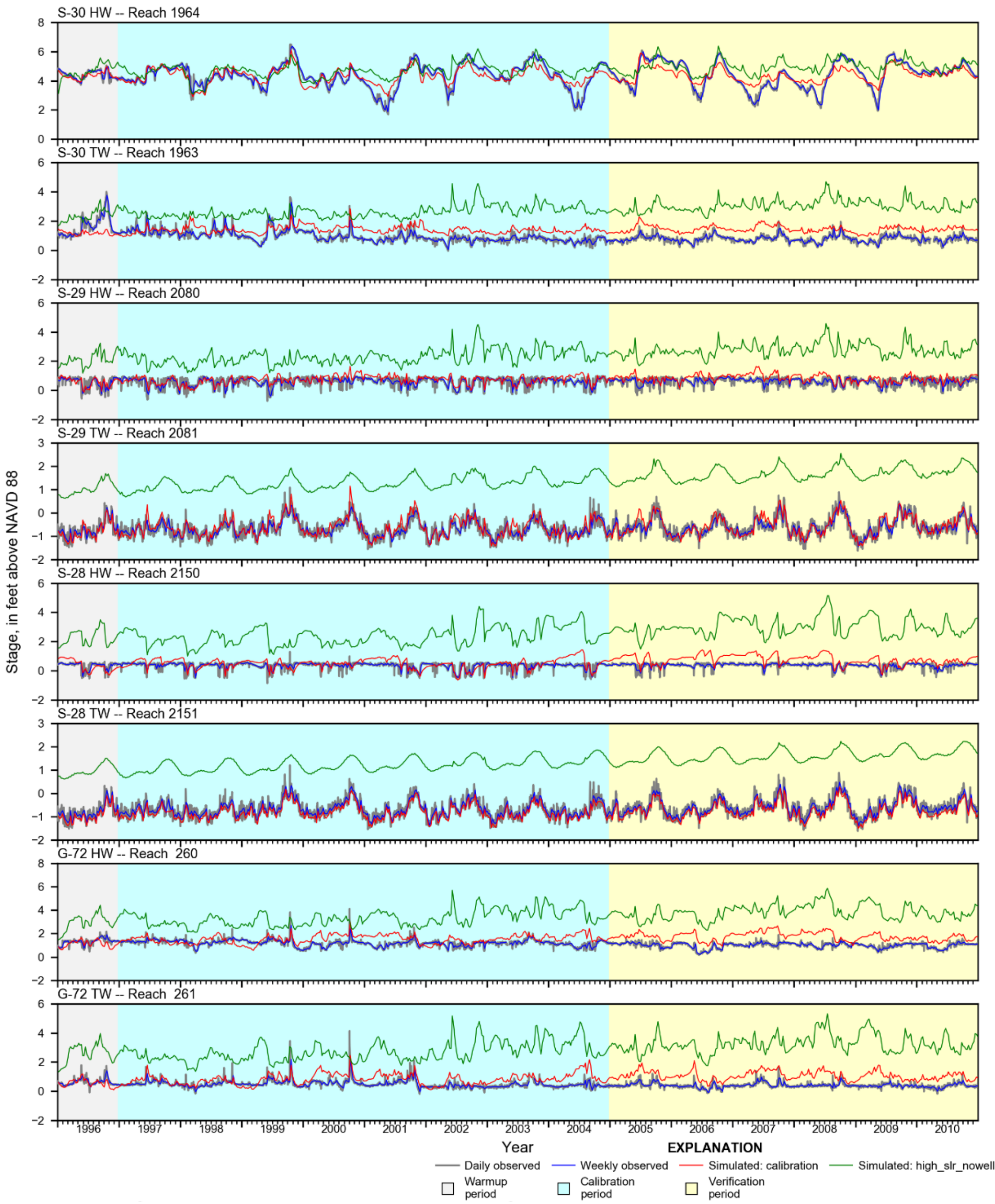


Figure 90. Observed and simulated stages at surface-water gages in the study area for the calibration run (red) and the High SLR + no pumpage sensitivity run.

Appendix II. Updating Existing Rainfall Maps

This appendix describes the work performed to update existing rainfall maps in the Florida Building Code. For this purpose, we have evaluated the most recent rainfall data and studies available from South Florida Water Management District (SFWMD), National Oceanographic and Atmospheric Administration (NOAA) and other agencies (i.e. Miami-Dade County) to develop 100-year, rainfall for durations of 1 hour up to 3 days. Based on this analysis, spatial maps of rainfall depth were produced.

For this task, we have assembled a dataset of rainfall data up to year 2018 and developed a time series of annual extremes for various durations of 1 hour up to 7 days. We used the extreme value analysis methods using the statistical software packages in R (open source statistical software package) to determine the design rainfall magnitudes for 100-year return period for various durations. The resulting values were mapped across the Miami-Dade County using appropriate spatial interpolation/smoothing methods to produce the rainfall loading maps using GIS tools. For further validation of the maps, they will be compared with the published data available from SFWMD and NOAA.

Finally, the LOCA downscaled climate data product by the University of California-San Diego was evaluated to determine potential future changes in rainfall extremes over Miami-Dade County using quantile mapping techniques for bias-correction.

Historical rainfall data sources

Various sources of historical rainfall data were evaluated for inclusion into a dataset for depth-duration-frequency (DDF) analysis. These will be described in the next sections. In the end, only rainfall data from NOAA Atlas 14 Volume 9 and from the South Florida Water Management District's DBHydro database were used in this task (Figure 91). The chosen rainfall stations were based on balancing the desire of using the most recent annual maxima rainfall data available and the desire of including sufficient years in the DDF analysis for reliable fitting of parameters.

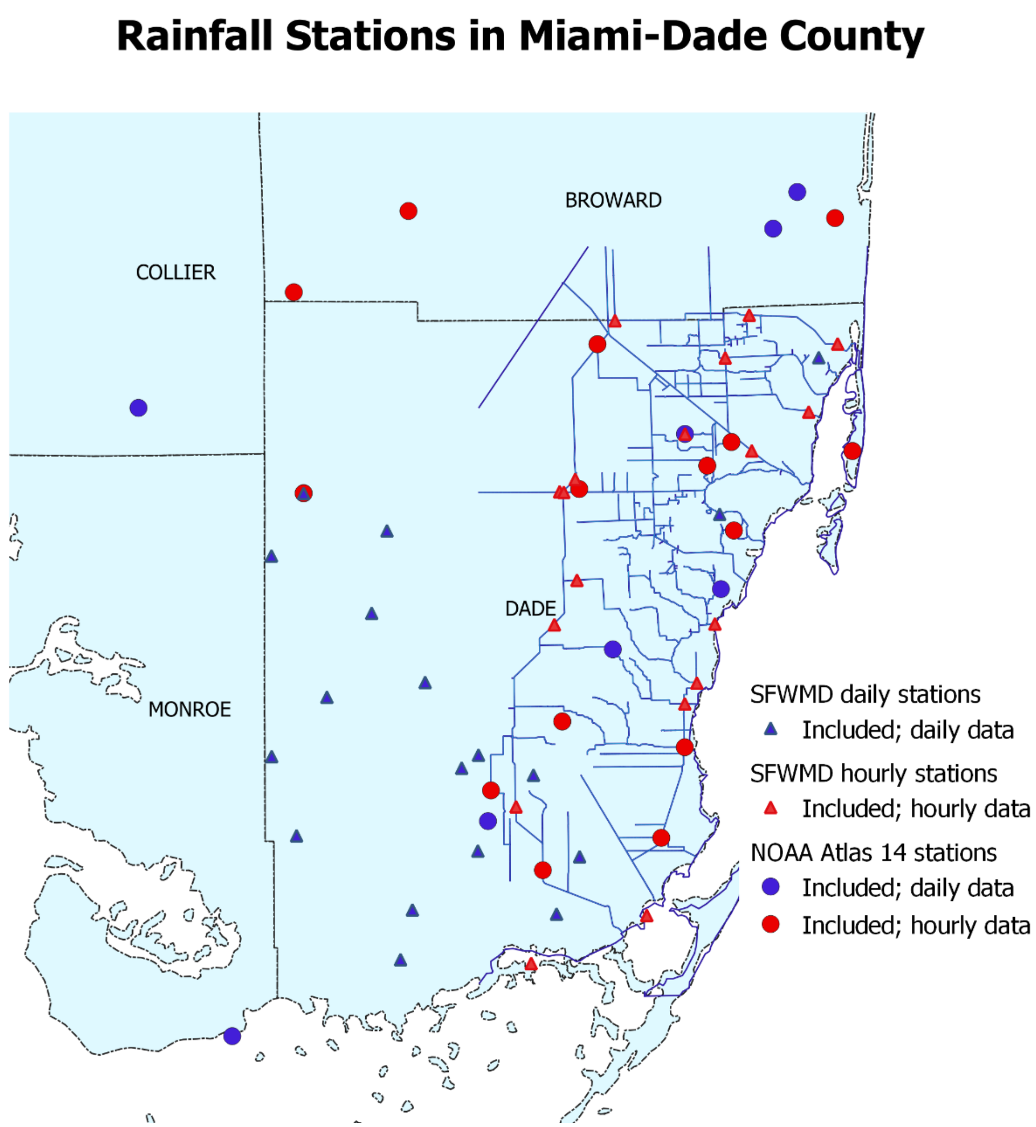


Figure 91. Rainfall stations in Miami-Dade County and vicinity

NOAA Atlas 14 Volume 9

NOAA Atlas 14 contains estimates of precipitation depth-duration-frequency (DDF) curves along with associated 90% confidence intervals for the United States and territories at both weather stations and as a gridded product with 30 arc-second resolution (approx. 0.5 mi). Supplementary information available as part of this product includes the annual maximum series (AMS) data used in developing the DDF curves, analysis of the AMS seasonality and trends, and the temporal distribution of heavy precipitation. The results are published through the Precipitation Frequency Data Server (PFDS) at http://hdsc.nws.noaa.gov/hdsc/pfds. The AMS data is generally available up to the years 2011-2012, depending on the station.

Volume 9 of NOAA Atlas 14 covers the Southeastern states including Florida. The methodology used in developing the DDF/IDF curves is documented at

http://www.nws.noaa.gov/oh/hdsc/PF_documents/Atlas14_Volume9.pdf.

AMS series were downloaded from PFDS for 23 weather stations in the vicinity of Miami-Dade County (Figure 91, Figure 92, and Table 12). Sources of weather station data for the Miami-Dade County used in Atlas 14 include: NOAA-National Climatic Data Center NCDC (prefix 08) and SFWMD-DBHydro database (prefix 90) (see Table 4.2.1 in NOAA, 2013). Periods of records at these stations can go back as far as 1840 and end in 2011-2012 (See Appendix 1, and Appendix A.1 of NOAA, 2013). A total of 15 NOAA Atlas 14 stations in the county have hourly AMS data available (Table 12).

Precipitation is recorded at clock-based (*constrained*) intervals of 15-min, 1-hour or 1-day (these are called "base duration") at these weather stations. Data at the base duration were accumulated over the durations of interest (1-hour, 2-hour, 3-hour, 6-hour, 12-hour, 1-day, 2-day, 3-day, 4-day, 7-day, 10-day, 20-day, 30-day, 45-day and 60-day) to develop *constrained* AMS for each duration. Due to the use of clock-based precipitation measurements, the *constrained* AMS series underestimates actual maxima (which in theory should be based on moving windows of a certain duration). In order to convert the *constrained* AMS series to *unconstrained* AMS values to be used in DDF development, NOAA Atlas 14 estimated correction factors which were applied to durations of 1-6 hours and 1-7 days (Table 13 and Table 14; from Tables 4.5.1 and 4.5.2 in NOAA, 2013). To avoid any confusion, all the AMS data is provided by NOAA as *constrained* values for all durations regardless of the base monitoring timestep of the original data (S. Pavlovic, NOAA, pers. comm. 9/16/2016).

Code was developed in the R programming language in order to extract AMS data for the 23 weather stations of interest in the NOAA Atlas 14 dataset, apply correction factors, and extract the most recent 30 years of available AMS data at each station. A separate dataset was developed with AMS data for the 22 stations (14 hourly, 8 daily) with sufficient years of AMS data up to the year 2005 (marked with '*' in Table 12). One station had less than 20 years of AMS data available for the period ending in 2005, so it was eliminated of the second dataset. This second dataset was used in bias-correction of projected precipitation extremes based on the LOCA statistically downscaled dataset, as explained later in this document.

Table 12. NOAA Atlas 14 stations used in this project. Station ids with a '' have 20-30 years of AMS data available up to the year 2005.*

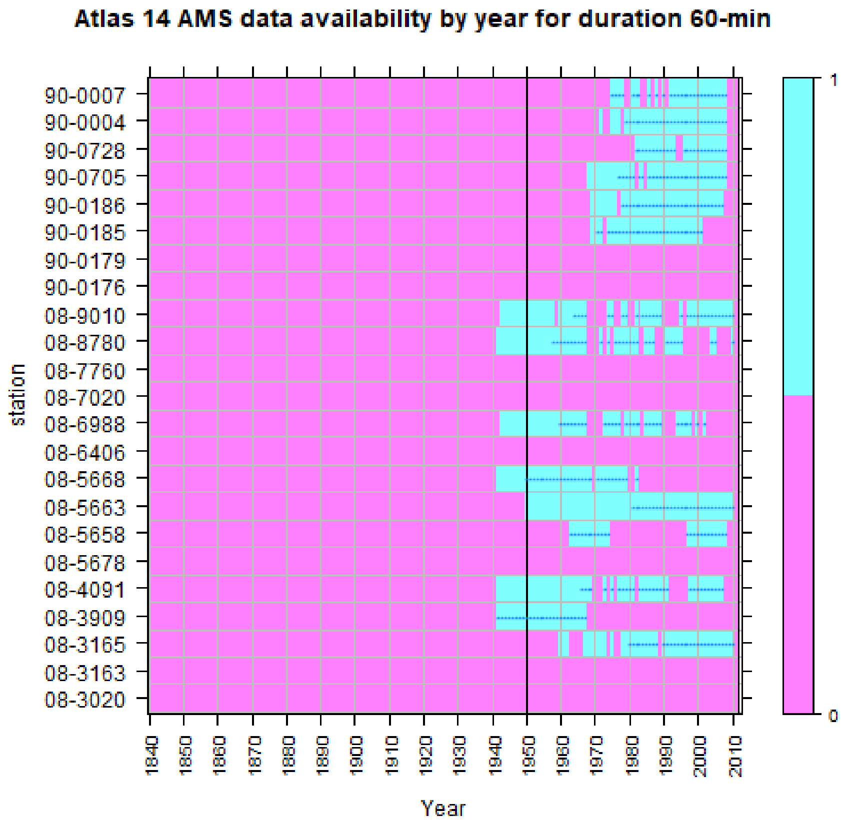
STATION ID	STATION NAME	AGENCY	LAT (degrees)	LONG (degrees)	MIN DURATION
08-3165*	FT LAUDERDALE INTL AP	NCDC	26.0719	-80.1536	hourly
08-3909*	HIALEAH	NCDC	25.8175	-80.2858	hourly
08-4091*	HOMESTEAD EXP STN	NCDC	25.5	-80.5	hourly
08-5658	MIAMI BEACH	NCDC	25.8064	-80.1336	hourly
08-5663*	MIAMI INTL AP	NCDC	25.7906	-80.3164	hourly
08-5668*	MIAMI WSO CITY	NCDC	25.7167	-80.2833	hourly
90-0185*	MRF122	SFWMD FL	25.47	-80.3464	hourly
90-0186*	MRF123	SFWMD FL	25.3669	-80.3764	hourly
08-6988*	PENNSUCO 5 WNW	NCDC	25.9297	-80.4539	hourly
90-0705*	S18C-R	SFWMD FL	25.3306	-80.525	hourly
90-0728*	S332-R	SFWMD FL	25.4217	-80.5897	hourly
08-8780*	TAMIAMI TRL 40 MI BEND	NCDC	25.7608	-80.8242	hourly
08-9010*	TRAIL GLADE RANGES	NCDC	25.7647	-80.4775	hourly
90-0004*	3AS+R	SFWMD FL	26.0821	-80.6915	hourly
90-0007*	3ASW+R	SFWMD FL	25.9898	-80.8362	hourly
08-3020*	FLAMINGO RS	NCDC	25.1422	-80.9144	daily
08-3163*	FT LAUDERDALE	NCDC	26.1019	-80.2011	daily
08-5678*	MIAMI 12 SSW	NCDC	25.65	-80.3	daily
90-0176*	MRF114	SFWMD FL	26.0603	-80.2317	daily
90-0179*	MRF117	SFWMD FL	25.8269	-80.3442	daily
08-6406*	OASIS RS	NCDC	25.8581	-81.0319	daily
08-7020*	PERRINE 4W	NCDC	25.5819	-80.4361	daily
08-7760*	ROYAL PALM RANGER STA	NCDC	25.3867	-80.5936	daily

Table 13. Correction factors applied to constrained AMS data across hourly durations.

Duration (hours)	1	2	3	6	>6
Correction factor	1.09	1.04	1.02	1.01	1.00

Table 14. Correction factors applied to constrained AMS data across daily durations.

Duration (days)	1	2	3	4	7	>7
Correction factor	1.12	1.04	1.03	1.02	1.01	1.00



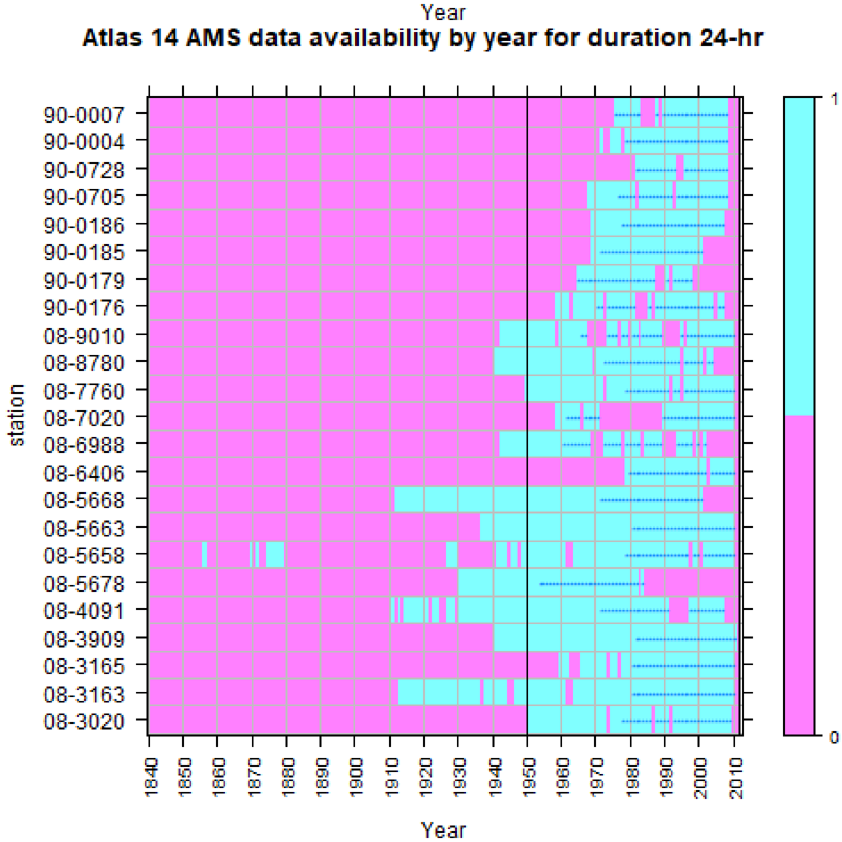


Figure 92. Availability of Atlas 14 AMS data for 1-hour (top) and 1-day (bottom) durations. Cyan boxes indicate years with valid AMS data. Blue dots indicate the last 30 years of available data.

South Florida Water Management District's DBHydro daily rainfall data

Rainfall data from the South Florida Water Management District's (SFWMD) DBHydro database were obtained for 30 stations in the vicinity of Miami-Dade County with sufficiently long periods of record (>20-30 years) (Figure 91). After further investigation, it was found that nine (9) of these stations were duplicates of NOAA Atlas 14 stations even when they often did not plot on top of each other due to differences in the provided significant figures for coordinates. This left 21 rainfall stations for further analysis. The rainfall data at these DBHydro stations is reported on a daily timestep and comes from various agencies including the SFWMD, the United States Geological Survey, and Everglades National Park. Data at some of the stations can start as early as 1941, but is generally available since the 1980s at most stations (Figure 93). The data is provided with daily data quality qualifiers (Table 15), which can be used to assess whether a particular value is reliable or not.

In order to develop a reliable timeseries of annual maxima (AMS) at each DBHydro rainfall station, it was necessary to first assess whether there was enough daily data present during each year. Too many missing values would bias the calculated annual maxima. Therefore, we followed the same criteria used by NOAA (Figure 4.3.1 of NOAA Atlas 14 Volume 9, 2013) to extract annual maxima for durations of 1-7 days. The calculated annual maxima for durations 1-7 days in a certain a year is considered reliable if the following conditions are met:

- Less than 20% of daily data is missing
- Less than 20% of wet season data is missing (wet season defined by NOAA as the months of March-October for daily durations).
- Less than 33% of daily data was accumulated for periods over 1 day
- Less than 15% of daily wet season data was accumulated for periods over 1 day

These criteria were programmed in R, and resulted in one station (DBKey 5815) being dropped completely due to frequent rainfall accumulations over multiple days. The next step was calculating the AMS timeseries each of the remaining 20 stations for durations of 1-7 days, while making sure that annual maxima for a certain duration were equal to or exceeded the annual maxima for the previous lower duration as done in NOAA Atlas 14. Subsequently, the *constrained* series were converted to *unconstrained* by applying the correction factors in Table 14. An additional check was performed again to make sure that *unconstrained* annual maxima for a certain duration were equal to or exceeded the annual maxima for the previous lower duration.

The most recent 30 years of *unconstrained* AMS at each station will be used in the final analysis. NOAA Atlas 14 volume 9 only used rainfall stations with at least 30 years of valid AMS values. However, in areas with low station density, stations with as little as 20 years of data were included in the dataset. We follow the same approach here. The remaining 20 DBHydro stations all had between 19-30 years of valid AMS data available (Table 16) and therefore were included in our dataset. As a final check, a recursive version of the Grubbs' statistical test for outliers (Grubbs, 1950) was used in order to identify potential erroneous AMS values at each station and for each duration. Due to the relatively short data records (19-30 years), the validity of this test in properly identifying outliers is questionable. However, it provided a methodology for flagging and investigating high values.

AMS for the years 1997, 1999, and 2017 were consistently identified as outlier values at a large number of stations based on the application of the Grubbs' statistical test in R. Examination of the rainfall timeseries showed that these occurred on exactly the same dates at a large number of stations and were frequently associated with known storm or high rainfall events (e.g. Hurricane Irma in 2017). Most outliers were considered valid with the exception of some events during Hurricane Irma in stations in central Everglades National Park (DBKeys 6040, 6041, G6149, and G6152) where rainfall amounts of up to 19 in/day were recorded. These large rainfall amounts were in disagreement with the SFWMD's NEXRAD rainfall maps and with NOAA's official rainfall totals for Hurricane Irma (see Figure 12 of https://www.nhc.noaa.gov/data/tcr/AL112017_Irma.pdf, accessed May 6, 2019). Therefore, the 2017 AMS values for these four stations were changed to missing. As a result, both G6149 and G6152 did not have enough reliable annual maximum values available and were eliminated from the analysis. In the end, a total of 18 SFWMD daily rainfall stations remained in the dataset.

The years 1998 and 2000 were also outlier years at a few stations; however, these large rainfall values happened on the same date at nearby stations; therefore, they were also assumed to be valid. Finally, an isolated high AMS value at the rainfall station with DBKey H2005 in the year 2005 corresponded to a high rainfall event on 5/31/2005, which was corroborated by inspection of SFWMD's NEXRAD rainfall maps (https://apps.sfwmd.gov/nexrad2) and by comparison with NOAA ATLAS 14 station 08-8780. The final years with valid AMS values at each SFWMD station are shown in Figure 93.

A separate dataset was developed with AMS data for the four (4) daily stations with sufficient years of AMS data up to the year 2005 (marked with '*' in Table 16). This second dataset was used in bias-correction of projected precipitation extremes based on the LOCA statistically downscaled dataset, as explained later in this document.

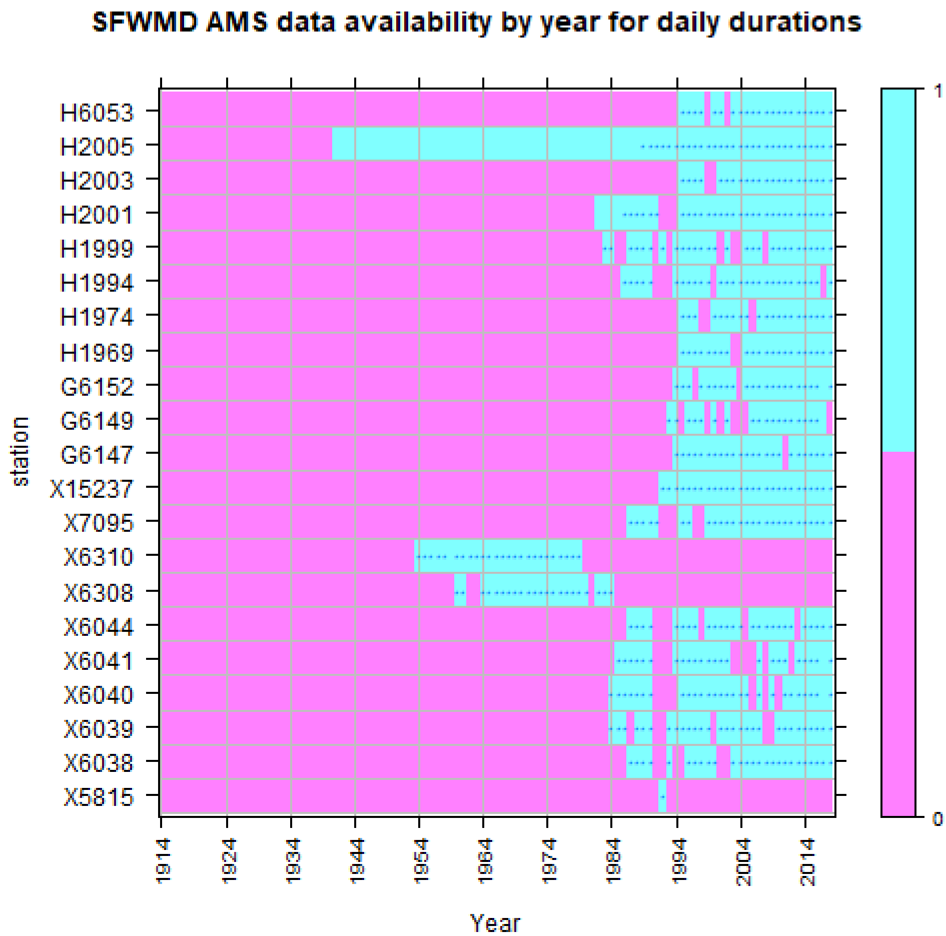


Figure 93. Availability of SFWMD rainfall data for daily durations. Cyan boxes indicate years with valid AMS data. Blue dots indicate the last 30 years of available data. Station labeled by its DBKey.

Table 15. Rainfall data qualifiers used in SFWMD's DBHydro database.

Qualifier	Meaning
A	Accumulated rainfall
M	Missing
N	Not yet available
X	Included in next amount marked A
P	Provisional data subject to revision
!	Normal limits exceeded

Table 16. DBhydro daily rainfall stations used in this project. DBKeys with a '' have 20-30 years of AMS data available up to the year 2005.*

DBKEY	STATION	AGENCY	LAT (degrees)	LONG (degrees)
6038	NP-P36	ENP	25.52833	-80.79528
6039	NP-P38	ENP	25.37056	-80.83361
6040	NP-203	ENP	25.62389	-80.73889
6041	NP-206	ENP	25.54500	-80.67194
6044	NP-201	ENP	25.71778	-80.71944
6308*	WHEELER_R	DADE	25.73528	-80.30111
6310*	STONEB_R	DADE	25.91306	-80.17528
7095	NP-EPR	ENP	25.28056	-80.50806
15237	NP-EV8	ENP	25.34583	-80.47889
G6147	NP-205	ENP	25.68944	-80.86472
H1969	R-127	ENP	25.35278	-80.60639
H1974	NP-N10	ENP	25.46194	-80.60528
H1994	NP-R3110	ENP	25.44722	-80.62639
H1999	NP-P35	ENP	25.46083	-80.86472
H2001*	NP-P37	ENP	25.28583	-80.68861
H2003	NP-ROB	ENP	25.43889	-80.53639
H2005*	NP-FMB	ENP	25.76056	-80.82417
H6053	NP-CHP	ENP	25.22917	-80.70361

South Florida Water Management District's DBHydro breakpoint (hourly) rainfall data

Breakpoint rainfall data from the South Florida Water Management District's (SFWMD) DBHydro database were obtained for 25 stations in the vicinity of Miami-Dade County with sufficiently long periods of record (>20-30 years). The rainfall data at these SFWMD stations is reported whenever there are breakpoints (changes) in the measurements. The breakpoint data spans the period 1997-present. The data is provided with quality qualifiers (Table 15), which can be used to assess whether a particular value is reliable or not.

The *runivg* (interval value generator) program developed by the SFWMD was run on the SFWMD network to compute hourly rainfall sums from the breakpoint data at these stations (Appendix A. C-shell script to run runivg program on the SFWMD network). The hourly timeseries generated by the program can only have no qualifier or an "M" (missing) qualifier. In the case when only partial breakpoint data is available during an hour, the program computes the sum, the value is given an "M" qualifier, and the percentage of missing data during the hour is given. In our analysis, we set every hourly value with an "M" qualifier and more than 10% missing data during the hour as missing (NA).

In order to develop a reliable timeseries of annual maxima (AMS) at each DBHydro hourly rainfall station, it was necessary to first assess whether there was enough hourly data present during each year. Too many missing values would bias the calculated annual maxima. Therefore, we followed the same criteria used by NOAA (Figure 4.3.1 of NOAA Atlas 14 Volume 9, 2013) to extract annual maxima for durations of 1 hour to 7 days. The calculated annual maximum for durations of 1 hour to 7 days at a station in a certain a year is considered reliable if the following conditions are met:

- Less than 20% of daily data is missing
- Less than 20% of wet season data is missing (wet season defined by NOAA as the months of March-October for daily durations and May-October for hourly durations).

These criteria were programmed in R, and resulted in five (5) stations (3AS3W3+R, MBTS+R, S12D+R, S179-R, S332-R) being dropped completely due to high frequency of missing hourly values. The next step was calculating the AMS timeseries each of the remaining 20 stations for durations of 1 hour to 7 days, while making sure that annual maxima for a certain duration were equal to or exceeded the annual maxima for the previous lower duration as done in NOAA Atlas 14. Subsequently, the *constrained* series were converted to *unconstrained* by applying the correction factors in Table 13. An additional check was performed again to make sure that *unconstrained* annual maxima for a certain duration were equal to or exceeded the annual maxima for the previous lower duration.

After further investigation, it was found that two (2) of the remaining stations were duplicates of NOAA Atlas 14 stations (S18C-R and S20F-R). The remaining 18 DBHydro hourly rainfall stations (Table 17) all had between 20-24 years of valid AMS data available (Figure 94) and were included in our dataset. As a final check, a recursive version of the Grubbs' statistical test for outliers (Grubbs, 1950) was used in order to identify potential erroneous AMS values at each station and for each duration. Due to the relatively short data records (20-24 years), the validity of this test in properly identifying outliers is questionable. However, it provided a methodology for flagging and investigating high values.

AMS for the years 1999, 2000, 2005 were consistently identified as outlier values at a large number of stations based on the application of the Grubbs' statistical test in R. Examination of the rainfall timeseries showed that these occurred on similar dates and times at a large number of stations and were frequently associated with known storm or high rainfall events (e.g. Hurricane Irene in 1999, TS. Leslie in 2000, Hurricane Katrina in 2005). Therefore, these large values were considered valid. Isolated outlier hours or days were corroborated by inspection of SFWMD's NEXRAD rainfall maps (https://apps.sfwmd.gov/nexrad2) and by comparison with other SFWMD and NOAA ATLAS 14 stations in the vicinity of the station in question.

Table 17. DBhydro breakpoint (aggregated to hourly) rainfall stations used in this project

DBKEY	STATION	AGENCY	LAT (degrees)	LONG (degrees)
IX715	JBTS+R	WMD	25.22444	-80.54
IY085	MDTS+R	WMD	25.27861	-80.395
IY095	MIAMI+R	WMD	25.82694	-80.3442
90593	S123-R	WMD	25.61028	-80.3078
90610	S177-R	WMD	25.40306	-80.5583
90605	S21-R	WMD	25.54306	-80.3308
90559	S21A-R	WMD	25.51944	-80.3461
90607	S26-R	WMD	25.80722	-80.2603
90581	S27-R	WMD	25.85111	-80.1883
90569	S28Z-R	WMD	25.91333	-80.2931
90560	S29-R	WMD	25.92861	-80.1514
90608	S29Z-R	WMD	25.96194	-80.2625
90570	S30-R	WMD	25.95639	-80.4317
IY618	S331W+R	WMD	25.61028	-80.5094
90567	S334-R	WMD	25.76167	-80.5022
90583	S335-R	WMD	25.77583	-80.4825
IY649	S336+R	WMD	25.76111	-80.4969
90584	S338-R	WMD	25.66083	-80.4808

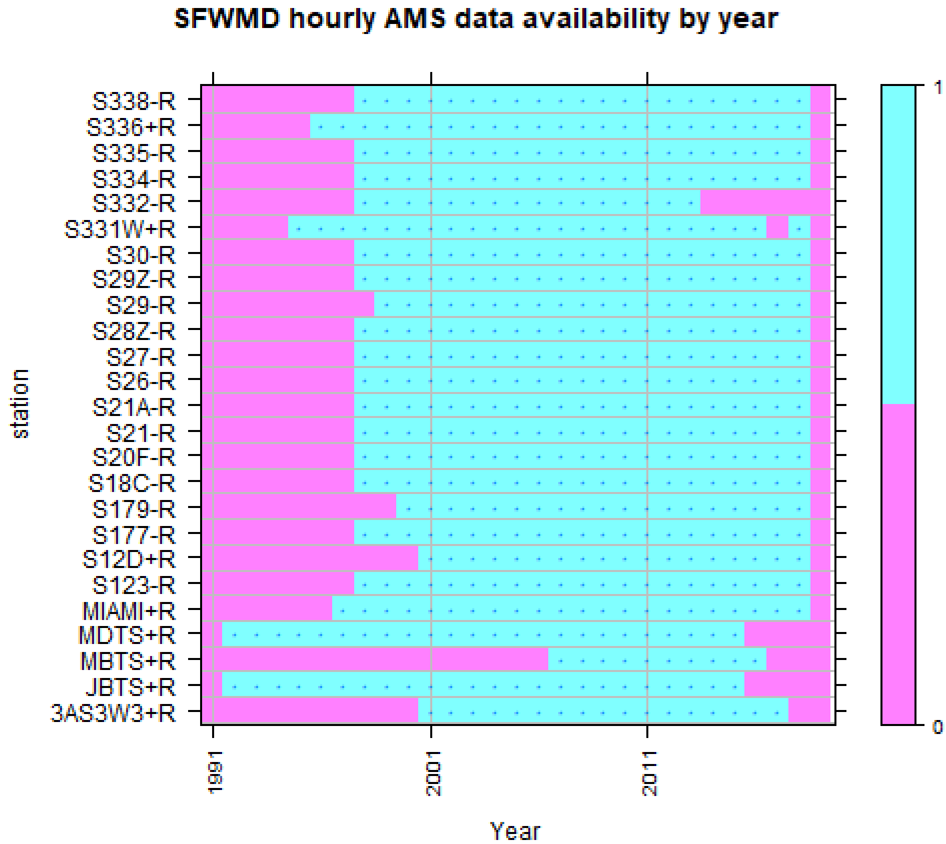


Figure 94. Availability of SFWMD rainfall data for hourly durations. Cyan boxes indicate years with valid AMS data. Blue dots indicate the last 20-30 years of available data. Station labeled by its name.

South Florida Water Management District's NEXRAD rainfall data

Since 2002, the SFWMD has been acquiring gage-corrected radar rainfall data on a 2 km x 2 km grid from various vendors (https://apps.sfwmd.gov/nexrad2/docs/aboutSFWMDNEXRADdata.pdf), and makes this data available to the public at various time intervals on a web interface located at https://apps.sfwmd.gov/nexrad2. The quality of the data and of the gage-correction methodology have improved over the years especially under the new vendor since 2007; however, the relatively short length of the dataset (16 years) limits its use in depth-duration-frequency analysis. In spite of this and despite the 2 km x 2 km grid resolution limitation, the interface has been an invaluable tool in aiding quality control and quality assurance of gage rainfall from other sources.

Miami Dade County rainfall data

Hourly historical rainfall data at 42 stations throughout Miami -Dade County for 1995-2019 were provided by the Miami-Dade Water and Sewer Department (WASD). The data provided is in raw format and has not undergone a rigorous quality assurance and quality control (QA/QC) process. Examination of the data showed numerous instances of negative values of various magnitudes (all of which are faulty measurements according WASD staff), instances of the same value repeating hour after hour, instances of extremely high daily values (e.g. 100-300 inches/day), instances of extremely high isolated hourly values (10+ inches/hour) many of which are on the last or first timestep of the day, among others.

An attempt was made to remove questionable values from the dataset, initially defined as those that exceeded the 1-in-100-year and 1-in-1000-year rainfall depths for hourly and daily durations from NOAA Atlas 14 stations in the county (1-in-100 year: 5.7 inches/hr, and 16 inches/day; 1-in-1000 year: 8.5 inches/hr, and 26 inches/day). However, many recorded values exceeded these limits by large amounts. Although in many instances the extremely high values happened during days when the SFWMD's NEXRAD maps showed high rainfall activity in the region, the magnitudes still seemed unreasonably high even though they were point values. Extremely high hourly and daily rainfall values also occurred at various stations during days with little to no rainfall activity in the region based on inspection of both the South Florida Water Management Model's 2-mi x 2-mi rainfall binary file (1965-2016 data) and the SFWMD NEXRAD maps (2002-2019). This reduces the trust in the data for the type of quantitative analysis used in this study.

Code was written in R with the purpose of automating a QA/QC process for the Miami-Dade WASD rainfall data. From 150 stations in total, 55 had valid hourly rainfall data from 1995-2019. Negative data values at these 55 stations were set as missing (NA). As a first pass, hourly rainfall values exceeding the 1-in-1000-year depth of 8.5 inches/yr based on NOAA Atlas 14 stations in the county, were also set to missing (NA). All hourly values for days exceeding the 1-in-1000-year daily rainfall depth of 26 inches/day were removed as well. Later on, if the data quality assessment turned out to be successful at some stations, a Grubbs' statistical test would be used for further outlier identification and possible exclusion from the annual maxima series.

These 55 stations were further assessed to determine whether there was enough daily data present during each year for the calculated annual maxima to be valid. Too many missing values would bias the calculated annual maxima. Therefore, we followed the same criteria used by NOAA (Figure 4.3.1 of NOAA Atlas 14 Volume 9, 2013) to extract annual maxima for durations of 1 hour to 7 days. The calculated annual maxima in a certain a year is considered reliable if the following conditions are met:

- Less than 20% of daily data is missing
- Less than 20% of wet season data is missing (wet season defined by NOAA as the months of March-October for daily durations and May-October for hourly durations).
- Less than 33% of daily data was accumulated for periods over 1 day – This criterion was not applicable to WASD rainfall data.

- Less than 15% of daily wet season data was accumulated for periods over 1 day – This criterion was not applicable to WASD rainfall data.

It was found that only 12 stations met all the criteria above for the minimum desired number of years (20 years). It was then observed that many of the remaining 12 stations had entire years, which had been designated as valid AMS years, with all values equal to zero. Once those years were eliminated from these stations, only 6 stations were left for analysis. Later, it was noticed how the computed AMS values for some years at some of these stations were extremely low even for durations of 7 days. Furthermore, quite often the calculated annual maxima were the same across durations, which seemed unreasonable based on experience at other nearby stations.

Due to the numerous issues found with the data, it was decided to not include this dataset as part of this project.

Florida State University's COAPS rainfall data

The state climatologist at Florida State University's Center for Ocean-Atmospheric Prediction Studies (COAPS) in the past provided us with NOAA NCDC daily rainfall data at what he considered the most reliable stations in the state of Florida. As part of this project, we were provided with more recent data up to the year 2017. Further investigation of this dataset showed that all the stations were already included in the NOAA Atlas 14 Volume 9 AMS dataset.

University of Florida's IFAS FAWN rainfall data

The University of Florida's Institute of Food and Agricultural Sciences (IFAS) Florida Automated Weather Network (FAWN) provides near-real time weather information directed towards agricultural users throughout the state of Florida (https://fawn.ifas.ufl.edu/). Historical rainfall, precipitation and other weather data is available for download at timesteps ranging from 15 minutes to daily, at https://fawn.ifas.ufl.edu/data/fawnpub/. Only two FAWN stations (Homestead and Fort Lauderdale) are located in the vicinity of Miami-Dade County. Of these two, only Homestead has sufficient years of data (1997-2018). However, further investigation showed that this is likely the same station as NOAA's Homestead Experimental Station (08-4091); therefore, it was excluded from our analysis.

GROWER network rainfall data

Within the last few years, a network of weather stations, under the name of GROWER, has been established in farms throughout the state. Data is available for download from the University of Florida's

IFAS at: https://fawn.ifas.ufl.edu/mffw/index.html. Due to the short period of data availability this dataset was not used in this project.

CoCoRaHS rainfall data

The Community Collaborative Rain, Hail and Snow Network (https://www.cocorahs.org) is a non-profit, community-based group of volunteers which measure and map precipitation over the United States. Data for Miami-Dade County was downloaded from: their export interface at http://data.cocorahs.org/cocorahs/export/exportmanager.aspx using this specific export link: http://data.cocorahs.org/export/exportreports.aspx?ReportType=MultiDay&dtf=1&Format=CSV&State=FL&County=MD&ReportDateType=reportdate&StartDate=1/1/1990&EndDate=4/4/2019&TimesInGMT=False. Rainfall data was only available sporadically between 2007-2019 at 27 stations in the county. Due to the limited data availability, this dataset was not used in this project.

Statistically-downscaled historical and projected model rainfall

SFWMD staff has in the past evaluated various statistically downscaled climate data products in terms of their ability to capture historical seasonal and long-term rainfall temporal and spatial patterns in south Florida. Based on analyses done by Irizarry and SFWMD staff, it was observed that the University of California's LOCA product generally did a better job than the US Bureau of Reclamation's BCCA (Bias-Correction Constructed Analogues) downscaled data product at capturing rainfall patterns in the state (Irizarry et al., 2016), particularly during extreme events. For this reason and also due to the fact that the LOCA product was used to guide the 4th US National Climate Assessment report (https://scenarios.globalchange.gov/), this dataset was chosen and further evaluated for providing future estimates of rainfall extremes as part of this project. A description of the LOCA dataset follows.

The University of California at San Diego has used Localized Constructed Analogues technique (LOCA) to downscale 32 global climate models from the World Climate Research Programme (WCRP) Coupled Model Intercomparison Project phase 5 (CMIP5) archive at a 1/16th degree (approx. 4.3 miles; 6.9 km) spatial resolution. LOCA covers North America from central Mexico through Southern Canada. LOCA is a statistical downscaling technique that uses past history to add improved fine-scale detail to global climate models (Pierce et al., 2014). First, a pool of candidate observed analog days is chosen by matching the model field to be downscaled to observed days over the region that is positively correlated with the point being downscaled, which leads to a natural independence of the downscaling results to the extent of the domain being downscaled. Then the one candidate analog day that best matches in the local area around the grid cell being downscaled is the single analog day used there.

Most grid cells are downscaled using only the single locally selected analog day, but locations whose neighboring cells identify a different analog day use a weighted combination of the center and adjacent analog days to reduce edge discontinuities. By contrast, existing constructed analog methods typically use a weighted average of the same 30 analog days for the entire domain. By greatly reducing this averaging, LOCA produces better estimates of extreme days, constructs a more realistic depiction of the spatial coherence of the downscaled field, and reduces the problem of producing too many light-precipitation days.

The historical period for LOCA is 1950-2005, and there are two future representative concentration pathway (RCP) scenarios available over the period 2006-2100 (although some models stop in 2099). RCP 4.5 and RCP 8.5, correspond to medium-low (4.5 W/m2) and high (8.5 W/m2) year 2000 radiative forcing values, respectively. The variables currently available are daily minimum and maximum temperature, and daily precipitation. Over the next year they will be running the VIC hydrological model with the downscaled data, which will give many more variables, such as snow cover, soil moisture, runoff, and humidity, all at a 1/16th degree spatial resolution on a daily timescale. More information on LOCA can be found at http://loca.ucsd.edu/.

Retrospective historical and predicted daily rainfall for 30 downscaled model runs for each of the two RCPs in LOCA (60 runs in total) were downloaded for use in this project. The downscaled CMIP5 runs in LOCA are listed in Table 18.

Table 18. CMIP5 models downscaled by the University of San Diego's LOCA project and used in this project.

Modeling Center (or Group)	Institute ID	Model Name
Commonwealth Scientific and Industrial Research Organization (CSIRO) and Bureau of Meteorology (BOM), Australia	CSIRO-BOM	ACCESS1.0 ACCESS1.3
Beijing Climate Center, China Meteorological Administration	BCC	BCC-CSM1.1
Canadian Centre for Climate Modelling and Analysis	CCCMA	CanESM2
National Center for Atmospheric Research	NCAR	CCSM4
Community Earth System Model Contributors	NSF-DOE-NCAR	CESM1(BGC) CESM1(CAM5)
Centre National de Recherches Météorologiques / Centre Européen de Recherche et Formation Avancée en Calcul Scientifique	CNRM-CERFACS	CNRM-CM5
Commonwealth Scientific and Industrial Research Organization in collaboration with Queensland Climate Change Centre of Excellence	CSIRO-QCCCE	CSIRO-Mk3.6.0
EC-Earth (European Earth System Model)	EC-EARTH	EC-EARTH
IAP (Institute of Atmospheric Physics, Chinese Acad emy of Sciences, Beijing, China) and THU (Tsinghua University)	LASG-CESS	FGOALS-g2
NIMR (National Institute of Meteorological Researc h, Seoul, South Korea) in association with the Met Office Hadley Centre, UK	NIMR/KMA	HADGEM2-AO
Met Office Hadley Centre, Fitzroy Road, Exeter, Dev on, EX1 3PB, UK	MOHC	HADGEM2-CC

Modeling Center (or Group)	Institute ID	Model Name
		HADGEM2-ES
NOAA Geophysical Fluid Dynamics Laboratory	NOAA GFDL	GFDL-CM3 GFDL-ESM2G GFDL-ESM2M
NASA Goddard Institute for Space Studies	NASA GISS	GISS-E2-H GISS-E2-R
Institut Pierre-Simon Laplace	IPSL	IPSL-CM5A-LR IPSL-CM5A-MR
Japan Agency for Marine-Earth Science and Technology, Atmosphere and Ocean Research Institute (The University of Tokyo), and National Institute for Environmental Studies	MIROC	MIROC-ESM MIROC-ESM-CHEM
Atmosphere and Ocean Research Institute (The University of Tokyo), National Institute for Environmental Studies, and Japan Agency for Marine-Earth Science and Technology	MIROC	MIROC5
Max-Planck-Institut für Meteorologie (Max Planck Institute for Meteorology)	MPI-M	MPI-ESM-MR MPI-ESM-LR
Meteorological Research Institute	MRI	MRI-CGCM3
Norwegian Climate Centre	NCC	NorESM1-M

**Note: Ensemble member r1i1p1 used for each model for both RCP45 and RCP85 with the exception of CCSM4_r6i1p1 for RCP45, CCSM4_r6i1p1 for RCP85, EC-EARTH_r8i1p1for RCP45, EC-EARTH_r2i1p1 for RCP85, GISS-E2-H_r6i1p3, GISS-E2-R_r6i1p1, GISS-E2-H_r2i1p1for RCP85, and GISS-E2-R_r2i1p1 for RCP85.*

DDF curve fitting across durations using At-site RFA

The Generalized Extreme Value (GEV) distribution family is frequently used in Extreme Value Theory to model block (e.g. seasonal or annual) maxima of rainfall and is described by the following cumulative distribution function (Fisher and Tippett, 1928; Jenkinson, 1955; Coles, 2001; Katz et al., 2002):

$$P(X \leq x) \;=\; F(x) \;= exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

Equation 9

where μ, σ, and ξ are the location, scale, and shape parameters, respectively, of the GEV, which can be fit to annual maximum data using maximum likelihood estimation (MLE) or L-moments methods. x can be either rainfall intensity (i) or depth (D). This distribution models the maxima of a series of independent and identically distributed observations and is an appropriate distribution for analyzing extreme values. It encapsulates three distinct extreme value distributions by means of the shape parameter: Gumbel (ξ=0), which is light tailed and unlimited; Fréchet (ξ>0), which has a lower limit at μ-σ/ξ and is heavy tailed; and the reverse Weibull (ξ<0), which has an upper limit at μ-σ/ξ and is short tailed.

In developing depth-duration-frequency (DDF) curves for a station or region, the Generalized Extreme Value distribution is often fit to annual maximum series (AMS) at various durations independently. However, it is possible for the fitted cumulative distribution functions (CDF) for consecutive durations to cross, which are reflected in the DDF curves as decreasing quantiles with increasing duration. This violates the physical constraint that rainfall depth quantiles at longer durations must exceed quantiles at shorter durations for a given return period. An objective in fitting GEV curves to the AMS (as opposed to just using empirical quantiles) is to extrapolate to large return periods (small exceedance frequencies) beyond the length of the AMS record. It has been shown that these extreme quantiles are very sensitive to the fitted parameters, especially to the shape parameter (ξ). The estimated shape parameter can have large errors and can be very noisy especially when estimated from short datasets using MLE. The large variation in estimated shape parameter across durations increases the chance of crossing CDFs.

Various methods exist in the literature to develop robust and consistent DDF/IDF curves based on annual maximum series (AMS) for different durations. These methods typically make assumptions about the variation of the GEV parameters in time (duration) and/or space as well as the relationship between these parameters. Some methods also pool data in time or space to improve the robustness of the GEV parameter estimates. In a previous study, Irizarry et al. (2016) employed the methodology of "At-Site Regional Frequency Analysis" (ASRFA) (Ayuso-Muñoz et al. 2015) to develop consistent DDF curves for rainfall stations in the state of Florida. It is a variation of the typical RFA methods where data for nearby stations to the station being fitted are grouped together in order to improve the GEV parameter fitting. ASRFA exchanges space (stations) for time (durations).

In ASRFA, all durations at a station are fit simultaneously by pooling AMS data for all durations at the particular station. R package {nsRFA} includes functions to implement this method. The quantile function as a function of the ASRFA method is given by:

$$x(F,i,d) = \left\{\mu^R(d) - \frac{\sigma^R(d)}{\xi^R(d)}\left[1-(-lnF)^{-\xi^R(d)}\right]\right\}\overline{x}(i,d)$$

Equation 10

$$\mu(i,d) = \mu^R\,\overline{x}(i,d)$$

Equation 11

$$\sigma(i,d) = \sigma^R\,\overline{x}(i,d)$$

Equation 12

$$\xi(i,d) = \xi^R(d)$$

Equation 13

Where *i is the* location of interest, *F* is the annual non-exceedance probability, which is related to the return period T_r by F = 1-1/(T_r) , *d is the* duration, $x(F,i,d)$ is the quantile function for the GEV at station *i* for duration *d*, (μ,σ,ξ) are the GEV location, scale and shape parameters, (μ^R,σ^R,ξ^R) are the regional (i.e. factor for all durations) GEV location, scale and shape parameters, $\overline{x}(i,d)$ is the mean annual maximum (MAM) at station *i* for duration of interest *d*.

As observed in Equation 10-Equation 13, the ASRFA method assumes that the GEV location and scale parameters are proportional to the MAM for a certain duration with the proportionality constant being the same across durations. Some theoretical limitations of the ASRFA approach are identified by Irizarry et al. (2016); however, the method was used here since it requires minimal user input and can be automated to produce consistent DDF curves at a large number of stations and across many data sources.

The ASRFA method was used in fitting consistent DDF curves to daily historical and downscaled-model AMS data at daily stations in Miami-Dade County for durations of 1, 2, 3, 4 and 7 days. For stations with hourly historical AMS data available, DDF curves were additionally fit for durations of 1, 2, 3, 6 and 12 hours.

DDF curves were fit for two different sets of historical observations. The first set consisted of a total of 59 stations with sufficient AMS data available up to the year 2018 (33 hourly and 26 daily stations). As an example, Figure 95 shows the GEV cumulative distribution function (CDF) fitted to the normalized annual maxima at station S30-R for durations from 1 hour to 7 days. It is evident that the At-site RFA method is appropriate for usage at this station due to the fact that the normalized AMS values do not depart much from the fitted CDF line. Figure 96 shows the GEV CDFs for the durations of interest, while Figure 97 shows the resulting DDF curves for the durations and return periods of interest. This first set was used to develop the main “official” maps of extremes for various durations and return periods.

Figure 98 shows contour maps of interpolated 1-in-100-year hourly rainfall totals based on thin plate spline (TPS) smoothing of the fitted DDF data at each station with sufficient AMS data available up to the year 2018. Fitted 1-in-100-year hourly rainfall totals range from 3.5 to 8.2 inches with most values below 6.5 inches with the exception of two outlier stations: S29-R and 08-4091. A smoothing or lambda factor of 0.02 was used in the Tps function of the fields package in R to generalize the surface and smooth out low and high outliers. This resulted in fitted values ranging from 4.8 to 5.7 inches for the 1-in-100-year hourly rainfall events.

Figure 99 shows that fitted 1-in-100-year daily rainfall totals range from 7.7 to 18.2 inches with most values below 15 inches with the exception of the same two outlier stations: S29-R and 08-4091. It is important to note that rainfall amounts of up to 19 in/day were recorded during Hurricane Irma in stations in central Everglades National Park (DBKeys 6040, 6041, G6149, and G6152), although these were considered suspect and removed from the dataset early on. After generalizing the surface using TPS with a smoothing factor of 0.02, the fitted values range from 8.1 to 13.7 inches for the 1-in-100-year daily rainfall events. Contour maps of 1-in-100 year rainfall totals for other daily and sub-daily durations are included in Appendix B. Contour maps of 1-in-100-year rainfall depths based on historical data. Table 19 shows the range in the 1-in-100 year fitted rainfall totals at individual stations for the durations of interest, and the rainfall totals based on the generalized surfaces developed using the TPS smoothing method with lambda=0.02. The at-station range is more conservative in that it includes high outliers. Professional judgment must be exercised when deciding on a set of design values.

A second set of historical DDF curves was developed from 26 stations with sufficient AMS data available up to the year 2005 (14 hourly and 12 daily stations). This second set was be used to bias-correct the LOCA statistically downscaled extreme precipitation projections for the period 2050-2079. This is due to the desire to limit the historical data for bias-correction to a period common to the LOCA historical period which ends in 2005.

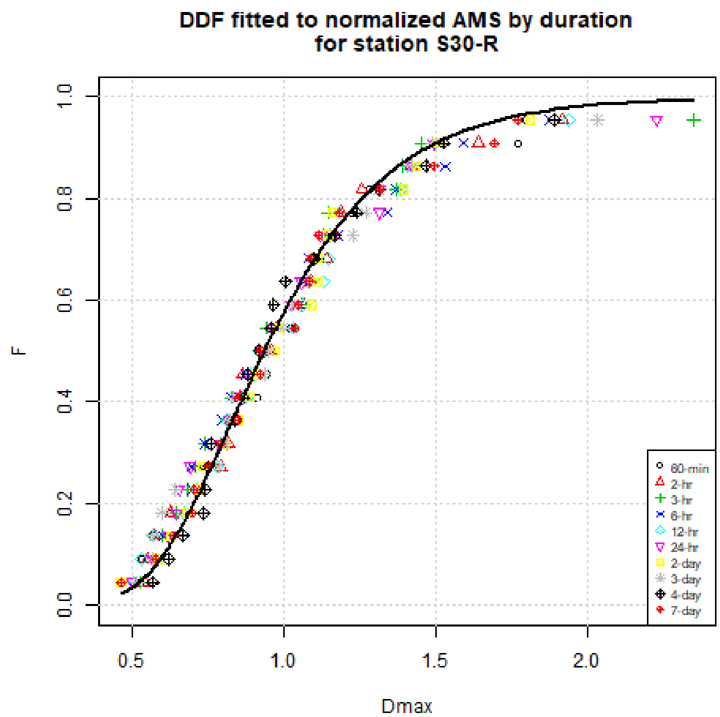


Figure 95. GEV CDF curve fitted to normalized annual maxima at station S30-R using the At-site RFA method.

GEV fits at station S30-R

for the historical period ending in 2018

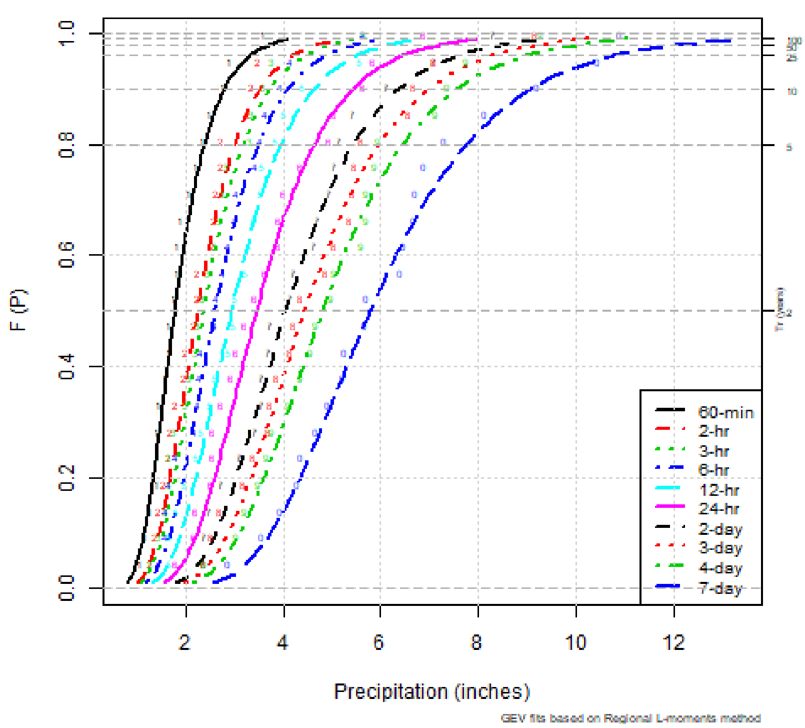


Figure 96. GEV CDFs fitted for various durations at station S30-R using the At-site RFA method.

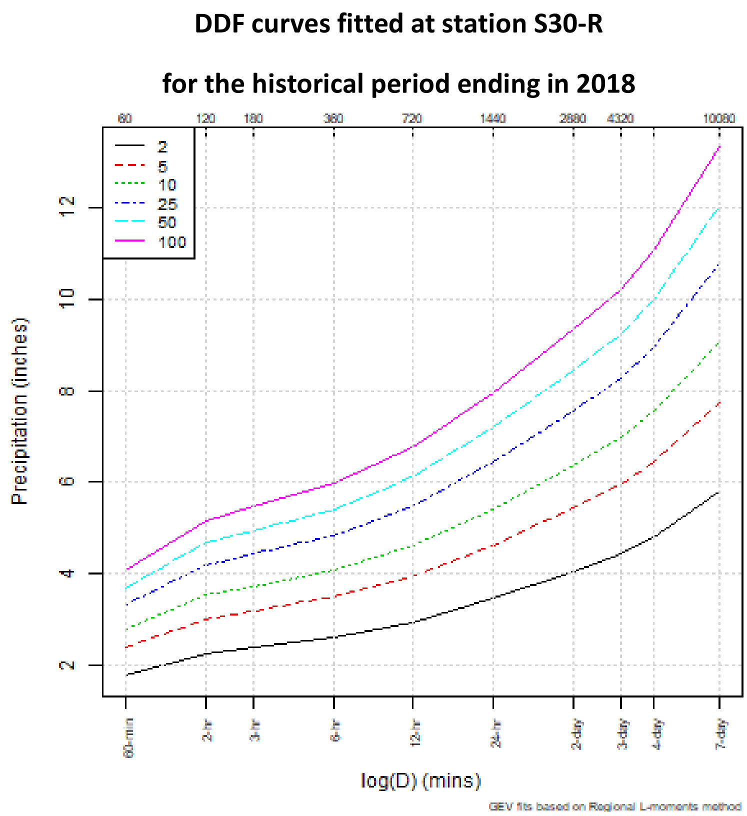


Figure 97. DDF curves fitted at station S30-R for various durations and return periods of interest based on the At-site RFA method.

Table 19. Range of 1-in-100-year rainfall totals fit at individual stations and those from generalized surface based on TPS method with lambda=0.02.

Duration	**Range of fit rainfall totals at stations (inches)**	**Range of rainfall totals from generalized surface based on TPS method with lambda=0.02 (inches)**
60-min	3.5-8.2	4.8-5.7
2-hour	4.7-10.8	6.1-7.6
3-hour	5.2-12.4	6.4-8.9
6-hour	5.9-15.3	7.2-10.6
12-hour	6.5-16.8	7.8-11.9
24-hour	7.7-18.2	8.1-13.7
2-day	9.3-21.4	9.5-16.3
3-day	9.8-22.9	10.7-18.0

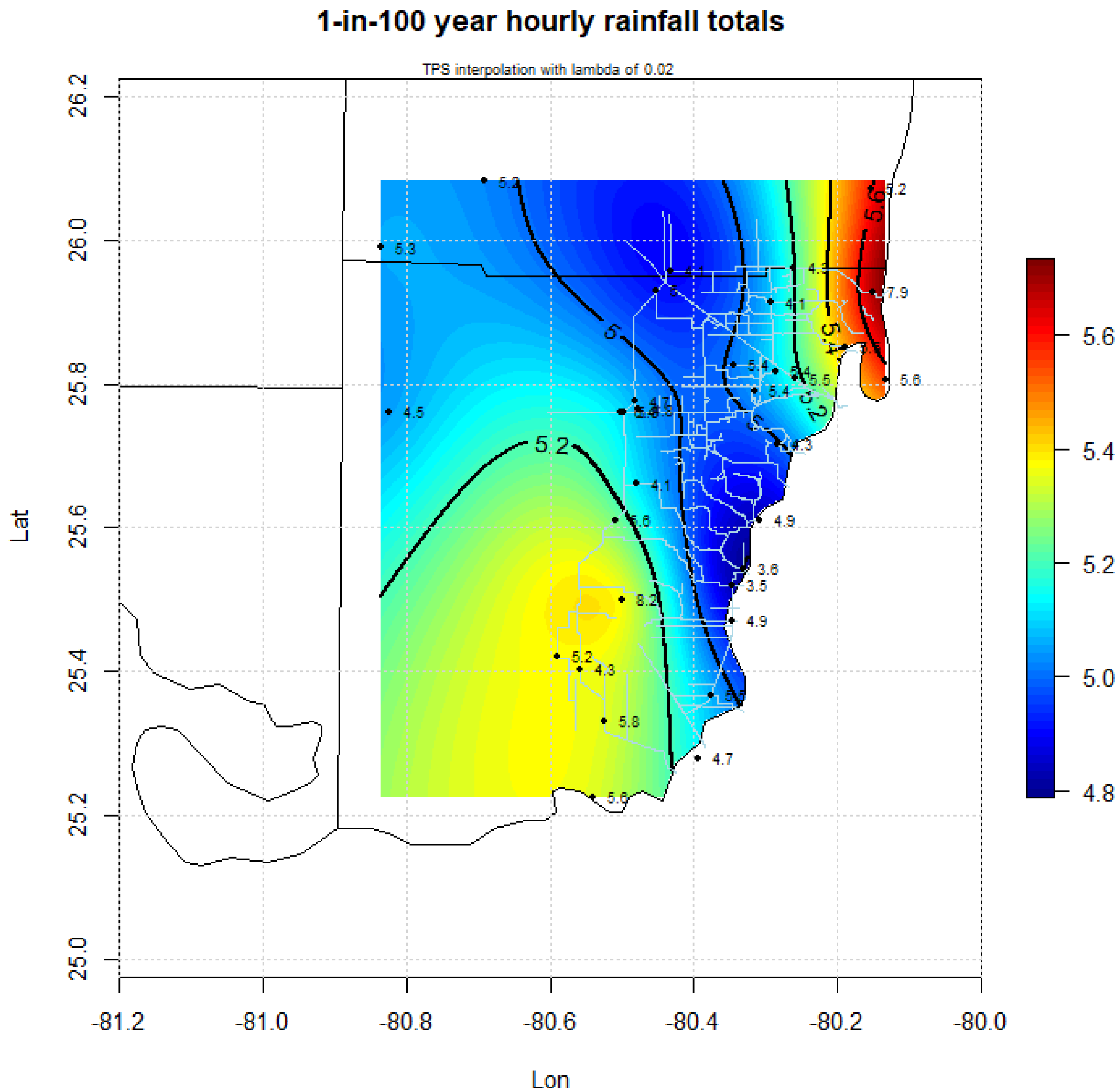


Figure 98. Interpolated 1-in-100-year hourly rainfall totals (inches) based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

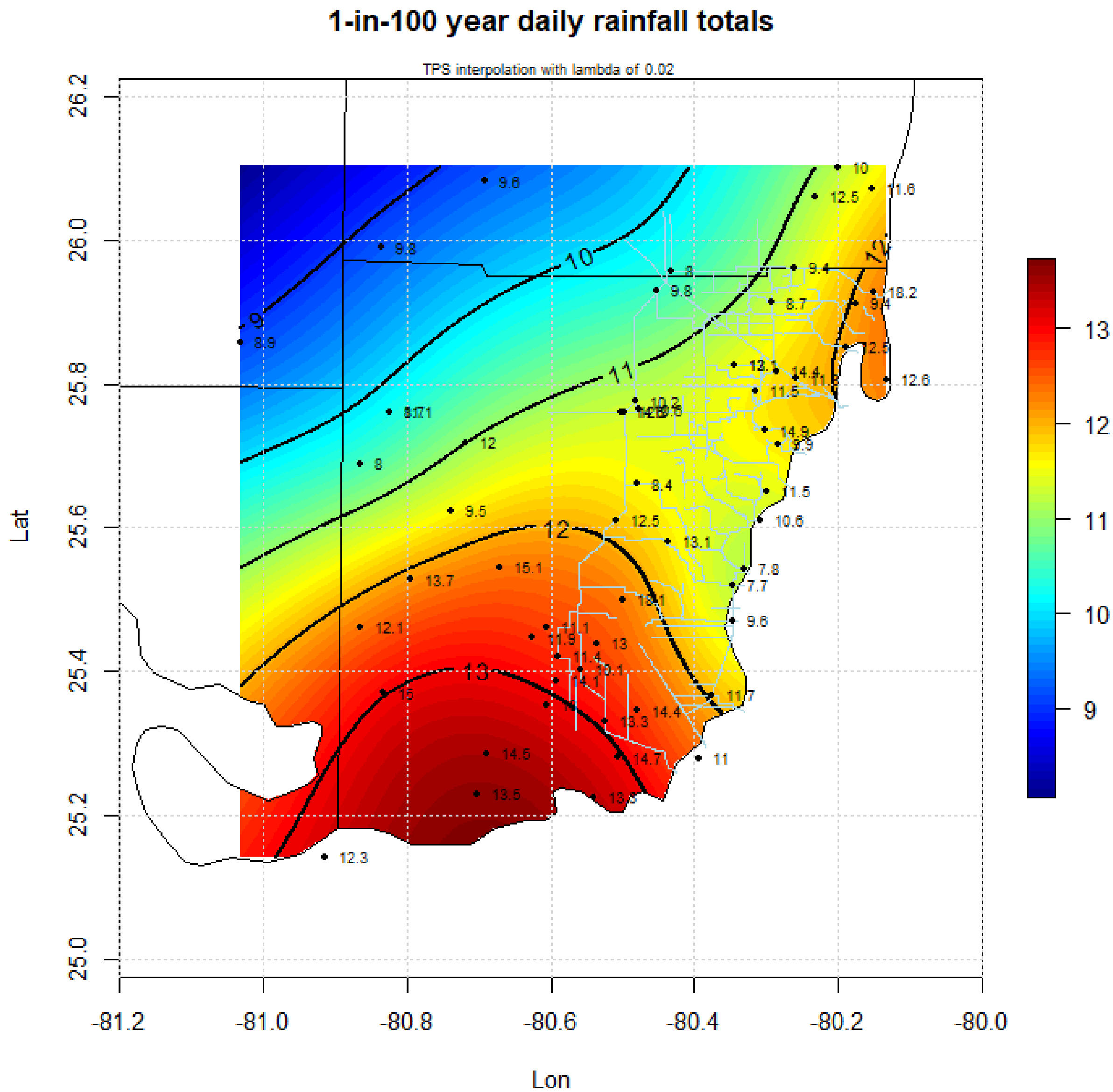


Figure 99. Interpolated 1-in-100-year daily rainfall totals (inches) based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

Quantile mapping for bias-correction of precipitation projections

Quantile mapping (QM), a CDF matching method (Panofsky and Brier, 1968), is typically applied to bias-correct entire precipitation timeseries from climate model simulations but the method can similarly be used to bias-correct annual maximum precipitation series (AMS) or DDF/IDF curves. The expression for quantile mapping is given by:

$$\hat{x}_{m-padj.} = F_{o-c}^{-1}\left[F_{m-c}\left(x_{m-p}\right)\right]$$

Equation 14

where $\hat{x}_{m-padj.}$ is the adjusted quantile for the model (m) projections (p) for the future period, F_{o-c} is the CDF of the observations (o) in the current baseline period (c) ,F_{m-c} is the CDF of the model (m) in the current baseline period (c) , x_{m-p} is the quantile for the model projections in the future baseline period. F^{-1} means the inverse of the CDF (i.e. the quantile function). The CDFs are developed based on data spanning decades and centered around some year of interest.

QM uses only information from the current period to correct for future biases. Therefore, it assumes that biases are stationary and that they will persist into the future. In other words, it assumes that $F_{o-c} \approx F_{o-p}$, so that as the mean changes, the variance and skew do not, which is unlikely under climate change. Furthermore, if a future projected value is outside the historical range, then some sort of extrapolation is required.

To avoid the limitations of QM, other methods have been developed such as Quantile Delta Mapping (QDM). As shown in Cannon et al. (2015), QM tends to inflate trends in precipitation extreme indices projected by GCMs, whereas QDM is not as prone to this problem. QDM preserves model-projected changes in quantiles, while simultaneously correcting for systematic biases across quantiles (Cannon et al., 2015). QDM also attempts to bridge the gap between point estimates for the observations vs. grid cell estimates in the model. However, it is important to note that changes in the mean may not be adequately preserved by QDM.

QDM can be applied in an additive form or a multiplicative form. Here we used the multiplicative version of quantile delta mapping (MQDM) to bias-correct future rainfall DDF curves derived from LOCA statistically downscaled CMIP5 model precipitation data with AMS values for various durations fit using the ASRFA method. The multiplicative form (MQDM) is better suited to correcting variables like precipitation where preserving relative changes is important in order to respect the Clausius-Clapeyron equation which relates the amount of atmospheric moisture to temperature changes simulated by the models. Figure 100 shows MQDM method based on hypothetical data.

Multiplicative QDM is given by:

$$\hat{x}_{m-padj.} = x_{m-p} * \left\{F_{o-c}^{-1}\left[F_{m-p}\left(x_{m-p}\right)\right] / F_{m-c}^{-1}\left[F_{m-p}\left(x_{m-p}\right)\right]\right\}$$

Equation 15

which is equivalent to:

$$\hat{x}_{m-padj.} = F^{-1}_{m-padj.}(G) = F^{-1}_{m-p}(G) * \{F^{-1}_{o-c}(G)/F^{-1}_{m-c}(G)\}$$

Equation 16

where *G* is the annual non-exceedance probability (CDF value) and is equal to *1-P*, *P* is the annual exceedance probability (AEP) which is related to the return period *T* by *1/P = T* (i.e. *G=1-1/T*), and F_{m-p} is the CDF for the model (m) projections (p) for the future period.

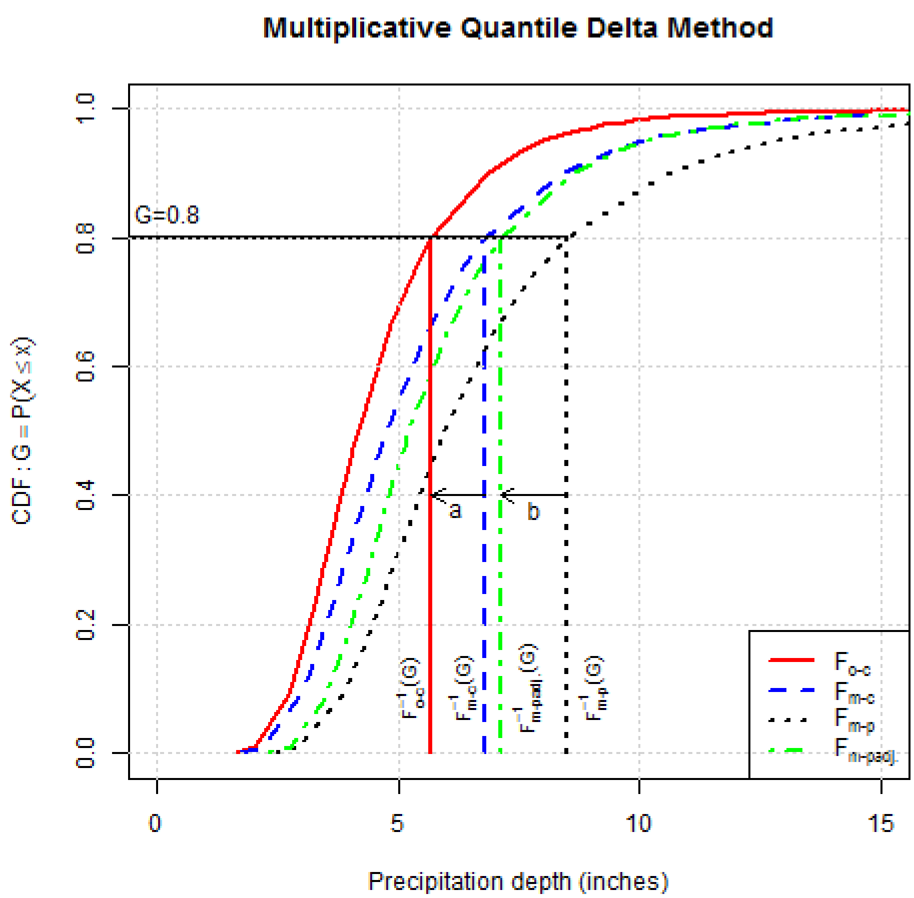


Figure 100. Diagram showing the Multiplicative Quantile Delta Method for hypothetical data.

F *is the non-exceedance probability of interest. The quantiles corresponding to* F *are given by CDF1*$^{-1}$*:* $F^{-1}_{o-c}(F)$ *for the observed current baseline, CDF2*$^{-1}$*:* $F^{-1}_{m-c}(F)$ *for the model current baseline, CDF3*$^{-1}$*:* $F^{-1}_{m-p}(F)$ *for the model projected (future) period. The corresponding adjusted quantile for the model projected (future) period is CDF4*$^{-1}$*:* $F^{-1}_{m-p\ adjust}(F) = F^{-1}_{m-p}(F)*\{F^{-1}_{o-c}(F)/F^{-1}_{m-c}(F)\}$. *The distances* a *and* b *are different in MQDM due to the use of a ratio in the bias correction equation. However,* a *and* b *would be equal in Additive Quantile Delta Method.*

At-Site RFA for the observational dataset in the current baseline period

In order to reduce biases in the application of MQDM, it is important that the timeseries of annual maxima used to derive the three CDFs of interest ($F_{o-c}, F_{m-c}, F_{m-p}$) using At-site RFA all have approximately equal length. It is well known that the GEV shape parameter tends to be underestimated with shorter record lengths where the largest extremes may not be captured. Ideally a record length of 30-40 years as a minimum is required for adequate GEV fitting. However, due to the sparsity of historical rainfall data in the county, we chose the last 20-30 years of available historical AMS data between 1950-2005 at each station as our current baseline (historical) period. The lumping of AMS data across durations in the ASRFA method effectively increases the data available for fitting the shape parameter at the expense of constraining it to be constant across durations. We found a total of 26 stations from NOAA Atlas 14 and SFWMD with 20-30 years of AMS data (12 daily and 14 hourly stations) within the chosen current baseline period, as described in previous sections.

Appendix C. Maps of At-site RFA parameters and DDF curves for the observational dataset in the current baseline period (Last 30 years up to 2005) includes maps of the fitted DDF values for hourly and daily durations and 1-in-100 year return period based on the chosen 20-30 years of observational AMS data at each of the 26 weather stations in Miami-Dade County (F_{o-c}). The fitted shape parameter, which is assumed constant between durations in ASRFA, is mostly positive with negative values at 5/26 stations.

As seen in Figure 113, the fitted hourly 1-in-100 year extremes range from 4.3-7.9 inches, which is generally consistent with the range of 3.5-8.2 inches obtained for the historical period ending in 2019 (Figure 98), although the spatial patterns differ somewhat. Overall the fitted daily 1-in-100 year extremes range from 8.7-17.8 in/day, which is generally consistent with the range of 7.7-18.2 in/day obtained for the historical period ending in 2019, although the spatial pattern differs (Figure 99). The pattern of higher daily extremes near the coast of Miami-Dade County seen in Figure 114 is similar to the pattern obtained by Irizarry et al. (2016). However, Figure 114 shows the highest daily extremes occurring in a northeast to southwest swath that is slightly inland from the coast, with lower values right along the coast. Coastal extremes in Miami-Dade County did not exceed 13.6 in/day in the previous DDF fits performed by Irizarry et al. (2016); however, it is worth nothing that the previous analysis used a smaller number of stations in the county and a different baseline period.

The spatial patterns in Figure 113 and Figure 114 are also very similar to those in the official NOAA Atlas 14 cartographic DDF maps for the hourly and daily 1-in-100 year events (see ftp://hdsc.nws.noaa.gov/pub/hdsc/data/se/fl100y60m.pdf and ftp://hdsc.nws.noaa.gov/pub/hdsc/data/se/fl100y24h.pdf) although the ranges are slightly larger in our fits. The official NOAA Atlas 14 fitted DDF values in the county range from 11.4-16.0 in/day for the 100-year daily event, and 4.4-5.7 in/hour for 100-year hourly event. The similarity in the spatial pattern with the official NOAA Atlas 14 cartographic maps is due to the fact that 22 out of the 26 stations in the historical dataset (ending in 2005) are from NOAA Atlas 14.

At-Site RFA for the downscaled model dataset in the current baseline period

The same 20-30 years selected for F_{o-c} derivation at each of the 26 weather stations identified in the previous station were used to develop F_{m-c} (model current-baseline CDFs) based on At-Site RFA methodology applied to model AMS data for the closest downscaled-model grid cell (Figure 101). Figure 102 shows goodness-of-fit statistics comparing fitted extremes from downscaled model output to fitted extremes from the observational dataset for durations (1, 2, 3, 4, and 7 days) and return periods of interest (2, 5, 10, 25, 100 years). It can be noticed that extremes are significantly underestimated in the downscaled model data with median ratios ($F_{m-c}^{-1}(G)/F_{o-c}^{-1}(G)$: modeled/observed DDF precipitation depths) ranging from 0.50-0.95 depending on duration and frequency of analysis (Table 20-Table 24). The largest errors mainly occur in the daily rainfall and propagates to longer durations. These large biases are also reflected in the Nash-Sutcliffe efficiency statistics which are often small and even negative for some models, meaning that the average of the observations is a better fit than the downscaled model. The range of extremes is also smaller in the downscaled models than in the observational dataset as reflected in an increasing absolute bias with return period (not shown) and in the standard deviation ratio (which is on average smaller than 1).

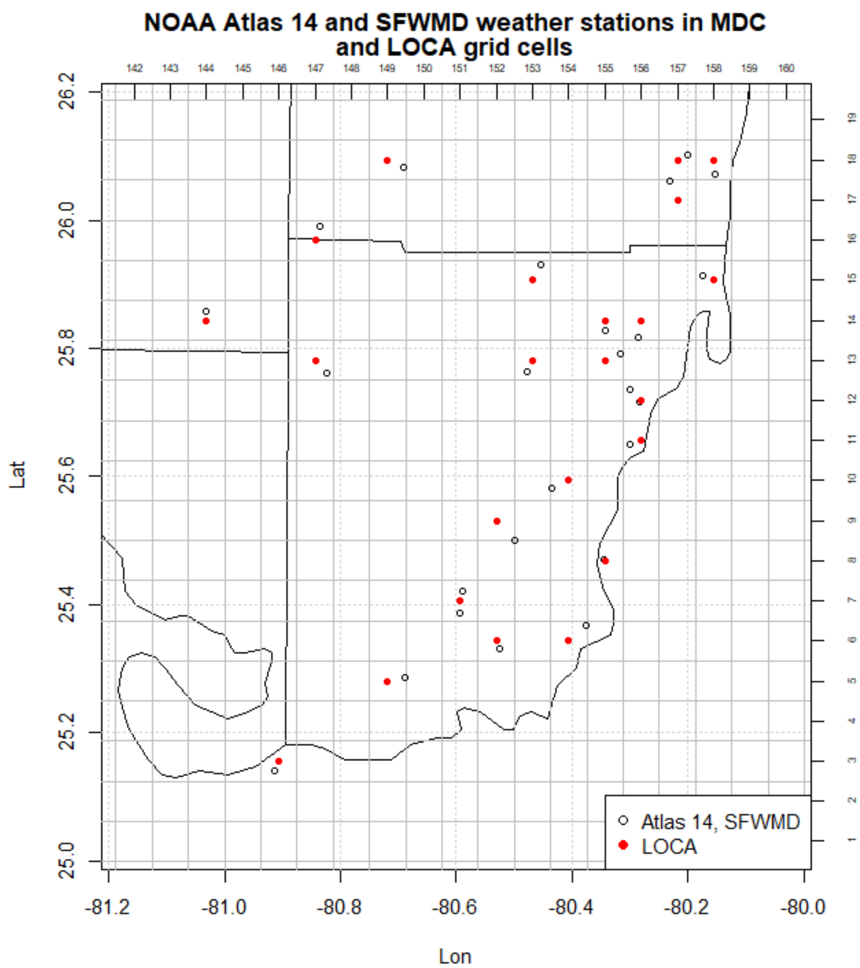


Figure 101. Location of NOAA Atlas 14 and SFWMD weather stations (open circles) and LOCA grid cells (red closed circles).

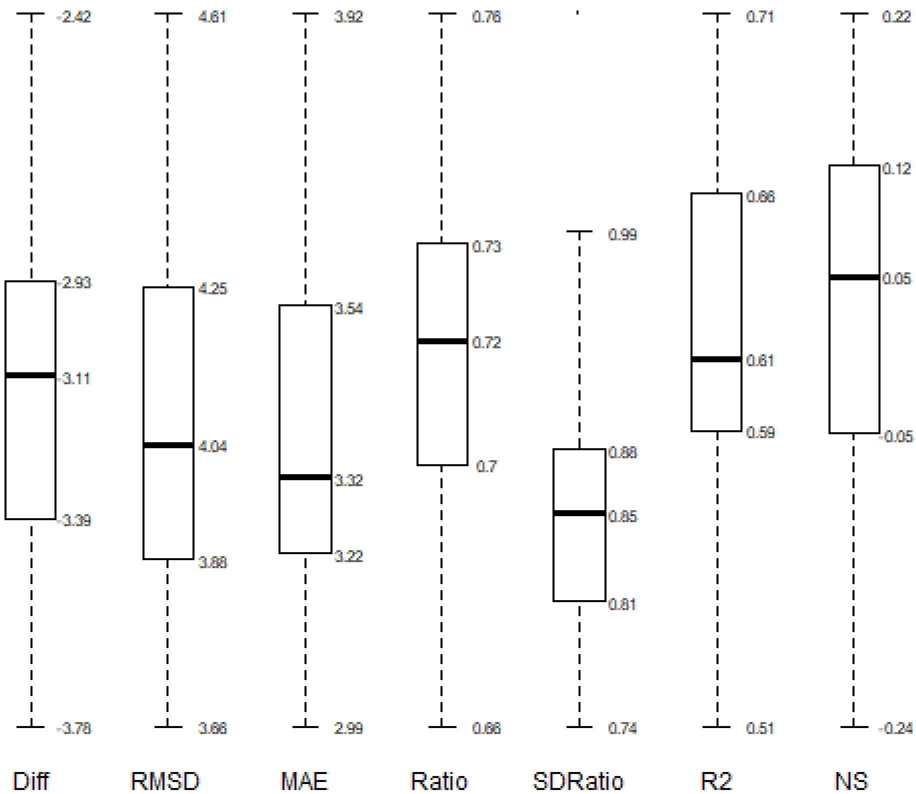


*Figure 102. Boxplot of goodness of fit-statistics for downscaled model versus observed DDF precipitation depth values across models for the current baseline period. Data for all durations and return periods has been pooled together. Box goes from Q1 to Q3 with Q2 in the middle and whiskers extend 1.5*IQR from the box. Bias: Model Bias (inches), RMSE: Root mean square error (inches), MAE: Mean absolute error (inches), Ratio: Average of modeled/observed values, SDRatio: Ratio of modeled to observed standard deviation, R2: Coefficient of determination (R^2), NS: Nash-Sutcliffe Efficiency.*

Figure 103 shows the Taylor diagram (Taylor, 2001) for all downscaled models when all durations and frequencies of interest are combined. All downscaled models behave similarly in terms of pattern correlation, centered root mean square difference, and standard deviation when compared to the observational dataset. When the durations and return periods of interest are analyzed individually (not shown), the spatial pattern correlation is much smaller (generally less than 0.5, and even negative for some models). The apparent improved performance obtained when all durations and return are analyzed is an artifact of lumping data of different magnitudes together.

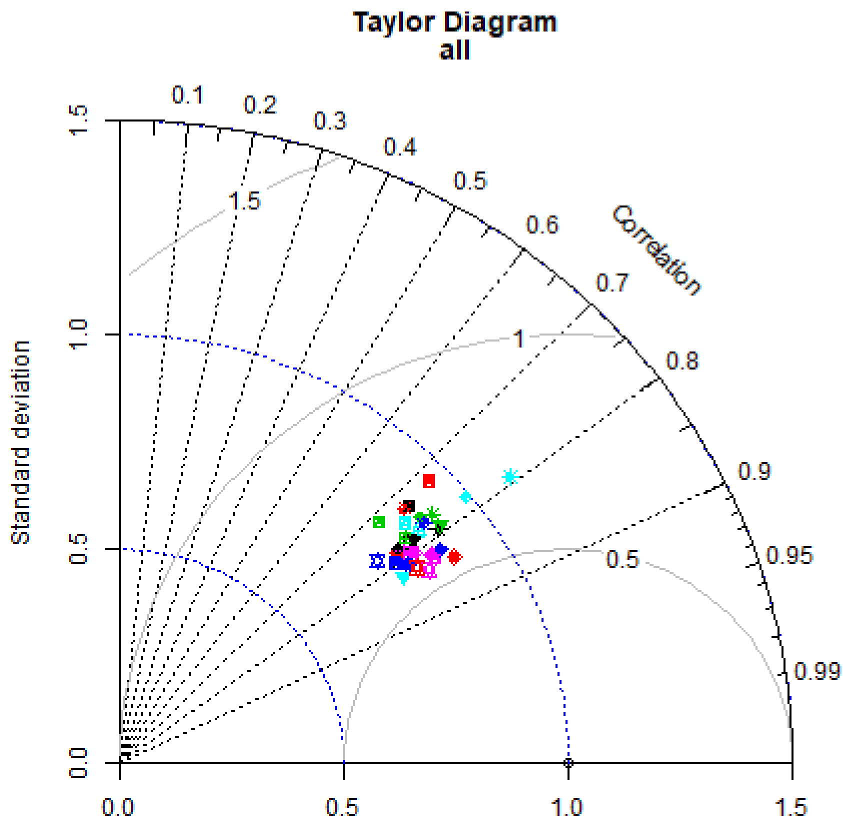


Figure 103. Taylor Diagram for downscaled model versus observed DDF precipitation depth values across all models for the current baseline period. Data for all durations and return periods of interest are pooled together. Light gray curves indicate the centered root mean squared difference between the model and the observations. The observational data is represented by the open-circle marker on the horizontal axis. The closer a model point is to the observational data point, the better the model performance. "In general, the Taylor diagram characterizes the statistical relationship between two fields, a "test" field (often representing a field simulated by a model) and a "reference" field (usually representing "truth", based on observations). Note that the means of the fields are subtracted out before computing their second order statistics, so the diagram does not provide information about overall biases, but solely characterizes the centered pattern error." (Taylor, 2001).

The biases in extremes found in the downscaled models, result from a combination of factors: 1) Comparison of point data in the observational dataset against areal data in the downscaled model dataset (approx. 18.5 mi^2 grid cell resolution), 2) Lack of corrections from constrained to unconstrained extremes in the downscaled model dataset, 3) Actual model biases. Based on Figure 1-5 of U.S. Weather Bureau's Technical Paper 29 (1958), areal reduction factors for an area approximately 18.5 mi^2 (grid cell size) in size and a 24-hour duration would be about 0.98. Based on Table 14, correction factors from constrained to unconstrained annual maximum series would range from 1.12 for 24-hour duration to 1.01 for 7-day duration. Combining these two sets of correction factors would result in model to observation ratios of 0.88 (i.e. 0.98/1.12), 0.94, 0.95, 0.96, 0.97 for 1, 2, 3, 4, and 7-day durations, respectively. These are much larger than most of the computed median ratios (0.50-0.95) between modeled/observed DDF precipitation depths described above, pointing to large actual model biases.

The inability of CMIP5 climate models at capturing (increases in) extreme precipitation in the latter part of the 20th century-early 21st century has been observed by Wuebbles et al. (2014) and Asadieh and Krakauer (2015) for the entire Continental US and North America, respectively. As indicated by SFEC (2016) this underestimation of extremes in the state of Florida is due to a "cold bias in the western Atlantic which limits the deep convection within cold fronts and tropical storms. The Atlantic Warm Pool (AWP) in the simulations is much smaller than observed. Our findings are consistent with Emanuel (2013) concerns with the CMIP models being able to simulate tropical cyclones and Kozar and Misra (2013) concerns with the cold bias in the western Atlantic." Based on our results for the state of Florida, these biases remain even after LOCA's statistical downscaling and general bias-correction of the CMIP5 climate model output.

Two LOCA runs best match the historical extremes for the durations and return periods of interest: GISS-E2-R_r6i1p1 and NorESM1-M_r1i1p1. However, even these two runs underestimate the daily 1-in-100-year event by 37%, but the 7-day 1-in-100-year event is only underestimated by 5-10% in these runs. Interestingly, these two LOCA model runs perform very differently when it comes to annual averages. GISS-E2-R_r6i1p1 underestimates the annual total rainfall in the period 1991-2005 by 6.68 in/year, while NorESM1-M_r1i1p1 underestimates it by just 1.09 in/yr. After bias-correcting, GISS-E2-R_r6i1p1 daily rainfall projections, future average annual rainfall under the RCP45 scenario for the period 2055-2069 is reduced by 2.3 in/yr compared to the observed 1991-2005 average, but rainfall in the RCP85 scenario is increased by 1.93 in/yr. Based on the daily bias-corrected daily NorESM1-M_r1i1p1 output, future annual total rainfall decreases by 7.46 in/yr in the RCP45 scenario and decreases by 10.26 in/yr under the RCP85 scenario. This highlights some of the uncertainties inherent in the statistically downscaled models and the difficulty in selecting output from a single model for planning.

Appendix D. Maps of At-site RFA DDF curves for the downscaled model dataset (LOCA) in the current baseline period (Last 30 years up to 2005) shows maps of the 5th, 50th and 95th percentile of DDF precipitation depths across the downscaled models in the current baseline period for a daily duration and a 100-year return period. The overall spatial pattern is one of higher values in a band going from the northwest corner of the county to the southeast corner. The band of higher extremes slightly inland from the coast that was found in the observational dataset and previous studies is not captured by the downscaled models. Comparison with the DDF maps from observational dataset shows major widespread underestimation of extremes in the downscaled models by as much as 100% in coastal areas of most models. Even the 95th percentile values are much lower than the observed fitted daily 100-year extremes (5-12 in/day vs. 8.7-17.8 in/day in the observational dataset). The MQDM technique will essentially remove these large biases by adjusting the projected future modeled extremes by the bias ratio estimated from the current baseline period.

Table 20. Differences in 24-hr DDF precipitation depths in inches (%) for various return periods for downscaled models (LOCA) versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	24-hr_2-year	24-hr_5-year	24-hr_10-year	24-hr_25-year	24-hr_50-year	24-hr_100-year
5%	-2.2 (-46%)	-3.06 (-46.8%)	-3.73 (-48%)	-4.66 (-49.2%)	-5.43 (-49.8%)	-6.28 (-50.2%)
10%	-2.18 (-45.6%)	-3.01 (-46.3%)	-3.63 (-46.6%)	-4.5 (-47.2%)	-5.23 (-47.7%)	-6.05 (-48.1%)
50%	-2.1 (-43.8%)	-2.88 (-44.1%)	-3.43 (-44%)	-4.26 (-44.9%)	-4.92 (-45%)	-5.64 (-45.2%)
90%	-2.01 (-41.9%)	-2.75 (-42.1%)	-3.27 (-42.2%)	-4 (-41.9%)	-4.67 (-42.4%)	-5.39 (-42.2%)
95%	-1.94 (-40.6%)	-2.65 (-40.9%)	-3.2 (-41.2%)	-3.93 (-41.2%)	-4.34 (-39.4%)	-4.75 (-37.3%)

Table 21. Differences in 2-day DDF precipitation depths in inches (%) for various return periods for downscaled models (LOCA) versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	2-day_2-year	2-day_5-year	2-day_10-year	2-day_25-year	2-day_50-year	2-day_100-year
5%	-1.9 (-33.7%)	-2.63 (-34.1%)	-3.23 (-35.2%)	-4.11 (-36.6%)	-4.85 (-37.3%)	-5.66 (-37.8%)
10%	-1.85 (-32.7%)	-2.6 (-33.8%)	-3.11 (-34%)	-3.9 (-34.6%)	-4.59 (-35.1%)	-5.36 (-35.6%)
50%	-1.7 (-30.1%)	-2.33 (-30.4%)	-2.8 (-30.4%)	-3.53 (-31.2%)	-4.16 (-31.7%)	-4.85 (-32%)
90%	-1.55 (-27.4%)	-2.11 (-27.4%)	-2.56 (-27.7%)	-3.17 (-27.9%)	-3.72 (-27.9%)	-4.32 (-27.8%)
95%	-1.53 (-27.1%)	-2.09 (-27%)	-2.5 (-27.2%)	-3.04 (-26.4%)	-3.28 (-24.2%)	-3.56 (-22.3%)

Table 22. Differences in 3-day DDF precipitation depths in inches (%) for various return periods for downscaled models (LOCA) versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	3-day_2-year	3-day_5-year	3-day_10-year	3-day_25-year	3-day_50-year	3-day_100-year
5%	-1.71 (-27.5%)	-2.36 (-28.1%)	-2.96 (-29.4%)	-3.82 (-30.8%)	-4.55 (-31.6%)	-5.35 (-32.1%)
10%	-1.65 (-26.9%)	-2.35 (-27.8%)	-2.86 (-28.2%)	-3.61 (-28.9%)	-4.26 (-29.2%)	-5.08 (-29.8%)
50%	-1.5 (-24.2%)	-2.04 (-24%)	-2.49 (-24.3%)	-3.16 (-25.1%)	-3.76 (-25.8%)	-4.44 (-25.9%)
90%	-1.32 (-21.2%)	-1.85 (-21.6%)	-2.2 (-21.2%)	-2.69 (-21.1%)	-3.18 (-21.1%)	-3.74 (-21%)
95%	-1.28 (-20.6%)	-1.76 (-20.6%)	-2.14 (-20.9%)	-2.49 (-19.1%)	-2.68 (-17.2%)	-2.85 (-15.1%)

Table 23. Differences in 4-day DDF precipitation depths in inches (%) for various return periods for downscaled models (LOCA) versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	4-day_2-year	4-day_5-year	4-day_10-year	4-day_25-year	4-day_50-year	4-day_100-year
5%	-1.62 (-24.4%)	-2.26 (-25.2%)	-2.84 (-26.4%)	-3.71 (-27.9%)	-4.44 (-28.8%)	-5.26 (-29.3%)
10%	-1.56 (-23.7%)	-2.22 (-24.5%)	-2.72 (-25.1%)	-3.51 (-26.2%)	-4.13 (-26.1%)	-4.98 (-27%)
50%	-1.41 (-21.3%)	-1.94 (-21.4%)	-2.35 (-21.5%)	-2.99 (-22.1%)	-3.55 (-22.7%)	-4.22 (-23.1%)
90%	-1.25 (-18.9%)	-1.73 (-19.2%)	-2.01 (-18%)	-2.53 (-18.3%)	-3.01 (-18.4%)	-3.55 (-18%)
95%	-1.23 (-18.7%)	-1.63 (-17.9%)	-1.99 (-17.9%)	-2.24 (-15.7%)	-2.37 (-13.8%)	-2.48 (-11.6%)

Table 24. Differences in 7-day DDF precipitation depths in inches (%) for various return periods for downscaled models (LOCA) versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	7-day_2-year	7-day_5-year	7-day_10-year	7-day_25-year	7-day_50-year	7-day_100-year
5%	-1.64 (-20.9%)	-2.34 (-22%)	-2.88 (-22.6%)	-3.76 (-24%)	-4.54 (-24.9%)	-5.41 (-25.7%)
10%	-1.58 (-20.1%)	-2.26 (-21.2%)	-2.84 (-22.4%)	-3.67 (-23.3%)	-4.36 (-24.2%)	-5.25 (-25.4%)
50%	-1.43 (-18.3%)	-1.94 (-18%)	-2.46 (-19.1%)	-3.05 (-19%)	-3.64 (-19.5%)	-4.36 (-20.1%)
90%	-1.26 (-16%)	-1.74 (-16.1%)	-1.99 (-15.3%)	-2.55 (-15.8%)	-3.06 (-16.1%)	-3.66 (-15.7%)
95%	-1.23 (-15.7%)	-1.66 (-15.4%)	-1.95 (-14.9%)	-2.13 (-12.6%)	-2.21 (-10.7%)	-2.24 (-8.4%)

At-Site RFA for the downscaled model dataset in the future projection period

A set of F_{m-p}(model projected/future CDF) curves were derived by applying At-site RFA to model-derived AMS data at each of the 26 weather station locations in Miami-Dade County for the 30-year period centered around 2065 (F_{m-p1}: 2050-2079). Maps of the 5th, 50th and 95th percentile of fitted precipitation extremes across the downscaled models in the period centered at 2065 for a daily duration and a 100-year return period are shown in Appendix E. Maps of At-site RFA DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079). Maps of the changes in extremes with respect to the fitted extremes for the downscaled models in the current baseline period (last 30 years up to 2005) are also included.

Table 25-Table 29 show the changes in precipitation depths in inches (%) for the durations and return periods of interest. It can be noticed that although some models predict increases in extremes, others predict decreases. Figure 104 shows that about half the of the models predict negative overall changes in extremes in the future period centered in 2065 compared to the current baseline period. However, the predicted increases are generally larger than the predicted decreases. The median changes are less than 0.5 inches for the county as a whole for the durations and return periods analyzed.

Percentage-wise, the predicted changes are more significant and these ratios are the foundation for the MQDM bias-correction method. Based on the 5th percentile, median, and 95th percentile of all models, the 24 hour/100-year precipitation depth is expected to change by -14.5%, +5.7%, and +44.7%. Comparing Table 25-Table 29 and Table 20-Table 24, it is evident that the projected changes in extremes are generally smaller than the biases in the downscaled models. For example, the projected changes in the 24-hour/100-year events range from -1.0 to +2.5 inches, whereas the corresponding downscaled model biases are on the order of -4.8 to -6.3 inches. Therefore, these relatively large percentage changes will have a significant impact on the adjusted 2065 extremes after MQDM implementation.

Table 25. Changes in 24-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus the current baseline period. 5-95th percentiles across models shown.

Perc.	24-hr_2-year	24-hr_5-year	24-hr_10-year	24-hr_25-year	24-hr_50-year	24-hr_100-year
5%	-0.42 (-16.1%)	-0.45 (-12.6%)	-0.52 (-12.1%)	-0.57 (-11%)	-0.74 (-10.6%)	-1.04 (-14.5%)
10%	-0.29 (-10.4%)	-0.37 (-10.5%)	-0.44 (-9.8%)	-0.5 (-8.7%)	-0.62 (-9.7%)	-0.83 (-10.1%)
50%	-0.01 (0.2%)	0.03 (1.6%)	0.01 (1.8%)	0.06 (2.5%)	0.14 (3.9%)	0.2 (5.7%)
90%	0.29 (11.8%)	0.5 (15%)	0.77 (19.9%)	1.22 (25%)	1.45 (27.2%)	1.65 (29.3%)
95%	0.35 (13.6%)	0.69 (19.6%)	0.91 (21.9%)	1.3 (28.3%)	1.83 (36.6%)	2.52 (44.7%)

Table 26. Changes in 2-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus the current baseline period. 5-95th percentiles across models shown.

Perc.	2-day_2-year	2-day_5-year	2-day_10-year	2-day_25-year	2-day_50-year	2-day_100-year
5%	-0.58 (-14.4%)	-0.67 (-13.4%)	-0.76 (-12.1%)	-0.99 (-11.8%)	-1.19 (-11.1%)	-1.63 (-13.6%)
10%	-0.48 (-12.8%)	-0.58 (-10.4%)	-0.73 (-10.7%)	-0.86 (-10.6%)	-1 (-10.3%)	-1.31 (-10.7%)
50%	-0.06 (-1%)	-0.02 (0.3%)	0.01 (1.6%)	0.08 (2.2%)	0.18 (3.9%)	0.19 (5.2%)
90%	0.42 (11.2%)	0.78 (16%)	1.16 (20.1%)	1.9 (26.5%)	2.18 (28.7%)	2.42 (31.2%)
95%	0.5 (14.6%)	1.1 (21.1%)	1.43 (23.5%)	1.96 (30%)	2.8 (38.6%)	3.84 (46.2%)

Table 27. Changes in 3-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus the current baseline period. 5-95th percentiles across models shown.

Perc.	3-day_2-year	3-day_5-year	3-day_10-year	3-day_25-year	3-day_50-year	3-day_100-year
5%	-0.68 (-14.7%)	-0.79 (-12.9%)	-0.88 (-12.2%)	-1.13 (-11.9%)	-1.41 (-11.4%)	-1.81 (-13.8%)
10%	-0.57 (-12.9%)	-0.67 (-10.8%)	-0.76 (-9.7%)	-0.99 (-9.4%)	-1.29 (-10.2%)	-1.66 (-10.9%)
50%	-0.02 (0%)	-0.02 (0.5%)	-0.03 (0.3%)	0.02 (1.2%)	0.13 (3.2%)	0.07 (5.4%)
90%	0.51 (12.1%)	0.93 (15.9%)	1.41 (20.4%)	2.2 (26.8%)	2.51 (28.1%)	2.97 (32.3%)
95%	0.6 (14%)	1.27 (22.9%)	1.78 (25.3%)	2.47 (31%)	3.4 (39.8%)	4.66 (46.2%)

Table 28. Changes in 4-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus the current baseline period. 5-95th percentiles across models shown.

Perc.	4-day_2-year	4-day_5-year	4-day_10-year	4-day_25-year	4-day_50-year	4-day_100-year
5%	-0.75 (-14.8%)	-0.88 (-13.3%)	-0.98 (-11.7%)	-1.15 (-11.7%)	-1.76 (-13.3%)	-2.05 (-14.9%)
10%	-0.62 (-12%)	-0.76 (-10.7%)	-0.86 (-10%)	-1.07 (-9.5%)	-1.34 (-10.4%)	-1.81 (-12%)
50%	-0.05 (-0.7%)	-0.02 (0.3%)	-0.01 (0.5%)	-0.01 (1.2%)	0.13 (3%)	0.09 (5%)
90%	0.48 (10.6%)	1 (15.1%)	1.58 (20.1%)	2.2 (24.2%)	2.76 (26.7%)	3.38 (30.7%)
95%	0.66 (13.5%)	1.27 (19.7%)	1.78 (22.5%)	2.78 (29.6%)	3.57 (37%)	4.84 (42.6%)

Table 29. Changes in 7-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus the current baseline period. 5-95th percentiles across models shown.

Perc.	7-day_2-year	7-day_5-year	7-day_10-year	7-day_25-year	7-day_50-year	7-day_100-year
5%	-0.86 (-13.9%)	-1.08 (-12.8%)	-1.24 (-12.4%)	-1.52 (-13.1%)	-2.32 (-14.9%)	-2.9 (-15.9%)
10%	-0.67 (-10.6%)	-0.94 (-11.2%)	-1.03 (-10.2%)	-1.37 (-10.6%)	-1.78 (-11.9%)	-2.24 (-13.6%)
50%	-0.08 (-0.8%)	-0.05 (0.5%)	0 (0.2%)	0.14 (1.7%)	0.24 (3.7%)	0.38 (5.9%)
90%	0.46 (8.6%)	1.1 (14.4%)	1.71 (18.3%)	2.77 (24%)	3.52 (28.6%)	4.41 (31.5%)
95%	0.78 (12.9%)	1.39 (18.3%)	2.21 (22.6%)	3.17 (28%)	4.33 (34%)	5.51 (39.6%)

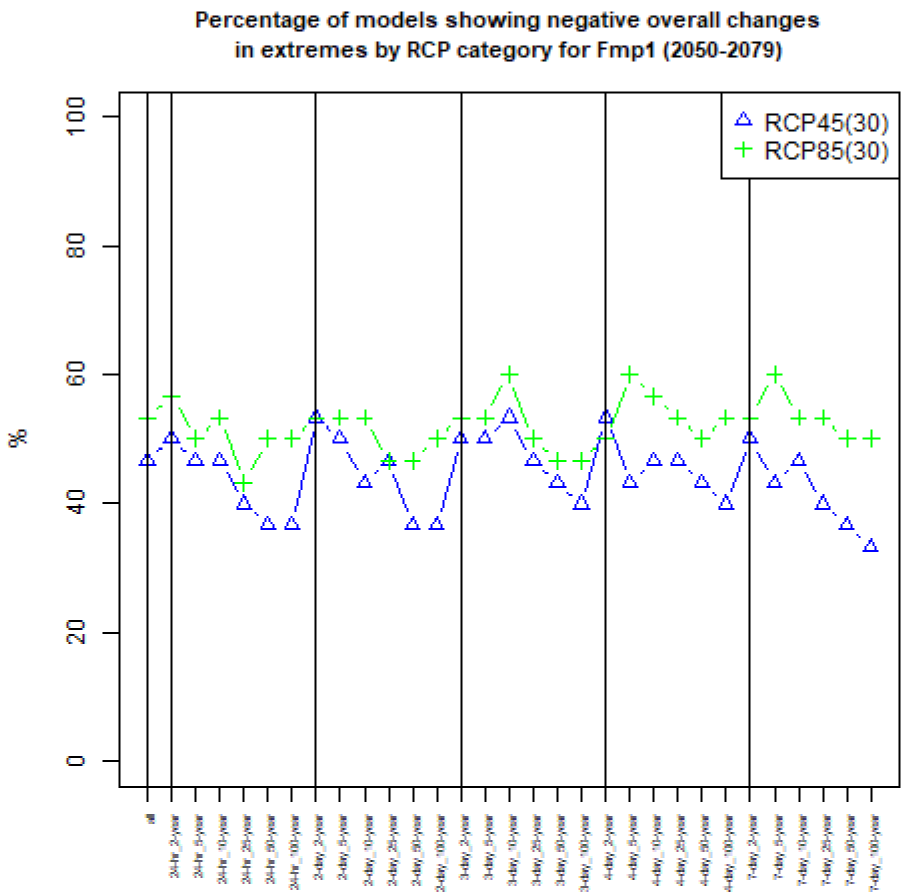


Figure 104. Percentage of downscaled models showing negative overall changes in extreme by RCP category for the future period centered at 2065. There is a total of 30 models for RCP45 and 30 for RCP85.

MQDM implementation to bias-correct the downscaled dataset for the future projection period

As shown in the previous sections, changes in extremes projected by the downscaled models for the future period 2050-2079 (centered in 2065) are smaller than the downscaled model biases (i.e. the trend signal is much smaller than the bias). This reduces the confidence in the projected changes in extremes even after adjustment with MQDM.

An unexpected issue was encountered when using MQDM to adjust projected DDF depth values for the two future periods of interest. It was found that about 1.5% of the adjusted curves ($F_{m-p\ adj.1}$) decrease with increasing return period. Further investigation showed that the issue arises more frequently when a negative shape parameter (and hence, a short-tail) for F_{m-p} and/or F_{o-c} is combined with a positive shape parameter (and hence, a long-tail) for F_{m-c}. This is illustrated in Figure 105 and can be confirmed from inspection of Equation 15-Equation 16. A semi-parametric method (SPM) was employed in order to smooth out the DDF curves and make them more consistent (see Irizarry et al., 2016 for more details). Figure 106 shows an example of the adjusted DDF curves after SPM smoothing.

Appendix F. Maps of adjusted DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079) shows maps of the 5th, 50th and 95th percentile of *adjusted* future DDF precipitation depths across the downscaled models for a daily duration and a 100-year return period for the future period centered at 2065 ($F_{m-padj.1}$: 2050-2079). Table 30-Table 34 show the changes in precipitation depths in inches (%) with respect to the observations for the current baseline period for the durations and return periods of interest. As expected, the percentage changes (i.e. ratios) are very close to those for the unadjusted series (Table 25-Table 29). Small differences are possibly due to the use of SPM at some stations as described above. As expected from Table 25-Table 29 and the large model biases, the 5th percentile of adjusted future precipitation is generally lower than the observed current baseline values, the 50th percentile is slightly higher for the longer return periods, while larger positive changes are estimated for the 95th percentile at all stations especially for the less frequent events. The median changes are less than 1 inch for the county as a whole for the durations and return periods analyzed.

Comparison of (Appendix F. Maps of adjusted DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079)) against the maps of unadjusted DDF depths for the same period (Appendix E. Maps of At-site RFA DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079)) shows that the daily depths nearly doubled due to the adjustments made by MQDM. Similarly, the differences against the appropriate current baseline values (F_{m-c} in the case of the unadjusted values; F_{o-c} in the case of the adjusted values) also doubled especially the highest differences. The spatial pattern now more closely resembles that of the observed current baseline (F_{o-c}) DDF maps in (Appendix C. Maps of At-site RFA parameters and DDF curves for the observational dataset in the current baseline period (Last 30 years up to 2005)). In particular, the band of higher values slightly inland from the coast is back.

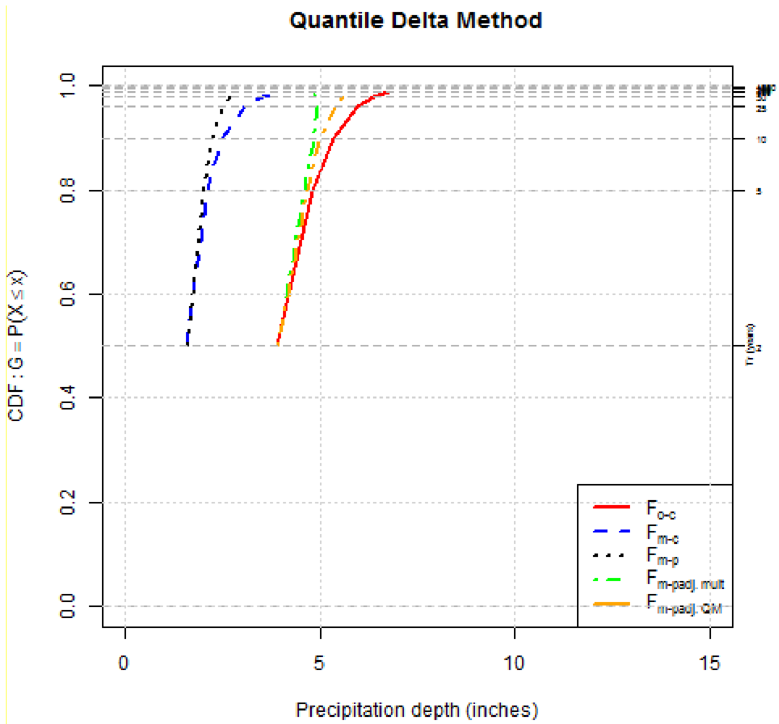


Figure 105. Decreasing adjusted quantiles with return period can result when using MQDM under certain shape parameter combinations. In such cases, QM could fix the issue the vast majority of the time.

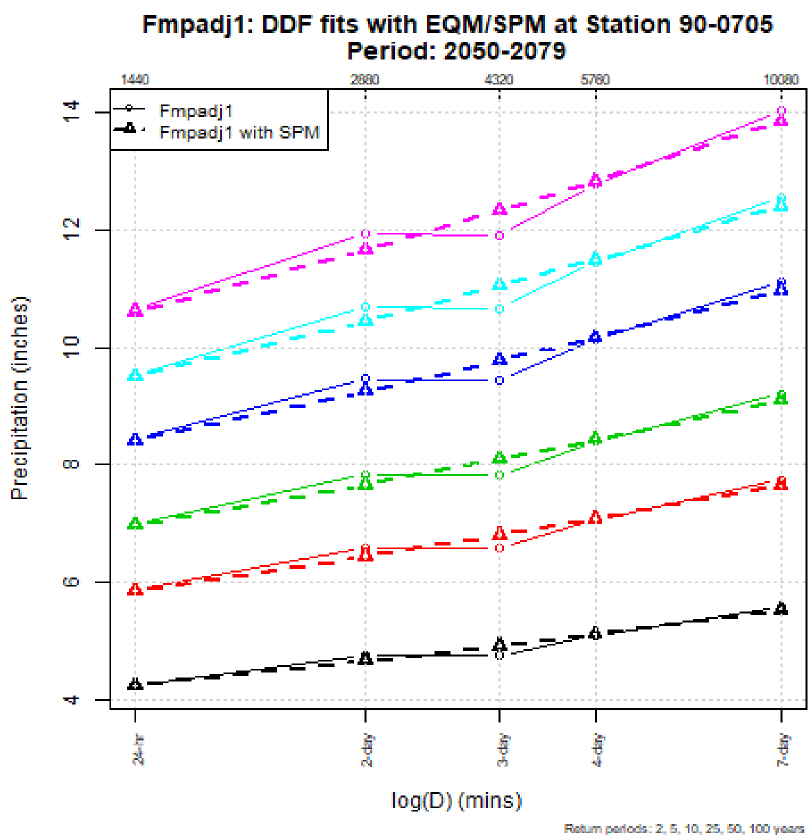


Figure 106. SPM smoothing of adjusted projected DDF curves for a sample station and model.

Table 30. Changes in adjusted 24-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	24-hr_2-year	24-hr_5-year	24-hr_10-year	24-hr_25-year	24-hr_50-year	24-hr_100-year
5%	-0.75 (-16.1%)	-0.83 (-12.6%)	-0.95 (-12.1%)	-1.03 (-10.8%)	-1.1 (-10.2%)	-1.4 (-12.8%)
10%	-0.51 (-10.6%)	-0.65 (-10.5%)	-0.74 (-9.8%)	-0.81 (-8.7%)	-0.91 (-9.6%)	-1.12 (-9.5%)
50%	0.02 (0.2%)	0.09 (1.4%)	0.12 (1.8%)	0.26 (2.5%)	0.46 (3.9%)	0.85 (5.7%)
90%	0.57 (11.7%)	0.97 (15%)	1.56 (20%)	2.32 (25%)	2.9 (27.2%)	3.58 (29.3%)
95%	0.64 (13.6%)	1.25 (19.6%)	1.67 (21.9%)	2.71 (28.4%)	3.89 (36.6%)	5.34 (44.8%)

Table 31. Changes in adjusted 2-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	2-day_2-year	2-day_5-year	2-day_10-year	2-day_25-year	2-day_50-year	2-day_100-year
5%	-0.8 (-14.4%)	-1.01 (-13.2%)	-1.12 (-12%)	-1.33 (-11.8%)	-1.43 (-11.1%)	-1.65 (-12.2%)
10%	-0.72 (-12.6%)	-0.8 (-10.4%)	-0.95 (-10.7%)	-1.12 (-10.6%)	-1.21 (-9.8%)	-1.5 (-10.4%)
50%	-0.04 (-1%)	0.03 (0.3%)	0.14 (1.6%)	0.31 (2.2%)	0.59 (4%)	1.03 (5.2%)
90%	0.6 (11.1%)	1.22 (16%)	1.84 (20.1%)	2.89 (26.4%)	3.59 (28.6%)	4.32 (30.9%)
95%	0.83 (14.3%)	1.58 (21.1%)	2.1 (23.5%)	3.31 (30%)	4.79 (38.6%)	6.4 (46.1%)

Table 32. Changes in adjusted 3-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	3-day_2-year	3-day_5-year	3-day_10-year	3-day_25-year	3-day_50-year	3-day_100-year
5%	-0.89 (-14.5%)	-1.07 (-12.9%)	-1.21 (-12.2%)	-1.36 (-11.5%)	-1.6 (-10.9%)	-2.01 (-13.1%)
10%	-0.78 (-12.7%)	-0.86 (-10.8%)	-0.92 (-9.7%)	-1.13 (-9.2%)	-1.27 (-9.9%)	-1.54 (-10.1%)
50%	0.02 (0%)	0.05 (0.4%)	0.03 (0.3%)	0.28 (1.8%)	0.53 (3.2%)	0.88 (5.5%)
90%	0.7 (11.3%)	1.29 (15.7%)	1.99 (20.4%)	3.24 (26.8%)	3.84 (27.9%)	4.92 (32.1%)
95%	0.86 (14%)	1.87 (22.9%)	2.51 (25.3%)	3.73 (31%)	5.38 (39.8%)	6.98 (46.2%)

Table 33. Changes in adjusted 4-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	4-day_2-year	4-day_5-year	4-day_10-year	4-day_25-year	4-day_50-year	4-day_100-year
5%	-0.96 (-14.6%)	-1.16 (-13.1%)	-1.24 (-11.7%)	-1.4 (-11.3%)	-1.86 (-11.8%)	-2.43 (-14.1%)
10%	-0.76 (-12%)	-0.93 (-10.7%)	-1.03 (-10%)	-1.13 (-9%)	-1.45 (-9.7%)	-1.74 (-11%)
50%	-0.02 (-0.6%)	0.04 (0.3%)	0.03 (0.4%)	0.29 (1.8%)	0.54 (3.3%)	0.85 (5.1%)
90%	0.7 (10.7%)	1.32 (15%)	2.11 (20.1%)	3.18 (24.6%)	3.97 (27.1%)	5.08 (31.2%)
95%	0.89 (13.5%)	1.74 (19.9%)	2.45 (23.1%)	3.79 (29.6%)	5.34 (37.2%)	6.93 (42.8%)

Table 34. Changes in adjusted 7-day DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	7-day_2-year	7-day_5-year	7-day_10-year	7-day_25-year	7-day_50-year	7-day_100-year
5%	-1.09 (-14.1%)	-1.34 (-12.8%)	-1.55 (-12.4%)	-1.91 (-12.8%)	-2.24 (-13.4%)	-3.04 (-15.9%)
10%	-0.81 (-10.6%)	-1.14 (-11.1%)	-1.26 (-10.1%)	-1.47 (-10.2%)	-1.94 (-11.6%)	-2.42 (-12.9%)
50%	-0.05 (-0.8%)	0.08 (0.5%)	0.07 (0.2%)	0.25 (1.4%)	0.74 (3.5%)	1.07 (5.6%)
90%	0.66 (8.4%)	1.55 (14.4%)	2.3 (18.3%)	3.59 (23.9%)	5.01 (28.6%)	6.22 (31.5%)
95%	0.99 (12.9%)	1.96 (18.2%)	2.76 (22.4%)	4.2 (28%)	5.75 (34%)	7.56 (39.5%)

Quantile mapping for temporal downscaling of precipitation projections

A modified version of the Quantile mapping (QM) equation can be used to temporally downscale extremes from daily to sub-daily durations based on the historical fitted DDF curve for the sub-daily duration of interest. The approach is similar to the methodology employed by Tetratech (2015).

$$\hat{x}_{m-padj.subdaily} = F^{-1}_{o-c\;subdaily}\left[F^{*}_{m-c\;daily}\left(\hat{x}_{m-padj.daily}\right)\right]$$

Equation 17

where $\hat{x}_{m-padj.subdaily}$ is the subdaily adjusted quantile for the model (m) projections (p) for the future period, $F_{o-c\;subdaily}$ is the CDF of the daily observations (o) in the current baseline period (c) , $F^{*}_{m-c\;daily}$ is the *bias-corrected* CDF of the daily model (m) data in the current baseline period (c) , $\hat{x}_{m-padj.daily}$ is the daily adjusted quantile for the model (m) projections (p) for the future. F^{-1} means the inverse of the CDF (i.e. the quantile function). The CDFs are developed based on data spanning decades and centered around some year of interest. It is important to note that the bias-corrected CDF of the daily model (m) data in the current baseline period (c), $F^{*}_{m-c\;daily}$, equals the CDF of the daily observations (o) in the current baseline period (o), $F_{o-c\;daily}$; therefore, Equation 17 can be re-written as:

$$\hat{x}_{m-padj.subdaily} = F^{-1}_{o-c\;subdaily}\left[F_{o-c\;daily}\left(\hat{x}_{m-padj.daily}\right)\right]$$

Equation 18

Equation 18 was applied to the adjusted future daily model quantiles for the return periods of interest to obtain sub-daily quantiles for durations of 1, 2, 3, 6, and 12 hours at the stations with hourly AMS data.

Appendix F. Maps of adjusted DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079) shows maps of the 5th, 50th and 95th percentile of *adjusted* future DDF precipitation depths across the downscaled models for hourly duration and a 100-year return period for the future period centered at 2065 ($F_{m-padj.1}$: 2050-2079). Table 35-Table 39 show the changes in precipitation depths in inches (%) with respect to the observations for the current baseline period for the sub-daily durations and return periods of interest. As expected, the percentage changes are very close to those for the daily durations from which the sub-daily extremes were derived (Table 30). The 5th percentile of overall adjusted future precipitation is generally lower than the observed current baseline values, the 50th percentile is slightly higher for the longer return periods, while larger positive changes are estimated for the 95th percentile at all stations especially for the less frequent events. The median changes are less than 1 inch for the county as a whole for the durations and return periods analyzed.

Table 35. Changes in adjusted 1-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	60-min_2-year	60-min_5-year	60-min_10-year	60-min_25-year	60-min_50-year	60-min_100-year
5%	-0.37 (-17.5%)	-0.37 (-12.8%)	-0.42 (-12.3%)	-0.47 (-11.4%)	-0.53 (-11.3%)	-0.75 (-13.3%)
10%	-0.22 (-10.5%)	-0.32 (-11.3%)	-0.38 (-10.8%)	-0.37 (-8.8%)	-0.46 (-9.4%)	-0.58 (-11.1%)
50%	0 (0.1%)	0.04 (1.5%)	0.11 (3%)	0.17 (3.5%)	0.26 (5.4%)	0.39 (7.1%)
90%	0.27 (12.8%)	0.45 (15.5%)	0.67 (19.8%)	1.09 (25.9%)	1.32 (27.8%)	1.67 (30.9%)
95%	0.32 (14.8%)	0.65 (22.6%)	0.86 (25%)	1.3 (31.1%)	1.78 (37.5%)	2.34 (43.8%)

Table 36. Changes in adjusted 2-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	2-hr_2-year	2-hr_5-year	2-hr_10-year	2-hr_25-year	2-hr_50-year	2-hr_100-year
5%	-0.46 (-17.5%)	-0.46 (-12.8%)	-0.52 (-12.3%)	-0.6 (-11.4%)	-0.68 (-11.3%)	-0.93 (-13.3%)
10%	-0.28 (-10.5%)	-0.4 (-11.3%)	-0.47 (-10.8%)	-0.47 (-8.8%)	-0.58 (-9.4%)	-0.74 (-11.1%)
50%	0.01 (0.1%)	0.04 (1.5%)	0.13 (3%)	0.22 (3.5%)	0.33 (5.4%)	0.5 (7.1%)
90%	0.34 (12.8%)	0.55 (15.5%)	0.83 (19.8%)	1.35 (25.9%)	1.63 (27.8%)	2.06 (30.9%)
95%	0.4 (14.8%)	0.8 (22.6%)	1.07 (25%)	1.62 (31.1%)	2.21 (37.5%)	2.9 (43.8%)

Table 37. Changes in adjusted 3-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	3-hr_2-year	3-hr_5-year	3-hr_10-year	3-hr_25-year	3-hr_50-year	3-hr_100-year
5%	-0.5 (-17.5%)	-0.5 (-12.8%)	-0.57 (-12.3%)	-0.65 (-11.4%)	-0.74 (-11.3%)	-1.02 (-13.3%)
10%	-0.3 (-10.5%)	-0.44 (-11.3%)	-0.52 (-10.8%)	-0.52 (-8.8%)	-0.64 (-9.4%)	-0.8 (-11.1%)
50%	0.01 (0.1%)	0.05 (1.5%)	0.15 (3%)	0.24 (3.5%)	0.37 (5.4%)	0.56 (7.1%)
90%	0.37 (12.8%)	0.6 (15.5%)	0.91 (19.8%)	1.49 (25.9%)	1.78 (27.8%)	2.26 (30.9%)
95%	0.44 (14.8%)	0.89 (22.6%)	1.18 (25%)	1.78 (31.1%)	2.42 (37.5%)	3.17 (43.8%)

Table 38. Changes in adjusted 6-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	6-hr_2-year	6-hr_5-year	6-hr_10-year	6-hr_25-year	6-hr_50-year	6-hr_100-year
5%	-0.59 (-17.5%)	-0.59 (-12.8%)	-0.67 (-12.3%)	-0.75 (-11.4%)	-0.84 (-11.3%)	-1.19 (-13.3%)
10%	-0.36 (-10.5%)	-0.52 (-11.3%)	-0.6 (-10.8%)	-0.6 (-8.8%)	-0.74 (-9.4%)	-0.91 (-11.1%)
50%	0.01 (0.1%)	0.06 (1.5%)	0.17 (3%)	0.27 (3.5%)	0.44 (5.4%)	0.69 (7.1%)
90%	0.44 (12.8%)	0.7 (15.5%)	1.08 (19.8%)	1.74 (25.9%)	2.08 (27.8%)	2.66 (30.9%)
95%	0.51 (14.8%)	1.05 (22.6%)	1.38 (25%)	2.08 (31.1%)	2.83 (37.5%)	3.71 (43.8%)

Table 39. Changes in adjusted 12-hr DDF precipitation depths in inches (%) for various return periods for the future period centered at 2065 versus observations in the current baseline period. 5-95th percentiles across models shown.

Perc.	12-hr_2-year	12-hr_5-year	12-hr_10-year	12-hr_25-year	12-hr_50-year	12-hr_100-year
5%	-0.68 (-17.5%)	-0.68 (-12.8%)	-0.77 (-12.3%)	-0.85 (-11.4%)	-0.95 (-11.3%)	-1.36 (-13.3%)
10%	-0.41 (-10.5%)	-0.6 (-11.3%)	-0.69 (-10.8%)	-0.7 (-8.8%)	-0.83 (-9.4%)	-1.04 (-11.1%)
50%	0.01 (0.1%)	0.07 (1.5%)	0.2 (3%)	0.3 (3.5%)	0.53 (5.4%)	0.81 (7.1%)
90%	0.51 (12.8%)	0.8 (15.5%)	1.25 (19.8%)	2.02 (25.9%)	2.42 (27.8%)	3.08 (30.9%)
95%	0.59 (14.8%)	1.21 (22.6%)	1.6 (25%)	2.4 (31.1%)	3.27 (37.5%)	4.28 (43.8%)

References

Asadieh B., Krakauer N. Y. 2015. Global trends in extreme precipitation: climate models versus observations. Hydrol. Earth Syst. Sci. **19**: 877–891.

Ayuso-Muñoz J. L., Ayuso-Ruiz P., Taguas E. V., García-Marín A. P., Estévez J., Pizarro-Tapia R. 2015. A more efficient rainfall intensity-duration-frequency relationship by using an at-site regional frequency analysis: Application at mediterranean climate locations. *Water Resources Management* **29**:3243–3263. DOI:10.1007/s11269-015-0993-z.

Coles S. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Springer-Verlag; 227.

Emanuel K. A. 2013: Downscaling CMIP5 climate models shows increased tropical cyclone activity over the 21st century. Proc. Nat. Acad. Sci. **110**. doi:10.1073/pnas.1301293110.

Fisher R. A., Tippett L. H. C. 1928. Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Proceedings of the Cambridge Philosophical Society* **24**:189-190.

Grubbs, F. E. 1950. Sample Criteria for testing outlying observations. Ann. Math. Stat. 21, 1, 27-58.

Irizarry M., Obeysekera J., Dessalegne T. 2016. Determination of Future Intensity-Duration-Frequency Curves for Level of Service Planning Projects. Task 2 - Deliverable 2.1 Conduct an extreme rainfall analysis in climate model outputs to determine temporal changes in IDF curves. Prepared for South Florida Water Management District.

Jenkinson A. F. 1955. The frequency distribution of the annual maximum (or minimum) of meteorological elements. *Quarterly Journal of the Royal Meteorological Society* **81**:158-171.

Katz R. W., Parlange M. B., Naveau P. 2002. Statistics of extremes in hydrology. *Advances in Water Resources* **25**: 1287–1304.

Kozar M., Misra V. 2013, Evaluation of twentieth-century Atlantic Warm Pool simulations in historical CMIP5 Runs. Climate Dynamics **41**(9/10):2375.

NOAA. 2013. Precipitation-Frequency Atlas of the United States, Southeastern States. NOAA Atlas 14 Volume 9 Version 2.0, S. Perica, D. Martin, S. Pavlovic, I. Roy, M. St. Laurent, C. Trypaluk, D. Unruh, M. Yekta, G. Bonnin, NOAA, National Weather Service, Silver Spring, MD.

Panofsky H. W., Brier G. W. 1968. *Some Applications of Statistics to Meteorology.* The Pennsylvania State University Press: Philadelphia.

Pierce D. W., Cayan D. R., Thrasher B. L. 2014. Statistical downscaling using Localized Constructed Analogs (LOCA). Journal of Hydrometeorology **15**:2558–2585.

South Florida Engineering and Consulting, LLC. 2016. Task 3 Report – Review of future rainfall extremes. Estimating the impact of projected global warming on magnitude and frequency of extreme rainfall events within the South Florida Water Management District.

Taylor K. E. 2001. Summarizing multiple aspects of model performance in a single diagram. J. Geophys. Res. **106**: 7183-7192.

Tetratech. 2015. Hydrologic Design Standards under Future Climate for Grand Rapids, Michigan. Prepared by Tetratech for the City of Grand Rapids, Michigan.

Wuebbles D. et al. 2014. CMIP5 climate model analyses. Climate extremes in the United States. Bulletin of the American Meteorological Society: 571-583.

Appendix A. C-shell script to run runivg program on the SFWMD network

```
#!/bin/csh -f

foreach station_id (`cat rfmiami.dat`)
  echo "Working on ...." $station_id
  if (-e tempfil ) /bin/rm tempfil
  if (-e ${station_id}_hourly.dat ) /bin/rm ${station_id}_hourly.dat
  touch tempfil
  echo 1 > tempfil
  echo $station_id >> tempfil
  echo H >> tempfil
  echo 199105222300","201904162300 >> tempfil
  echo SUM >> tempfil
  echo ${station_id}_hourly.dat >> tempfil

  echo "Calling runivg"
  #runivg at: /k_wmp/ka_db/dcvp/prod/bin/runivg
  runivg < tempfil

  echo "Done with runivg for ..." $station_id

end
```

Contents of rfmiami.dat:

```
3AS3W3+R
JBTS+R
MBTS+R
MDTS+R
MIAMI+R
S123-R
S12D+R
S177-R
S179-R
S18C-R
S20F-R
S21A-R
S21-R
S26-R
S27-R
S28Z-R
S29Z-R
S29-R
S30-R
S331W+R
S332-R
S334-R
S335-R
S336+R
S338-R
```

Appendix B. Contour maps of 1-in-100-year rainfall depths based on historical data

(Units: inches)

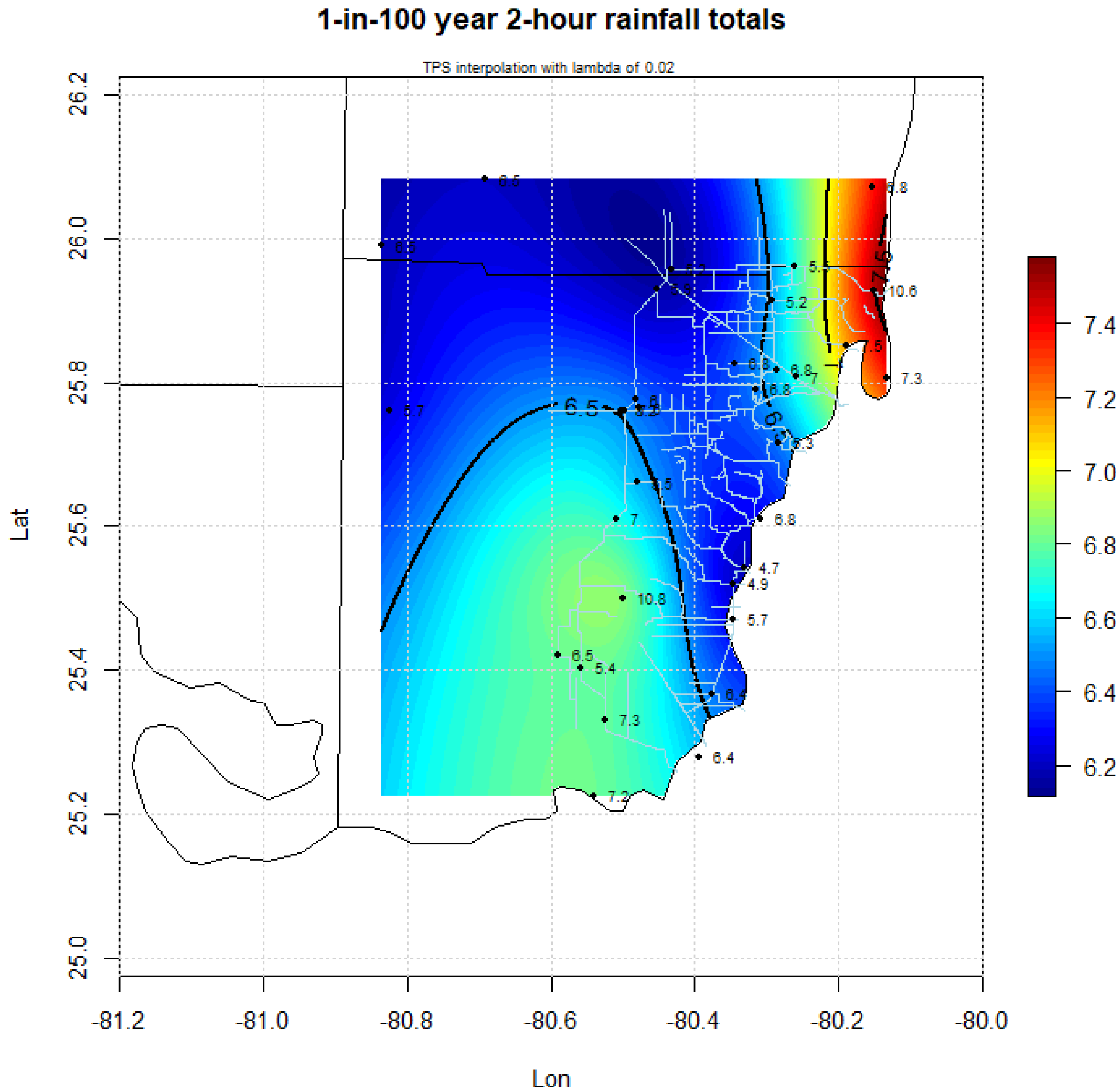


Figure 107. Interpolated 1-in-100-year 2-hour rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

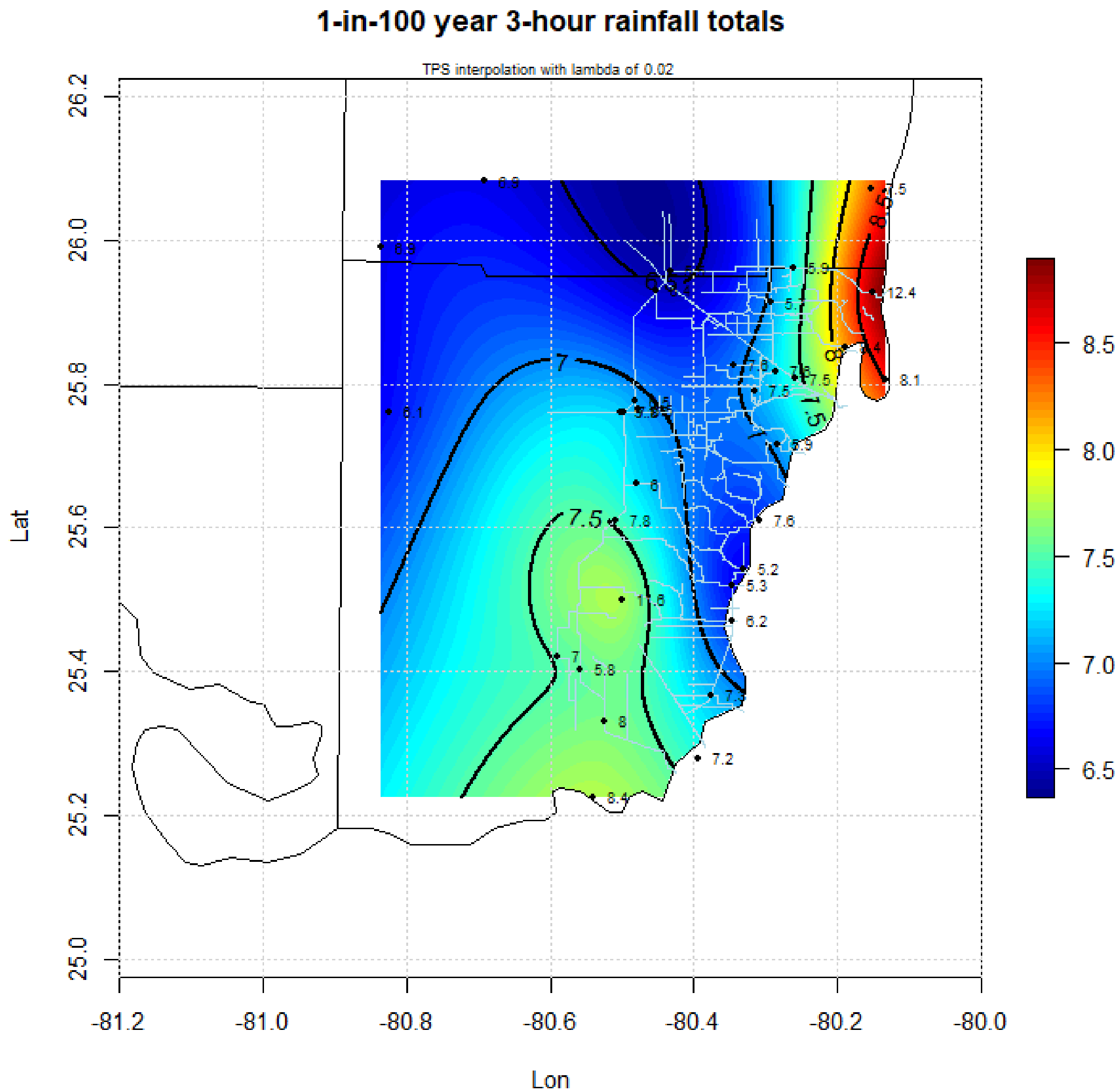


Figure 108. Interpolated 1-in-100-year 3-hour rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

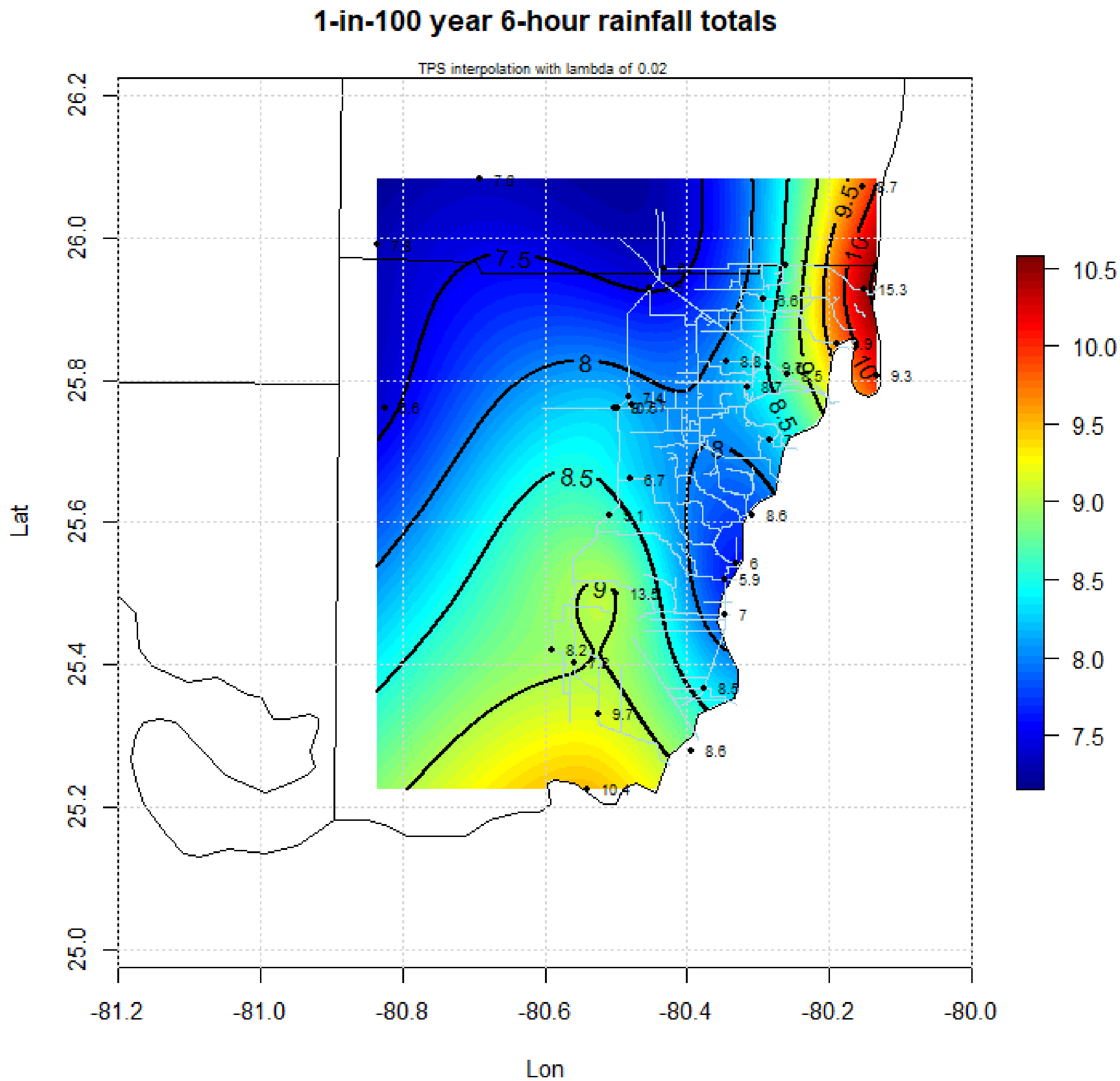


Figure 109. Interpolated 1-in-100-year 6-hour rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

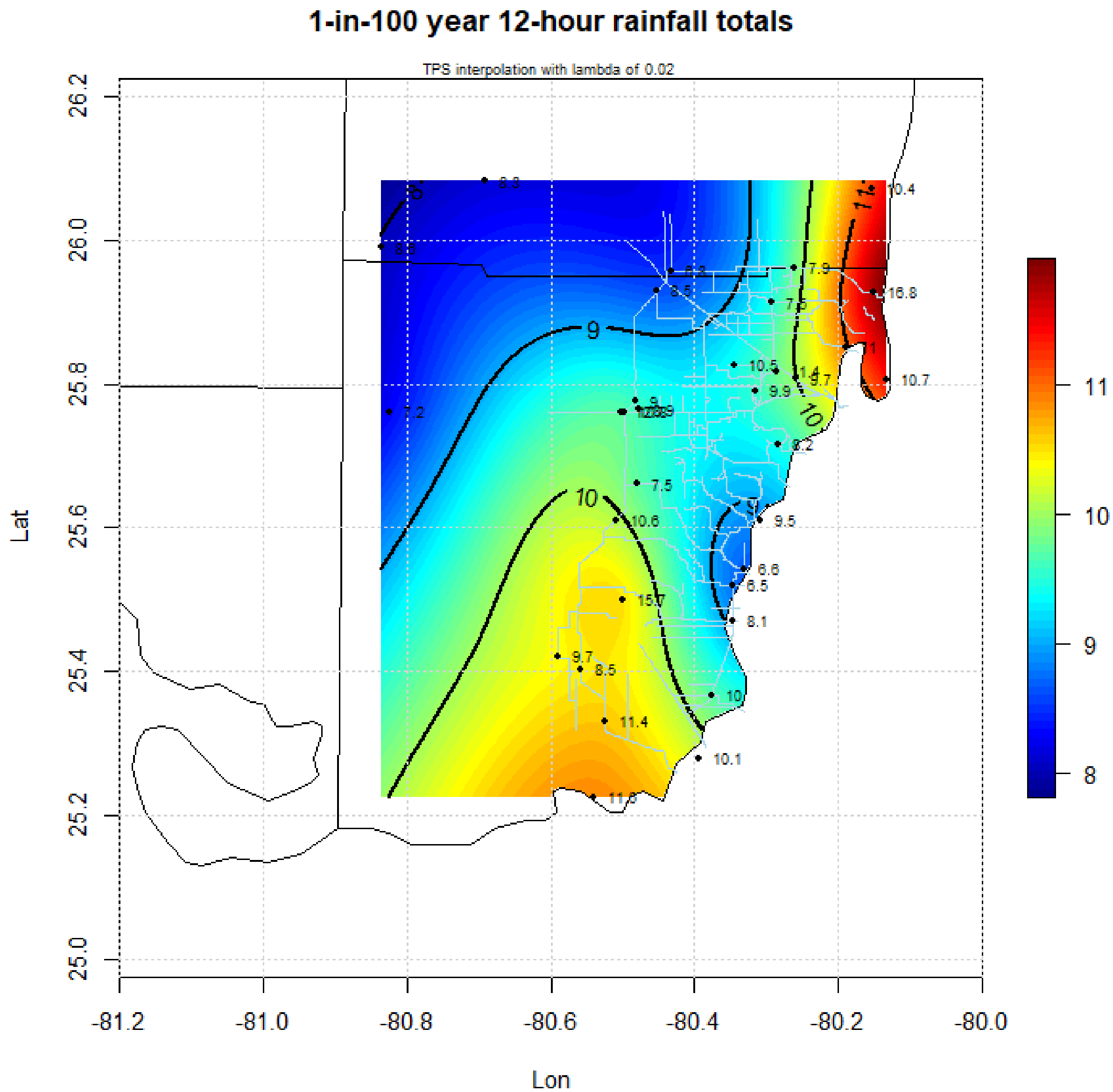


Figure 110. Interpolated 1-in-100-year 12-hour rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

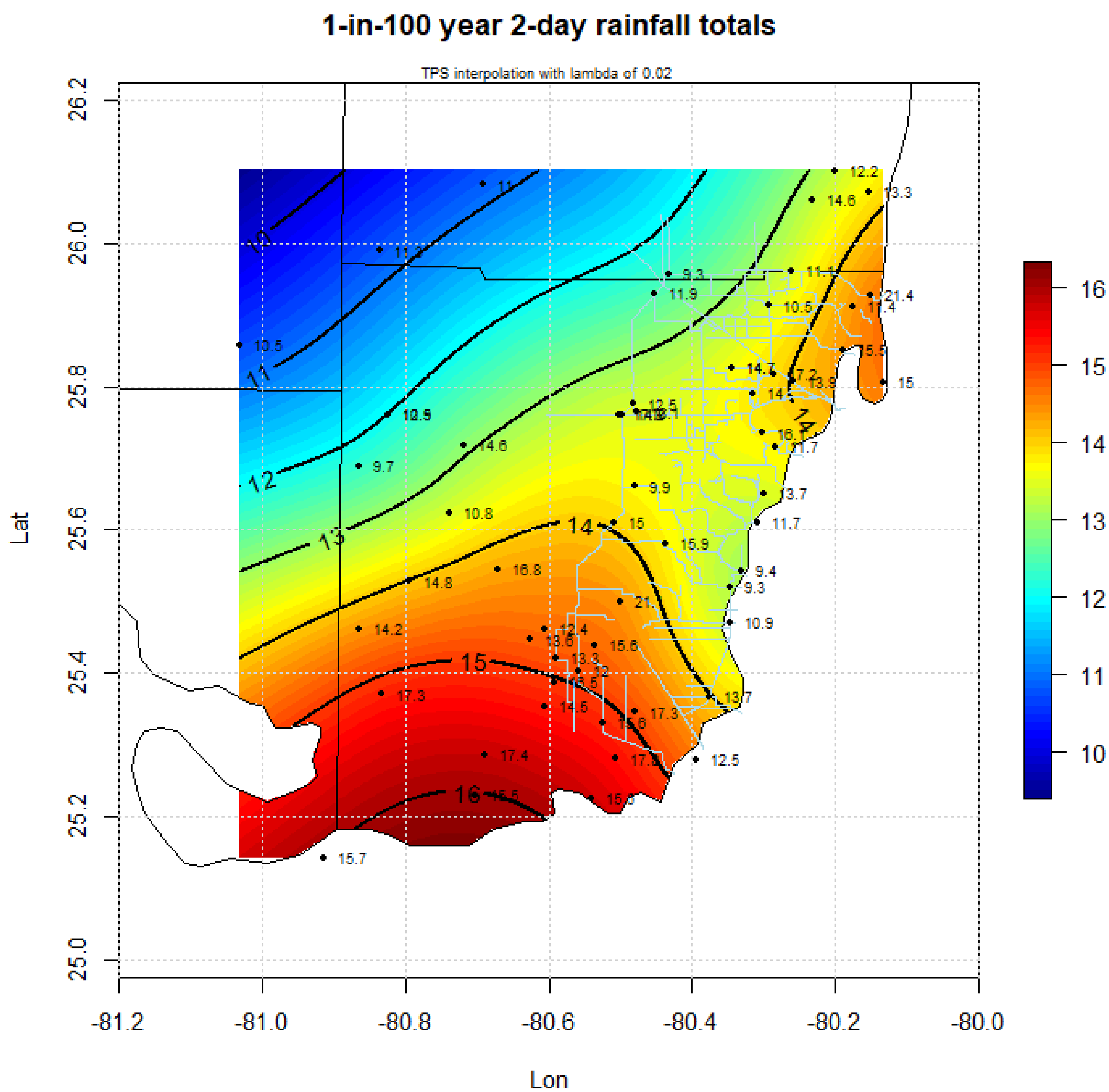


Figure 111. Interpolated 1-in-100-year 2-day rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

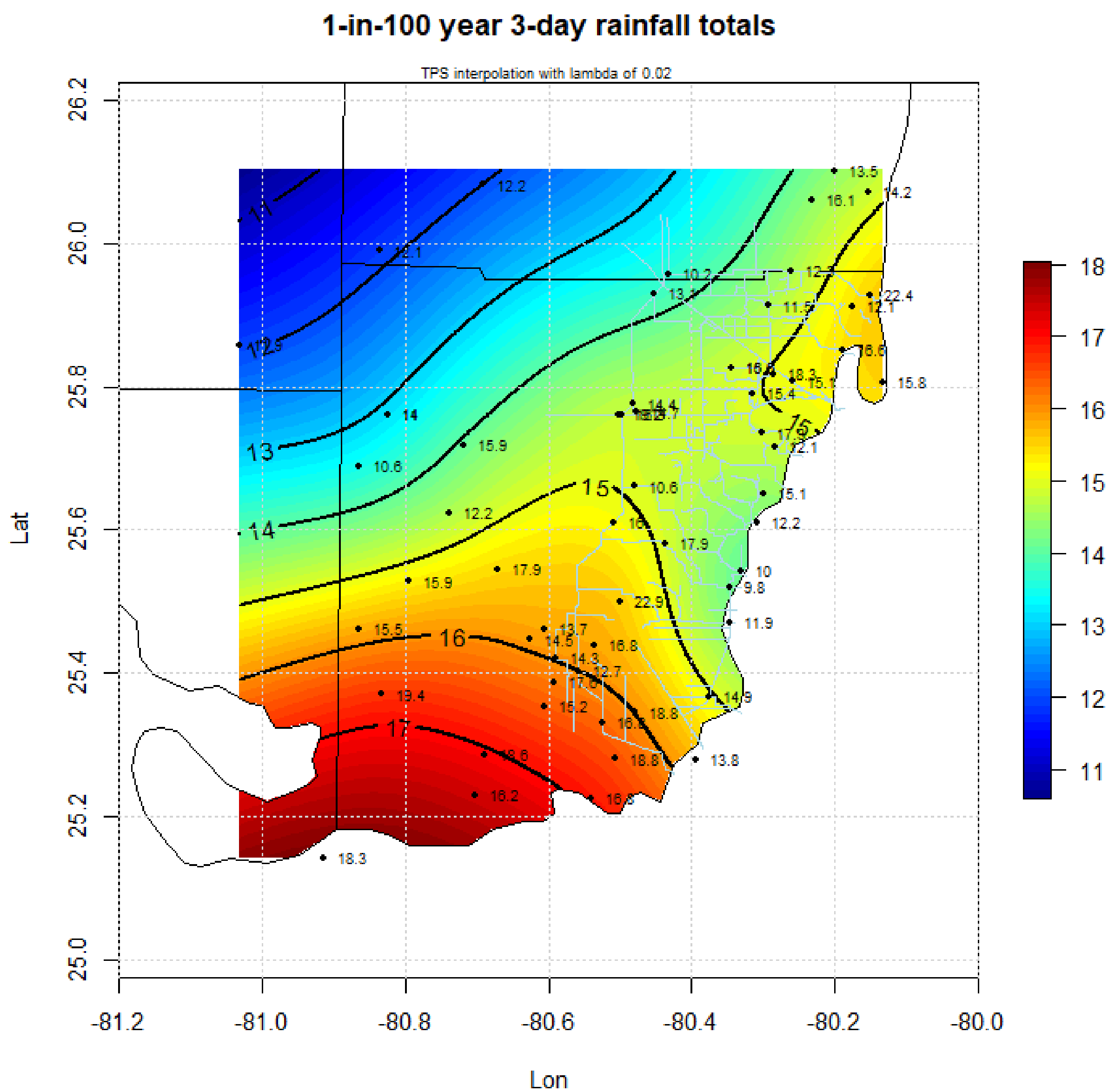


Figure 112. Interpolated 1-in-100-year 3-day rainfall totals (inches) based on based on TPS smoothing of station data (black dots) using a lambda value of 0.02.

Appendix C. Maps of At-site RFA parameters and DDF curves for the observational dataset in the current baseline period (Last 30 years up to 2005)

(Units: inches)

1-in-100-year hourly rainfall totals for the observational dataset in the current baseline period

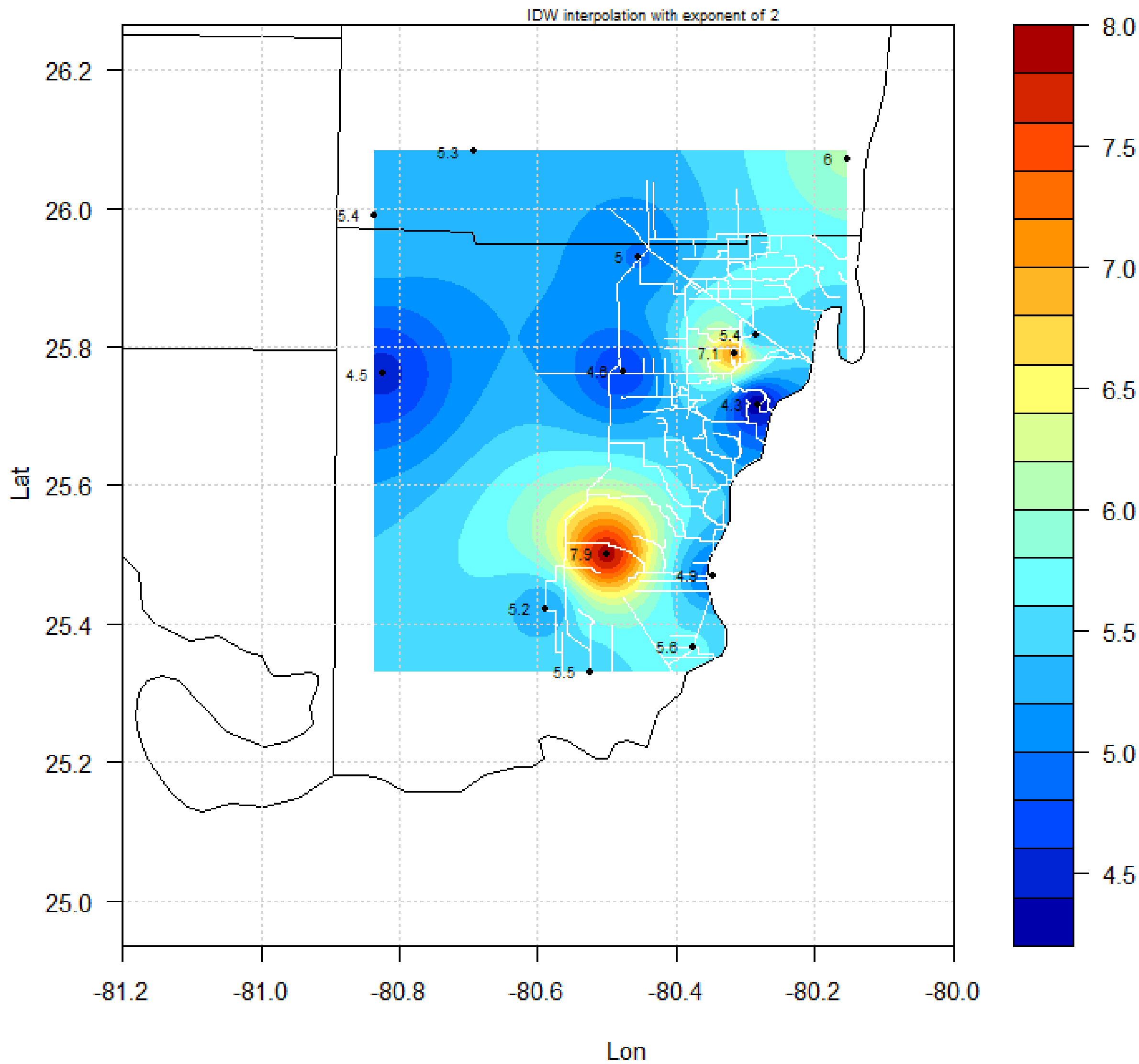


Figure 113. DDF precipitation depths (inches) fit to Atlas 14 AMS data in the current baseline period (last 30 years up to 2005, F_{o-c}) for hourly duration, 100-year return period.

1-in-100-year daily rainfall totals for the observational dataset in the current baseline period

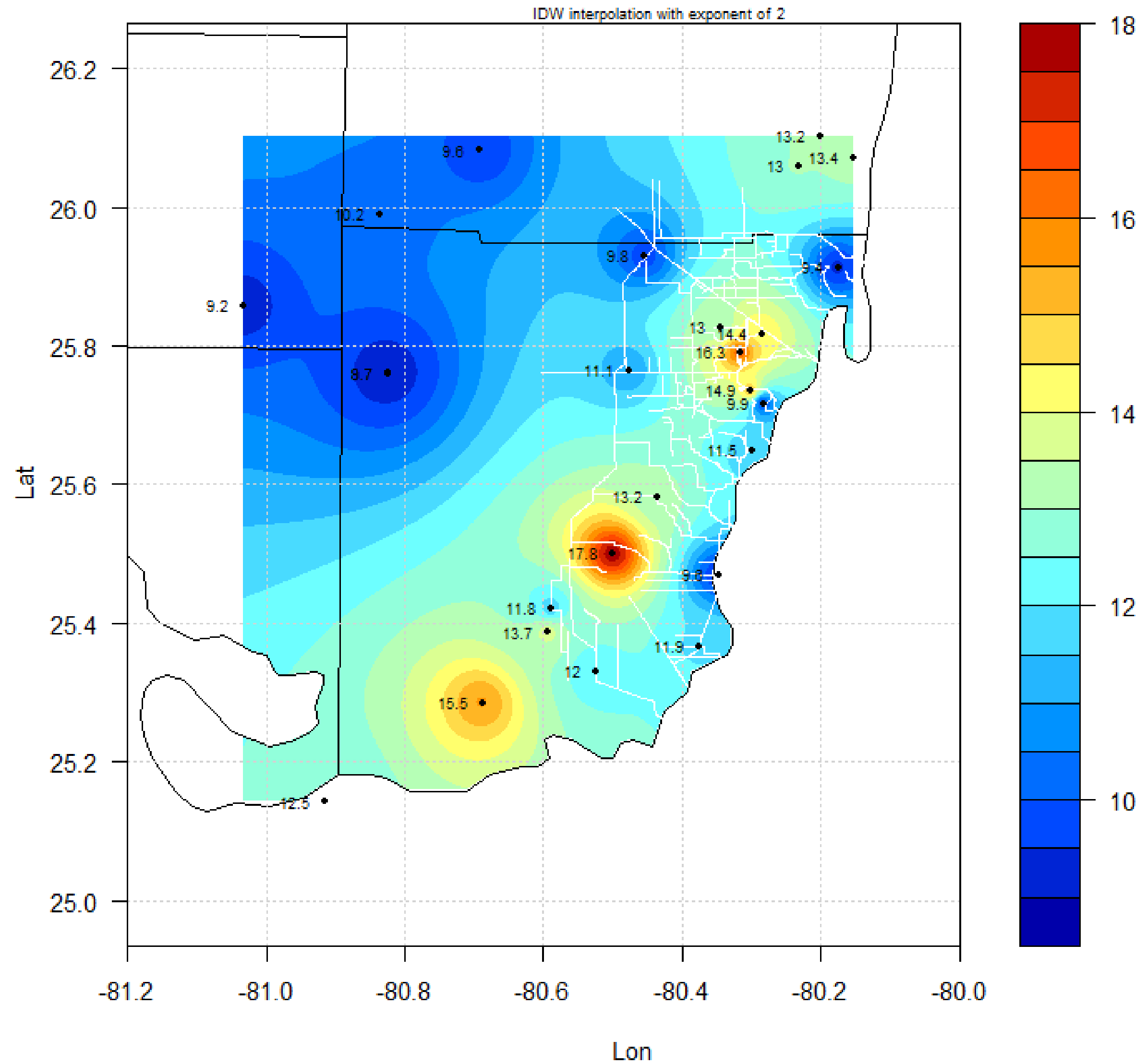


Figure 114. DDF precipitation depths (inches) fit to Atlas 14 AMS data in the current baseline period (last 30 years up to 2005, F_{o-c}) for daily duration, 100-year return period.

Appendix D. Maps of At-site RFA DDF curves for the downscaled model dataset (LOCA) in the current baseline period (Last 30 years up to 2005)

(Note: 5th, 50th, 95th percentiles can come from different models at different locations; Units: inches)

5th percentile of 1-in-100-year daily rainfall totals from LOCA for the current baseline period

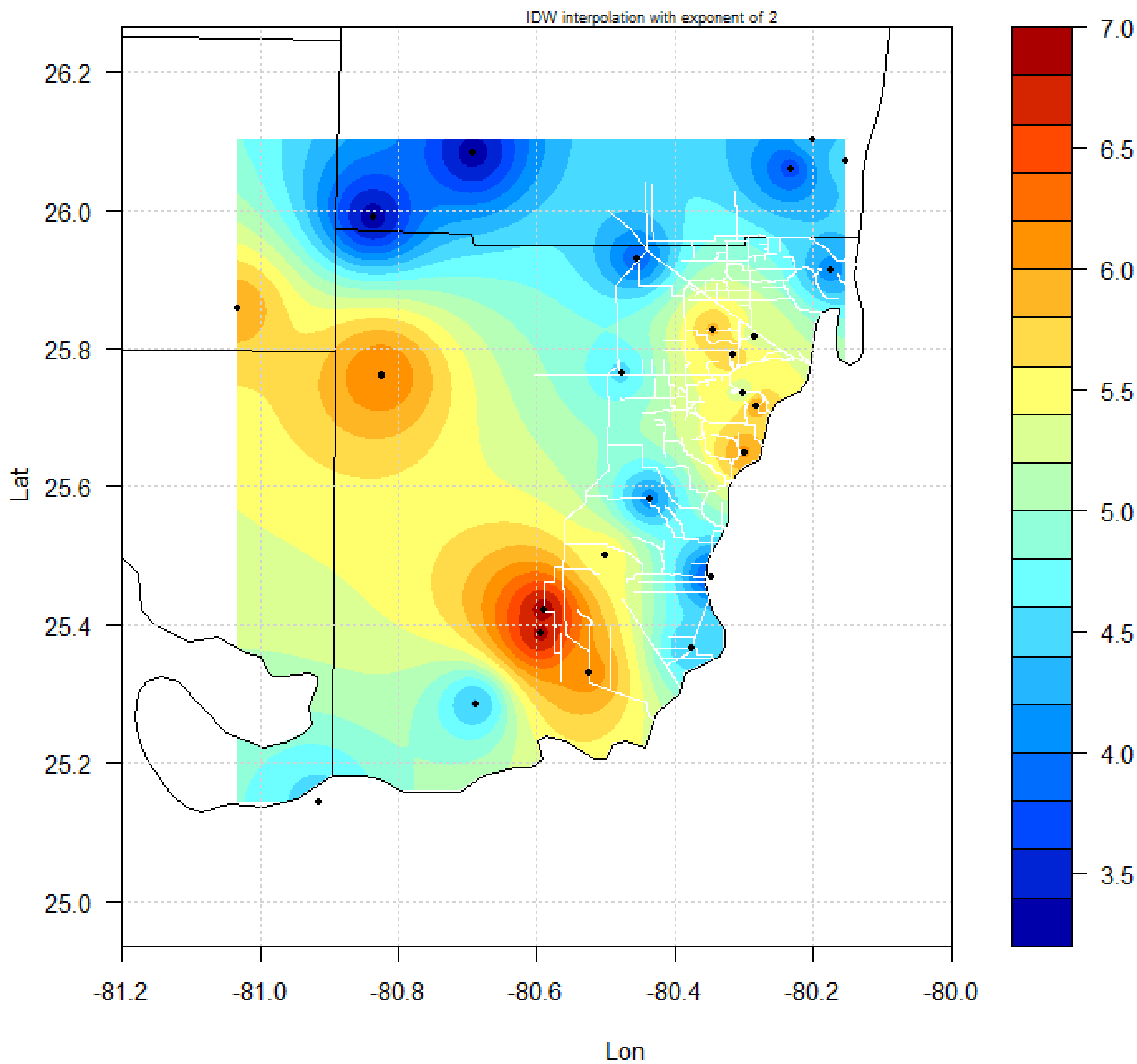


Figure 115.5th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the current baseline period (last 30 years up to 2005, F_{m-c}) for 24-hour duration, 100-year return period.

50th percentile of 1-in-100-year daily rainfall totals from LOCA

for the current baseline period

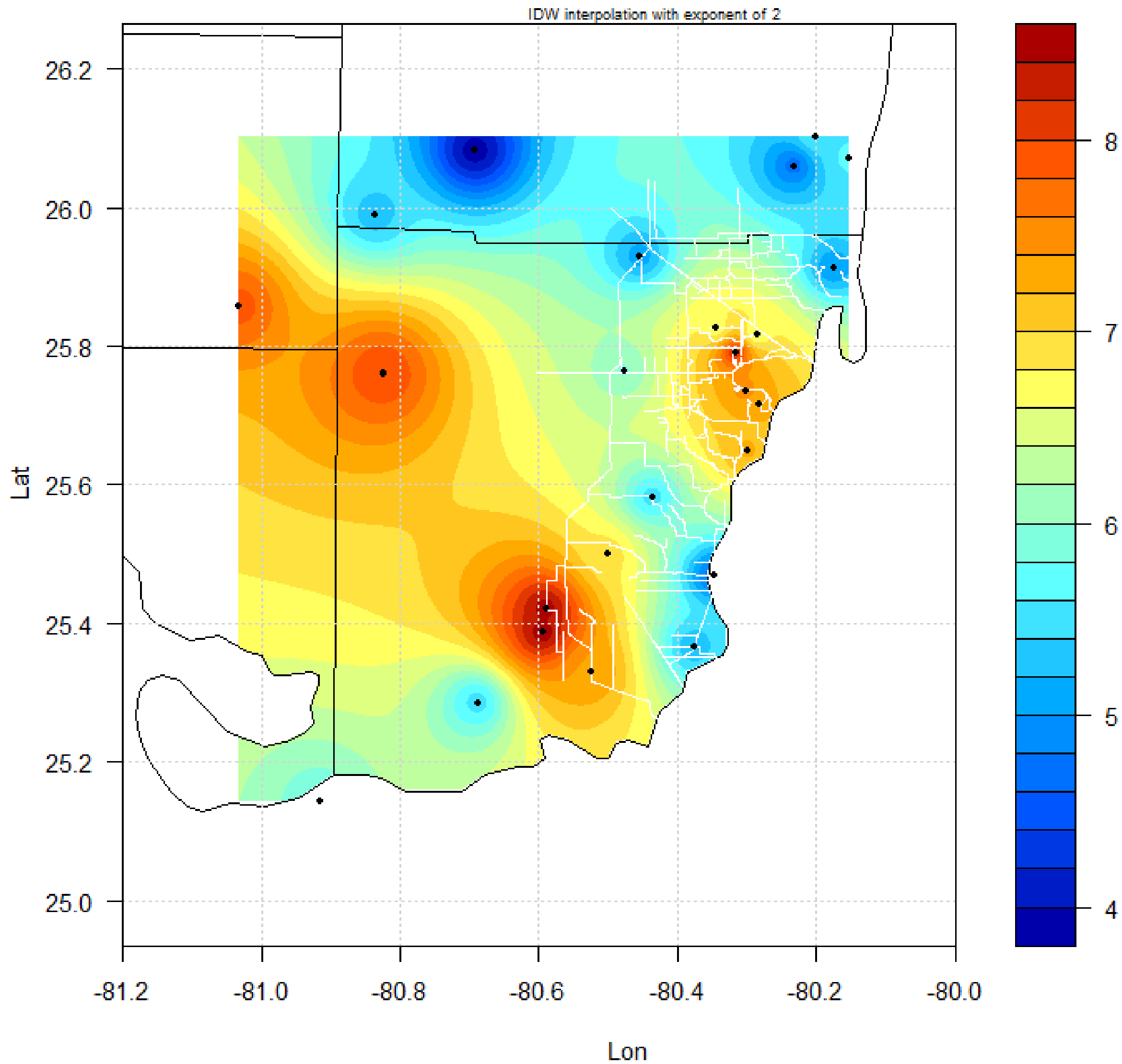


Figure 116. 50th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the current baseline period (last 30 years up to 2005, F_{m-c}) for 24-hour duration, 100-year return period.

95th percentile of 1-in-100-year daily rainfall totals from LOCA for the current baseline period

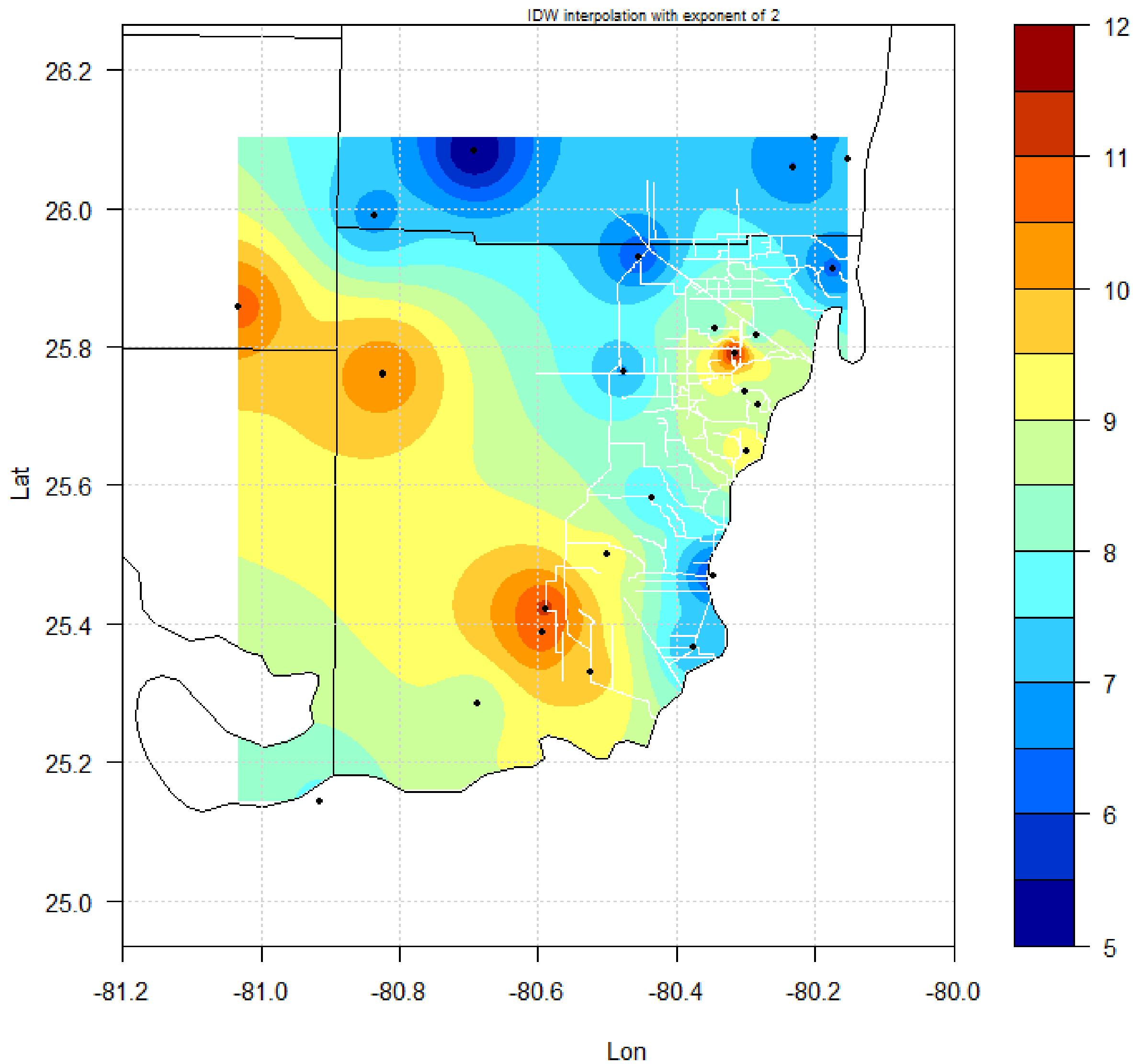


Figure 117. 95th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the current baseline period (last 30 years up to 2005, F_{m-c}) for 24-hour duration, 100-year return period.

5th percentile of differences in 1-in-100-year daily rainfall totals (LOCA – observations) for the current baseline period

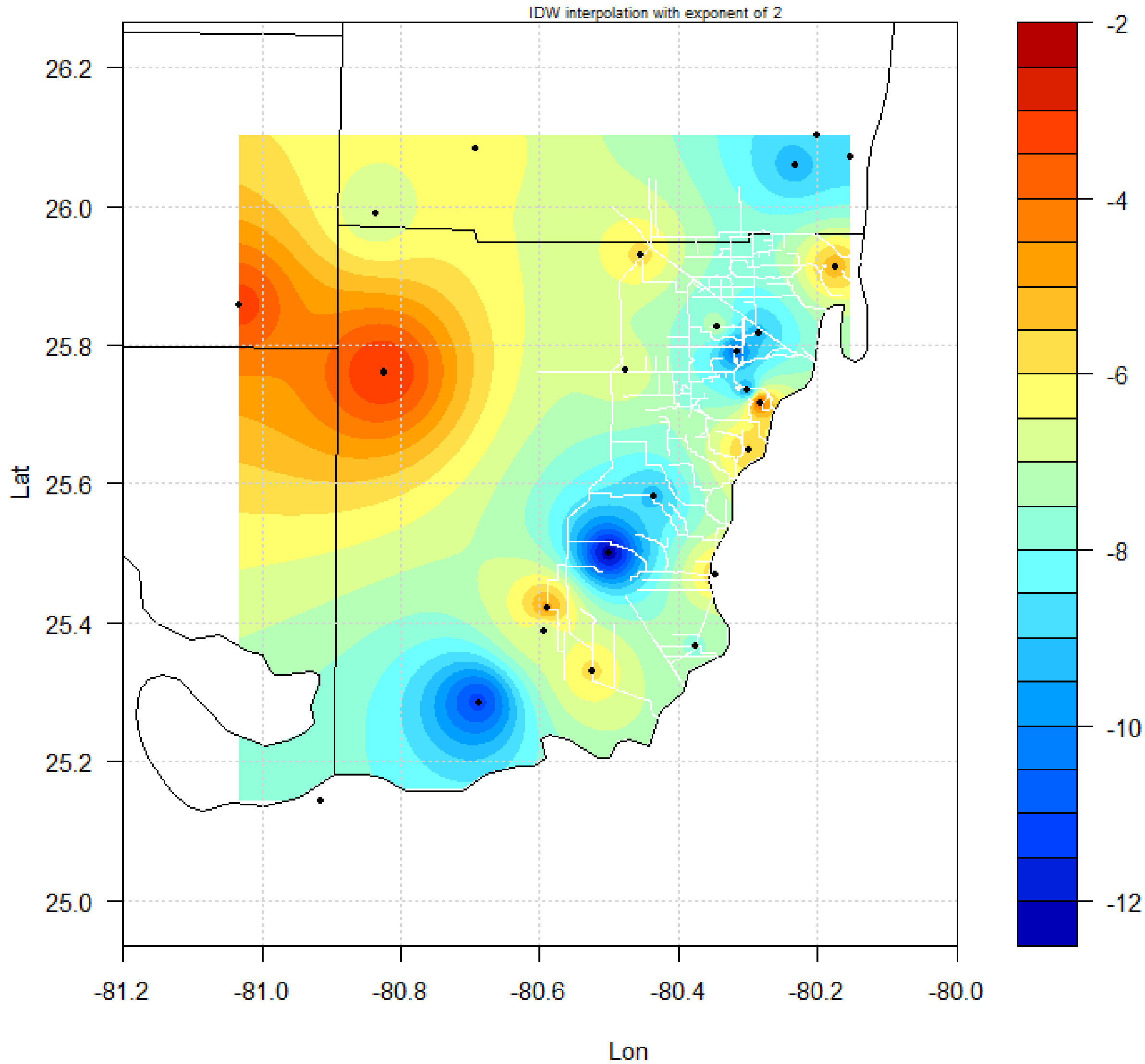


Figure 118. 5th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data minus DDF precipitation depths fit to Atlas 14 AMS data in the current baseline period (last 30 years up to 2005) (F_{m-c}- F_{o-c}) for 24-hour duration, 100-year return period.

50th percentile of differences in 1-in-100-year daily rainfall totals (LOCA – observations) for the current baseline period

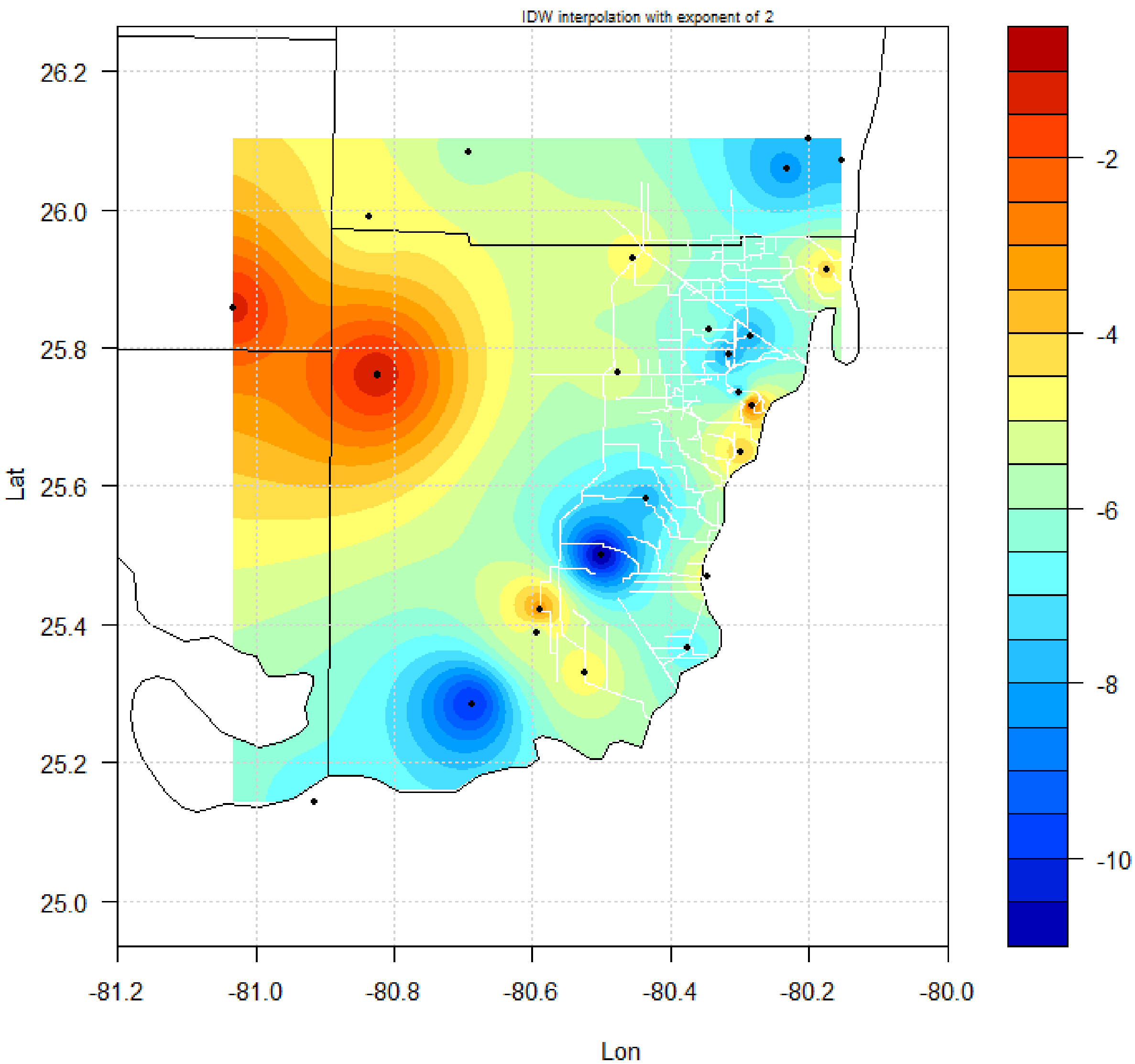


Figure 119. 50th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data minus DDF precipitation depths fit to Atlas 14 AMS data in the current baseline period (last 30 years up to 2005) (F_{m-c}- F_{o-c}) for 24-hour duration, 100-year return period.

95[th] percentile of differences in 1-in-100-year daily rainfall totals (LOCA – observations) for the current baseline period

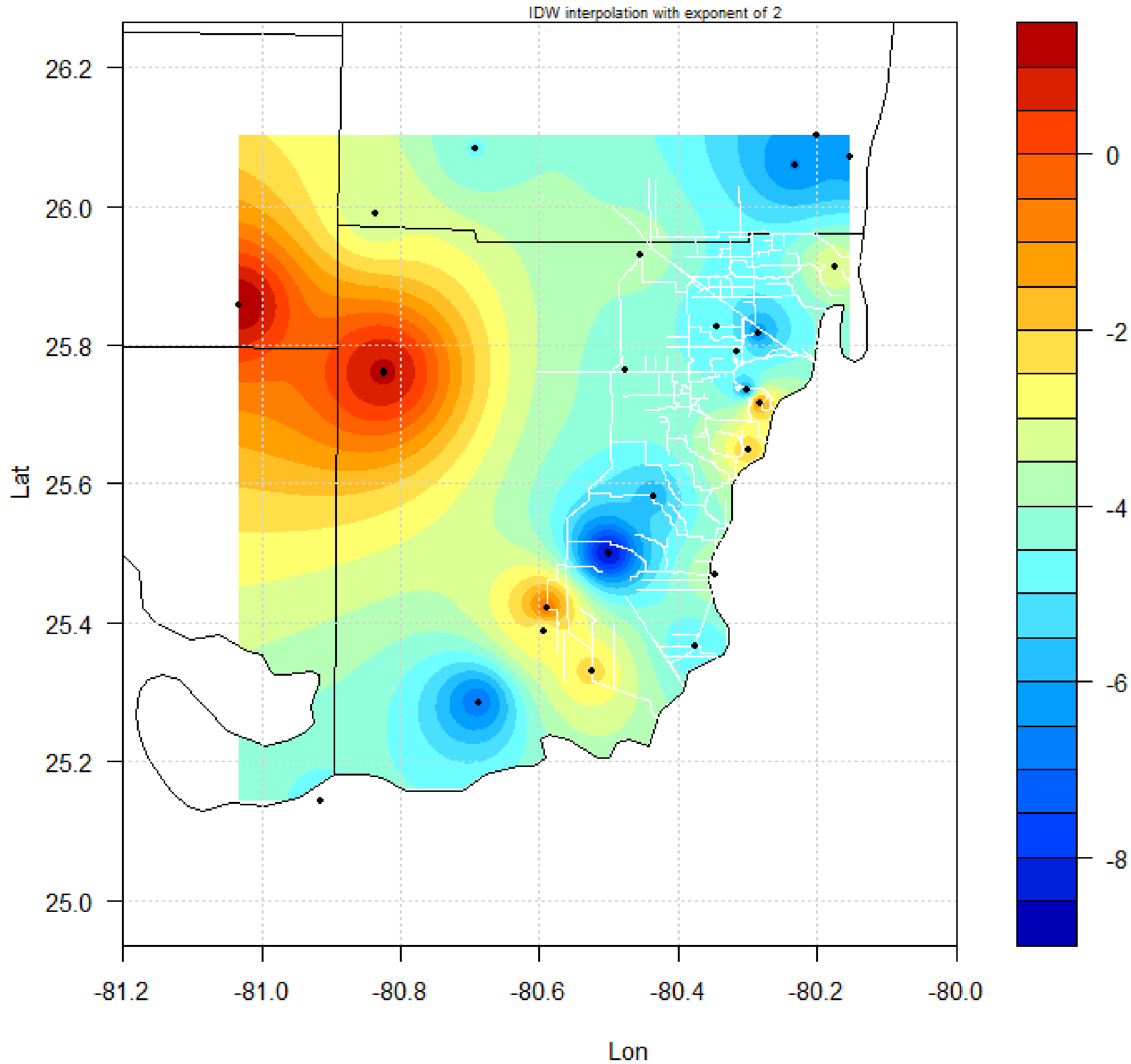


Figure 120. 95th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data minus DDF precipitation depths fit to Atlas 14 AMS data in the current baseline period (last 30 years up to 2005) (F_{m-c}- F_{o-c}) for 24-hour duration, 100-year return period.

Appendix E. Maps of At-site RFA DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079)

(Note: 5th, 50th, 95th percentiles can come from different models at different locations; Units: inches)

5th percentile of 1-in-100-year daily rainfall totals from LOCA

for the future projection period

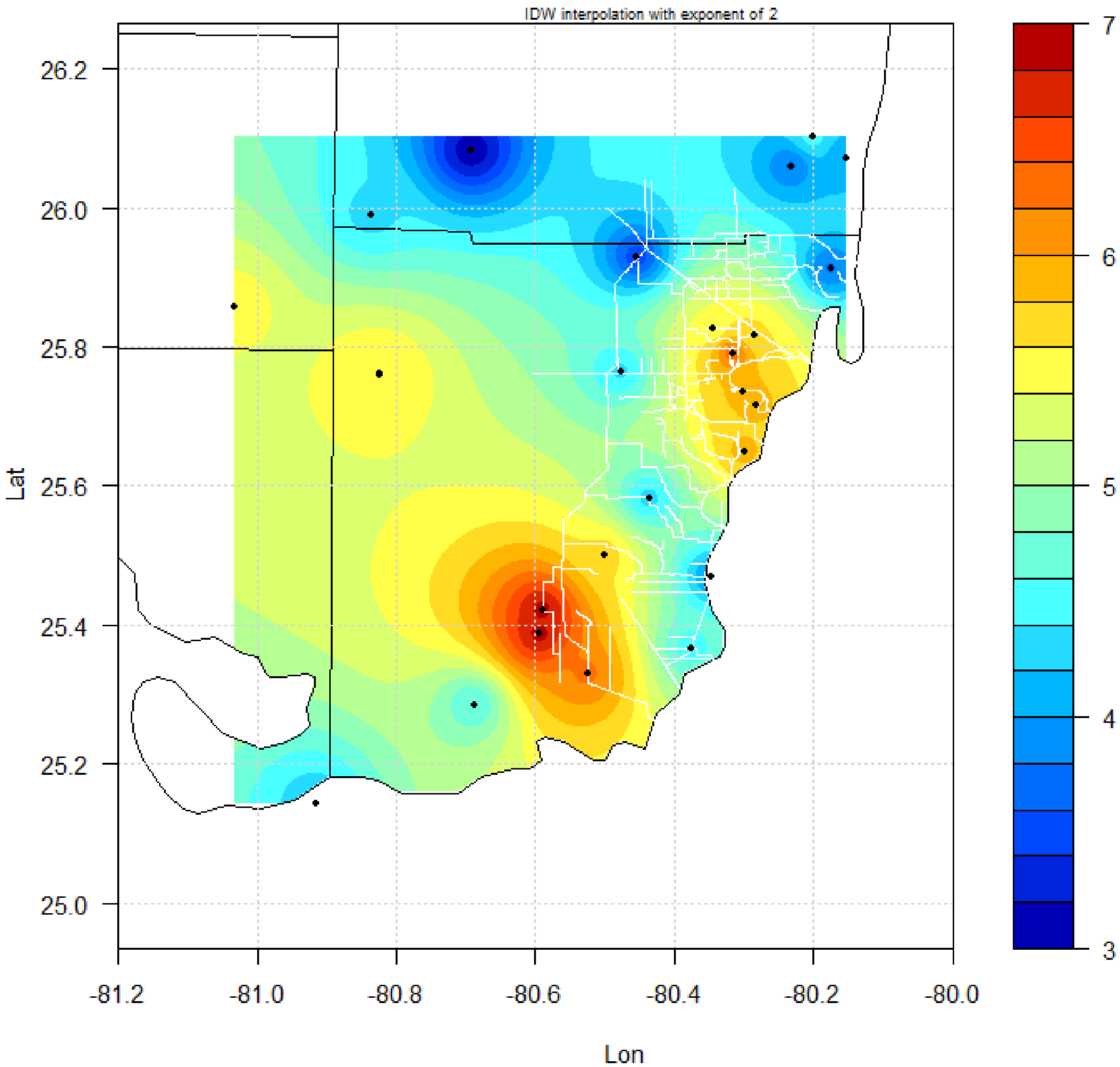


Figure 121. 5th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the future projection period centered in 2065 (2050-2079, F_{m-p1}) for 24-hour duration, 100-year return period.

50^{th} percentile of 1-in-100-year daily rainfall totals from LOCA for the future projection period

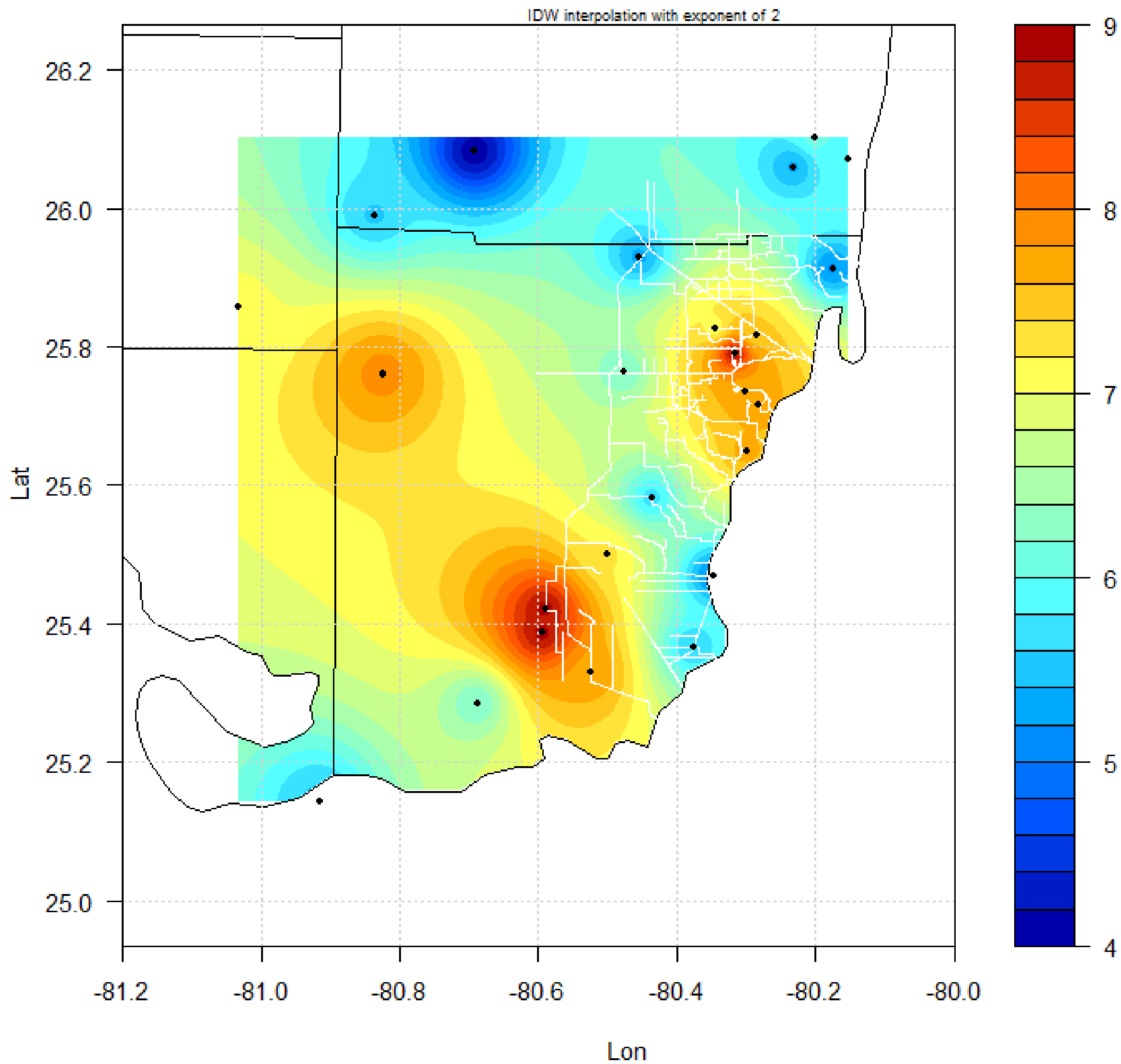


Figure 122. 50th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the future projection period centered in 2065 (2050-2079, F_{m-p1}) for 24-hour duration, 100-year return period.

95th percentile of 1-in-100-year daily rainfall totals from LOCA for the future projection period

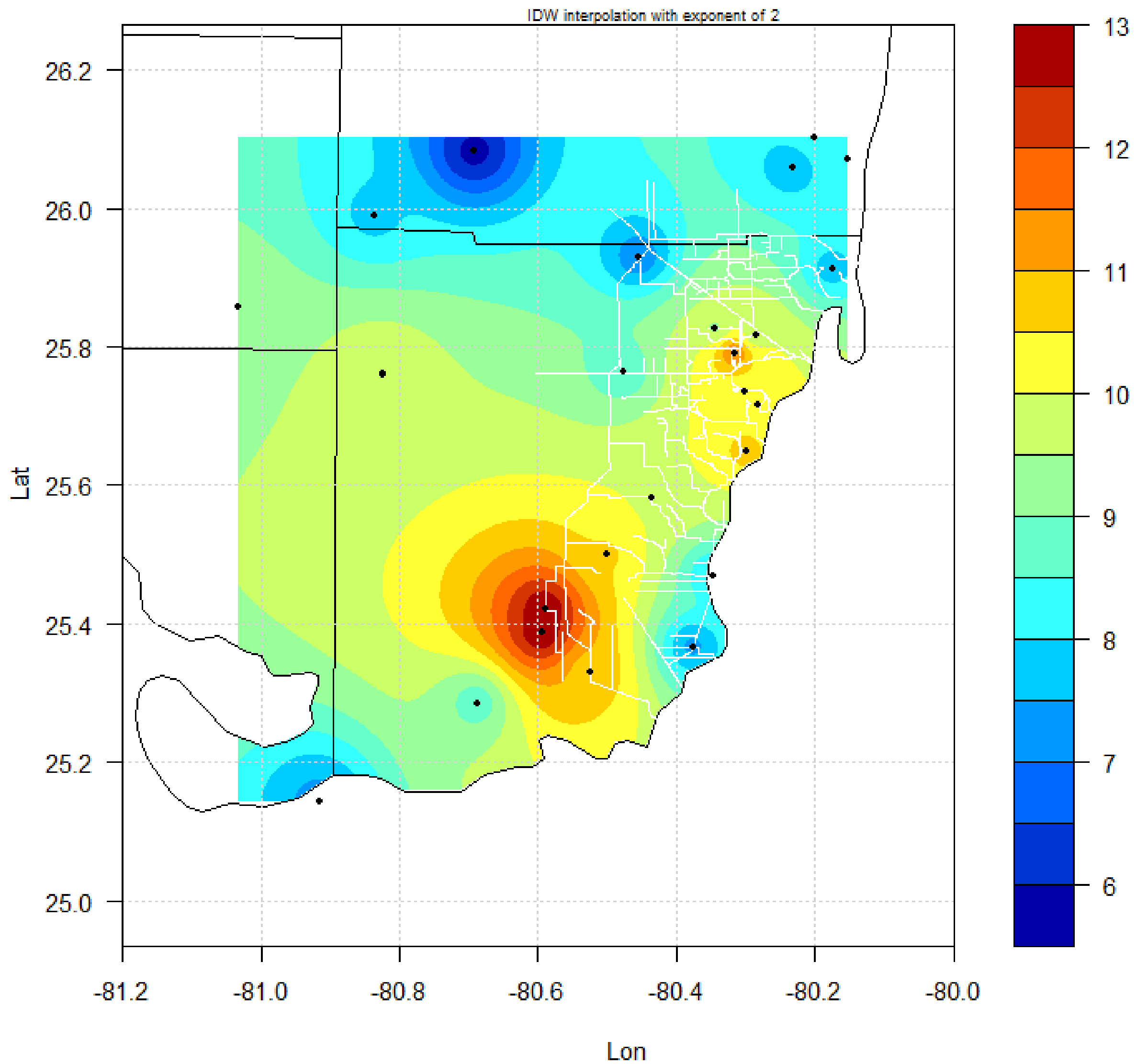


Figure 123. 95th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data in the future projection period centered in 2065 (2050-2079, F_{m-p1}) for 24-hour duration, 100-year return period.

5th percentile of differences in 1-in-100-year daily rainfall totals for future - current baseline period in LOCA

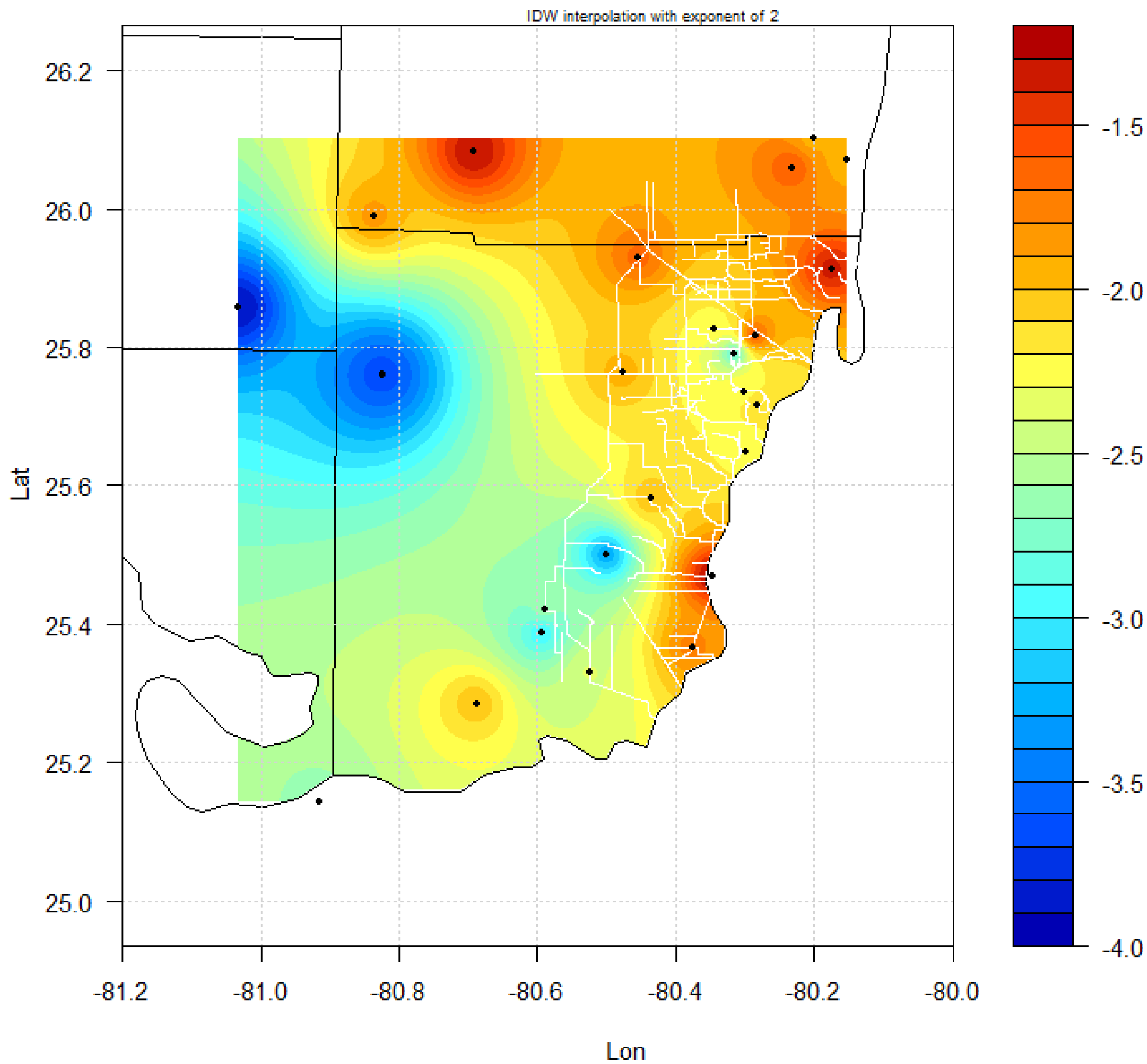


Figure 124. 5th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data for the future period centered in 2065 (2055-2079) minus the current baseline period (last 30 years up to 2005) ($F_{m-p1} - F_{m-c}$) for 24-hour duration, 100-year return period.

50th percentile of differences in 1-in-100-year daily rainfall totals for future - current baseline period in LOCA

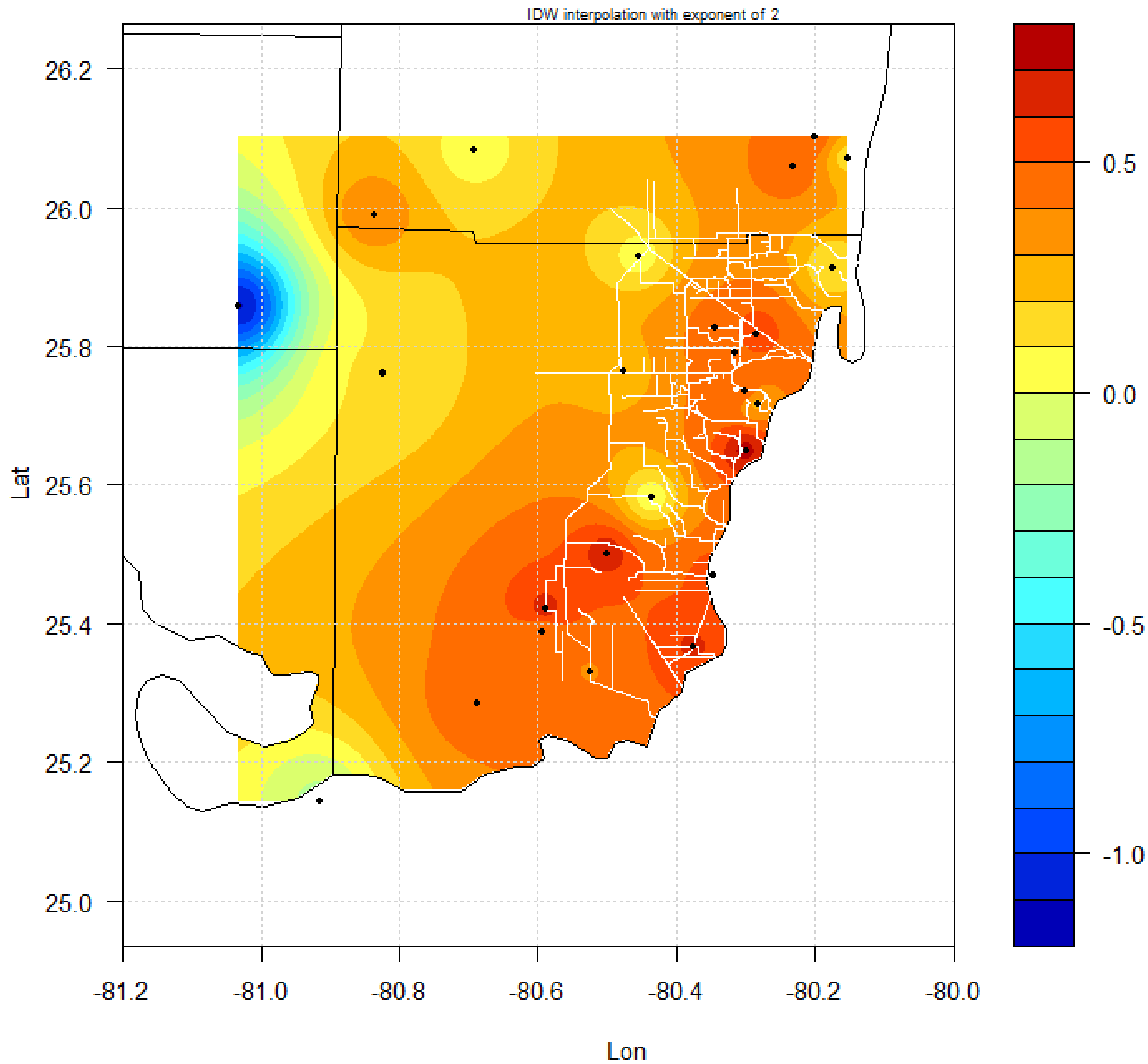


Figure 125. 50th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data for the future period centered in 2065 (2055-2079) minus the current baseline period (last 30 years up to 2005) ($F_{m-p1} - F_{m-c}$) for 24-hour duration, 100-year return period.

95th percentile of differences in 1-in-100-year daily rainfall totals for future - current baseline period in LOCA

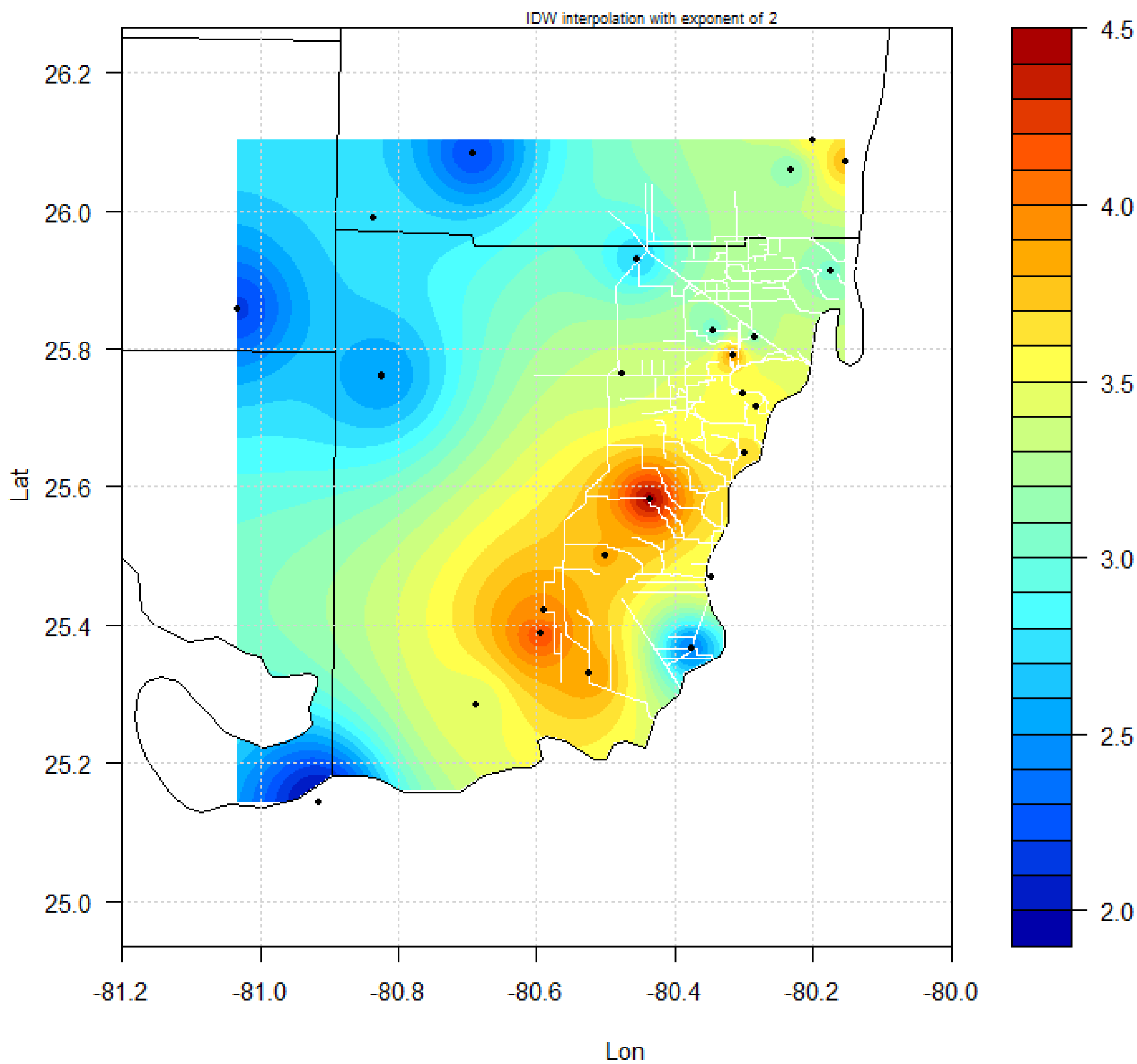


Figure 126. 95th percentile of DDF precipitation depths (inches) fit to downscaled model (LOCA) AMS data for the future period centered in 2065 (2055-2079) minus the current baseline period (last 30 years up to 2005) ($F_{m-p1} - F_{m-c}$) for 24-hour duration, 100-year return period.

Appendix F. Maps of adjusted DDF curves for the downscaled model dataset (LOCA) in the future period centered in 2065 (2050-2079)

(Note: 5th, 50th, 95th percentiles can come from different models at different locations; Units: inches)

5th percentile of adjusted 1-in-100-year daily rainfall totals from LOCA for the future projection period

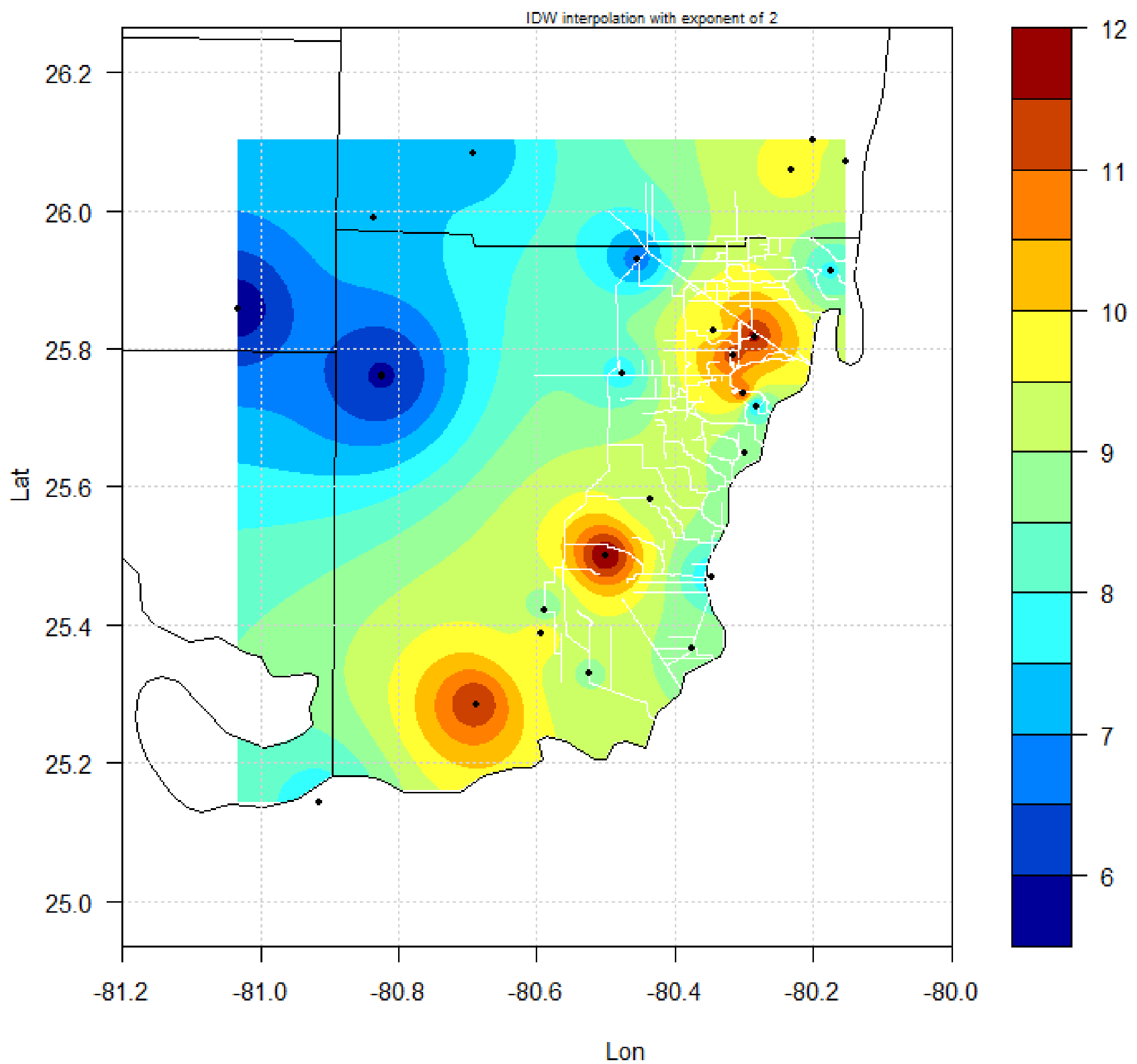


Figure 127. 5th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079, $F_{m-padj.1}$) for 24-hour duration, 100-year return period.

50th percentile of adjusted 1-in-100-year daily rainfall totals from LOCA for the future projection period

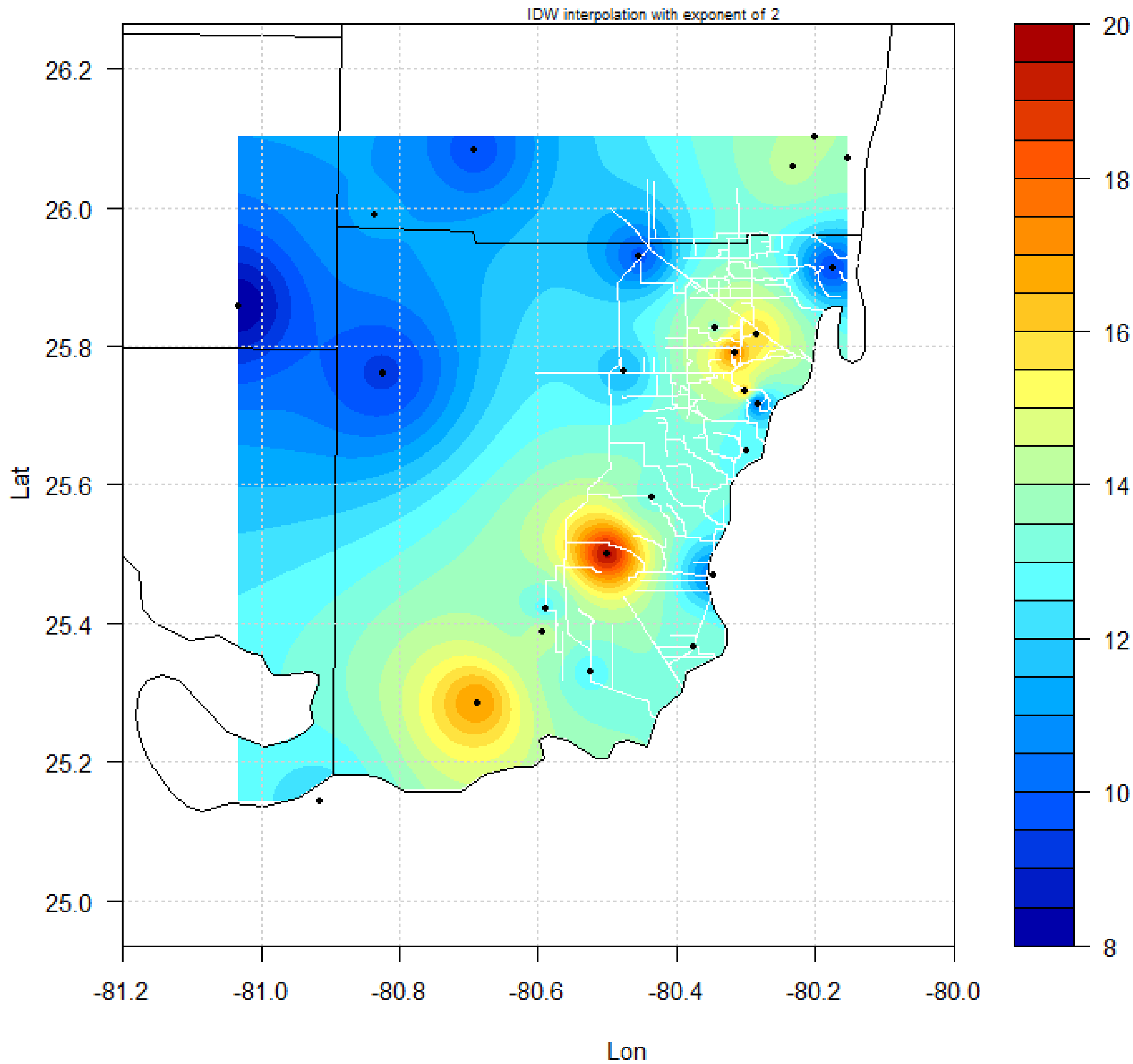


Figure 128. 50th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079, $F_{m-padj.1}$) for 24-hour duration, 100-year return period.

95th percentile of adjusted 1-in-100-year daily rainfall totals from LOCA for the future projection period

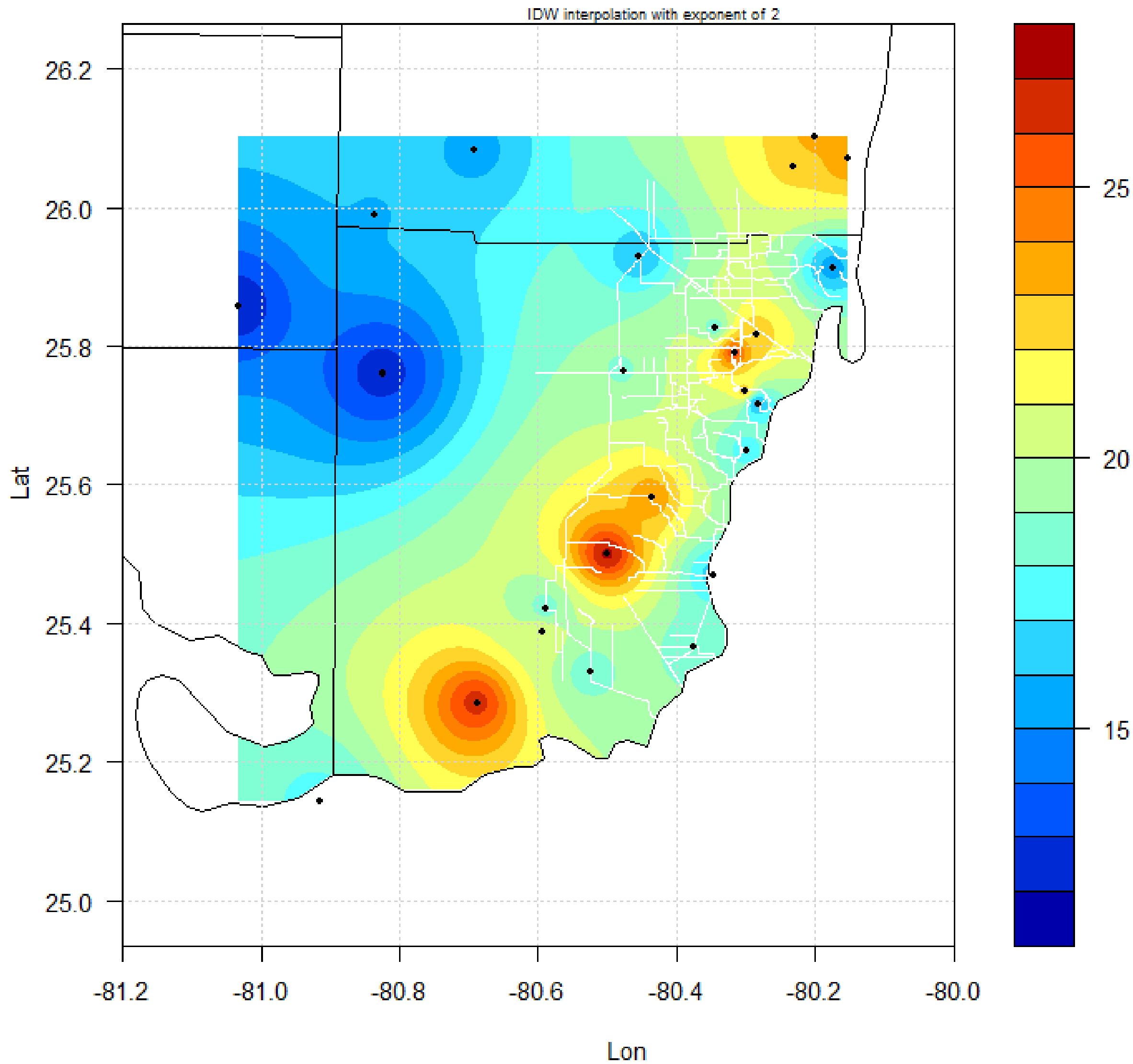


Figure 129. 95th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079, $F_{m-padj.1}$) for 24-hour duration, 100-year return period.

5th percentile of differences in 1-in-100-year daily rainfall totals

for future adjusted LOCA projections – observations in the current baseline period

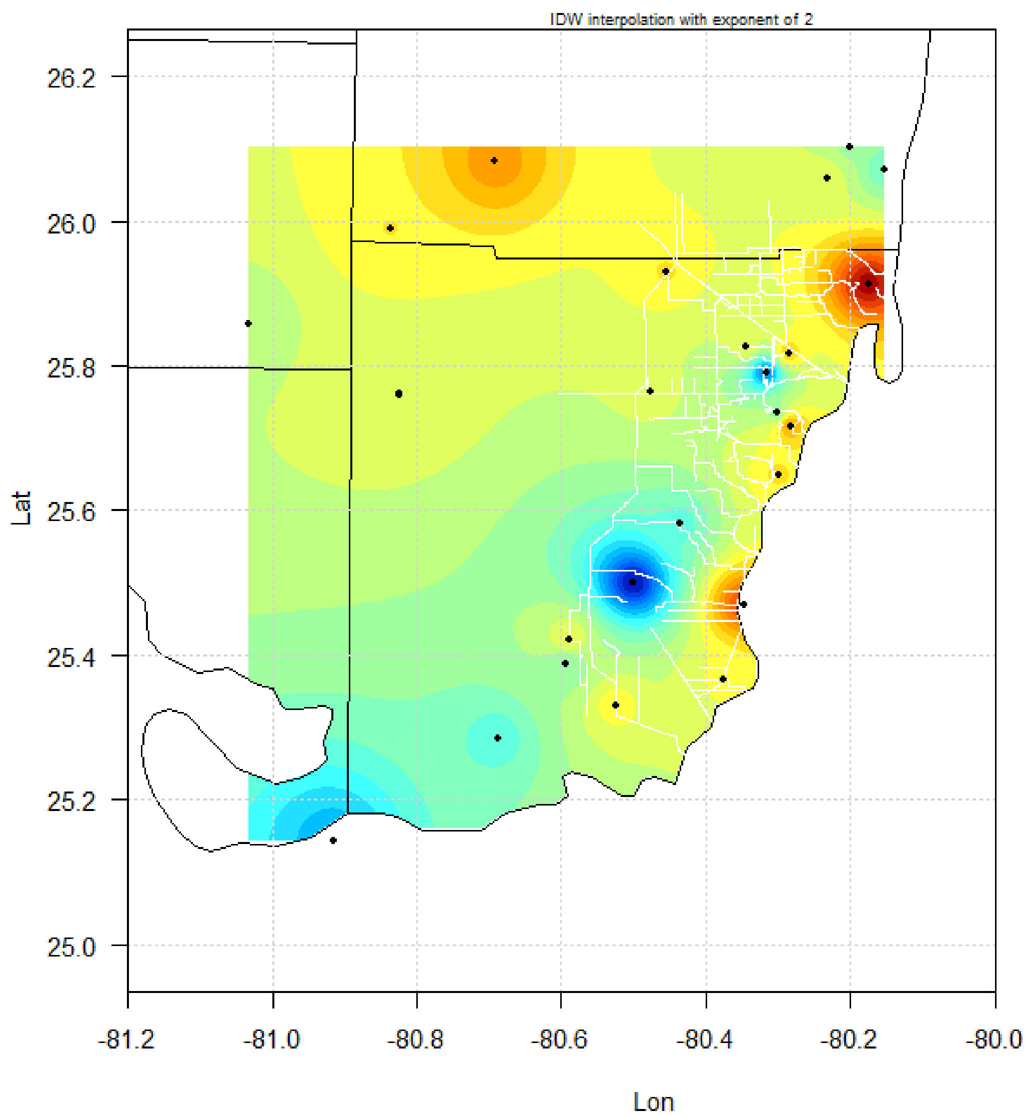


Figure 130. 5th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) ($F_{m-padj.1} - F_{o-c}$) for 24-hour duration, 100-year return period.

50th percentile of differences in 1-in-100-year daily rainfall totals for future adjusted LOCA projections – observations in the current baseline period

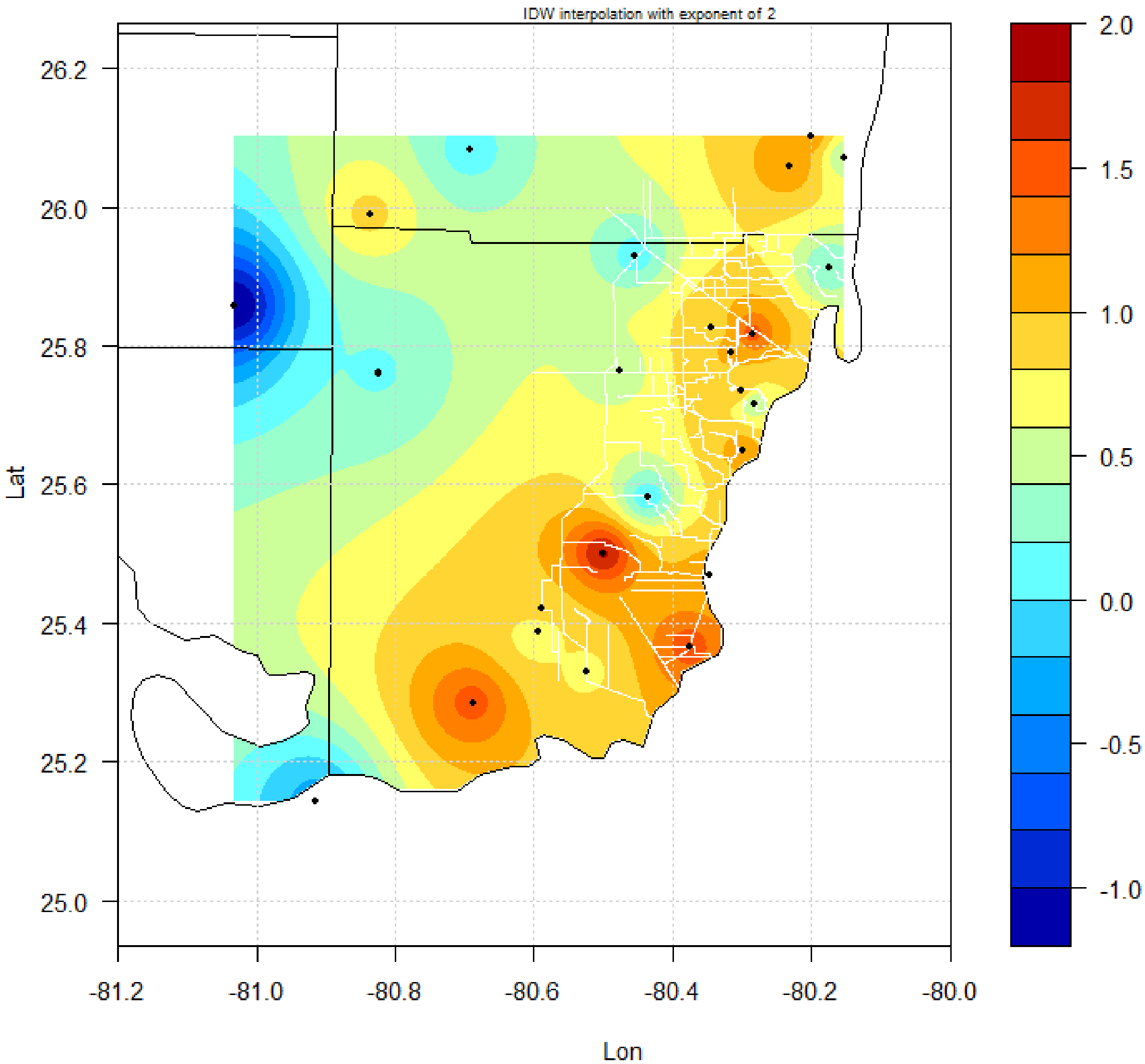


Figure 131. 50th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) $(F_{m-padj.1} - F_{o-c})$ *for 24-hour duration, 100-year return period.*

95th percentile of differences in 1-in-100-year daily rainfall totals for future adjusted LOCA projections – observations in the current baseline period

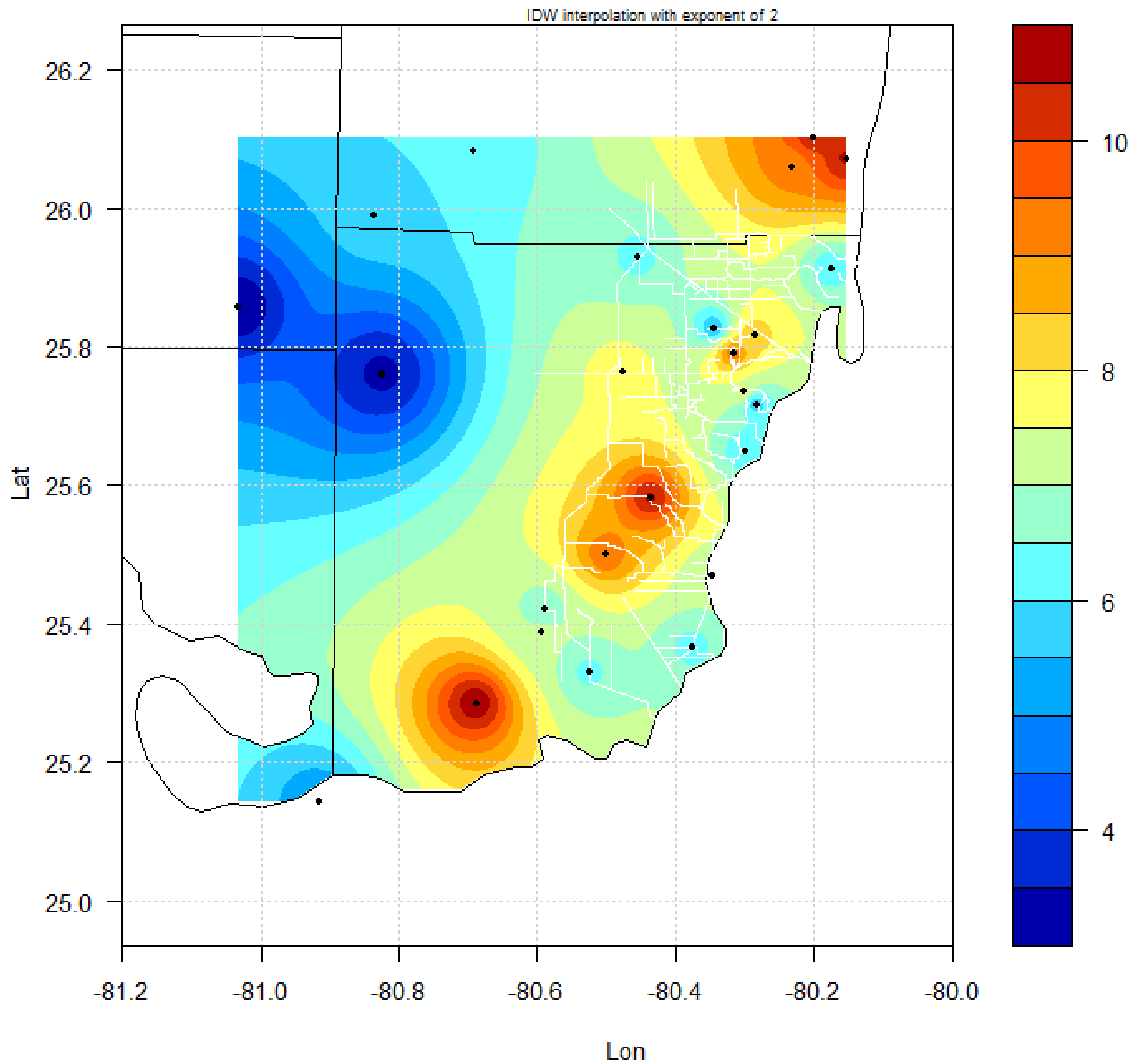


Figure 132. 95th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) $(F_{m-padj.1} - F_{o-c})$ for 24-hour duration, 100-year return period.

5th percentile of adjusted 1-in-100-year hourly rainfall totals from LOCA for the future projection period

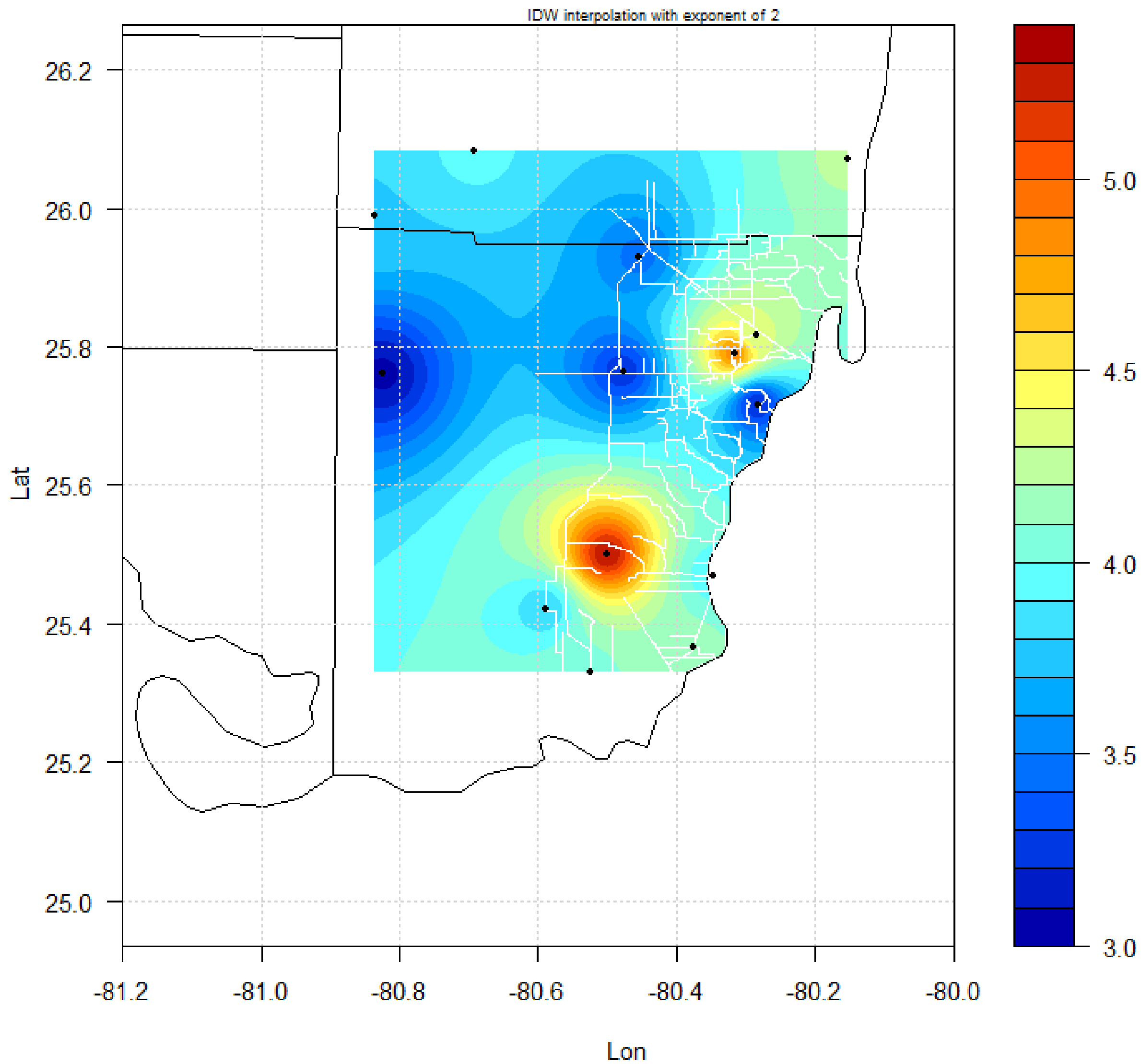


Figure 133. 5th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079) for 1-hour duration, 100-year return period.

50th percentile of adjusted 1-in-100-year hourly rainfall totals from LOCA for the future projection period

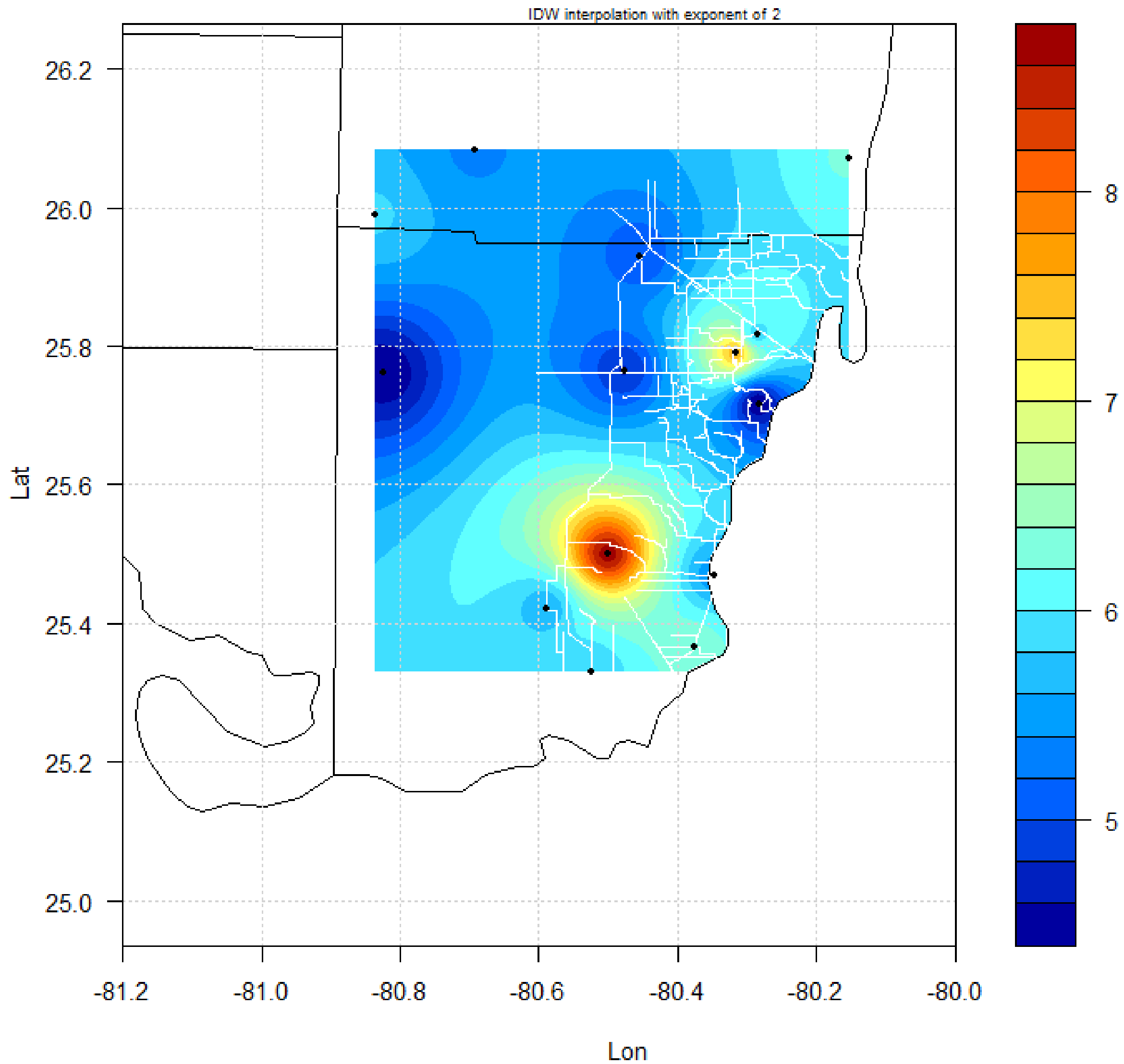


Figure 134. 50th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079) for 1-hour duration, 100-year return period.

95th percentile of adjusted 1-in-100-year hourly rainfall totals from LOCA for the future projection period

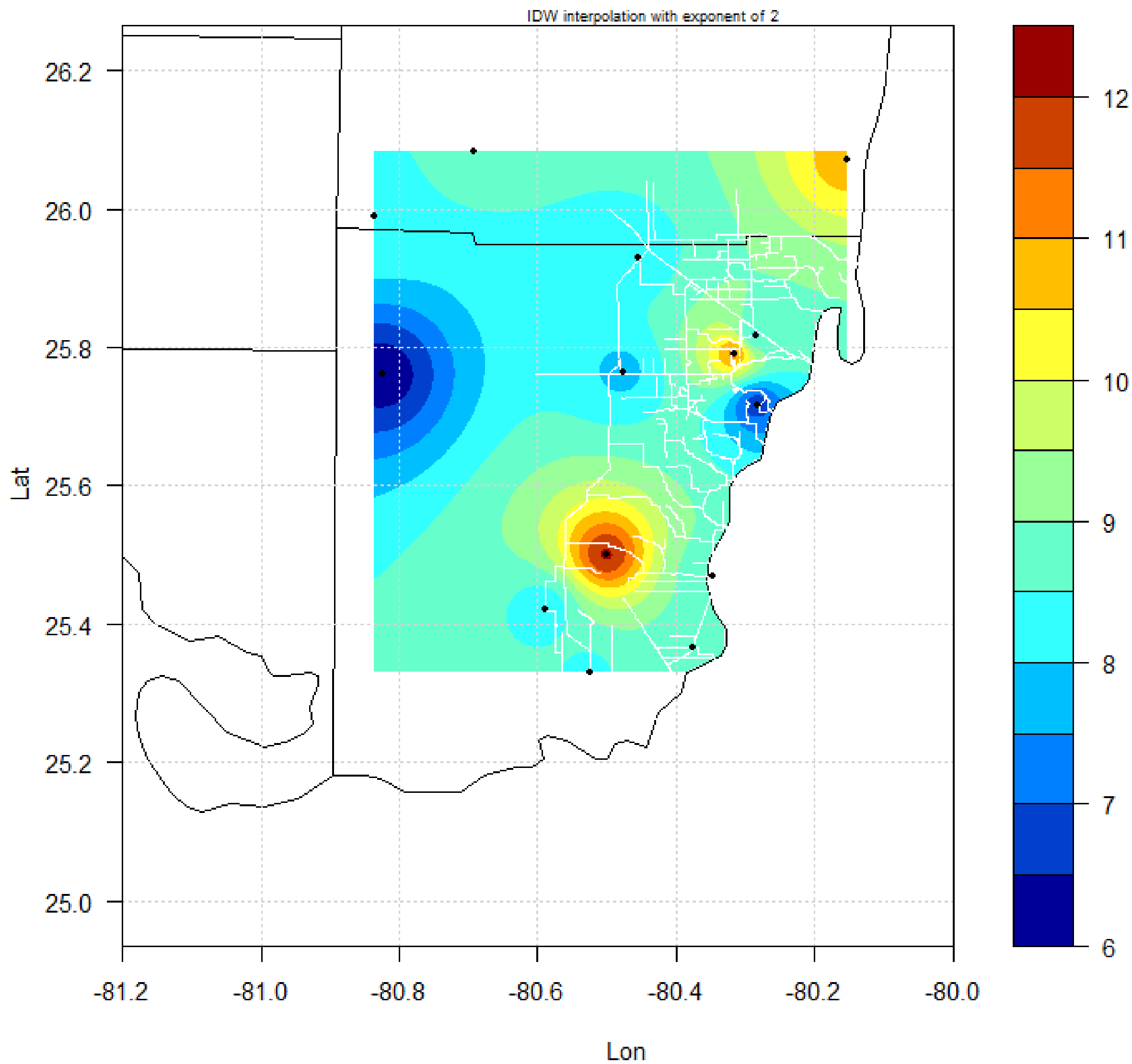


Figure 135. 95th percentile of adjusted DDF precipitation depths (inches) for the future projection period centered in 2065 (2050-2079) for 1-hour duration, 100-year return period.

5th percentile of differences in 1-in-100-year hourly rainfall totals

for future adjusted LOCA projections – observations in the current baseline period

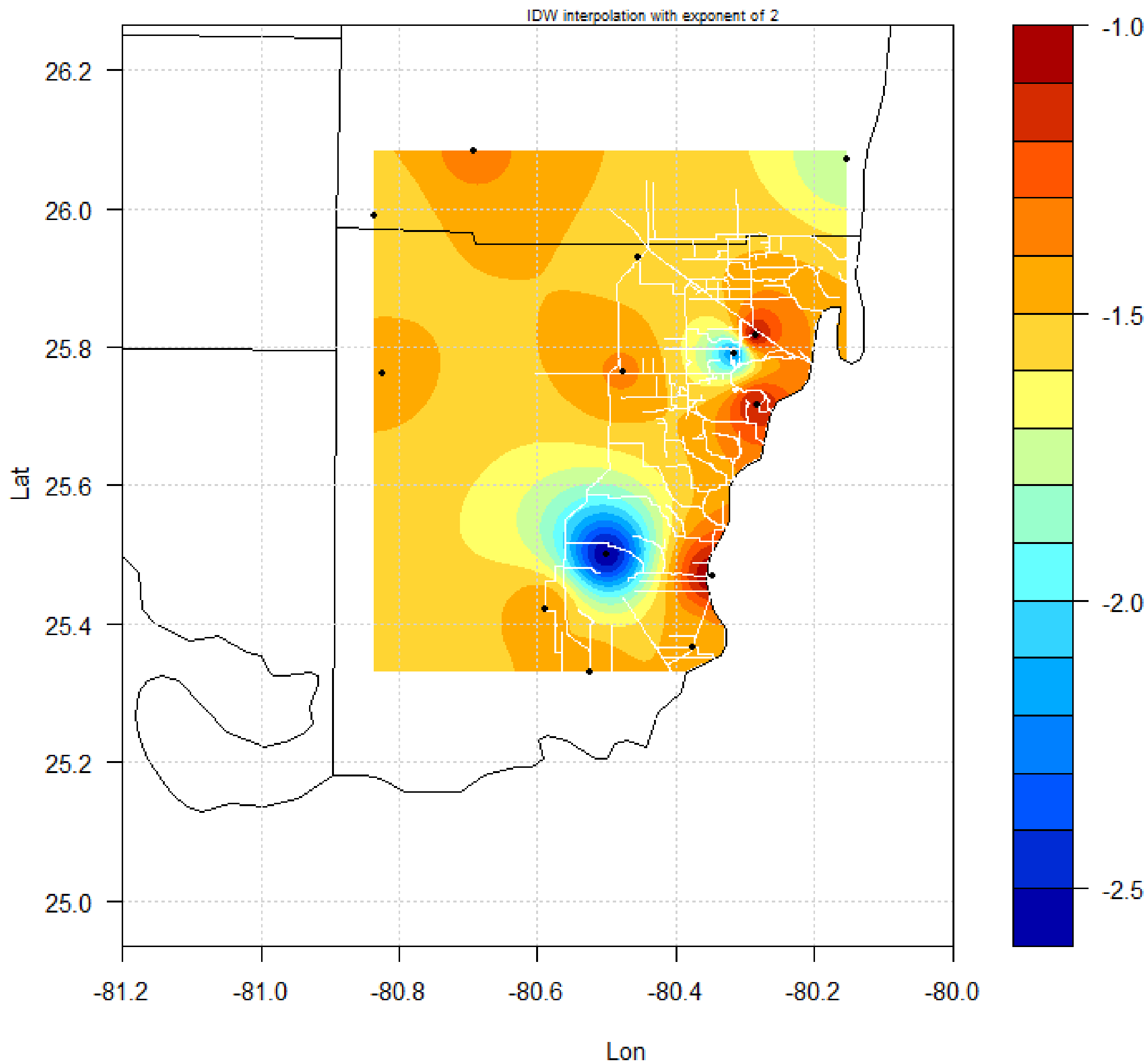


Figure 136. 5th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) for 1-hour duration, 100-year return period.

50th percentile of differences in 1-in-100-year hourly rainfall totals

for future adjusted LOCA projections – observations in the current baseline period

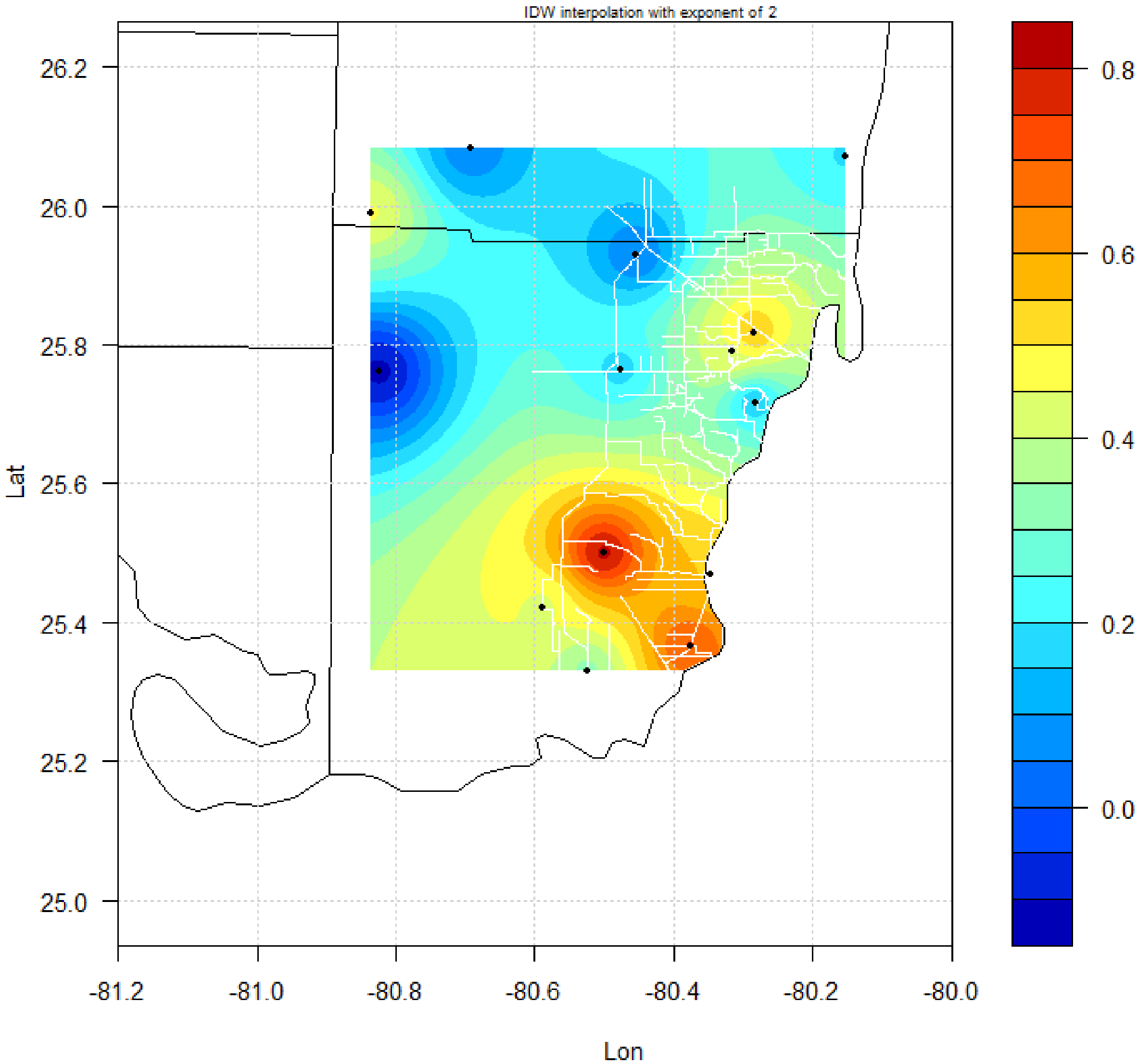


Figure 137. 50th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) for 1-hour duration, 100-year return period.

95th percentile of differences in 1-in-100-year hourly rainfall totals for future adjusted LOCA projections – observations in the current baseline period

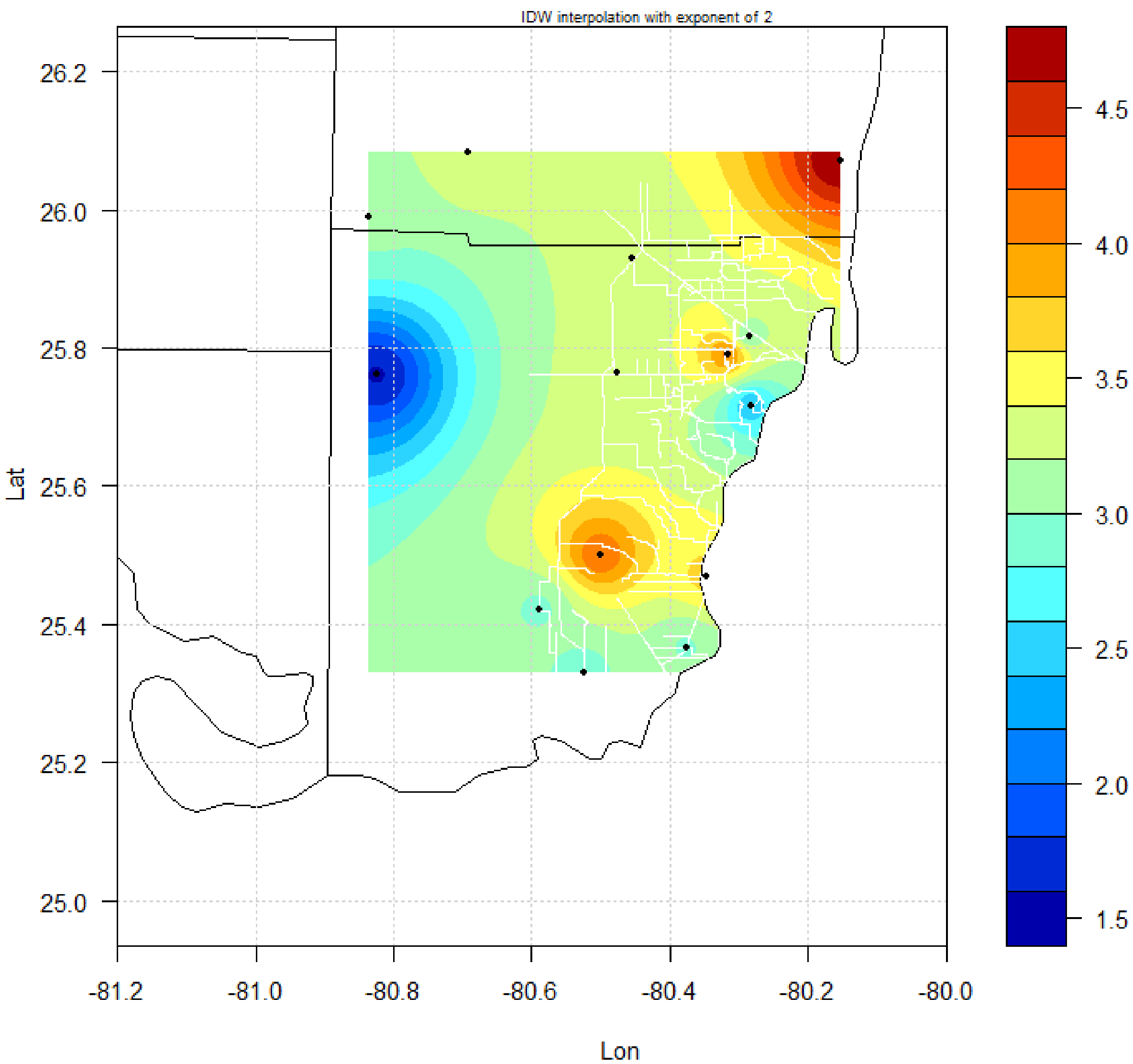


Figure 138. 95th percentile of adjusted DDF precipitation depths (inches) for the period centered in 2065 (2050-2079) minus DDF precipitation depths fit to observational data in the current baseline period (last 30 years up to 2005) for 1-hour duration, 100-year return period.

Appendix G. R code used in the extreme rainfall analysis

The R code used in the rainfall analysis (Task II) is included in this appendix. In the analysis of historical observational data up to the year 2019, only functions getAMSobs and grubbs.flag and fitGEV were called. In the bias-correction analysis of LOCA data the following functions were called in order.

- getAMSobs and grubbs.flag
- station2cellmap
- subset_loca
- getAMS
- fitGEVall which calls fitGEV
- doEQM
- computeGOFquants was called three times: to compare $F_{o\text{-}c}$ vs. $F_{m\text{-}c}$, $F_{m\text{-}c}$ vs. $F_{m\text{-}p1}$, and $F_{o\text{-}c}$ vs. $F_{m\text{-}padj.1}$
- contourmap_Tps – was used to smooth out at-station values of fitted extremes

```
############################################################################
getAMSobs <- function(){
#Function to compute AMS for various observational rainfall datasets
############################################################################

#Libraries used
library(lubridate)
library(zoo)
library(lattice)
library(tidyr)
library(outliers)
library(ggplot2)
library(stringr)
library(data.table)
library(abind)

datadir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/"
#Number of years to get data for
numys=30
#Cutoff year before which to get the numys
cutyr=2019
################################################################################
#Compute AMS from SFWMD daily rainfall data
################################################################################
setwd(paste(datadir,"/SFWMD/",sep=""))


#Definition of wet season in NOAA Atlas 14 volume 9
#For daily durations
wetseas=seq(3,10,1)
wetseasd=245

#Read in SFWMD rainfall data
#Note: Blank fields (corresponding to missing or not yet available values) are read as NA
SFWMDrain=read.csv(paste(datadir,"/SFWMD/SFWMD_data.csv",sep=""))
nstas=((ncol(SFWMDrain)-3)/2)

yrs=SFWMDrain[,2]
uyrs=unique(yrs)
ndays=365+1*leap_year(unique(yrs))
mos=SFWMDrain[,3]

#Durations of interest for SFWMD data
sdursdays=c(1,2,3,4,7)
ndurs=length(sdursdays)
ams=array(dim=c(ndurs,length(uyrs),nstas))
amsoutin=array(dim=c(length(uyrs),nstas))
amsoutin[,]=0


#SFWMD DBHydro data qualifiers
#A: accumulated rainfall
#M: Missing
#N: Not yet available
#X: Included in next amount marked A
#P: Provisional data subject to revision
#!: Normal limits exceeded

#Go through each SFWMD structure and get AMS
```

```
#Based on following flowchart in Fig. 4.3.1 of NOAA Atlas 14 vol. 9 (p.11)
for (s in 1:nstas) {
  print(paste("s=",s))
  sdata=SFWMDrain[,(2*s+2)]
  squal=SFWMDrain[,(2*s+3)]
  sdata[squal=='X']=NA
  databyqual=tapply(sdata,list('yrs'=yrs,'squal'=squal),function(x) length(x))

  #Overall percentage of zero values
  szero=tapply(sdata,yrs,function(x) sum(x==0,na.rm=TRUE))
  szero=100*szero/ndays
  #print(range(szero,na.rm=TRUE))

  #Overlal percentage of missing and zero values
  szeromiss=tapply(sdata,yrs,function(x) sum(x==0)+sum(is.na(x)))
  szeromiss=100*szeromiss/ndays
  #print(range(szeromiss,na.rm=TRUE))

  #range(sdata[squal=='M' | squal=='N'],na.rm=TRUE)
  #Overall percentage missing
  smiss=tapply(sdata,yrs,function(x) sum(is.na(x)))
  smiss=100*smiss/ndays
  print(range(szero[smiss<=2]))

  #Percentage missing for wet season
  swmiss=tapply(sdata,list('yrs'=yrs,'mos'=mos),function(x) sum(is.na(x)))
  swmiss=100*rowSums(swmiss[,wetseas],na.rm=TRUE)/wetseasd

  #Percentage accumulated
  if (is.na(match("X",colnames(databyqual)))) {
    sacc=vector(mode="integer",length=length(uyrs))
       swacc=vector(mode="integer",length=length(uyrs))
  }
  else {
    sacc=100*rowSums(databyqual[,c("X","A")],na.rm=TRUE)/ndays

    #Percentage accumulated for wet season
    swacc=tapply(sdata,list('yrs'=yrs,'mos'=mos,'squal'=squal),function(x) length(x))
    swacc=100*rowSums(swacc[,wetseas,c("X","A")],na.rm=TRUE)/wetseasd

    #Get runs with qualifiers "X" or "A", to do this first replace all "A" with "X"
    #squal2=squal
    #squal2[squal2=="A"]="X"
    #sruns=rle(as.vector(squal2))
    #saxruns=sruns$lengths[which(sxaruns$values=="X")]
    #Gives location of last element of the run
    #srunends=cumsum(sruns$length)
    #runyrs=yrs[srunends[which(sxaruns$values=="X")]]
    #Percentage of runs less than 1 day duration
    #Threshold run lengths are 1 day for durations of 1, 2 and 3 days
    #srunl=100*tapply(saxruns,list('yrs'=runyrs),function(x)
sum(x<1))/tapply(saxruns,list('yrs'=runyrs),function(x) length(x))
    #This is irrelevant, srunl will be equal to 0 for all years for durations of 1, 2,
and 3 days based on the definition of a run
    #runmos=mos[srunends[which(sxaruns$values=="X")]]
    #temp1=tapply(saxruns,list('yrs'=runyrs,'mos'=runmos),function(x) sum(x<1))
    #temp2=tapply(saxruns,list('yrs'=runyrs,'mos'=runmos),function(x) length(x))

#swrunl=100*rowSums(temp1[,wetseas],na.rm=TRUE)/rowSums(temp2[,wetseas],na.rm=TRUE)
    #swrunl[is.na(swrunl)]=0
  }
```

```
  #Check whether to compute AMS for the station for the particular year based on the
conditions above
  #amsoutin is 0 if there is not enough data to compute AMS for that year
  #amsoutin is 1 if there is enough data to compute AMS for that year
  iy=0
  for (y in uyrs) {
    iy=iy+1
      if (smiss[iy]<20 && swmiss[iy]<20 && sacc[iy] < 33 && swacc[iy] < 15) {
        amsoutin[iy,s]=1
    }
  }

  print(paste("SFWMD station DBKEY: ",colnames(SFWMDrain)[(2*s+2)],", AMS valid values:
",sum(amsoutin[,s]),sep=""))

  for (u in 1:ndurs) {
    print(paste("u = ",u,sep=""))
    k=sdursdays[u]
      #rs=tapply(sdata,list('yrs'=yrs),function(x)               ave(x,FUN=function(x)
c(rollsum(x,k),rep(NA,k-1)),k=k))
    rs=tapply(sdata,list('yrs'=yrs),function(x,k) c(rollsum(x,k),rep(NA,k-1)),k=k)
    ams[u,,s]=unlist(lapply(rs,function(x) max(as.numeric(x),na.rm=TRUE)))
    rm(rs)
  }
}
ams[ams<0]=NA

colnames(amsoutin)=colnames(SFWMDrain[seq(4,ncol(SFWMDrain)-1,2)])
rownames(amsoutin)=uyrs
dimnames(ams)[[1]]=paste(sdursdays,"-day",sep="")
dimnames(ams)[[2]]=uyrs
dimnames(ams)[[3]]=colnames(SFWMDrain[seq(4,ncol(SFWMDrain)-1,2)])

ams2=ams
#Exclude years with not enough values for accurate AMS
exclind=which(amsoutin==0,arr.ind=TRUE)
ams2[cbind(rep(1:3,each=1647),rep(exclind[,1],3),rep(exclind[,2],3))]=NA
#Check that for a certain year the AMS totals for n days are greater than the totals for
(n-1) days
#If not, set the total for n days to the total for (n-1) days
for (s in 1:nstas) {
  for (u in 1:(ndurs-1)) {
    ams2[(u+1),,s]=pmax(ams2[u,,s],ams2[(u+1),,s],na.rm=TRUE)
  }
}

#Convert the AMS from constrained to unconstrained using NOAA ATLAS14 conversion factors
amsunc=ams2
amsunc[1,,]=ams2[1,,]*1.12 #1-day
amsunc[2,,]=ams2[2,,]*1.04 #2-day
amsunc[3,,]=ams2[3,,]*1.03 #3-day
amsunc[4,,]=ams2[4,,]*1.02 #4-day
amsunc[5,,]=ams2[5,,]*1.01 #7-day
#Again, check that for a certain year the AMS totals for n days are greater than the
totals for (n-1) days
#If not, set the total for n days to the total for (n-1) days
for (s in 1:nstas) {
  for (u in 1:(ndurs-1)) {
    amsunc[(u+1),,s]=pmax(amsunc[u,,s],amsunc[(u+1),,s],na.rm=TRUE)
  }
}

#Cut off data based on cutoff year cutyr
```

```
amsoutin=amsoutin[which(as.numeric(rownames(amsoutin))<=cutyr),]
amsunc=amsunc[,which(as.numeric(rownames(amsoutin))<=cutyr),]
ams2=ams2[,which(as.numeric(rownames(amsoutin))<=cutyr),]

last30s=apply(amsunc[1,,],2,function(x)
{nvalyrs=min(numys,sum(!is.na(x)));c(rep(NA,numys-
nvalyrs),as.numeric(tail(names(x)[!is.na(x)],nvalyrs)))})
#Remove 1959 from DBKey X6310 due to AMS value being accumulated over 7 days
X6310_last30s=last30s[,"X6310"]
X6310_last30s[which(X6310_last30s==1959)]=NA
valyrs=sort(X6310_last30s)
X6310_last30s=c(rep(NA,numys-length(valyrs)),valyrs)
last30s[,"X6310"]=X6310_last30s
#Remove 2017 (Hurricane Irma) from DBKeys X6040, X6041, G6149, G6152
X6040_last30s=last30s[,"X6040"]
X6040_last30s[which(X6040_last30s==2017)]=NA
valyrs=sort(X6040_last30s)
X6040_last30s=c(rep(NA,numys-length(valyrs)),valyrs)
last30s[,"X6040"]=X6040_last30s

X6041_last30s=last30s[,"X6041"]
X6041_last30s[which(X6041_last30s==2017)]=NA
valyrs=sort(X6041_last30s)
X6041_last30s=c(rep(NA,numys-length(valyrs)),valyrs)
last30s[,"X6041"]=X6041_last30s

G6149_last30s=last30s[,"G6149"]
G6149_last30s[which(G6149_last30s==2017)]=NA
valyrs=sort(G6149_last30s)
G6149_last30s=c(rep(NA,numys-length(valyrs)),valyrs)
last30s[,"G6149"]=G6149_last30s

G6152_last30s=last30s[,"G6152"]
G6152_last30s[which(G6152_last30s==2017)]=NA
valyrs=sort(G6152_last30s)
G6152_last30s=c(rep(NA,numys-length(valyrs)),valyrs)
last30s[,"G6152"]=G6152_last30s

#Make levelplots of data availability
startyr=min(uyrs)
endyr=max(uyrs)
rngst=1:nstas
png(paste("SFWMD_AMS_data_avail_stas_",min(rngst),"_to_",max(rngst),".png",sep=""))
plot.new()
print(levelplot(amsoutin[,rngst],xlab="Year",ylab="station",cuts=1,at=c(0,0.5,1),color
key=list(at=c(0,0.5,1),tick.number=1),
          scales=list(x=list(at=seq(startyr,endyr,10)-
startyr+1,labels=seq(startyr,endyr,10),rot=90),y=list(cex=1)),
          main=(paste("SFWMD   AMS   data   availability   by   year   for   daily
durations",sep="")),
          aspect="xy",panel = function(...){
            panel.levelplot(...)
            panel.abline(h=seq(rngst)-0.5,col="grey")
            panel.abline(v=seq(startyr,endyr,10)-startyr+1,col="grey")
            #panel.abline(v=c(1950,2018)-startyr+1,col="black")
              panel.points(as.vector((last30s[,rngst])-startyr+1),rep(rngst-
min(rngst)+1,each=numys),pch='*')
          }))
dev.off()

#Extract last 20-30 years of VALID AMS data only
sfwmd_amsunc30=array(dim=c(ndurs,numys,nstas))
dimnames(sfwmd_amsunc30)[[1]]=paste(sdursdays,"-day",sep="")
```

```
dimnames(sfwmd_amsunc30)[[2]]=1:numys
dimnames(sfwmd_amsunc30)[[3]]=colnames(SFWMDrain[seq(4,ncol(SFWMDrain)-1,2)])
sfwmd_amscon30=array(dim=c(ndurs,numys,nstas))
dimnames(sfwmd_amscon30)[[1]]=paste(sdursdays,"-day",sep="")
dimnames(sfwmd_amscon30)[[2]]=1:numys
dimnames(sfwmd_amscon30)[[3]]=colnames(SFWMDrain[seq(4,ncol(SFWMDrain)-1,2)])

for (s in 1:nstas) {
  inds=match(last30s[,s],colnames(amsunc))
  sfwmd_amsunc30[,(1:length(inds)),s]=amsunc[,inds,s]
  sfwmd_amscon30[,(1:length(inds)),s]=ams2[,inds,s]
}

#Number of valid AMS values per station:
(colSums(!is.na(last30s),na.rm=TRUE))
#Eliminate the first station (5815) when cutyr=2019 since it only has 1 valid AMS value
#sfwmd_amscon30=sfwmd_amscon30[,,-1]
#sfwmd_amsunc30=sfwmd_amsunc30[,,-1]
#last30s=last30s[,-1]
#When cutyr=2005, only keep stations with more than 20 years of data available
#This is more generic
stations_to_keep=which(colSums(!is.na(last30s))>=20)
#Also remove station G6152 (NP-P33)
stations_to_keep=stations_to_keep[- which(names(stations_to_keep)=="G6152")]
sfwmd_amscon30=sfwmd_amscon30[,,stations_to_keep]
sfwmd_amsunc30=sfwmd_amsunc30[,,stations_to_keep]
last30s=last30s[,stations_to_keep]

save(sfwmd_amscon30,sfwmd_amsunc30,last30s,file=paste("SFWMD_daily_AMS_before_",cutyr,
".RData",sep=""))

#Note: grubbs.flag test was done manually for all stations and all durations
#The following years came out as outliers at many stations and therefore,
#the corresponding values were not considered outliers: 1997, 1999, 2017
#Identified outliers in 1998 and 2000 were compared against values at
#nearby stations from the SFWMD and NOAA and deemed reasonable
#2005 outlier at station 19 (DBKey H2005) also observed at NOAA station 08-8780
#and corroborated from SFWMD NEXRAD map.
#1959 outlier at DBKey 6310 is a true outlier since it is accumulation of
#7 days of data; therefore, it is removed earlier in the script



################################################################################
#Compute AMS from SFWMD breakpoint rainfall data
################################################################################
#setwd(paste(datadir,"/SFWMD/",sep=""))

#a=read.table("breakptRF.txt",quote="",stringsAsFactors=FALSE)
#stas=unique(a[,1])
#a$station=a$V1
#a$datetime=as.POSIXct(paste(a[,2],"
",str_pad(a[,3],4,side="left",pad="0"),sep=""),tryFormats=c("%Y%m%d %H%M"),tz="EST")
#a$times=strftime(a$datetime,format="%H:%M",tz="EST")
#a$dates=strftime(a$datetime,format="%m/%d/%Y",tz="EST")
#a$value=as.numeric(gsub("[^0-9.-]","",a[,4]))
#a$qual=gsub("[0-9.-]","",a[,4])
#a=a[,c("station","datetime","times","dates","value","qual")]

#for (s in stas) {
#  sdata=a[a$station==s,]
#  sdata[which(sdata$qual %in% c("N","M","?","U")),"value"]=NA
#  sdata=sdata[(2:dim(sdata)[1]),]
```

```
#  df=data.frame(datetime=sdata$datetime,value=sdata$value)
#  z=read.zoo(df,tz="EST")
#  hrs=trunc(time(z),"hours")
#  dt=data.table(value=sdata$value,hrs=hrs)
#  hrsum=dt[, sum(value), keyby=hrs]
#  rng=range(hrs)
#  tt=seq(trunc(rng[1],"days"),trunc(rng[2]+24*60*60,"days"),by="hours")
#  tt=tt[!(format(tt) %in% format(hrs))]
#  eee=zoo(hrsum$V1,hrsum$hrs)
#  hrsum2=merge(eee,zoo(,tt),fill=0)

  #sdata[which(is.na(sdata$value)),"value"]=-901

#write.table(sdata,file=paste(s,"_breakpoint.csv",sep=""),sep=",",row.names=FALSE,quot
e=FALSE)
#}


################################################################################
#Compute AMS from SFWMD hourly rainfall data
################################################################################
#Note: Don't do this part when cutyr=2005 since no station will have enough data left

setwd(paste(datadir,"/SFWMD/",sep=""))
fils=list.files(".","*hourly.dat$")

snames=vector(length=length(fils),mosde="character")
sfwmd_hrlysta_keep=vector(length=length(fils),mode="integer")

#Overall range of years
minyr=1991
maxyr=2019
rngyrs=seq(minyr,maxyr)

#Create large dataframe to store data for all stations
sfwmd_hrlydata=data.frame(datehrs=seq(from=as.POSIXct("1991-1-1 0:00", tz="EST"),
     to=as.POSIXct("2019-12-31 23:00", tz="EST"),
     by="hour") )

#Determine whether there's enough data for a "reasonable" AMS calculation for that year
sfhamsoutin=array(dim=c(length(rngyrs),length(fils)))
sfhamsoutin[,]=0

s=0
for (fil in fils) {
  s=s+1

hrlydata=read.fortran(fil,format=c("A10","13X","A3","2X","A12","1X","A12","4X","F6","1
X","A1","8X","F6"),

col.names=c("staid","stat","sdate","edate","rain","qual","perc"))
  hrlydata[,1]=gsub(" ", "", hrlydata[1,1], fixed = TRUE)
  hrlydata$sdate=as.POSIXct(hrlydata$sdate,tryFormats=c("%Y%m%d%H%M"),tz="EST")
  hrlydata$edate=as.POSIXct(hrlydata$edate,tryFormats=c("%Y%m%d%H%M"),tz="EST")
  #For any M if perc > 10%, change the value to NA
  hrlydata[which(hrlydata[,"perc"]>10),"rain"]=NA

  #Add year, month, day
  hrlydata$Year=as.POSIXlt(hrlydata$sdate)$year+1900
  hrlydata$Month=as.POSIXlt(hrlydata$sdate)$mo+1
  hrlydata$Day=as.POSIXlt(hrlydata$sdate)$mday

  #Save station name
```

```
  snames[s]=hrlydata[1,1]

  #Save station data
  sfwmd_hrlydata[,s+1]=NA
  colnames(sfwmd_hrlydata)[s+1]=snames[s]

sfwmd_hrlydata[which(sfwmd_hrlydata$datehrs%in%hrlydata$sdate),snames[s]]=hrlydata$rai
n


  #Based on following flowchart in Fig. 4.3.1 of NOAA Atlas 14 vol. 9 (p.11)
  #SFWMD hourly does not have accumulated data, so flowchart is simplified
  #Percentage missing for entire year
  uyrs=unique(hrlydata$Year)
  ndays=365+1*leap_year(uyrs)
  sfwmdhavail=tapply(hrlydata$rain,list('yrs'=hrlydata$Year),function(x)
sum(!is.na(x)))
  sfwmdhmiss=100*(ndays*24-sfwmdhavail)/(ndays*24)

  #Percentage missing for wet season
  #Wet season defined as May-Oct for sub-daily durations
  hwetseas=seq(5,10,1)
  hwetseasd=184

sfwmd_whavail=tapply(hrlydata$Year,list('yrs'=hrlydata$Year,'mos'=hrlydata$Month),func
tion(x) sum(!is.na(x)))
  sfwmd_whmiss=100*(hwetseasd*24-
rowSums(sfwmd_whavail[,hwetseas],na.rm=TRUE))/(hwetseasd*24)
  #sum(sfwmd_whmiss<=20)

  #Wet season defined as Mar-Oct for daily durations
  dwetseas=seq(3,10,1)
  dwetseasd=245

sfwmd_wdavail=tapply(hrlydata$Year,list('yrs'=hrlydata$Year,'mos'=hrlydata$Month),func
tion(x) sum(!is.na(x)))
  sfwmd_wdmiss=100*(dwetseasd*24-
rowSums(sfwmd_wdavail[,dwetseas],na.rm=TRUE))/(dwetseasd*24)
  #sum(sfwmd_wdmiss<=20)

  #nvalidyrs=sum((sfwmdhmiss<=20)*(sfwmd_whmiss<=20)*(sfwmd_wdmiss<=20))
  #Determine whether there's enough data for a "reasonable" AMS calculation for that
year
  for (iy in rngyrs) {
    if (iy%in%names(sfwmdhmiss)) {
         if(sfwmdhmiss[names(sfwmdhmiss)==iy]<=20                                    &
sfwmd_whmiss[names(sfwmd_whmiss)==iy]<=20 & sfwmd_wdmiss[names(sfwmd_wdmiss)==iy] <=20)
{
           sfhamsoutin[(iy-min(rngyrs)+1),s]=1
         }
    }
  }

  nvalidyrs=sum(sfhamsoutin[,s])

  if (nvalidyrs>=20) {
         sfwmd_hrlysta_keep[s]=1
         print(paste("Keep station: ",snames[s],", # valid years: ",nvalidyrs,sep=""))
   }
   else {
         sfwmd_hrlysta_keep[s]=0
         print(paste("Remove station: ",snames[s],", # valid years: ",nvalidyrs,sep=""))
```

```
  }

}

dimnames(sfhamsoutin)[[1]]=rngyrs
dimnames(sfhamsoutin)[[2]]=snames

#Get last 30 years of data for each station
numys=30
last30sh=apply(sfhamsoutin,2,function(x)  {nvalyrs=min(numys,sum(x==1));c(rep(NA,numys-
nvalyrs),as.numeric(tail(names(x)[x==1],nvalyrs)))})

#Make levelplots of data availability
startyr=minyr
endyr=maxyr
rngst=1:length(fils)
png(paste("SFWMD_hourly_AMS_data_avail_stas_",min(rngst),"_to_",max(rngst),".png",sep=
""))
plot.new()
print(levelplot(sfhamsoutin[,rngst],xlab="Year",ylab="station",cuts=1,at=c(0,0.5,1),co
lorkey=list(at=c(0,0.5,1),tick.number=1),
          scales=list(x=list(at=seq(startyr,endyr,10)-
startyr+1,labels=seq(minyr,endyr,10),rot=90),y=list(cex=1)),
          main=(paste("SFWMD hourly AMS data availability by year",sep="")),
          aspect="xy",panel = function(...){
            panel.levelplot(...)
            panel.abline(h=seq(rngst)-0.5,col="grey")
            panel.abline(v=seq(startyr,endyr,10)-startyr+1,col="grey")
            #panel.abline(v=c(1950,2018)-startyr+1,col="black")
              panel.points(as.vector((last30sh[,rngst])-startyr+1),rep(rngst-
min(rngst)+1,each=numys),pch='*')
          }))
dev.off()

#Only keep stations with more than 20 years of valid AMS data
stations_to_keep=which(apply(sfhamsoutin,2,sum)>=20)
#Exclude S18C-R and S20F-R since they're already in ATLAS14
stations_to_keep=stations_to_keep[-which(names(stations_to_keep)%in%c("S18C-R","S20F-
R"))]
nstas=length(stations_to_keep)
sfwmd_hrlydata=sfwmd_hrlydata[,c(1,stations_to_keep+1)] #1 is offset since r3wasd has
date, time, date.time in cols 1-3
sfhamsoutin=sfhamsoutin[,stations_to_keep]
last30sh=last30sh[,which(colnames(last30sh)%in%names(stations_to_keep))]

#Get constrained AMS
shdurs=c(1,2,3,6,12,24,48,72,96,168) #durations of interest in hours
shdursnames=c("1-hr","2-hr","3-hr","6-hr","12-hr","1-day","2-day","3-day","4-day","7-
day")
ndurs=length(shdurs)
ams2h=array(dim=c(ndurs,length(rngyrs),nstas))
for (u in 1:ndurs) {
  print(paste("u = ",u,sep=""))
  k=shdurs[u]
  for (s in 1:nstas) {
    print(paste("s = ",s,sep=""))

rs=tapply(sfwmd_hrlydata[,(s+1)],list('yrs'=(as.POSIXlt(sfwmd_hrlydata$datehrs)$year+1
900)),function(x,k) c(rollsum(x,k),rep(NA,k-1)),k=k)
    ams2h[u,,s]=unlist(lapply(rs,function(x) max(as.numeric(x),na.rm=TRUE)))
    rm(rs)
  }
}
```

```
#Exclude years with not enough values for accurate AMS
exclind=which(sfhamsoutin==0,arr.ind=TRUE)
ams2h[cbind(rep(1:ndurs,each=dim(exclind)[1]),rep(exclind[,1],ndurs),rep(exclind[,2],n
durs))]=NA

#Check that for a certain year the AMS totals for n days are greater than the totals for
(n-1) days
#If not, set the total for n days to the total for (n-1) days
for (s in 1:nstas) {
  for (u in 1:(ndurs-1)) {
    ams2h[(u+1),,s]=pmax(ams2h[u,,s],ams2h[(u+1),,s],na.rm=TRUE)
  }
}

#Apply factors to go from constrained to constrained observations
#no corrections applied beyond 12 hours
corrfac=c(1.09,1.04,1.02,1.01,1.00,1.00,1.00,1.00,1.00,1.00)
amsunch=array(dim=c(ndurs,length(rngyrs),nstas))
for (u in 1:ndurs) {
  amsunch[u,,]=ams2h[u,,]*corrfac[u]
}

#Again, check that for a certain year the AMS totals for n days are greater than the
totals for (n-1) days
#If not, set the total for n days to the total for (n-1) days
for (s in 1:nstas) {
  for (u in 1:(ndurs-1)) {
    amsunch[(u+1),,s]=pmax(amsunch[u,,s],amsunch[(u+1),,s],na.rm=TRUE)
  }
}

dimnames(ams2h)[[1]]=dimnames(amsunch)[[1]]=shdursnames
dimnames(ams2h)[[2]]=dimnames(amsunch)[[2]]=rngyrs
dimnames(ams2h)[[3]]=dimnames(amsunch)[[3]]=names(stations_to_keep)

#Extract last 20-30 years of VALID AMS data only
sfwmdh_amsunc30=array(dim=c(ndurs,numys,length(stations_to_keep)))
sfwmdh_amscon30=array(dim=c(ndurs,numys,length(stations_to_keep)))
dimnames(sfwmdh_amsunc30)[[1]]=dimnames(sfwmdh_amscon30)[[1]]=shdursnames
dimnames(sfwmdh_amsunc30)[[2]]=dimnames(sfwmdh_amscon30)[[2]]=1:numys
dimnames(sfwmdh_amsunc30)[[3]]=dimnames(sfwmdh_amscon30)[[3]]=dimnames(amsunch)[[3]]


for (s in 1:nstas) {
  inds=match(last30sh[,s],colnames(amsunch))
  sfwmdh_amsunc30[,(1:length(inds)),s]=amsunch[,inds,s]
  sfwmdh_amscon30[,(1:length(inds)),s]=ams2h[,inds,s]
}

#Number of valid AMS values per station:
(colSums(!is.na(last30sh),na.rm=TRUE))

save(sfwmdh_amscon30,sfwmdh_amsunc30,last30sh,file=paste("SFWMD_hrly_AMS_before_",cuty
r,".RData",sep=""))


#Note: grubbs.flag test was done manually for all stations and all durations
#The following years came out as outliers at many stations and therefore,
#the corresponding values were not considered outliers:  1999 (H. Irene),
#2000 (TS Leslie), 2005 (H Katrina).
#Identified outliers not associated to a named storm were compared against values at
#nearby stations from SFWMD and NOAA and corroborated from SFWMD NEXRAD map.
```

```
#####################################################################################
#Read in MDWASD data
#####################################################################################
# setwd(paste(datadir,"/MDC/",sep=""))

# WASDrain=read.csv("MDCRainGaugeHourlyall.csv",stringsAsFactors=FALSE)
# WASDrain$Reading.Date=as.Date(WASDrain$Reading.Date,"%m/%d/%Y")
#
r1wasd=reshape(WASDrain,idvar=c("Station.Id","Reading.Date"),varying=list(3:26),v.name
s="Rain",direction="long")
#
r2wasd=reshape(r1wasd,v.names="Rain",idvar=c("Reading.Date","time"),timevar="Station.I
d",direction="wide")
# r2wasd=r2wasd[order(r2wasd$Reading.Date),]
#         r2wasd=cbind(r2wasd[,1:2],as.POSIXct(paste(r2wasd[,1],"        ",(r2wasd[,2]-
1),":00",sep=""),tz="EST"),r2wasd[,3:ncol(r2wasd)])
# colnames(r2wasd)[3]="Date.time"

# #Make negative values equal to NA (missing)
# r2wasd[,c(4:ncol(r2wasd))][r2wasd[,(4:ncol(r2wasd))] < 0] = NA

# #Initially make hourly values greater than 1-in-1000 hourly rainfall for NOAA
# #ATLAS14 stations in MDC (8.5 in/hr) equal to NA
# r2wasd[,c(4:ncol(r2wasd))][r2wasd[,(4:ncol(r2wasd))] > 8.5] = NA

# #Fill-in missing dates
#      WASDdates=seq(as.POSIXct("1995-01-01      0:00",tz="EST"),as.POSIXct("2019-03-28
23:00",tz="EST"),"hours")

# r3wasd=as.data.frame(array(dim=c(length(WASDdates),ncol(r2wasd))))
# dimnames(r3wasd)[[2]]=dimnames(r2wasd)[[2]]
# r3wasd[,1]=as.Date(WASDdates,tz="EST")
# r3wasd[,2]=as.POSIXlt(WASDdates)$hour
# r3wasd[,3]=WASDdates
# inds=which(r3wasd$Date.time %in% r2wasd$Date.time)

# r3wasd[inds,c(4:ncol(r2wasd))]=r2wasd[,(4:ncol(r2wasd))]

# #Get daily values and if a daily total exceeds the 1-in-1000 daily rainfall
# #for NOAA ATLAS14 stations in MDC (26 in/day), then set it to NA
#                             wasd_daily=apply(r3wasd[,(4:ncol(r3wasd))],2,function(x)
tapply(x,list('date'=r3wasd$Reading.Date),
                 # function(x) if (sum(!is.na(x)) == 0) (NA) else (sum(x,na.rm=TRUE)) )
)
# wasd_daily[wasd_daily > 26] = NA
# WASDdays=unique(sort(as.Date(WASDdates,tz="EST")))

# #Now remove stations with all daily values equal to NA from both wasd_daily and r3wasd
# stas_to_remove = -c(which(apply(wasd_daily,2,function(x) sum(!is.na(x))) == 0))
# wasd_daily=wasd_daily[,stas_to_remove]
# r3wasd=r3wasd[,stas_to_remove-3] #-3 is offset since r3wasd has date, time, date.time
in cols 1-3

# #Remove all hourly values for days when daily is NA
# for (i in 1:ncol(wasd_daily)) {
  # #WASDdays[which(is.na(wasd_daily[,i]))]
  # print(paste("station #",i,",station name:",colnames(wasd_daily)[i]))
  # r3wasd[r3wasd[,1]%in%WASDdays[which(is.na(wasd_daily[,i]))],(i+3)]=NA
# }


# yrs=as.numeric(format(r3wasd$Reading.Date,"%Y"))
```

```
# mos=as.numeric(format(r3wasd$Reading.Date,"%m"))
# uyrs=unique(yrs)
# ndays=365+1*leap_year(unique(yrs))

# #Based on following flowchart in Fig. 4.3.1 of NOAA Atlas 14 vol. 9 (p.11)
# #MDCWASD does not have accumulated data, so flowchart is simplified
# #Percentage missing for entire year
#                                 wasdmiss=apply(r3wasd[,(4:ncol(r3wasd))],2,function(x)
tapply(x,list('yrs'=yrs),function(x) sum(is.na(x))))
# wasdmiss=100*wasdmiss/(ndays*24)
# #apply(wasdmiss,2,function(x) sum(x<=20))

# #Additional checks of zeros since it seems like a lot of missing values are set to 0
in WASD dataset
#                                 wasdzero=apply(r3wasd[,(4:ncol(r3wasd))],2,function(x)
tapply(x,list('yrs'=yrs),function(x) sum(x==0,na.rm=TRUE)))
# wasdzero=100*wasdzero/(ndays*24)
#                             wasdzeromiss=apply(r3wasd[,(4:ncol(r3wasd))],2,function(x)
tapply(x,list('yrs'=yrs),function(x) sum(x==0,na.rm=TRUE)+sum(is.na(x))))
# wasdzeromiss=100*wasdzeromiss/(ndays*24)

# #Percentage missing for wet season
# #Wet season defined as May-Oct for sub-daily durations
# hwetseas=seq(5,10,1)
# hwetseasd=184
#                               wasd_whmiss=lapply(r3wasd[,(4:ncol(r3wasd))],function(x)
tapply(x,list('yrs'=yrs,'mos'=mos),function(x) sum(is.na(x))))
#                              wasd_whmiss=simplify2array(lapply(wasd_whmiss,function(x)
100*rowSums(x[,hwetseas],na.rm=TRUE)/(hwetseasd*24)))
# #apply(wasd_whmiss,2,function(x) sum(x<=20))

# #Wet season defined as Mar-Oct for daily durations
# dwetseas=seq(3,10,1)
# dwetseasd=245
#                               wasd_wdmiss=lapply(r3wasd[,(4:ncol(r3wasd))],function(x)
tapply(x,list('yrs'=yrs,'mos'=mos),function(x) sum(is.na(x))))
#                              wasd_wdmiss=simplify2array(lapply(wasd_wdmiss,function(x)
100*rowSums(x[,dwetseas],na.rm=TRUE)/(dwetseasd*24)))
# #apply(wasd_wdmiss,2,function(x) sum(x<=20))

# #Determine whether there's enough data for a "reasonable" AMS calculation for that
year
# wamsoutin=array(dim=dim(wasdmiss))
# wamsoutin[,]=0
# dimnames(wamsoutin)=dimnames(wasdmiss)
# for (iy in 1:nrow(wasdmiss)) {
  # for (s in 1:ncol(wasdmiss)) {
    #  if  (wasdmiss[iy,s]<=20  &  wasd_whmiss[iy,s]<=20  &  wasd_wdmiss[iy,s]  <=20  &
wasdzeromiss[iy,s]<100) {
        # wamsoutin[iy,s]=1
      # }
  # }
# }


# #6 stations to keep so far
#   #stations_to_keep=which(apply(wasdmiss,2,function(x)   sum(x<=20))   >=   20   &
apply(wasd_whmiss,2,function(x) sum(x<=20)) >= 20 &
# #     apply(wasd_wdmiss,2,function(x) sum(x<=20)) >= 20)
# stations_to_keep=which(apply(wamsoutin,2,sum)>=20)
# r3wasd=r3wasd[,c(1:3,stations_to_keep+3)] #3 is offset since r3wasd has date, time,
date.time in cols 1-3
# wasd_daily=wasd_daily[,stations_to_keep]
```

```
# wamsoutin=wamsoutin[,stations_to_keep]

# #Get constrained AMS
# wdurs=c(1,2,3,6,12,24,48,72,96,168) #durations of interest in hours
#  wdursnames=c("1-hr","2-hr","3-hr","6-hr","12-hr","1-day","2-day","3-day","4-day","7-day")
# wasd_amscon=array(dim=c(length(wdurs),length(uyrs),length(stations_to_keep)))
# for (u in 1:length(wdurs)) {
  # print(paste("u = ",u,sep=""))
  # k=wdurs[u]
  # for (s in 1:length(stations_to_keep)) {
    # print(paste("s = ",s,sep=""))
    #  rs=tapply(r3wasd[,(s+3)],list('yrs'=yrs),function(x,k)  c(rollsum(x,k),rep(NA,k-1)),k=k)
    # wasd_amscon[u,,s]=unlist(lapply(rs,function(x) max(as.numeric(x),na.rm=TRUE)))
    # rm(rs)
  # }
# }

# #Exclude years with not enough values for accurate AMS
# exclind=which(wamsoutin==0,arr.ind=TRUE)
#
wasd_amscon[cbind(rep(1:length(wdurs),each=dim(exclind)[1]),rep(exclind[,1],length(wdurs)),rep(exclind[,2],length(wdurs)))]=NA

# #Get percentage of daily totals equal to zero
# dyrs=format(WASDdays,"%Y")
#    wasddzero=apply(wasd_daily,2,function(x)    tapply(x,list('dyrs'=dyrs),function(x) sum(x==0,na.rm=TRUE)))
# wasddzero=100*wasddzero/(ndays)

# #Exclude years with more than 90% of days with zero rainfall
# exclind=which(wasddzero>=90,arr.ind=TRUE)
#
wasd_amscon[cbind(rep(1:length(wdurs),each=dim(exclind)[1]),rep(exclind[,1],length(wdurs)),rep(exclind[,2],length(wdurs)))]=NA


# #Check that for a certain year the AMS totals for n days are greater than the totals for (n-1) days
# #If not, set the total for n days to the total for (n-1) days
# for (s in 1:length(stations_to_keep)) {
  # for (u in 1:(length(wdurs)-1)) {
    # wasd_amscon[(u+1),,s]=pmax(wasd_amscon[u,,s],wasd_amscon[(u+1),,s],na.rm=TRUE)
  # }
# }

# #Apply factors to go from constrained to constrained observations
# #no corrections applied beyond 12 hours
# corrfac=c(1.09,1.04,1.02,1.01,1.00,1.00,1.00,1.00,1.00,1.00)
# wasd_amsunc=array(dim=c(length(wdurs),length(uyrs),length(stations_to_keep)))
# for (u in 1:(length(wdurs))) {
  # wasd_amsunc[u,,]=wasd_amscon[u,,]*corrfac[u]
# }

# #Again, check that for a certain year the AMS totals for n days are greater than the totals for (n-1) days
# #If not, set the total for n days to the total for (n-1) days
# for (s in 1:length(stations_to_keep)) {
  # for (u in 1:(length(wdurs)-1)) {
    # wasd_amsunc[(u+1),,s]=pmax(wasd_amsunc[u,,s],wasd_amsunc[(u+1),,s],na.rm=TRUE)
  # }
# }
```

```
# dimnames(wasd_amscon)[[1]]=dimnames(wasd_amsunc)[[1]]=wdursnames
# dimnames(wasd_amscon)[[2]]=dimnames(wasd_amsunc)[[2]]=uyrs
# dimnames(wasd_amscon)[[3]]=dimnames(wasd_amsunc)[[3]]=names(stations_to_keep)

# #Calculated AMS data for the remaining 6 stations after QA/QC is highly suspect
# #due to low totals for some years for large durations and repeated totals
# #across many durations, so MDWASD data will be excluded from analysis
# save(wasd_amscon,wasd_amsunc,file="WASD_AMS.RData")

############################################################################
#Read in NOAA ATLAS 14 AMS data for stations in and around MDC
############################################################################
durs=c("5-min","10-min","15-min","30-min","60-min","2-hr","3-hr","6-hr","12-hr",
       "24-hr","2-day","3-day","4-day","7-day","10-day","20-day","30-day","45-
day","60-day");
durint=5:14 #durations of interest
subdurs=durs[durint]
startyr=1840
endyr=2012
nyrs=endyr-startyr+1
allyrs=startyr:endyr
ndurs=length(subdurs)
setwd("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/ATLAS14/AMS")

load(file="FL_Atlas14_AMScorr.RData",verbose=TRUE)
#AMS4 (already corrected to unconstrained), AMSDATE, staid2

MDC_ATLAS14=read.table("MDC_ATLAS14.txt",stringsAsFactors = FALSE)
#find index of AMS4 corresponding to each station listed in MDC_ATLAS14
listids=match(MDC_ATLAS14[,1],staid2)

# Plot AMS data availability at each station for each durations
# for durations 60-min to 3-day (#5 to #12)
av2=array(dim=c(nyrs,length(listids),length(durs[durint])))
dimnames(av2)[[1]]=allyrs
dimnames(av2)[[2]]=staid2[listids]
dimnames(av2)[[3]]=durs[durint]
av2[,,]=0
#Get last 30 years of data
#Use the same last 30 years for all stations
last30=array(dim=c(numys,length(listids)))
#last30=array(dim=c(length(subdurs),numys,length(listids)))
#dimnames(last30)[[1]]=durs[durint]
#dimnames(last30)[[2]]=1:numys
#dimnames(last30)[[3]]=staid2[listids]
#id=0
#for (d in subdurs) {
#  id=id+1
#  # av2 has 0 where no AMS data is available and 1 if AMS data is available
#  for (i in 1:length(listids)){
#    sdata=AMS4[[listids[i]]]
#    if (any(colnames(sdata)==d)) {
#      av2[as.numeric(rownames(sdata))-startyr+1,i,id]=1*!is.na(sdata[,d])
#       nvalyrs=min(numys,sum(!is.na(sdata[,d])))
#       filly=numys-nvalyrs
#
last30[id,,i]=c(rep(NA,filly),tail(as.numeric(rownames(sdata)[!is.na(sdata[,d])]),nval
yrs))
#      }
#  }
```

```
#  m=1
#  rngst=1:length(listids)
#
png(paste("Atlas14_AMS_data_avail_stas_",min(rngst),"_to_",max(rngst),"_",d,".png",sep
=""))
#  plot.new()
#
print(levelplot(av2[,rngst,d],xlab="Year",ylab="station",cuts=1,at=c(0,0.5,1),colorkey
=list(at=c(0,0.5,1),tick.number=1),
#                                          scales=list(x=list(at=seq(startyr,endyr,10)-
startyr+1,labels=seq(startyr,endyr,10),rot=90),y=list(cex=1)),
#          main=(paste("Atlas 14 AMS data availability by year for duration ",d,sep="")),
#            aspect="xy",panel = function(...){
#              panel.levelplot(...)
#              panel.abline(h=seq(rngst)-0.5,col="grey")
#              panel.abline(v=seq(startyr,endyr,10)-startyr+1,col="grey")
#              panel.abline(v=c(1950,2012)-startyr+1,col="black")
#                     panel.points(as.vector((last30[d,,rngst])-startyr+1),rep(rngst-
min(rngst)+1,each=numys),pch='*')
#            }))
#  dev.off()
#}

for (i in 1:length(listids)) {
  sdata=AMS4[[listids[i]]]
  sdata=sdata[as.numeric(rownames(sdata))<=cutyr,]
  sumd=rowSums(sdata[,which(colnames(sdata)%in%subdurs)])
  nvalyrs=min(numys,sum(!is.na(sumd)))
  print(paste("station=",MDC_ATLAS14[i,1],", nvalyrs=",nvalyrs))
  filly=numys-nvalyrs
  last30[,i]=c(rep(NA,filly),tail(as.numeric(names(sumd)[!is.na(sumd)]),nvalyrs))
}

dimnames(last30)[[1]]=1:numys
dimnames(last30)[[2]]=staid2[listids]

id=0
for (d in subdurs) {
  id=id+1
  # av2 has 0 where no AMS data is available and 1 if AMS data is available
  for (i in 1:length(listids)){
    sdata=AMS4[[listids[i]]]
    if (any(colnames(sdata)==d)) {
      av2[as.numeric(rownames(sdata))-startyr+1,i,id]=1*!is.na(sdata[,d])
    }
  }

  last30d=last30
  if (any(colSums(av2[,,dimnames(av2)[[3]]==d])==0)) {
    last30d[,which(colSums(av2[,,dimnames(av2)[[3]]==d])==0)]=NA
  }
  m=1
  rngst=1:length(listids)

png(paste("Atlas14_AMS_data_avail_stas_",min(rngst),"_to_",max(rngst),"_",d,".png",sep
=""))
  plot.new()

print(levelplot(av2[,rngst,d],xlab="Year",ylab="station",cuts=1,at=c(0,0.5,1),colorkey
=list(at=c(0,0.5,1),tick.number=1),
            scales=list(x=list(at=seq(startyr,endyr,10)-
startyr+1,labels=seq(startyr,endyr,10),rot=90),y=list(cex=1)),
```

```
            main=(paste("Atlas  14  AMS  data  availability  by  year  for  duration
",d,sep="")),
            aspect="xy",panel = function(...){
              panel.levelplot(...)
              panel.abline(h=seq(rngst)-0.5,col="grey")
              panel.abline(v=seq(startyr,endyr,10)-startyr+1,col="grey")
              panel.abline(v=c(1950,2012)-startyr+1,col="black")
                      panel.points(as.vector((last30d[,rngst])-startyr+1),rep(rngst-
min(rngst)+1,each=numys),pch='*')
            }))
  dev.off()
}



#Extract last 20-30 years of VALID AMS data only at NOAA Atlas 14 stations
atlas14_amsunc30=array(dim=c(length(subdurs),numys,length(listids)))
#id=0
#for (d in subdurs) {
#  id=id+1
#  # av2 has 0 where no AMS data is available and 1 if AMS data is available
#  for (i in 1:length(listids)){
#    sdata=AMS4[[listids[i]]]
#    cind=match(d,colnames(sdata))
#    if (!is.na(cind)) {
#         rinds=match(last30[id,,i],as.numeric(rownames(sdata)))
#         atlas14_amsunc30[id,((numys-length(rinds)+1):numys),i]=sdata[rinds,cind]
#      }
#  }
#}


# av2 has 0 where no AMS data is available and 1 if AMS data is available
id=0
for (d in subdurs) {
  id=id+1
  for (i in 1:length(listids)){
    sdata=AMS4[[listids[i]]]
    cind=match(d,colnames(sdata))
    if (!is.na(cind)) {
      rinds=match(last30[,i],as.numeric(rownames(sdata)))
      atlas14_amsunc30[id,((30-length(rinds)+1):numys),i]=sdata[rinds,cind]
    }
  }
}

#Check that for a certain year the AMS totals for a duration are greater than the totals
#for the previous shorter duration
#If not, set the total for the duration to the total for the previous shorter duration
for (i in 1:length(listids)) {
  for (u in 1:(ndurs-1)) {

atlas14_amsunc30[(u+1),,i]=pmax(atlas14_amsunc30[u,,i],atlas14_amsunc30[(u+1),,i],na.r
m=TRUE)
  }
}

stations_to_keep=which(colSums(!is.na(last30))>=20)
atlas14_amsunc30=atlas14_amsunc30[,,stations_to_keep]
last30=last30[,stations_to_keep]

dimnames(atlas14_amsunc30)[[1]]=durs[durint]
dimnames(atlas14_amsunc30)[[2]]=dimnames(last30)[[1]]
```

```
dimnames(atlas14_amsunc30)[[3]]=dimnames(last30)[[2]]



###############################################################################
#Compute AMS from hourly FAWN data at station 440 (Homestead)
#To do: Check if this is the same as NOAA's 08-4091 (Homestead Exp. Stn.)
###############################################################################
#setwd(paste(datadir,"/FAWN",sep=""))
#write("Data for FAWN station 440",file="data440.csv")
#for (y in 1997:2018){
#  fils=list.files(paste("./",y,"_hourly/",sep=""))
#  for (fil in fils) {
#    eee=read.csv(paste("./",y,"_hourly/",fil,sep=""))
#
       write.table(eee[eee[,1]=="440",c(1,2,19)],file="data440.csv",append=TRUE,col.na
mes=FALSE,row.names=FALSE,sep=",")
#  }
#}

#data440=read.csv("data440.csv",skip=1,header=FALSE,stringsAsFactors=FALSE)
#data440[,2]=as.POSIXct(data440[,2],format="%Y-%m-%d %H:%M:%OS",tz="EST")
#colnames(data440)=c("station","date_time","rain")
#datehrs=seq(from=as.POSIXct("1998-1-1 0:00", tz="EST"),
#     to=as.POSIXct("2018-12-31 23:00", tz="EST"),
#     by="hour"
#   )

#mydf=as.data.frame(data440 %>% complete(date_time=datehrs))

#yrs=as.numeric(format(mydf$date_time,"%Y"))
#mos=as.numeric(format(mydf$date_time,"%m"))
#uyrs=unique(yrs)
#ndays=365+1*leap_year(unique(yrs))

#Based on following flowchart in Fig. 4.3.1 of NOAA Atlas 14 vol. 9 (p.11)
#FAWN does not have accumulated data, so flowchart is simplified
#Percentage missing for entire year
#fmiss=tapply((mydf[,3]),list('yrs'=yrs),function(x) sum(is.na(x)))
#fmiss=100*fmiss/(ndays*24)
#All years with less than 20% of data missing

#Percentage missing for wet season
#Wet season defined as May-Oct for sub-daily durations
#hwetseas=seq(5,10,1)
#hwetseasd=184
#fwhmiss=tapply(mydf[,3],list('yrs'=yrs,'mos'=mos),function(x) sum(is.na(x)))
#fwhmiss=100*rowSums(fwhmiss[,hwetseas],na.rm=TRUE)/(hwetseasd*24)
#All years with less than 20% of wet season data missing

#Wet season defined as Mar-Oct for daily durations
#dwetseas=seq(3,10,1)
#dwetseasd=245
#fwdmiss=tapply(mydf[,3],list('yrs'=yrs,'mos'=mos),function(x) sum(is.na(x)))
#fwdmiss=100*rowSums(fwdmiss[,dwetseas],na.rm=TRUE)/(dwetseasd*24)
##All years with less than 20% of wet season data missing

#All years valid, then compute AMS
#fdurs=c(1,2,3,6,12,24,48,72) #durations of interest in hours
#fdursnames=c("1-hr","2-hr","3-hr","6-hr","12-hr","1-day","2-day","3-day")
#factors to go from constrained to constrained observations
#no corrections applied beyond 12 hours
#corrfac=c(1.09,1.04,1.02,1.01,1.00,1.00,1.00,1.00)
```

```
#fawn_ams=array(dim=c(length(fdurs),length(uyrs),1))
#for (u in 1:length(fdurs)) {
#  print(paste("u = ",u,sep=""))
#  k=fdurs[u]
#  rs=tapply(mydf[,3],list('yrs'=yrs),function(x,k) c(rollsum(x,k),rep(NA,k-1)),k=k)
#  fawn_ams[u,,1]=unlist(lapply(rs,function(x) max(as.numeric(x),na.rm=TRUE)))
#  rm(rs)
#}
#dimnames(fawn_ams)[[1]]=fdursnames
#dimnames(fawn_ams)[[2]]=uyrs
#dimnames(fawn_ams)[[3]]="440"

#Get unconstrained AMS from constrained AMS and correction factors from NOAA Atlas 14
Volume 9
#fawn_amsunc=fawn_ams
#Make sure every AMS for a duration is greater than AMS for the previous smaller duration
#for (u in 1:length(fdurs)) {
#  print(paste("u = ",u,sep=""))
#  if (u==1) {
#    fawn_amsunc[u,,1]=fawn_ams[u,,1]*corrfac[u]
#  } else {
#    fawn_amsunc[u,,1]=pmax(fawn_ams[u,,1]*corrfac[u],fawn_amsunc[(u-1),,1])
#  }
#}

#Same as NOAA station, so exclude

#######################################################################
# Merge SFWMD and NOAA ATLAS 14 datasets into a single dataset
#######################################################################
mdc_amsunc30=abind(atlas14_amsunc30,sfwmdh_amsunc30,along=3)
temp=abind(array(dim=c(5,dim(sfwmd_amsunc30)[2],dim(sfwmd_amsunc30)[3])),sfwmd_amsunc3
0,along=1)
mdc_amsunc30=abind(mdc_amsunc30,temp,along=3)

dimnames(mdc_amsunc30)[[1]]=dimnames(atlas14_amsunc30)[[1]]
dimnames(mdc_amsunc30)[[2]]=dimnames(atlas14_amsunc30)[[2]]
dimnames(mdc_amsunc30)[[3]]=c(dimnames(atlas14_amsunc30)[[3]],dimnames(sfwmdh_amsunc30
)[[3]],dimnames(sfwmd_amsunc30)[[3]])

mdc_last30=abind(last30,last30sh,last30s,along=2)
dimnames(mdc_last30)[[1]]=dimnames(last30)[[1]]
dimnames(mdc_last30)[[2]]=c(dimnames(last30)[[2]],dimnames(last30sh)[[2]],dimnames(las
t30s)[[2]])

#When cutyr=2005 do this instead
#temp=abind(array(dim=c(5,dim(sfwmd_amsunc30)[2],dim(sfwmd_amsunc30)[3])),sfwmd_amsunc
30,along=1)
#mdc_amsunc30=abind(atlas14_amsunc30,temp,along=3)
#dimnames(mdc_amsunc30)[[1]]=dimnames(atlas14_amsunc30)[[1]]
#dimnames(mdc_amsunc30)[[2]]=dimnames(atlas14_amsunc30)[[2]]
#dimnames(mdc_amsunc30)[[3]]=c(dimnames(atlas14_amsunc30)[[3]],dimnames(sfwmd_amsunc30
)[[3]])
#mdc_last30=abind(last30,last30s,along=2)
#dimnames(mdc_last30)[[1]]=dimnames(last30)[[1]]
#dimnames(mdc_last30)[[2]]=c(dimnames(last30)[[2]],dimnames(last30s)[[2]])


########################################################################
# Split up data into hourly stations and daily stations and create
# AMS4-like list
########################################################################
```

```
mdc_hourly30=mdc_amsunc30[,,which(apply(mdc_amsunc30[1,,],2,function(x)
sum(!is.na(x))) > 0)]
mdc_daily30=mdc_amsunc30[,,which(apply(mdc_amsunc30[1,,],2,function(x)  sum(!is.na(x)))
== 0)]

mdc_last30h=mdc_last30[,which(apply(mdc_amsunc30[1,,],2,function(x)  sum(!is.na(x)))  >
0)]
mdc_last30d=mdc_last30[,which(apply(mdc_amsunc30[1,,],2,function(x)  sum(!is.na(x))) ==
0)]

rm(AMS4)
AMS4=list()
for (i in 1:dim(mdc_amsunc30)[3]) {
  AMS4[[i]] = t(mdc_amsunc30[,,i])
  rownames(AMS4[[i]])=mdc_last30[,i]
}
names(AMS4) = dimnames(mdc_amsunc30)[[3]]
staid2=names(AMS4)

AMS4h=list()
for (i in 1:dim(mdc_hourly30)[3]) {
  AMS4h[[i]] = t(mdc_hourly30[,,i])
  rownames(AMS4h[[i]])=mdc_last30h[,i]
}

names(AMS4h) = names(which(apply(mdc_amsunc30[1,,],2,function(x) sum(!is.na(x))) > 0))
staid2h=names(AMS4h)

AMS4d=list()
for (i in 1:dim(mdc_daily30)[3]) {
  AMS4d[[i]] = t(mdc_daily30[,,i])
  rownames(AMS4d[[i]])=mdc_last30d[,i]
}

names(AMS4d)=names(which(apply(mdc_amsunc30[1,,],2,function(x) sum(!is.na(x))) == 0))
staid2d=names(AMS4d)

if (!dir.exists(paste(datadir,"/Obs_datasets/before_",cutyr,sep=""))) {
  dir.create(paste(datadir,"/Obs_datasets/before_",cutyr,sep=""))
}
setwd(paste(datadir,"/Obs_datasets/before_",cutyr,sep=""))
save(mdc_amsunc30,mdc_last30,mdc_hourly30,mdc_daily30,mdc_last30h,mdc_last30d,AMS4,AMS
4h,AMS4d,staid2,staid2h,staid2d,file="MDC_amsunc30.RData")



}
```

grubbs.flag <- function(x) {

```
  library(outliers)
  outliers <- NULL
  test <- x
  grubbs.result <- grubbs.test(test)
  pv <- grubbs.result$p.value
  while(pv < 0.05) {
    outliers <- c(outliers,as.numeric(strsplit(grubbs.result$alternative," ")[[1]][3]))
    test <- x[!x %in% outliers]
    grubbs.result <- grubbs.test(test)
    pv <- grubbs.result$p.value
  }
  return(data.frame(X=x,Outlier=(x %in% outliers)))
}
```

```
###########################################################################

station2cellmap <- function() {
# Function to map Atlas 14 weather stations to closest LOCA downscaled data grid point

###########################################################################

library(fields)
library(maps)
library(zoo)

datadir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/"

setwd(paste(datadir,"/Obs_datasets/before_2019",sep=""))

# Read weather station file
stas=read.csv(paste(datadir,"/ATLAS14/noaa_atlas14_included_stations.csv",sep=""),fill
=FALSE,stringsAsFactors=FALSE)
stas2=read.csv(paste(datadir,"/SFWMD/sfwmd_hourly_included_stations.csv",sep=""),fill=
FALSE,stringsAsFactors=FALSE)
stas3=read.csv(paste(datadir,"/SFWMD/sfwmd_included_stations.csv",sep=""),fill=FALSE,s
tringsAsFactors=FALSE)

# Weather station lats and lons
unordered_mystasNames=c(stas$STATION.ID,stas2$STATION,stas3$DBKEY)
unordered_stasLat=c(stas$LAT..degrees,stas2$LAT..degrees,stas3$LAT..degrees)
unordered_stasLon=c(stas$LONG..degrees,stas2$LONG..degrees,stas3$LONG..degrees)

#Read in AMS file for station names
load(file="MDC_amsunc30.RData")
staid2=dimnames(mdc_amsunc30)[[3]]
save(staid2,file="stationids.RData")

#Get lat and lon for stations in the order they're listed in mds_amsunc30
stasLat=unordered_stasLat[match(staid2,unordered_mystasNames)]
stasLon=unordered_stasLon[match(staid2,unordered_mystasNames)]


# LOCA CMIP5 grid cells lats and lons
ulats=seq(from=25.03125,to=32.03125,by=1/16)
ulons=seq(from=-89.96875,to=-78.96875,by=1/16)
nlat=length(ulats)
nlon=length(ulons)
ulats2=rep(ulats,each=nlon)
ulons2=rep(ulons,nlat)
ulatslons2 = as.matrix(cbind(ulons2,ulats2))
ulats3=matrix(ulats2,ncol=nlat)
ulons3=matrix(ulons2,ncol=nlat)

# Get list of LOCA active cells
load("Z:/miriza/Work/R/LOCA_dataset/Data/Active_LOCA_gridcells.RData")

# Find the closest active LOCA CMIP5 grid cell to each weather station
# id is the index of the station in ulatslons2
id=vector(mode="numeric",length=length(staid2))
# ilonclosest and ilatclosest are the first and second dimension indices
# to be used to access LOCA data
ilonclosest=vector(mode="numeric",length=length(staid2))
ilatclosest=vector(mode="numeric",length=length(staid2))
for (i in 1:length(staid2)) {
  lons = stasLon[i]
  lats = stasLat[i]
  p <- cbind(lons,lats)
```

```
  r = rdist.earth(ulatslons2,p)
#  id[i] <- apply(r,2,which.min)
#  lonlatclosest=ulatslons2[id[i],]
#  ilonclosest[i]=match(lonlatclosest[i,1],ulons)
#  ilatclosest[i]=match(lonlatclosest[i,2],ulats)
  o = apply(r,2,function(x) order(x))
  # If the closest cell is inactive (NA) go through other closest cells until
  # an active one is found
  for (l in 1:length(o)) {
    id[i] = o[l]
    # lonlatclosest gives the longitude and latitude of the closest
    # active LOCA CMIP5 gridcell to each weather station
    lonlatclosest=ulatslons2[id[i],]
    ilonclosest[i]=match(lonlatclosest[1],ulons)
    ilatclosest[i]=match(lonlatclosest[2],ulats)
    if (!is.na(actcells[ilonclosest[i],ilatclosest[i]])) break
  }
}

lonclosest=ulons[ilonclosest]
latclosest=ulats[ilatclosest]
save(stasLat,stasLon,ilonclosest,ilatclosest,lonclosest,latclosest,file="station2cellm
ap.RData")


png("Lat_mapping_check.png")
plot(stasLat,latclosest,xlab="Weather station lat",ylab="LOCA grid cell lat",
     main="Check that latitudes are close")
lines(range(cbind(stasLat,latclosest)),range(cbind(stasLat,latclosest)),col="red",lty=
2)
legend("bottomright",legend=c("Data","1:1"),lty=c(NA,2),pch=c(1,NA),col=c("black","red
"))
grid()
dev.off()

png("Lon_mapping_check.png")
plot(stasLon,lonclosest,xlab="Weather station lon",ylab="LOCA grid cell lon",
     main="Check that longitudes are close")
lines(range(cbind(stasLon,lonclosest)),range(cbind(stasLon,lonclosest)),col="red",lty=
2)
legend("bottomright",legend=c("Data","1:1"),lty=c(NA,2),pch=c(1,NA),col=c("black","red
"))
grid()
dev.off()

png("Mapping_check_FLmap.png",height=720,width=720,pointsize=15)
map('county',"Florida",xlim=c(-81.2,-80),ylim=c(25,26.2))
map.axes()
grid()
points(stasLon,stasLat,cex=0.8)
points(lonclosest,latclosest,cex=0.8,pch=19,col="red")
title(main=c("NOAA Atlas 14 and SFWMD weather stations in MDC","and closest LOCA grid
cell centers"),xlab="Lon",ylab="Lat")
legend(x="bottomright",c("Atlas 14, SFWMD","LOCA"),col=c("black","red"),pch=c(1,19))
dev.off()

png("LOCA_gridcells.png",height=720,width=720,pointsize=15)
map('county',"Florida",xlim=c(-81.2,-80),ylim=c(25,26.2))
map.axes()
grid()
abline(h=seq(from=25.03125-1/32,to=32.03125+1/32,by=1/16),col="gray")
abline(v=seq(from=-89.96875-1/32,to=-78.96875+1/32,by=1/16),col="gray")
points(stasLon,stasLat,cex=0.8)
```

```
points(lonclosest,latclosest,cex=0.8,pch=19,col="red")
axis(3,at=ulons[seq(1,180,1)],labels=seq(1,180,1),cex.axis=0.5,tck=0.02,mgp=c(3,0.3,0)
)
axis(4,at=ulats[seq(1,120,1)],labels=seq(1,120,1),las=3,cex.axis=0.5)
title(main=c("NOAA Atlas 14 and SFWMD weather stations in MDC","and LOCA grid
cells"),xlab="Lon",ylab="Lat")
legend(x="bottomright",c("Atlas 14, SFWMD","LOCA"),col=c("black","red"),pch=c(1,19))
dev.off()

png("Active_LOCA_gridcells.png",height=720,width=720,pointsize=15)
map('state',"Florida",xlim=c(-88,-80),ylim=c(25,32))
map.axes()
grid()
abline(h=seq(from=25.03125-1/32,to=32.03125+1/32,by=1/16),col="gray")
abline(v=seq(from=-89.96875-1/32,to=-78.96875+1/32,by=1/16),col="gray")
#points(stasLon,stasLat,cex=0.3,pch=19,col="red")
points(ulons3*actcells,ulats3*actcells,cex=0.3,pch=19,col="red")
LOK_Boundary=read.csv(paste(datadir,"LOK_Boundary.csv",sep=""))
lines(LOK_Boundary$x,LOK_Boundary$y)
axis(3,at=ulons[seq(10,180,10)],labels=seq(10,180,10),cex.axis=0.5,tck=0.02,mgp=c(3,0.
3,0))
axis(4,at=ulats[seq(10,120,10)],labels=seq(10,120,10),las=3,cex.axis=0.5)
title(main=c("LOCA active grid cells in Florida"),xlab="Lon",ylab="Lat")
legend(x="bottomleft",c("Active"),col=c("red"),pch=c(19))
dev.off()

png("Active_LOCA_gridcells_MDC.png",height=720,width=720,pointsize=15)
map('county',"Florida",xlim=c(-81.2,-80),ylim=c(25,26.2))
map.axes()
grid()
abline(h=seq(from=25.03125-1/32,to=32.03125+1/32,by=1/16),col="gray")
abline(v=seq(from=-89.96875-1/32,to=-78.96875+1/32,by=1/16),col="gray")
#points(stasLon,stasLat,cex=0.3,pch=19,col="red")
points(ulons3*actcells,ulats3*actcells,cex=0.3,pch=19,col="red")
LOK_Boundary=read.csv(paste(datadir,"LOK_Boundary.csv",sep=""))
lines(LOK_Boundary$x,LOK_Boundary$y)
axis(3,at=ulons[seq(1,180,1)],labels=seq(1,180,1),cex.axis=0.5,tck=0.02,mgp=c(3,0.3,0)
)
axis(4,at=ulats[seq(1,120,1)],labels=seq(1,120,1),las=3,cex.axis=0.5)
title(main=c("LOCA active grid cells in MDC"),xlab="Lon",ylab="Lat")
legend(x="bottomleft",c("Active"),col=c("red"),pch=c(19))
dev.off()


#############################################################################
}
```

```
################################################################################

contourmap                                                                    <-
function(statis,ids,direc="./",main,res=1000,idp=2,posonly=TRUE,pval=N
ULL,labs="none",zlim=NULL){
# Uses IDW to interpolate the data over FL and then contour it
# statis: data to contour
# ids: IDs of the stations
# direc: Directory where to save the png file
# main: title for plot and file name
# res: resolution of grid for interpolation prior to contouring
# idp: exponent for IDW interpolation
# posonly: whether variable only has positive values
# pval: pval associated with the data (default is NULL, i.e. none)
#       If given it must be the same length as statis and stations with a
#       significant pval (<0.05) are labeled with an '*'
# labs: "none" so only station locations are plotted (default)
#       "names" so stations are plotted and labeled with their names
#       "data" so stations are plotted and labeled with the data values
#       "dn" so stations are plotted and labeled with their names and data values
# zlim: Limits for colormap use NULL to have code compute them automatically from data
ranges
#       Enter a pair of values otherwise (e.g. c(-0.5,0.5))

##############################################################################

library(maps)
library(akima)
library(sp)
library(ggplot2)
library(maptools)
library(gstat)
library(colorRamps)
library(raster)

# First eliminate stations with missing (NA) data.
if (!is.null(pval)) {
  tokeep=is.finite(statis)&is.finite(pval)
  statis=statis[tokeep]
  ids=ids[tokeep]
  pval=pval[tokeep]
} else {
  tokeep=is.finite(statis)
  statis=statis[tokeep]
  ids=ids[tokeep]
  pval=pval[tokeep]
}
# Number of stations left
nstas=length(ids)
print(names(statis))
print (length(statis))

datadir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/"

# Load FL boundary
FL_Boundary=read.csv(paste(datadir,"/Code/FL_Boundarydetailed.csv",sep=""))

#Load canals
cnls=shapefile("Z:/miriza/Work/FIU/FL_Building_Code/Data/USGS_MODFLOW/ancillary/ancill
ary/gis/umd_swr_hydrography.shp")
cnlslatlon=spTransform(cnls,CRS="+proj=longlat        +datum=WGS84        +ellps=WGS84
+towgs84=0,0,0")
```

```
## Note: To draw contours first need to interpolate values to a grid using IDW from
gstat package
## Options for contours include contour, contourplot (lattice), filled.contour,
contourLines
## Get station locations
# Read weather station file
stas=read.csv(paste(datadir,"/ATLAS14/noaa_atlas14_included_stations.csv",sep=""),fill
=FALSE,stringsAsFactors=FALSE)
stas2=read.csv(paste(datadir,"/SFWMD/sfwmd_hourly_included_stations.csv",sep=""),fill=
FALSE,stringsAsFactors=FALSE)
stas3=read.csv(paste(datadir,"/SFWMD/sfwmd_included_stations.csv",sep=""),fill=FALSE,s
tringsAsFactors=FALSE)

# Weather station lats and lons
unordered_mystasNames=c(stas$STATION.ID,stas2$STATION,stas3$DBKEY)
unordered_stasLat=c(stas$LAT..degrees,stas2$LAT..degrees,stas3$LAT..degrees)
unordered_stasLon=c(stas$LONG..degrees,stas2$LONG..degrees,stas3$LONG..degrees)


#Get lat and lon for stations in the order they're listed in mds_amsunc30
stasLat=unordered_stasLat[match(names(statis),unordered_mystasNames)]
stasLon=unordered_stasLon[match(names(statis),unordered_mystasNames)]

print(paste(length(stasLat),length(stasLon)))
mydata=data.frame(cbind(statis,stasLon,stasLat))
names(mydata)=c("statis","x","y")
coordinates(mydata) = ~x + y

#Define labels for plot
a=character(nstas)
pch=rep(16,nstas)
if (!is.null(pval)) {
  a[pval<0.05]="*"
  pch[pval<0.05]=15
}

if (labs=="none") lab=rep(NULL,nstas)
if (labs=="names") lab=paste(ids,sep="")
if (labs=="data") lab=paste(round(statis,1),a,sep="")
if (labs=="dn") lab=paste(ids,": ",round(statis,1),a,sep="")

#Determine colormap and z-limits
if (posonly) {
  colorpal=matlab.like
  if (is.null(zlim)) zlim=range(statis, finite=TRUE)
} else {
  zabmax=max(abs(statis))
  colorpal=blue2red
  if (is.null(zlim)) zlim=c(-zabmax,zabmax)
}
#Interpolate data using IDW
grd=expand.grid(x=seq(min(stasLon),max(stasLon),length=res),
                y=seq(min(stasLat),max(stasLat),length=res))
coordinates(grd) = ~x + y
gridded(grd) = TRUE
IDW <- idw(formula=statis~1,locations=mydata,newdata=grd,idp=idp)
IDW.output = as.data.frame(IDW)  # output is defined as a data table
names(IDW.output)[1:3] <- c("lon", "lat", "var1.pred")  # give names to the modelled
variables

#Reformat the output for mapping
xcoord=IDW.output$lon
```

```
ycoord=IDW.output$lat
zcoord=matrix(IDW.output$var1.pred,nrow=res,ncol=res)
mycoords=list(x=xcoord,y=ycoord)
#Blank out areas outside of FL_Boundary
zcoord[!point.in.polygon(xcoord,ycoord,FL_Boundary[,1],FL_Boundary[,2])]=NA

png(paste(direc,"/contourmap_",main,".png",sep=""),height=720,width=720,pointsize=15)
filled.contour(x=seq(min(stasLon),max(stasLon),length=res),y=seq(min(stasLat),max(stas
Lat),length=res),
       z=zcoord,xlim=c(-81.2,-80),ylim=c(25,26.2),
       zlim=zlim,asp=1,color.palette=colorpal,
       plot.axes={map('county',"Florida",add=TRUE);axis(1);axis(2);grid();
                            lines(cnlslatlon,col="white");
                 points(stasLon,stasLat,pch=pch,cex=0.6);
                 text(stasLon,stasLat,labels=lab,cex=0.6,pos=2)}
       )
title(main=main,xlab="Lon",ylab="Lat")
mtext(paste("IDW interpolation with exponent of ",idp,sep=""),side=3,line=0,cex=0.6)
if (!is.null(pval)) mtext("* Significant at the 0.05 level",side=1,adj=1,line=3,cex=0.6)
dev.off()
}
```

```
##############################################################################

subset_loca <- function() {
# R script to subset netCDF LOCA data for Florida
# and then subset it for the Atlas 14 stations in FL

##############################################################################

#Main variables
vn="pr"
vnl="Precip"
data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data"
pdates=seq(as.Date("2006/1/1"), as.Date("2100/12/31"),"days")
pyrs=as.numeric(format(pdates,'%Y'))
hdates=seq(as.Date("1950/1/1"), as.Date("2005/12/31"),"days")
hyrs=as.numeric(format(hdates,'%Y'))
hyrss=seq(1950,2005,1)

#Latitudes and longitudes of interest (in Florida)
lats=c(25,32)
lons=c(-90,-79)

library(RNetCDF)
library(ncdf4)
#library(geoknife)
library(fields)
library(maps)
library(pheno)
library(akima)


setwd(data_dir)

print(paste("Working on",vn))

projs=read.table("loca_projections.txt",stringsAsFactors=FALSE)
nprojs=nrow(projs)
fns=paste(vn,"_",projs[,1],sep="")
modelp=apply(projs,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelpbase=apply(projs,1,function(x) strsplit(x,"_")[[1]][1])
modelprip=apply(projs,1,function(x) strsplit(x,"_")[[1]][2])
modelprcp=apply(projs,1,function(x) strsplit(x,"_")[[1]][3])
modelpbases=unique(modelpbase)

hist=read.table("loca_historical.txt",stringsAsFactors=FALSE)
fnsh=paste(vn,"_",hist[,1],sep="")
modelh=apply(hist,1,function(x) paste(c(strsplit(x,"_")[[1]][1:2]),collapse="_"))
modelhbase=apply(hist,1,function(x) strsplit(x,"_")[[1]][1])
modelhrip=apply(hist,1,function(x) strsplit(x,"_")[[1]][2])

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
onids.RData")
nstas=length(staid2)

# Load ids of closest LOCA grid cells to Atlas 14 stations
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
on2cellmap.RData")

# Extract subset of data of interest for projections
for (i in 1:nrow(projs)) {
  print(paste("i=",i,sep=""))
```

```
  ncfile<-open.nc(paste("Z:/miriza/Work/R/LOCA_dataset/Data/",fns[i],"_2006-
2100.nc",sep=""))
  #if (i == 1) {
  #  lat=var.get.nc(ncfile,"lat")
  #  lon=var.get.nc(ncfile,"lon")
  #}
  time=var.get.nc(ncfile,"time")
  tss=length(time)
  psub=matrix(nrow=tss,ncol=nstas)

  units=att.get.nc(ncfile,fns[i],"units")
  print(units)
  if (units == "kg m-2 s-1") {
    conv=141.7323*24  #mm/s to in/day
  } else if (units == "mm") {
    conv=1/25.4  #mm to in (per day of course)
  } else {
    stop("Different type of units")
  }

  scale=try(att.inq.nc(ncfile,fns[i],"scale_factor"),silent=TRUE)
  if (class(scale) == "try-error") {
    scale=1
  } else {
    scale=att.get.nc(ncfile,fns[i],"scale_factor")
  }
  print(scale)

  for (j in 1:nstas) {
    print(paste("proj j=",j,sep=""))


psub[,j]=conv*scale*var.get.nc(ncfile,fns[i],start=c(ilonclosest[j],ilatclosest[j],1),
count=c(1,1,tss))
  }
  psub=as.data.frame(psub)
  colnames(psub)=staid2
  psub$Date=pdates[1:tss]
  psub$Year=pyrs[1:tss]
  close.nc(ncfile)

  # Merge with historical data if available
  idh=which(modelhbase %in% modelpbase[i])
  if (length(idh) == 0) {
    print(paste("No historical data found for model:",modelp[i]," (i=",i,")",sep=""))
  } else if (length(idh) > 1) {
    stop(paste("Multiple    historical    data    found    for    model:",modelp[i],"
(i=",i,")",sep=""))
  } else {
    ncfile2<-open.nc(paste("Z:/miriza/Work/R/LOCA_dataset/Data/",fnsh[idh],"_1950-
2005.nc",sep=""))
    time2=var.get.nc(ncfile2,"time")
    tss2=length(time2)
    psub2=matrix(nrow=tss2,ncol=nstas)

    units2=att.get.nc(ncfile2,fnsh[idh],"units")
    print(units2)
    if (units2 == "kg m-2 s-1") {
      conv2=141.7323*24  #mm/s to in/day
    } else if (units2 == "mm") {
      conv2=1/25.4  #mm to in (per day of course)
    } else {
      stop("Different type of units")
```

```
    }

    scale2=try(att.inq.nc(ncfile2,fnsh[idh],"scale_factor"),silent=TRUE)
    if (class(scale2) == "try-error") {
      scale2=1
    } else {
      scale2=att.get.nc(ncfile2,fnsh[idh],"scale_factor")
    }
    print(scale2)

    for (j in 1:nstas) {
      print(paste("hist j=",j,sep=""))

psub2[,j]=conv2*scale2*var.get.nc(ncfile2,fnsh[idh],start=c(ilonclosest[j],ilatclosest
[j],1),count=c(1,1,tss2))
    }
    psub2=as.data.frame(psub2)
    colnames(psub2)=staid2
    psub2$Date=hdates[1:tss2]
    psub2$Year=hyrs[1:tss2]
    close.nc(ncfile2)

    psub=rbind(psub2,psub)
  }

  ibase=which(modelpbases %in% modelpbase[i])

save(psub,file=paste("./",toupper(modelprcp)[i],"/",vnl,"_",toupper(modelprcp)[i],"_",
min(psub$Year),"_",max(psub$Year),"_model_",ibase,"_subset.RData",sep=""))


write.table(projs[i,],file=paste("./",toupper(modelprcp)[i],"/Projections5.txt",sep=""
),
              append=TRUE,quote=FALSE,row.names=FALSE,col.names=FALSE)

}

}
```

```
##############################################################################

getAMS <- function() {

# Get AMS series for model data

##############################################################################

library(zoo)

#Main variables
vn="pr"
vnl="Precip"
data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data"

setwd(data_dir)


RCPs=c("RCP45","RCP85")
udurs=c("24-hr","2-day","3-day","4-day","7-day","10-day","20-day","30-day","45-
day","60-day");
udurs2=c("1 day","2 days","3 days","4 days","7 days","10 days","20 days","30 days","45
days","60 days");
udursmins=c(1440,2880,4320,5760,10080,14400,28800,43200,64800,86400)
udursdays=c(1,2,3,4,7,10,20,30,45,60)
ndurs=length(udursdays)

# Define dates and years for subset dataset
udates=seq(as.Date("1950/1/1"), as.Date("2099/12/31"),"days")
uyrs=as.numeric(format(udates,'%Y'))
nyrs=length(unique(uyrs))

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
onids.RData")
nstas=length(staid2)

for (r in 1:length(RCPs)) {
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))

fils=paste("Precip_",RCPs[r],"_1950_2100_model_",c(1:length(projs)),"_subset.RData",se
p="")

  for (f in 1:length(fils)) {
    print(paste("f = ",f,sep=""))
    load(paste("./",RCPs[r],"/",fils[f],sep=""))
    # Exclude the year 2100 from all files
    psub=psub[(psub$Year%in%uyrs),]
    ams=array(dim=c(ndurs,nyrs,nstas))

    for (u in 1:ndurs) {
      print(paste("u = ",u,sep=""))
      k=udursdays[u]

      rs=apply(psub[,1:nstas],2,function(x)             ave(x,psub$Year,FUN=function(x)
c(rollsum(x,k),rep(NA,k-1)),k=k))
      ams[u,,]=apply(rs,2,function(x) tapply(as.numeric(x),psub$Year,max,na.rm=TRUE))
      rm(rs)


    } #end u
```

```
save(ams,file=paste("./",RCPs[r],"/AMS_",RCPs[r],"_1950_2099_model_",f,".RData",sep=""
))
    rm(ams)

  } #end f
} #end r

} #end function
# Get AMS series for model data

library(zoo)

#Main variables
vn="pr"
vnl="Precip"
data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data"

setwd(data_dir)


RCPs=c("RCP45","RCP85")
udurs=c("24-hr","2-day","3-day","4-day","7-day","10-day","20-day","30-day","45-
day","60-day");
udurs2=c("1 day","2 days","3 days","4 days","7 days","10 days","20 days","30 days","45
days","60 days");
udursmins=c(1440,2880,4320,5760,10080,14400,28800,43200,64800,86400)
udursdays=c(1,2,3,4,7,10,20,30,45,60)
ndurs=length(udursdays)

# Define dates and years for subset dataset
udates=seq(as.Date("1950/1/1"), as.Date("2099/12/31"),"days")
uyrs=as.numeric(format(udates,'%Y'))
nyrs=length(unique(uyrs))

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
onids.RData")
nstas=length(staid2)

for (r in 1:length(RCPs)) {
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))

fils=paste("Precip_",RCPs[r],"_1950_2100_model_",c(1:length(projs)),"_subset.RData",se
p="")

  for (f in 1:length(fils)) {
    print(paste("f = ",f,sep=""))
    load(paste("./",RCPs[r],"/",fils[f],sep=""))
    # Exclude the year 2100 from all files
    psub=psub[(psub$Year%in%uyrs),]
    ams=array(dim=c(ndurs,nyrs,nstas))

    for (u in 1:ndurs) {
      print(paste("u = ",u,sep=""))
      k=udursdays[u]

      rs=apply(psub[,1:nstas],2,function(x)             ave(x,psub$Year,FUN=function(x)
c(rollsum(x,k),rep(NA,k-1)),k=k))
      ams[u,,]=apply(rs,2,function(x) tapply(as.numeric(x),psub$Year,max,na.rm=TRUE))
      rm(rs)
```

```
    } #end u

save(ams,file=paste("./",RCPs[r],"/AMS_",RCPs[r],"_1950_2099_model_",f,".RData",sep=""
))
    rm(ams)

  } #end f
} #end r

} #end function
```

```
##############################################################################
fitGEVall <- function() {
##############################################################################

  source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/fitGEV.R")

  type="bysite"
  method="RegLmom"
  cutyr=2005

  #Durations of interest for hourly data
  subdurs=c("60-min","2-hr","3-hr","6-hr","12-hr","24-hr","2-day","3-day","4-day","7-
day")

  #Run fitGEV for observed data (current period): Foc
  #using last 30 years of data

datadir=paste("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_"
,cutyr,sep="")
  setwd(datadir)
  load("MDC_amsunc30.RData")
  #Start with fitting GEV's to stations with hourly data


fitGEV(AMS4,type=type,method=method,yrs=NULL,syr=1940,eyr=2005,frac=0.0,subdurs=subdur
s,nstas=length(AMS4),
         dataset="MDC_obs",lab="Foc",smooth=FALSE,doplots=TRUE,contmaps=TRUE)

  #Run fitGEV for modeled data
  fitGEVwrap(type=type,method=method,yrs=t(mdc_last30),syr=1940,eyr=2005,lab="Fmc")
  #30 years centered on 2060-2069
  fitGEVwrap(type=type,method=method,syr=2050,eyr=2079,lab="Fmp1")

  #Note: Then must run EQM for the stations
}

##############################################################################

fitGEVwrap                                                                       <-
function(type="bysite",method="RegLmom",yrs=NULL,syr=1950,eyr=2012,lab
="Fmc") {
# Function to fitGEV to model AMS data
# (get Fmc and Fmp: CDF of model AMS data for the current period and future periods)
# type and method of GEV fitting (see header of fitGEV.R)
# yrs: If null, then get all data between years syr and eyr
#       Otherwise, it is a matrix with nstas rows and nyears columns with every row
listing
#      the years of analysis for a station.
# syr: Start year for analysis (cut off data before syr; make it very small to use POR)
# eyr: End year for analysis (cut off data after; make it very large to use POR)
# lab: How to label the plots

##############################################################################

source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/fitGEV.R")

data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data/"

setwd(data_dir)
```

```
RCPs=c("RCP45","RCP85")

# Durations
durs=c("24-hr","2-day","3-day","4-day","7-day","10-day","20-day","30-day","45-
day","60-day");
durs2=c("1 day","2 days","3 days","4 days","7 days","10 days","20 days","30 days","45
days","60 days");
dursmins=c(1440,2880,4320,5760,10080,14400,28800,43200,64800,86400)
dursdays=c(1,2,3,4,7,10,20,30,45,60)
ndurs=length(durs)
#Durations of interest
subdurs=c("24-hr","2-day","3-day","4-day","7-day")

# Define dates and years for subset dataset
udates=seq(as.Date("1950/1/1"), as.Date("2099/12/31"),"days")
uyrs=as.numeric(format(udates,'%Y'))
yrindx=1950:2099
nyrs=length(unique(uyrs))

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
onids.RData")
nstas=length(staid2)

for (r in 1:length(RCPs)) { #
  print(paste("r = ",r,sep=""))
  setwd(data_dir)
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))

  for (f in 1:length(projs)) { #i=143, f=9 gives error in HW.tests; i=122, f=13
    print(paste("f = ",f,sep=""))
    setwd(data_dir)
    load(paste("./",RCPs[r],"/AMS_",RCPs[r],"_1950_2099_model_",f,".RData",sep=""))

    # Create AMS4-equivalent list
    dimnames(ams)=list(durs,yrindx,staid2)
    AMS4=lapply(seq_len(dim(ams)[3]), function(i) t(ams[,,i]))
    names(AMS4)=staid2

    # Call fitGEV
    myDir=paste("./",RCPs[r],"/model_",f,"/",sep="")
    dir.create(file.path(myDir), showWarnings = FALSE)
    setwd(file.path(myDir))


fitGEV(AMS4,type=type,method=method,yrs=yrs,syr=syr,eyr=eyr,frac=0.0,subdurs=subdurs,n
stas=length(AMS4),
           dataset="FL_LOCA",lab=lab,smooth=FALSE,doplots=TRUE,contmaps=TRUE)

    rm(ams,AMS4)

  } #end f
} # end r

}
```

```
################################################################################

fitGEV <-
function(AMS4,type="bysite",method="RegLmom",yrs=NULL,syr=1800,eyr=2100,frac=0.0,subdurs=c("24-hr","2-day","3-day","4-day","7-day"),nstas=15,dataset="FL_Atlas14",lab="Foc",smooth=FALSE,doplots=FALSE,contmaps=FALSE) {
# Function to fitGEV to observed corrected AMS data (get Foc: CDF of observations for the current period)
# or modeled current or future (Fmc, Fmp)
# AMS4 = list with AMS data, it has at least 242 elements and each element is a matrix with one row per
#        year of data and one column per duration. It may have more than 242 elements for type="bydur"
#        method="RegLmom" if stations outside the state are used as part of the ROI of a station, but
#        the code is only run for the first 242 stations (those in FL).
#        The names of the elements of the list are the names of the weather stations.
# type = "bysite" fitting all durations at the same time (e.g. using Regional L-moments)
#        "bysite" can be applied to all durations but since it is based on separability assumption
#          it is better to apply to limited range of durations of interest for better fit
#        "bysite" only solved by method="RegLmom"
# method = "RegLmom" fitting by method of regional L-moments
# yrs: If null, then get all data between years syr and eyr
#       Otherwise, it is a matrix with nstas rows and nyears columns with every row listing
#      the years of analysis for a station.
# syr: Start year for analysis (cut off data before syr; make it very small to use POR)
# eyr: End year for analysis (cut off data after; make it very large to use POR)
# frac: Minimum fraction of years with AMS data to do GEV fitting
#      (i.e. if station has at least frac*(syr-eyr+1) AMS values
#            available-->do GEV fitting, otherwise skip it)
#       Make frac equal to 0 to use whatever data is available between syr and eyr
# subdurs: Subset of durations of interest over which to do the fitting
# dataset: The name of the dataset, e.g. "FL_Atlas14", "FL_USBR"
# lab: "Foc", "Fmc", "Fmp", "Fmp1", etc. How to label the plots
# smooth: Whether to smooth out DDF curves using cubic splines (IDF curves output will be unsmoothed)
# doplots: Whether to create any plots
# contmaps: Whether to create contour maps of variables or not (can be very slow so set to FALSE by default)

################################################################################

# Install R packages
library(lmom)
library(nsRFA)
library(ismev)
library(rootSolve)
library(moments)
library(extRemes)
library(car)
library(kSamples)

source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/contourmap.R")

#setwd("F:/ATLAS14/AMS/")
#load("F:/ATLAS14/stationids.RData")
#load("F:/ATLAS14/AMS/FLALGAMS_Atlas14_AMScorr.RData")
#AMS4,AMSDATE3,staid2 has 242 stations in FL
```

```
#and the remaining 40 are in AL, GA, MS and are used in
#RegLmom with official Atlas 14 ROI
staid2all=names(AMS4)
staid2=staid2[1:nstas]
print(paste("AMS4 has",length(AMS4),"elements"))
print("Station ids:")
print(staid2all)

#Durations
durs=c("5-min","10-min","15-min","30-min","60-min","2-hr","3-hr","6-hr","12-hr","24-
hr","2-day","3-day","4-day",
"7-day","10-day","20-day","30-day","45-day","60-day");
dursmins=c(5,10,15,30,60,120,180,360,720,1440,2880,4320,5760,10080,14400,28800,43200,6
4800,86400)
ndurs=length(durs)
#dursint: Durations of interest for which to compute idf curves (1-7 days in minutes)
#dursint=24*60*(1:7)
#subdurs: Subset of durations to extract (that bound durations of interest)
#subdurs=c("24-hr","2-day","3-day","4-day","7-day")
#subdurs=durs[max(which(dursmins<=min(dursint))):min(which(dursmins>=max(dursint)))]
nsubdurs=length(subdurs)

#Years of interest
#Get overall starting and ending year if matrix with years of interest is specified
if (!is.null(yrs)) {
  syr=min(yrs,na.rm=TRUE)
  eyr=max(yrs,na.rm=TRUE)
}

#Return periods
#Tr=c(2,5,10,25,50,100,200,500,1000)
Tr=c(2,5,10,25,50,100)
#Non exceedance probabilities
pne=1-1/Tr
nts=length(Tr)
#Return periods of interest
subTr=c(2,5,10,25,50,100)
#Non exceedance probabilities of interest
pnes=1-1/subTr
ntss=length(subTr)

# Initialize matrices (to NA)
# Values will remain NA for stations without enough data according to frac rule
k=alfa=xi=matrix(nrow=nstas,ncol=nsubdurs,dimnames=list(staid2,subdurs))
intTrmax=intTrmin=matrix(nrow=nstas,ncol=nsubdurs,dimnames=list(staid2,subdurs))
# Goodness-of-fit over durations of interest
sBias=sRMSE=sMAE=sR2=sNS=matrix(nrow=nstas,ncol=1,dimnames=list(staid2))

if (type=="bysite") {

  if (method=="RegLmom") {
    crits=nps=t3R=adbpval=H1=matrix(nrow=nstas,ncol=1,dimnames=list(staid2))
    DISC=INDS=list()
  }

}

GEVpars=list()
IDF=list()
DDF=list()

for (i in 1:nstas) {
  print(paste("i=",i,",staid2=",staid2[i],sep=""))
```

```
  # Cut off data based on yrs or (syr and eyr)
  # If years of analysis specified, use those
  if (!is.null(yrs)) {
    # If no years to extract then go to the next station
    if (all(is.na(yrs[i,]))) {
      next
    } else {
    # Otherwise, extract the years
      sdata=AMS4[[i]][na.omit(match(yrs[i,],rownames(AMS4[[i]]))),]
      #print("here1")
      #print(sdata)
    }
  # Otherwise, use starting and ending year
  } else {
    sdata=AMS4[[i]][rownames(AMS4[[i]])<=eyr & rownames(AMS4[[i]])>=syr,]
  }

  # Also eliminate durations with less than frac of data between syr and eyr
  sdata=sdata[,apply(sdata,2,function(x) sum(!is.na(x))) >= frac*(eyr-syr+1)]

  # Get data for durations of interest (and only durations with values)
  sdata=sdata[,which(colnames(sdata)%in%subdurs)]
  sdata=sdata[,apply(sdata,2,function(x) sum(!is.na(x))) >= 15]

  # If no data left then go to the next station
  if (length(sdata) == 0) next
  #print("here2")

  if (type=="bysite") {
    if (method=="RegLmom") {
        environment(regLmomAS)=environment()
        rLmom=regLmomAS(variation="RFA")
        GEVp2=rLmom$GEVp2 #Already comes with sign of k corrected to match gev.fit
        H1[i]=rLmom$H1
        t3R[i]=rLmom$t3R
        nps[i]=rLmom$nps
        adbpval[i]=rLmom$adbpval
        DISC[[i]]=rLmom$disc
        crits[i]=rLmom$crits
    }

    rownames(GEVp2)=c("xi","alfa","k")
    colnames(GEVp2)=colnames(sdata)
    xi[i,colnames(sdata)]=GEVp2["xi",]
    alfa[i,colnames(sdata)]=GEVp2["alfa",]
    k[i,colnames(sdata)]=GEVp2["k",]
    GEVpars[[i]]=GEVp2
    ddf=mapply(invF.GEV,as.list(data.frame(pne)),xi=GEVp2[1,],alfa=GEVp2[2,],k=-
GEVp2[3,])

idf=ddf/(matrix(rep(dursmins[match(colnames(sdata),durs)],length(pne)),ncol=ncol(sdata
),byrow=TRUE)/60)
  }

  colnames(ddf)=colnames(sdata)
  rownames(ddf)=Tr
  colnames(idf)=colnames(sdata)
  rownames(idf)=Tr

  DDF[[i]]=ddf
  print(ddf)
  IDF[[i]]=idf
```

```
  if (doplots) {
  png(paste(dataset,"_",lab,"_DDF_",staid2[i],"_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""))

matplot(t(matrix(rep(dursmins[match(colnames(sdata),durs)],length(pne)),ncol=ncol(sdat
a),byrow=TRUE)),t(ddf),
          type="l",log="x",,xlab="log(D) (mins)",ylab="Precipitation (inches)",
          main=c(paste(lab,": DDF fits at Station ",staid2[i],sep=""),
          paste("Period: ",syr,"-",eyr,sep="")),axes=FALSE)

#matlines(t(matrix(rep(dursmins[match(colnames(sdata),durs)],length(pne)),ncol=ncol(sd
ata),byrow=TRUE))
  #,t(ddfsmooth),lwd=2)
  axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
  axis(side=2)
  axis(side=3,at=dursmins,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
  abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
  legend("topleft",legend=Tr,lty=1:5,col=1:6,cex=0.8)
  box()
  mtext("GEV fits based on Regional L-moments method",side=1,adj=1,line=4,cex=0.6)
  dev.off()

  png(paste(dataset,"_",lab,"_IDF_",staid2[i],"_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)

matplot(t(matrix(rep(dursmins[match(colnames(sdata),durs)],length(pne)),ncol=ncol(sdat
a),byrow=TRUE)),t(idf),
          type="l",log="x",,xlab="log(D)     (mins)",ylab="Precipitation     intensity
(inches/hour)",
          main=c(paste(lab,": IDF fits at Station ",staid2[i],sep=""),
          paste("Period: ",syr,"-",eyr,sep="")),axes=FALSE)
  axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
  axis(side=2)
  axis(side=3,at=dursmins,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
  abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
  legend("topright",legend=Tr,lty=1:5,col=1:6,cex=0.8)
  box()
  mtext("GEV fits based on Regional L-moments method",side=1,adj=1,line=4,cex=0.6)
  dev.off()
  }

  sample_xs=mapply(function(x) sort(x),

x=as.list(data.frame(sdata[,which(colnames(sdata)%in%subdurs)])),SIMPLIFY=FALSE)
  sample_ffs=mapply(function(x) ppoints(sort(x),a=0),

x=as.list(data.frame(sdata[,which(colnames(sdata)%in%subdurs)])),SIMPLIFY=FALSE)
  fit_xs=mapply(invF.GEV,sample_ffs,xi=GEVp2[1,which(colnames(sdata)%in%subdurs)],
         alfa=GEVp2[2,which(colnames(sdata)%in%subdurs)],k=-
GEVp2[3,which(colnames(sdata)%in%subdurs)], SIMPLIFY=FALSE)
  sBias[i]=mean(unlist(fit_xs)-unlist(sample_xs))
  sRMSE[i]=RMSE(unlist(sample_xs),unlist(fit_xs))
  sMAE[i]=MAE(unlist(sample_xs),unlist(fit_xs))
  sR2[i]=cor(unlist(sample_xs),unlist(fit_xs))^2
  sNS[i]=R2(unlist(sample_xs),unlist(fit_xs))#same                                   as
NSE(unlist(fit_xs),unlist(sample_xs))

  ffs=matrix(rep(seq(0.01,0.99,by=0.01),dim(sdata)[2]),ncol=dim(sdata)[2])
  xs=mapply(invF.GEV,as.list(data.frame(ffs)),xi=GEVp2[1,],alfa=GEVp2[2,],k=-
GEVp2[3,])
  ffs2=mapply(function(x) ecdf(x)(x),x=as.list(data.frame(sdata)))

  if (doplots){
```

```
  png(paste(dataset,"_",lab,"_",staid2[i],"_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
  matplot(xs,ffs,type="l",main=c(paste(lab,": GEV fits at Station ",staid2[i],sep=""),
          paste("Period: ",syr,"-",eyr,sep="")),xlab="Precipitation (inches)",
          ylab="F (P)",lwd=2)
  matpoints(sdata,ffs2,cex=0.5)
  legend("bottomright",legend=colnames(GEVp2),lty=1:5,lwd=2,col=1:6,cex=0.8)
  grid()
  abline(h=c(0,1,pne[1:6]),col="gray70",lty=2)
  axis(side=4,at=pne[1:6],labels=Tr[1:6],cex.axis=0.5,las=1)
  mtext("Tr (years)",side=4,cex=0.5)
  mtext("GEV fits based on Regional L-moments method",side=1,adj=1,line=4,cex=0.6)
  dev.off()
  }

  #Check whether curves intersect and at which return period Tr
  # intTr: if GEV fits intersect between pne of 0.001 and 1-1/max(Tr))
  # then save the ***approximate*** Tr of intersection here
  #This gives an idea of the number of intersections PRE-SMOOTHING
  for (u in 1:nsubdurs) {
    #only do it for durations pairs (30 mins vs. 15 mins) or more
    if (match(subdurs[u],durs)>=4 & u>1) {
      #print(paste("u=",u))
      if (!is.na(xi[i,u]) & !is.na(xi[i,u-1])) {
        fun=function(fr) invF.GEV(fr,xi=xi[i,u],alfa=alfa[i,u],k=-k[i,u]) -
                        invF.GEV(fr,xi=xi[i,u-1],alfa=alfa[i,u-1],k=-k[i,u-1])
        xroot=try(uniroot.all(fun, c(0.001,1-1/max(Tr)),n=10000))
        if (class(xroot)!="try-error" && length(xroot)>0) {
          intTrmin[i,u]=1/(1-min(xroot))
          intTrmax[i,u]=1/(1-max(xroot))
        }
      }
    }
  }#end u

} #end i


save(GEVpars,xi,alfa,k,IDF,DDF,sBias,sRMSE,sMAE,sNS,sR2,staid2,
     file=paste(dataset,"_",lab,"_GEVpars_",type,"_",method,"_",syr,"-
",eyr,".RData",sep=""))

save(intTrmax,intTrmin,staid2,
     file=paste(dataset,"_",lab,"_intersections_",type,"_",method,"_",syr,"-
",eyr,".RData",sep=""))

if (method=="RegLmom") save(H1,t3R,nps,crits,adbpval,DISC,INDS,staid2,

file=paste(dataset,"_",lab,"_regLmoms_",type,"_",method,"_",syr,"-
",eyr,".RData",sep=""))

# Plot GEV curve intersections
if (doplots) {
png(paste(dataset,"_",lab,"_GEV_curve_intersections_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
plot(colSums(!is.na(intTrmax),na.rm=TRUE),axes=FALSE,xlab="Duration",ylab="#         of
intersections")
axis(1,at=1:nsubdurs,labels=colnames(intTrmax),las=3)
axis(2)
grid()
box()
dev.off()
#plot(rowSums(!is.na(intTrmax),na.rm=TRUE),xlab="station #",ylab="# of intersections")
```

```
}


if (contmaps) {
  # Make contour maps of DDF data
  environment(DDFcontours)=environment()
  DDFcontours()
  # Make contour maps of each parameter for each duration
  # (starting at 15 mins since no data for 5 and 10 mins)
  for (u in 1:nsubdurs) {
    #only do it for durations of 15 mins or more
    if (match(subdurs[u],durs)>=3) {
      contourmap(xi[,u],staid2,direc=getwd(),

main=paste(dataset,"_",lab,"_GEV_locpar_",subdurs[u],"_",type,method,"_",syr,"-
",eyr,sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")
      contourmap(alfa[,u],staid2,direc=getwd(),

main=paste(dataset,"_",lab,"_GEV_scalepar_",subdurs[u],"_",type,method,"_",syr,"-
",eyr,sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")
      contourmap(k[,u],staid2,direc=getwd(),

main=paste(dataset,"_",lab,"_GEV_shapepar_",subdurs[u],"_",type,method,"_",syr,"-
",eyr,sep=""),
                 res=1000,idp=2,posonly=FALSE,pval=NULL,labs="none",zlim=c(-
max(abs(k),na.rm=TRUE),max(abs(k),na.rm=TRUE)))
    }
  }#end u
}

if (doplots) {

png(paste(dataset,"_",lab,"_GEV_locpar_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(matrix(rep(dursmins[match(subdurs,durs)],nstas),nrow=nstas,ncol=nsubdurs,byr
ow=TRUE)),
        t(xi),type="l",log="x",xlab="log(D) (mins)",ylab="Location parameter",
        main=c(paste(lab,": GEV location parameter as function of duration",sep=""),
        paste("at   all   MDC   stations   (",method,")   for   period:   ",syr,"-
",eyr,sep="")),axes=FALSE)
axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
axis(side=2)
axis(side=3,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
lines(dursmins[match(subdurs,durs)],colMeans(xi,na.rm=TRUE),lwd=3,lty=2)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.05,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.5,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.95,na.rm=TRUE),lwd=3)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

png(paste(dataset,"_",lab,"_GEV_locpar_alldur_loglog_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(matrix(rep(dursmins[match(subdurs,durs)],nstas),nrow=nstas,ncol=nsubdurs,byr
ow=TRUE)),
        t(xi),type="l",log="xy",xlab="log(D) (mins)",ylab="log(Location parameter)",
        main=c(paste(lab,": GEV location parameter as function of duration",sep=""),
        paste("at   all   MDC   stations   (",method,")   for   period:   ",syr,"-
",eyr,sep="")),axes=FALSE)
axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
```

```
axis(side=2)
axis(side=3,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
lines(dursmins[match(subdurs,durs)],colMeans(xi,na.rm=TRUE),lwd=3,lty=2)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.05,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.5,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(xi,2,quantile,probs=0.95,na.rm=TRUE),lwd=3)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

png(paste(dataset,"_",lab,"_GEV_scalepar_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(matrix(rep(dursmins[match(subdurs,durs)],nstas),nrow=nstas,ncol=nsubdurs,byr
ow=TRUE)),
        t(alfa),type="l",log="x",xlab="log(D) (mins)",ylab="Scale parameter",
        main=c(paste(lab,": GEV scale parameter as function of duration",sep=""),
        paste("at    all    MDC    stations    (",method,")    for    period:    ",syr,"-
",eyr,sep="")),axes=FALSE)
axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
axis(side=2)
axis(side=3,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
lines(dursmins[match(subdurs,durs)],colMeans(alfa,na.rm=TRUE),lwd=3,lty=2)
lines(dursmins[match(subdurs,durs)],apply(alfa,2,quantile,probs=0.05,na.rm=TRUE),lwd=3
)
lines(dursmins[match(subdurs,durs)],apply(alfa,2,quantile,probs=0.5,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(alfa,2,quantile,probs=0.95,na.rm=TRUE),lwd=3
)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

png(paste(dataset,"_",lab,"_GEV_scalepartolocpar_ratio_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(matrix(rep(dursmins[match(subdurs,durs)],nstas),nrow=nstas,ncol=nsubdurs,byr
ow=TRUE)),
        t(alfa)/t(xi),type="l",log="x",xlab="log(D)                 (mins)",ylab="Scale
parameter/location parameter",
        main=c("Ratio of scale/location as function of duration",
        paste("at    all    MDC    stations    (",method,")    for    period:    ",syr,"-
",eyr,sep="")),axes=FALSE)
axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
axis(side=2)
axis(side=3,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
lines(dursmins[match(subdurs,durs)],colMeans(alfa/xi,na.rm=TRUE),lwd=3,lty=2)
lines(dursmins[match(subdurs,durs)],apply(alfa/xi,2,quantile,probs=0.05,na.rm=TRUE),lw
d=3)
lines(dursmins[match(subdurs,durs)],apply(alfa/xi,2,quantile,probs=0.5,na.rm=TRUE),lwd
=3)
lines(dursmins[match(subdurs,durs)],apply(alfa/xi,2,quantile,probs=0.95,na.rm=TRUE),lw
d=3)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

png(paste(dataset,"_",lab,"_GEV_locpar_vs_scalepar_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(xi),
        t(alfa),type="l",xlab="Location parameter",ylab="Scale parameter",
        main=c("GEV location parameter vs. GEV scale parameter",
        paste("at all MDC stations (",method,") for period: ",syr,"-",eyr,sep="")))
```

```
abline(v=axTicks(side=1),h=axTicks(side=2),col="lightgray",lty="dotted")
lines(colMeans(xi,na.rm=TRUE),colMeans(alfa,na.rm=TRUE),lwd=3,lty=2)
lines(apply(xi,2,quantile,probs=0.05,na.rm=TRUE),apply(alfa,2,quantile,probs=0.05,na.r
m=TRUE),lwd=3)
lines(apply(xi,2,quantile,probs=0.5,na.rm=TRUE),apply(alfa,2,quantile,probs=0.5,na.rm=
TRUE),lwd=3)
lines(apply(xi,2,quantile,probs=0.95,na.rm=TRUE),apply(alfa,2,quantile,probs=0.95,na.r
m=TRUE),lwd=3)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

#png(paste(dataset,"_",lab,"_GEV_locpar_vs_scalepar_durupto2hr_",type,method,"_",syr,"
-",eyr,".png",sep=""),)
#matplot(t(xi[,1:6]),
#        t(alfa[,1:6]),type="l",xlab="Location parameter",ylab="Scale parameter",
#        main=c("GEV location parameter vs. GEV scale parameter",
#        paste("at all MDC stations (",method,") for period: ",syr,"-",eyr,sep="")))
#abline(v=axTicks(side=1),h=axTicks(side=2),col="lightgray",lty="dotted")
#lines(colMeans(xi[,1:6],na.rm=TRUE),colMeans(alfa[,1:6],na.rm=TRUE),lwd=3,lty=2)
#lines(apply(xi[,1:6],2,quantile,probs=0.05,na.rm=TRUE),apply(alfa[,1:6],2,quantile,pr
obs=0.05,na.rm=TRUE),lwd=3)
#lines(apply(xi[,1:6],2,quantile,probs=0.5,na.rm=TRUE),apply(alfa[,1:6],2,quantile,pro
bs=0.5,na.rm=TRUE),lwd=3)
#lines(apply(xi[,1:6],2,quantile,probs=0.95,na.rm=TRUE),apply(alfa[,1:6],2,quantile,pr
obs=0.95,na.rm=TRUE),lwd=3)
#legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
#box()
#dev.off()


png(paste(dataset,"_",lab,"_GEV_locpar_vs_scalepar_alldur_loglog_",type,method,"_",syr
,"-",eyr,".png",sep=""),)
matplot(t(xi),
        t(alfa),type="l",log="xy",xlab="log(Location       parameter)",ylab="log(Scale
parameter)",
        main=c("GEV location parameter vs. GEV scale parameter",
        paste("at all MDC stations (",method,") for period: ",syr,"-",eyr,sep="")))
abline(v=axTicks(side=1),h=axTicks(side=2),col="lightgray",lty="dotted")
lines(colMeans(xi,na.rm=TRUE),colMeans(alfa,na.rm=TRUE),lwd=3,lty=2)
lines(apply(xi,2,quantile,probs=0.05,na.rm=TRUE),apply(alfa,2,quantile,probs=0.05,na.r
m=TRUE),lwd=3)
lines(apply(xi,2,quantile,probs=0.5,na.rm=TRUE),apply(alfa,2,quantile,probs=0.5,na.rm=
TRUE),lwd=3)
lines(apply(xi,2,quantile,probs=0.95,na.rm=TRUE),apply(alfa,2,quantile,probs=0.95,na.r
m=TRUE),lwd=3)
legend("top",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

png(paste(dataset,"_",lab,"_GEV_shapepar_alldur_",type,method,"_",syr,"-
",eyr,".png",sep=""),)
matplot(t(matrix(rep(dursmins[match(subdurs,durs)],nstas),nrow=nstas,ncol=nsubdurs,byr
ow=TRUE)),
        t(k),type="l",log="x",xlab="log(D) (mins)",ylab="Shape parameter",
        main=c(paste(lab,": GEV shape parameter as function of duration",sep=""),
        paste("at   all   MDC   stations   (",method,")   for   period:   ",syr,"-
",eyr,sep="")),axes=FALSE,ylim=c(-0.52,0.52))
axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
axis(side=2)
axis(side=3,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
lines(dursmins[match(subdurs,durs)],colMeans(k,na.rm=TRUE),lwd=3,lty=2)
```

```
lines(dursmins[match(subdurs,durs)],apply(k,2,quantile,probs=0.05,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(k,2,quantile,probs=0.5,na.rm=TRUE),lwd=3)
lines(dursmins[match(subdurs,durs)],apply(k,2,quantile,probs=0.95,na.rm=TRUE),lwd=3)
abline(h=0)
legend("bottom",legend=c("mean","P5, P50, P95"),lty=c(2,1),lwd=3)
box()
dev.off()

}


}

#######################################################################
```

regLmomAS <- function(variation="RFA"){

```
# At-site regional Lmoments (by duration)
# The environment for this function is set to that of the parent environment so it
# can see all the variables in the parent, but not modify them directly
# variation: "RFA" to group normalized annual maxima across stations in the ROI
#            (normalizing variable is MAM), compute Lmoments for the group as
#            well as GEV parameters and then convert those back to at station
#            estimates
#            "A14" to compute regional Lmoments for each station in the ROI
#            independently and then do weighted average to compute at station
#            estimate

############################################################################

library(nsRFA)

mam=colMeans(sdata,na.rm=TRUE)

GEVp2=matrix(nrow=3,ncol=ncol(sdata))
colnames(GEVp2)=colnames(sdata)

dat=data.frame(V1=rep(colnames(sdata),each=nrow(sdata)),Dmax=as.vector(sdata))
y=as.data.frame(dat[rowSums(is.na(dat))==0,])
mvals=tapply(y$Dmax,y$V1,mean,na.rm=TRUE)
if (variation=="RFA") {
  nps=length(unique(y$V1))
  #Note: nps must be greater than or equal to 5
  #Anderson-Darling test for homogeneity
  adbpval=1-ADbootstrap.test(y$Dmax,y$V1,index=1)["P"]
  y$Dmax=y$Dmax/unsplit(tapply(y$Dmax,y$V1,mean,na.rm=TRUE),y$V1)
  #Discordancy measure of AMS normalized by index (MAM)
  disc=discordancy(y$Dmax,y$V1)
  #H values for AMS normalized by index (MAM)
  Hws=HW.tests(y$Dmax,y$V1)
  H1=Hws[1]
  #print(Hws)
  #estimate and plot regional growth curve
  regLM=nsRFA::Lmoments(y$Dmax)
  t3R=regLM[4]
  pars=par.GEV(regLM[1],regLM[2],regLM[4])
  #changing sign of shape par k for consistency with MLE estimated shape parameter (by
gev.fit)
  GEVp2[1,]=pars$xi*mam
  GEVp2[2,]=pars$alfa*mam
  GEVp2[3,]=-pars$k
  #plot L-moment diagram
  #Lmoment.ratio.diagram()
  #points(regLM[4],regLM[5],pch=19,col="red")
```

```
  png(paste(dataset,"_",lab,"_DDF_",staid2[i],"_alldur_norm_",type,method,"_",syr,"-
",eyr,".png",sep=""))
  FF=F.GEV(y$Dmax,pars$xi,pars$alfa,pars$k)
  regionalplotpos(y$Dmax,y$V1,xlab="Dmax",main=c("DDF  fitted  to  normalized  AMS  by
duration",
                                                  paste("for                    station
",staid2[i],sep="")))
  w=as.data.frame(cbind(y$Dmax,FF))
  names(w)=c("Dmax","FF")
  sDmax=sort(w$Dmax[!is.na(w$FF)])
  sFF=sort(w$FF[!is.na(w$FF)])
  lines(sDmax[!is.na(sFF)],sFF[!is.na(sFF)],lwd=2)
  leg=unique(y$V1)
  legend("bottomright",legend=leg,pch=c(1:length(unique(y$V1))),
         col=c(1:length(unique(y$V1))), cex=0.7)
  dev.off()
} else if (variation=="A14") {
  nps=length(unique(y$V1))
  #Discordancy measure of AMS
  #disc=discordancy(y$Dmax,y$V1)
  #Anderson-Darling test for homogeneity
  adbpval=1-ADbootstrap.test(y$Dmax,y$V1,index=1)["P"]
  #H values for AMS
  Hws=HW.tests(y$Dmax,y$V1)
  H1=Hws[1]
  regLM=regionalLmoments(y$Dmax,y$V1)
  lambda1=mam
  lambda2=regLM[3]*mam
  t3R=rep(regLM[4],length(mam))
  GEVp2=mapply(function(lambda1,lambda2,tau3)
as.numeric(par.GEV(lambda1,lambda2,tau3)),
                    lambda1=lambda1,lambda2=lambda2,tau3=t3R)
  t3R=regLM[4]
  #changing sign of shape par k for consistency with MLE estimated shape parameter (by
gev.fit)
  GEVp2[3,]=-GEVp2[3,]
}

Z=qf(.1/nps,3,nps-4,lower.tail=FALSE)
crits=(nps-1)*Z/(nps-4+3*Z)
#crits matches value from function criticalD()

#print(disc)
#print(paste(crits,nps,Z))
return(list(GEVp2=GEVp2,H1=H1,t3R=t3R,disc=disc,crits=crits,nps=nps,adbpval=adbpval))

}


#################################################################################
```

GOFstatscompare <- function() {

```
# Compare GOF statistics for the different methods
# Every list has the following components: (sRMSE,sMAE,sR2,sNS)

setwd("F:/ATLAS14/AMS/")
load("F:/ATLAS14/AMS/GOF_statistics.RData")
Stat_lists=c("bydur_Lmom","bydur_MLE","bydur_RegLmom","bydur_RegLmom_officialROI_A14",
"bysite_RegLmom","CPM_Lmom","CPM_MLE","scaling_optim","unified_MLE")

# Get staid2 (ids of weather stations)
load("F:/ATLAS14/stationids.RData")
nstas=length(staid2)
sBias=sRMSE=sMAE=sR2=sNS=matrix(nrow=nstas,ncol=length(Stat_lists))
```

```
#colnames(sRMSE)=Stat_lists
#colnames(sMAE)=Stat_lists
#colnames(sR2)=Stat_lists

for (o in 1:length(Stat_lists)){
  sBias[,o]=get(paste("Stats_",Stat_lists[o],sep=""))$sBias
  sRMSE[,o]=get(paste("Stats_",Stat_lists[o],sep=""))$sRMSE
  sMAE[,o]=get(paste("Stats_",Stat_lists[o],sep=""))$sMAE
  sR2[,o]=get(paste("Stats_",Stat_lists[o],sep=""))$sR2
  sNS[,o]=get(paste("Stats_",Stat_lists[o],sep=""))$sNS
}

png("GOF_statistics_boxplot.png",height=1000,width=720,pointsize=20)
nf=layout((c(1,2,3,4,5)),heights=c(2,2,2,2,2))

#pars=par(mar=c(5.1,4.1,4.1,10),xpd=TRUE)
par(mar=c(0,4.1,2,2.1),mgp=c(2,1,0))
bxp=boxplot(sBias,xaxt="n",ylab="Bias  (inches)",main="Comparison  of  Goodness-of-fit
statistics across methods")
abline(h=axTicks(side=2),col="lightgray",lty="dotted")
text(x=rep(1:length(Stat_lists),1),y=as.vector(t(bxp$stats[3,])+0.02),labels=round(as.
vector(t(bxp$stats[3,])),2),
    col="red",font=2)

par(mar=c(1,4.1,1,2.1),mgp=c(2,1,0))
bxp=boxplot(sRMSE,xaxt="n",ylab="RMSE (inches)")
abline(h=axTicks(side=2),col="lightgray",lty="dotted")
text(x=rep(1:length(Stat_lists),1),y=as.vector(t(bxp$stats[3,])+0.10),labels=round(as.
vector(t(bxp$stats[3,])),2),
    col="red",font=2)

par(mar=c(1,4.1,0,2.1),mgp=c(2,1,0))
bxp=boxplot(sMAE,xaxt="n",ylab="MAE (inches)")
abline(h=axTicks(side=2),col="lightgray",lty="dotted")
text(x=rep(1:length(Stat_lists),1),y=as.vector(t(bxp$stats[3,])+0.03),labels=round(as.
vector(t(bxp$stats[3,])),2),
    col="red",font=2)

par(mar=c(1,4.1,0,2.1),mgp=c(2,1,0))
bxp=boxplot(sR2,xaxt="n",ylab=expression("R"^"2"))
abline(h=axTicks(side=2),col="lightgray",lty="dotted")
text(x=rep(1:length(Stat_lists),1),y=as.vector(t(bxp$stats[3,])-
0.005),labels=round(as.vector(t(bxp$stats[3,])),2),
    col="red",font=2)

par(mar=c(2,4.1,0,2.1),mgp=c(2,1,0))
bxp=boxplot(sNS,ylab="NSE")
#axis(1,at=1:length(Stat_lists),labels=Stat_lists,cex.axis=1,las=3)
abline(h=axTicks(side=2),col="lightgray",lty="dotted")
text(x=rep(1:length(Stat_lists),1),y=as.vector(t(bxp$stats[3,])-
0.01),labels=round(as.vector(t(bxp$stats[3,])),2),
     col="red",font=2)

#par(fig=c(0,1,0,1),oma=c(0,0,0,0),mar=c(0,0,0,0),new=TRUE)
#plot(0,0,type="n",bty="n",xaxt="n",yaxt="n",col="white")
#legend.col=c(rep(0,3),rep(1,3),rep(2,3))
#legend("bottom",legend=paste(c(1:9),":",Stat_lists,sep=""),xpd=TRUE,inset=c(0,0),bty=
"n",
#       cex=1,col=legend.col,ncol=3)
dev.off()

}
```

```
##############################################################################

DDFcontours <- function() {
#This function create contour maps for fitted DDF values
#It can see the following variables from the main program:
#DDF is a list with each element of the list being a matrix with one row per return
period
#and one column per duration
#subdurs: durations of interest
#subTr: return periods of interest

##############################################################################

source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/contourmap.R")

nsubdurs=length(subdurs)
ntss=length(subTr)

#Create matrix from DDF list
ddfm=array(dim=c(nstas,nsubdurs,ntss),dimnames=list(staid2,subdurs,subTr))
#oo=lapply(seq_len(length(DDF)),                                           function(i)
ddfm[i,,]=t(DDF[[i]][as.character(subTr),subdurs]))
for (i in 1:nstas){
  if (!is.null(DDF[[i]])) ddfm[i,match(colnames(DDF[[i]]),subdurs),]=t(DDF[[i]])
}


for (u in 1:nsubdurs) {
  for (t in 1:ntss) {
    contourmap(ddfm[,u,t],staid2,direc=getwd(),
               main=paste(dataset,"_",lab,"_DDF_",subdurs[u],"_",subTr[t],"-
year_",type,method,"_",syr,"-",eyr,sep=""),
               res=1000,idp=2,posonly=TRUE,pval=NULL,labs="data")
  }
}

}

##############################################################################

copytable <- function(x, ...) {

##############################################################################

  library(xtable)
  f <- tempfile(fileext=".html")
  print(xtable(x, ...), "html", file = f)
  browseURL(f)
}
```

```
###############################################################################
```

doEQM <- function () {

```
#This function does EQM for bias-correction and QM for temporal downscaling
###############################################################################

data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data/"
setwd(data_dir)

source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/contourmap.R")

## Open log file and write header
logfile = paste("EQM_",format(Sys.time(),"%Y-%m-%d_%I_%M_%S_%p"),".log",sep="")
write(paste("Log file for EQM:",sep=""),file=logfile)

logfile2 = paste("EQM_2",format(Sys.time(),"%Y-%m-%d_%I_%M_%S_%p"),".log",sep="")
write(paste("Log file for EQM:",sep=""),file=logfile2)

RCPs=c("RCP45","RCP85")
nmodels=60
RCP45=1:30
RCP85=31:60

#Quantiles of interest
probs=c(0.05,0.5,0.95)

# Durations
durs=c("24-hr","2-day","3-day","4-day","7-day","10-day","20-day","30-day","45-
day","60-day");
dursmins=c(1440,2880,4320,5760,10080,14400,28800,43200,64800,86400)
ndurs=length(durs)
#Durations of interest
subdurs=c("24-hr","2-day","3-day","4-day","7-day")
nsubdurs=length(subdurs)

#Subdaily durations of interest for temporal downscaling
hdurs=c("60-min","2-hr","3-hr","6-hr","12-hr")
nhdur=length(hdurs)
hdursmins=c(60,120,180,360,720)


#Return periods
Tr=c(2,5,10,25,50,100,200,500,1000)
#Exceedance prob.
pe=1/Tr
#Non exceedance probabilities
pne=1-1/Tr
nts=length(Tr)
#Return periods of interest
subTr=c(2,5,10,25,50,100)
ntss=length(subTr)
pes=1/subTr
pnes=1-1/subTr
durshr=(matrix(rep(dursmins[match(subdurs,durs)],ntss),ncol=ntss,byrow=FALSE)/60)
hdurshr=(matrix(rep(hdursmins,ntss),ncol=ntss,byrow=FALSE)/60)

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stati
onids.RData")
nstas=length(staid2)
```

```
dataset1="MDC_obs"
dataset2="FL_LOCA"

curr=c(1940,2005)
currm=c(1942,2005)
proj1=c(2050,2079)

type="bysite"
method="RegLmom"

#Observational datasets
lab="Foc"
load(paste("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005
/",
           dataset1,"_",lab,"_GEVpars_",type,"_",method,
           "_",curr[1],"-",curr[2],".RData",sep=""))
xioc=xi
alfaoc=alfa
koc=k
ddfoc=array(dim=c(nstas,nsubdurs,ntss),dimnames=list(staid2,subdurs,subTr))
for (i in 1:nstas){
  if (!is.null(DDF[[i]])) ddfoc[i,,]=t(DDF[[i]][as.character(subTr),subdurs])
}
ddfoch=array(dim=c(nstas,nhdur,ntss),dimnames=list(staid2,hdurs,subTr))
for (i in 1:nstas){
  if (!is.null(DDF[[i]])) {
    if (all(hdurs%in%colnames(DDF[[i]]))) {
      ddfoch[i,,]=t(DDF[[i]][as.character(subTr),hdurs])
       }
  }
}
rm(GEVpars,xi,alfa,k)


DDFmpadj1=array(dim=c(nmodels,nstas,nsubdurs,ntss))
m=0
for (r in 1:length(RCPs)) {
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))

  for (f in 1:length(projs)) {
    m=m+1
    print(paste("f = ",f,", m = ",m,sep=""))

    lab="Fmc"
    load(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_GEVpars_",type,"_",
         method,"_",currm[1],"-",currm[2],".RData",sep=""))
    #load(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_GEVpars_",type,"_",
    #     method,"_",curr[1],"-",curr[2],"_allyrs.RData",sep=""))
    ximc=xi
    alfamc=alfa
    kmc=k
    rm(GEVpars,xi,alfa,k)

    lab="Fmp1"
    load(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_GEVpars_",type,"_",
         method,"_",proj1[1],"-",proj1[2],".RData",sep=""))
    ximp1=xi
    alfamp1=alfa
    kmp1=k
    rm(GEVpars,xi,alfa,k)

    for (i in 1:nstas) {
```

```
        print(paste("i = ",i,sep=""))
        for (u in 1:nsubdurs) {
          print(paste("u = ",u,sep=""))
          if (!is.na(xioc[i,subdurs[u]])) {

GEVparsoc=c(xi=xioc[i,subdurs[u]],alfa=alfaoc[i,subdurs[u]],k=koc[i,subdurs[u]])
           #GEVparsoc=NULL

GEVparsmc=c(xi=ximc[i,subdurs[u]],alfa=alfamc[i,subdurs[u]],k=kmc[i,subdurs[u]])

GEVparsmp1=c(xi=ximp1[i,subdurs[u]],alfa=alfamp1[i,subdurs[u]],k=kmp1[i,subdurs[u]])
           #Comment out first GEVparsoc above (set GEVparsoc to NULL instead)
           #and add #xoc=ddfoc[i,u,] in the calls to EQM below to use
           #Official Atlas 14 DDF curves as xoc

EQMmpadj1=EQM(GEVparsoc,GEVparsmc,GEVparsmp1,type="ratio",Tr=subTr,logfile=logfile)
           DDFmpadj1[m,i,u,]=EQMmpadj1
          }
        } #end u
      } # end i
  } # end f
}# end r

#Flag inconsistent values
incons1=NULL
for (m in 1:nmodels) {
  for (i in 1:nstas) {
    if ( sum(!is.na(DDFmpadj1[m,i,,]))) {
      if (any(apply(DDFmpadj1[m,i,,],1,is.unsorted))) {
        print(paste("m=",m,",i=",i,",1",sep=""))
        incons1=cbind(incons1,c(m,i,1))
      }
      if (any(apply(DDFmpadj1[m,i,,],2,is.unsorted))) {
        print(paste("m=",m,",i=",i,",2",sep=""))
        incons1=cbind(incons1,c(m,i,2))
      }
    }

  }
}


#Use SPM to fix inconsistent values
sepfun <- function(pars,durshr) {
  aT=pars[1:ntss]
  eta=pars[ntss+1]
  #eta=0.75
  bd=1/(durshr[,1]^eta)
  ddf=(bd*durshr[,1])%*%t(aT)
  return(ddf)
}

optfun <- function(pars,dat,durshr) {
  ddffit=sepfun(pars,durshr)
  res1=(ddffit-dat)^2
  return(mean(c(res1)))
}

environment(sepfun)=environment()
environment(optfun)=environment()

DDFmpadj11=DDFmpadj1
lab="Fmpadj1"
```

```
for (inc in 1:ncol(incons1)) {
  par0=c(colMeans(DDFmpadj1[incons1[1,inc],incons1[2,inc],,]/(durshr^(1-0.8))),0.8)
  SPMopt=optim(par=par0,
              fn=optfun,dat=DDFmpadj1[incons1[1,inc],incons1[2,inc],,],durshr=durshr,
              control=list(maxit=10000))
  prevval=SPMopt$value
  par0=c(colMeans(DDFmpadj1[incons1[1,inc],incons1[2,inc],,]/(durshr^(1-
SPMopt$par[ntss+1]))),SPMopt$par[ntss+1])
  conv=FALSE
  while (conv==FALSE) {
    print(paste("conv=",conv))
    SPMopt=optim(par=par0,

fn=optfun,dat=DDFmpadj1[incons1[1,inc],incons1[2,inc],,],durshr=durshr,
                control=list(maxit=10000))
    newval=SPMopt$value
    percchange=(newval-prevval)/prevval*100
    if (abs(percchange) < 0.1) {
      conv=TRUE
    } else {
      par0=c(colMeans(DDFmpadj1[incons1[1,inc],incons1[2,inc],,]/(durshr^(1-
SPMopt$par[ntss+1]))),SPMopt$par[ntss+1])
      prevval=newval
    }
  }
  print(paste("incons1                 =                 ",inc,",conv                =
",SPMopt$convergence,",eta=",round(SPMopt$par[ntss+1],2),sep=""))
  DDFmpadj11[incons1[1,inc],incons1[2,inc],,]=sepfun(SPMopt$par,durshr)

ylim=range(cbind(DDFmpadj1[incons1[1,inc],incons1[2,inc],,],DDFmpadj11[incons1[1,inc],
incons1[2,inc],,]))

png(paste("SPM_adjust_",lab,"_m_",incons1[1,inc],"_sta_",incons1[2,inc],"_",staid2[inc
ons1[2,inc]],".png",sep=""))
  matplot(durshr*60,DDFmpadj1[incons1[1,inc],incons1[2,inc],,],type="b",log="x",
          xlab="log(D) (mins)",ylab="Precipitation (inches)",
          main=c(paste(lab,":     DDF     fits     with     EQM/SPM     at     Station
",staid2[incons1[2,inc]],sep=""),
          paste("Period: ",proj1[1],"-",proj1[2],sep="")),
          axes=FALSE,ylim=ylim,pch=1,lty=1,lwd=1)
  matplot(durshr*60,DDFmpadj11[incons1[1,inc],incons1[2,inc],,],type="b",log="x",
          ylim=ylim,pch=2,lty=2,lwd=2,add=TRUE)
  axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
  axis(side=2)
  axis(side=3,at=dursmins,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
  abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
  lty=c(1:2)
  pch=c(1:2)
  lwd=c(1:2)
  legend("topleft",legend=c(lab,paste(lab,"with
SPM")),lty=lty,pch=pch,lwd=lwd,cex=0.8)
  mtext( paste("Return periods:",toString(subTr),"years"),side=1,adj=1,line=4,cex=0.6)
  box()
  dev.off()
}


#Plot all adjusted DDFs
m=0
for (r in 1:length(RCPs)) {
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))
```

```
  for (f in 1:length(projs)) {
    m=m+1
    print(paste("f = ",f,", m = ",m,sep=""))

    for (i in 1:nstas) {
      print(paste("i = ",i,sep=""))
      lab="Fmpadj1"
      if ( sum(!is.na(DDFmpadj1[m,i,,]))) {
        ylim=range(cbind(DDFmpadj1[m,i,,],DDFmpadj11[m,i,,]))
        png(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_DDF_",
                  staid2[i],"_alldur_EQM_",proj1[1],"-",proj1[2],".png",sep=""))
        matplot(dursshr*60,DDFmpadj1[m,i,,],type="b",log="x",
                xlab="log(D) (mins)",ylab="Precipitation (inches)",
                main=c(paste(lab,": DDF fits with EQM at Station ",staid2[i],sep=""),
                paste("Period: ",proj1[1],"-",proj1[2],sep="")),
                axes=FALSE,ylim=ylim,pch=1,lty=1,lwd=1)
        matplot(durshr*60,DDFmpadj11[m,i,,],type="b",log="x",
                ylim=ylim,pch=2,lty=2,lwd=2,add=TRUE)
        axis(side=1,at=dursmins,labels=durs,las=3,cex.axis=0.7)
        axis(side=2)
        axis(side=3,at=dursmins,cex.axis=0.7,tck=0.02,mgp=c(3,0,0))
        abline(v=dursmins,h=axTicks(side=2),col="lightgray",lty="dotted")
        lty=c(1:2)
        pch=c(1:2)
        lwd=c(1:2)
        legend("topleft",legend=c(lab,paste(lab,"with
SPM")),lty=lty,pch=pch,lwd=lwd,cex=0.8)
        mtext(                                                    paste("Return
periods:",toString(subTr),"years"),side=1,adj=1,line=4,cex=0.6)
        box()
        dev.off()
      }

     }#end i
  }#end f
}#end r


#Get quantiles of interest
quantsmp1=apply(DDFmpadj11,c(2,3,4),quantile,probs=probs,na.rm=TRUE)
dimnames(quantsmp1)[[2]]=staid2
dimnames(quantsmp1)[[3]]=subdurs
dimnames(quantsmp1)[[4]]=subTr

#
#Contourmaps of quantiles of interest
for (u in 1:nsubdurs) {
  for (p in 1:length(probs)) {#
    for (t in 1:ntss) {
      lab="Fmpadj1"
      contourmap(quantsmp1[p,,u,t],staid2,direc=getwd(),
                 main=paste(dataset2,"_",lab,"_DDF_",subdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",proj1[1],"-",proj1[2],sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")

      DDFdiff1=quantsmp1[p,,u,t]-ddfoc[,u,t]
      contourmap(DDFdiff1,staid2,direc=getwd(),
                  main=paste(dataset2,"_",lab,"-
",dataset1,"_Foc_DDF_",subdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",proj1[1],"-",proj1[2],sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")

    }#end t
```

```
  }#end p
}#end u

lab="Fmpadj1"
dimnames(DDFmpadj11)[[1]]=paste("model",seq(1:60),sep="")
dimnames(DDFmpadj11)[[2]]=staid2
dimnames(DDFmpadj11)[[3]]=subdurs
dimnames(DDFmpadj11)[[4]]=subTr
save(DDFmpadj11,quantsmp1,staid2,
     file=paste("stats_quants_",lab,"_alldurallTr.RData",sep=""))


#Do the temporal downscaling
DDFmpadj1_hrly=array(dim=c(nmodels,nstas,nhdur,ntss))
yrindx=1950:2099
m=0
for (r in 1:length(RCPs)) {
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))

  for (f in 1:length(projs)) {
    m=m+1
    print(paste("f = ",f,", m = ",m,sep=""))

    lab="Fmc"
    load(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_GEVpars_",type,"_",
         method,"_",currm[1],"-",currm[2],".RData",sep=""))
    ximc=xi
    alfamc=alfa
    kmc=k
    rm(GEVpars,xi,alfa,k)

    lab="Fmp1"
    load(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_GEVpars_",type,"_",
         method,"_",proj1[1],"-",proj1[2],".RData",sep=""))
    ximp1=xi
    alfamp1=alfa
    kmp1=k
    rm(GEVpars,xi,alfa,k)

    load(paste("./",RCPs[r],"/AMS_",RCPs[r],"_1950_2099_model_",f,".RData",sep=""))
    # Create AMS4-equivalent list
    dimnames(ams)=list(durs,yrindx,staid2)
    AMS4=lapply(seq_len(dim(ams)[3]), function(x) t(ams[,,x]))
    names(AMS4)=staid2

    for (i in 1:nstas) {
       print(paste("i = ",i,sep=""))
         #First, get Xmpadj_daily corresponding to Xmp for proj1 years
         #xmp1=AMS4[[i]][rownames(AMS4[[i]])%in%(seq(proj1[1],proj1[2])),"24-hr"]
         if (!is.na(xioc[i,"24-hr"])) {
         GEVparsoc=c(xi=xioc[i,"24-hr"],alfa=alfaoc[i,"24-hr"],k=koc[i,"24-hr"])
         #GEVparsoc=NULL
         GEVparsmc=c(xi=ximc[i,"24-hr"],alfa=alfamc[i,"24-hr"],k=kmc[i,"24-hr"])
         GEVparsmp1=c(xi=ximp1[i,"24-hr"],alfa=alfamp1[i,"24-hr"],k=kmp1[i,"24-hr"])

#xmpadj1=EQM(GEVparsoc,GEVparsmc,GEVparsmp1,type="ratio",xmp=xmp1,logfile=logfile2)
         }
       for (u in 1:nhdur) {
         print(paste("u = ",u,sep=""))
         if (!is.na(xioc[i,hdurs[u]])) {

GEVparsocsd=c(xi=xioc[i,hdurs[u]],alfa=alfaoc[i,hdurs[u]],k=koc[i,hdurs[u]])
```

```
                #Need  to  use  GEVparsoc  as  GEVparsmc  since  GEVparsmc  has  not  been
adjusted!
                #When GEVparsmc is adjusted, it basically becomes GEVparsoc

#QMmpadj1=temp_QM(GEVparsoc,GEVparsocsd,GEVparsoc,GEVparsmp1,xmpadj=xmpadj1)

QMmpadj1=temp_QM(GEVparsoc,GEVparsocsd,GEVparsmc,GEVparsmp1,Tr=subTr,logfile=logfile2)
           DDFmpadj1_hrly[m,i,u,]=QMmpadj1
         }
       } #end u
          #Plot the adjusted DDFs for all durations
       lab="Fmpadj1"
       if ( sum(!is.na(DDFmpadj1_hrly[m,i,,]))) {
         ylim=range(rbind(DDFmpadj1_hrly[m,i,,],DDFmpadj11[m,i,,]))
         png(paste("./",RCPs[r],"/model_",f,"/",dataset2,"_",lab,"_DDF_",
                   staid2[i],"_alldur_fromhourly_EQM_",proj1[1],"-
",proj1[2],".png",sep=""))

matplot(rbind(hdurshr,durshr)*60,rbind(DDFmpadj1_hrly[m,i,,],DDFmpadj11[m,i,,]),type="
b",log="x",
                      xlab="log(D) (mins)",ylab="Precipitation (inches)",
                 main=c(paste(lab,": DDF fits with EQM at Station ",staid2[i],sep=""),
                 paste("Period: ",proj1[1],"-",proj1[2],sep="")),
                 ylim=ylim,pch=1,lty=1,lwd=1,axes=FALSE)

axis(side=1,at=c(hdursmins,dursmins[1:nsubdurs]),labels=c(hdurs,durs[1:nsubdurs]),las=
3,cex.axis=0.7)
         axis(side=2)

axis(side=3,at=c(hdursmins,dursmins[1:nsubdurs]),cex.axis=0.7,tck=0.02,mgp=c(3,0,0))

abline(v=c(hdursmins,dursmins[1:nsubdurs]),h=axTicks(side=2),col="lightgray",lty="dott
ed")
         lty=c(1)
         pch=c(1)
         lwd=c(1)
         legend("topleft",legend=c(lab),lty=lty,pch=pch,lwd=lwd,cex=0.8)
         mtext(                                                          paste("Return
periods:",toString(subTr),"years"),side=1,adj=1,line=4,cex=0.6)
         box()
         dev.off()
          }
     } # end i
  } # end f
}# end r

quantsmp1h=apply(DDFmpadj1_hrly,c(2,3,4),quantile,probs=probs,na.rm=TRUE)
dimnames(quantsmp1h)[[2]]=staid2
dimnames(quantsmp1h)[[3]]=hdurs
dimnames(quantsmp1h)[[4]]=subTr

#Contourmaps of quantiles of interest
for (u in 1:nhdur) {
  for (p in 1:length(probs)) {#
    for (t in 1:ntss) {
      lab="Fmpadj1"
      contourmap(quantsmp1h[p,,hdurs[u],t],staid2,direc=getwd(),
                 main=paste(dataset2,"_",lab,"_DDF_",hdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",proj1[1],"-",proj1[2],sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")

      DDFdiff1=quantsmp1h[p,,hdurs[u],t]-ddfoch[,hdurs[u],t]
      contourmap(DDFdiff1,staid2,direc=getwd(),
```

```
                  main=paste(dataset2,"_",lab,"-
",dataset1,"_Foc_DDF_",hdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",proj1[1],"-",proj1[2],sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")

    }#end t
  }#end p
}#end u

dimnames(DDFmpadj1_hrly)[[1]]=paste("model",seq(1:60),sep="")
dimnames(DDFmpadj1_hrly)[[2]]=staid2
dimnames(DDFmpadj1_hrly)[[3]]=hdurs
dimnames(DDFmpadj1_hrly)[[4]]=subTr
save(DDFmpadj1_hrly,hdurs,quantsmp1h,staid2,
     file=paste("stats_quants_",lab,"_hrlyduralltr.RData",sep=""))

}

############################################################################
```

**EQM <-
function(GEVparsoc,GEVparsmc,GEVparsmp,type="diff",xmp=NULL,xoc=NULL,
Tr=c(2,5,10,25,50,100,200,500,1000),logfile="logfile.txt"){**

```
#Function to do Quantile Delta Mapping
#GEVparsoc: GEV parameters for the observations in the current period
#           Will not be used when xoc is given in which case GEVparsoc can be set to
NULL
#GEVparsmc: GEV paramters for the model in the current period
#GEVparsmp: GEV parameters for the model in the future (projected) period
#type: "diff" uses an additive model for EQM
#      "ratio" uses a multiplicative model for EQM
#xmp: Future extreme precipitation values to adjust
#     Set to NULL to use return period instead
#xoc: Observed current baseline values
#     Used instead of GEVparsoc when
#     observed current baseline values do not come from a GEV distribution
#     (e.g. when using Official Atlas 14 DDF values)
#     Set to NULL to use GEVparsoc
#     Only used when xmp==NULL
#Tr: return periods of interest
#   Set to NULL to use xmp instead

############################################################################

library(extRemes)

if (!is.null(xmp)) {
  if (type=="diff") {

xmpadj=xmp+qevd(pevd(xmp,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                         GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV")
-

qevd(pevd(xmp,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                         GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
  } else {

xmpadj=xmp*qevd(pevd(xmp,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                         GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV")
/

qevd(pevd(xmp,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
```

```
                         GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
  }

  return(xmpadj)

} else {
  p = 1-1/Tr
  if (type=="diff") {
    if (is.null(xoc)) {
      qmpadj=qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV") +
             qevd(p,GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV") -
             qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
    } else {
      qmpadj=qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV") +
             unlist(list(xoc)) -
             qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
    }
    #Use QM when there are inconsistencies
    if (is.unsorted(qmpadj)) {

write(paste("unsorted:",GEVparsmp["k"],GEVparsoc["k"],GEVparsmc["k"]),file=logfile,app
end=TRUE)
      if (is.null(xoc)) {

qmpadj=qevd(pevd(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                    GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),
                    GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV")
      } else {

qmpadj=approx(x=p,y=xoc,xout=pevd(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["
k"],type="GEV"),

GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),method="linear",rule=2)$y
      }
      #for (o in 2:length(qmpadj)) {
      #  if (qmpadj[o]<qmpadj[o-1]) qmpadj[o]=qmpadj[o-1]+0.01
      #}
    }


  } else {
    if (is.null(xoc)) {
      qmpadj=qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV") *
             qevd(p,GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV") /
             qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
    } else {
      qmpadj=qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV") *
             unlist(list(xoc)) /
             qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")
    }

    #Use QM when there are inconsistencies
    if (is.unsorted(qmpadj)) {

write(paste("unsorted:",GEVparsmp["k"],GEVparsoc["k"],GEVparsmc["k"]),file=logfile,app
end=TRUE)
      if (is.null(xoc)) {

qmpadj=qevd(pevd(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                    GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),
                    GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV")
      } else {
```

```
qmpadj=approx(x=p,y=xoc,xout=pevd(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["
k"],type="GEV"),

GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),method="linear",rule=2)$y
      }
      #for (o in 2:length(qmpadj)) {
      #  if (qmpadj[o]<qmpadj[o-1]) qmpadj[o]=qmpadj[o-1]+0.01
      #}
    }


  }

  return(qmpadj)
}

##png("EQM_issue2.png")
#plot(qevd(p,GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV"),p,type="l",
#     xlim=c(0,15),ylim=c(0,1),lty=1,lwd=2,col="red",main="Quantile Delta Method",
#      xlab="Precipitation depth (inches)",ylab=expression(CDF: G==P(X<=x)))
##,ylim=c(0.97,1.0)
#lines(qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),p,col="blue
",lty=2,lwd=2)
#lines(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),p,col="blac
k",lty=3,lwd=2)
#lines(qmpadj2,p,col="green",lty=4,lwd=2)
#lines(qmpadj,p,col="orange",lty=4,lwd=2)
#legend("bottomright",col=c("red","blue","black","green","orange"),lty=1:4,legend=c(ex
pression(F["o-c"]),expression(F["m-c"]),
#             expression(F["m-p"]),expression(F["m-padj.  mult"]),expression(F["m-padj.
QM"])),lwd=2)
##p2=pevd(xmp,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV")
##lines(xmp,p2,lty=2)
##lines(xmpadj,p2,lty=2,col="green")
#axis(side=4,at=1-1/Tr,labels=Tr,cex.axis=0.5,las=1)
#abline(h=c(0,1,1-1/Tr),col="gray70",lty=2)
#mtext("Tr (years)",side=4,cex=0.5)
#grid()
##dev.off()


}


############################################################################
```

**temp_QM <-
function(GEVparsoc,GEVparsocsd,GEVparsmc,GEVparsmp,xmpadj=NULL,
Tr=c(2,5,10,25,50,100,200,500,1000),logfile="logfile2.txt"){**

```
############################################################################

#Function to do quantile mapping for temporal downscaling
#GEVparsoc: GEV parameters for the daily observations in the current period
#           Not used when xmpadj is defined
#GEVparsocsd: GEV parameters for the sub-daily observations in the current period
#GEVparsmc: GEV paramters for the model in the current period (daily duration)
#GEVparsmp: GEV parameters for the model in the future (projected) period (daily
duration)
#           Not used when xmpadj is defined
#xmpadj: Future ADJUSTED extreme precipitation values
```

```
#     Set to NULL to use return period instead
#Tr: return periods of interest
#   Set to NULL to use xmpadj instead

library(extRemes)

if (!is.null(xmpadj)) {

  qmpadjsd=qevd(pevd(xmpadj,
              GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),
              GEVparsocsd["xi"],GEVparsocsd["alfa"],GEVparsocsd["k"],type="GEV")
  return(qmpadjsd)

} else {
  p = 1-1/Tr
  qmpadj=qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV") *
         qevd(p,GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV") /
         qevd(p,GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV")

  #Use QM when there are inconsistencies
  if (is.unsorted(qmpadj)) {

write(paste("unsorted:",GEVparsmp["k"],GEVparsoc["k"],GEVparsmc["k"]),file=logfile,app
end=TRUE)

qmpadj=qevd(pevd(qevd(p,GEVparsmp["xi"],GEVparsmp["alfa"],GEVparsmp["k"],type="GEV"),
                GEVparsmc["xi"],GEVparsmc["alfa"],GEVparsmc["k"],type="GEV"),
                GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV")
  }
  #Here GEVparsoc takes the place of GEVparsmc since after bias correction
  #GEVparsmc=GEVparsoc
  qmpadjsd=qevd(pevd(qmpadj,
              GEVparsoc["xi"],GEVparsoc["alfa"],GEVparsoc["k"],type="GEV"),
              GEVparsocsd["xi"],GEVparsocsd["alfa"],GEVparsocsd["k"],type="GEV")


  return(qmpadjsd)
}


}
##########################################################################
```

```
##########################################################################

computeGOFquants   <-   function(subdurs=c("24-hr","2-day","3-day","4-day","7-day"),       subTr=c(2,5,10,25,50,100),probs=c(0.05,0.50,0.95), dataset1="FL_LOCA",lab1="Fmc",syr1=1950,eyr1=2008,dataset2="FL_LOCA",lab2="Fmp2",syr2=2040,eyr2=2079) {

#Function to compute goodness-of-fit statistics and quantiles of DDF curves

###########################################################################

#     dataset1="FL_Atlas14",lab1="Foc",syr1=1950,eyr1=2008,
#                       dataset2="FL_LOCA",lab2="Fmc",syr2=1950,eyr2=2008) {

source("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Code/contourmap.R")

library(plotrix)
library(nsRFA)

data_dir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/LOCA_dataset/Data/"

setwd(data_dir)

RCPs=c("RCP45","RCP85")

#Durations of interest
nsubdurs=length(subdurs)

#Return periods of interest
ntss=length(subTr)

# Get staid2 (ids of weather stations)
load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/stationids.RData")
nstas=length(staid2)

nmodels=60
nvals=1+nsubdurs*ntss

#Initialize matrices of goodness-of-fit statistics
mDiff=mRatio=mSD=mSDRatio=mRMSDCRatio=oSD=mRMSD=mRMSDC=mMAE=mR=mNS=matrix(nrow=nmodels,ncol=nvals)

#Initialize arrays of DDF and GEV parameters
mxi1=malfa1=mk1=mxi2=malfa2=mk2=array(dim=c(nmodels,nstas,nsubdurs))
mddf1=mddf2=array(dim=c(nmodels,nstas,nsubdurs,ntss))

# Get Foc (reference)
if (lab1=="Foc") {

load("Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/Obs_datasets/before_2005/MDC_obs_Foc_GEVpars_bysite_RegLmom_1940-2005.RData")
  #Create matrix from DDF list
  ddf1=array(dim=c(nstas,nsubdurs,ntss),dimnames=list(staid2,subdurs,subTr))
  #oo=lapply(seq_len(length(DDF)),                                    function(i) ddf2[i,,]=t(DDF[[i]][as.character(subTr),subdurs]))
  for (i in 1:nstas){
    if (!is.null(DDF[[i]])) ddf1[i,,]=t(DDF[[i]][as.character(subTr),subdurs])
  }
```

```
#load("C:/Users/miriza/Documents/Work/SFWMD_Contract_IDF/ATLAS14/IDF/FL_Atlas14_AMS_ID
F.RData")
  #DDF_official_ATLAS14=IDF
  #ddf1=DDF_official_ATLAS14[,10:14,1:6]

  oSD=sd(na.omit(as.vector(ddf1)))
  for (u in 1:nsubdurs) {
    for (t in 1:ntss) {
      oSD=c(oSD,sd(na.omit(as.vector(ddf1[,u,t]))))
    }
  }
}

if (lab2=="Fmpadj1" | lab2=="Fmpadj2") {
  load(paste("stats_quants_",lab2,"_allduralltr.RData",sep=""))
  mddf2=get(paste("DD",lab2,substr(lab2,nchar(lab2),nchar(lab2)),sep=""))
}


# Get all the models (Fmc or Fmp)
im=0 #counter for the models
allprojs=NULL
for (r in 1:length(RCPs)) { #
  print(paste("r = ",r,sep=""))
  projs=readLines(paste("./",RCPs[r],"/Projections5.txt",sep=""))
  allprojs=c(allprojs,projs)


  for (f in 1:length(projs)) {
    print(paste("f = ",f,sep=""))
    im=im+1

    if (lab1!="Foc") {
      load(paste("./",RCPs[r],"/model_",f,
                 "/",dataset1,"_",lab1,"_GEVpars_bysite_RegLmom_",syr1,"-
",eyr1,".RData",sep=""))
      #load(paste("./",RCPs[r],"/model_",f,
      #                       "/",dataset1,"_",lab1,"_GEVpars_bysite_RegLmom_",syr1,"-
",eyr1,"_allyrs.RData",sep=""))


      #Populate the overall arrays
      mxi1[im,,]=xi
      malfa1[im,,]=alfa
      mk1[im,,]=k

      #Create matrix from DDF list
      ddf1=array(dim=c(nstas,nsubdurs,ntss),dimnames=list(staid2,subdurs,subTr))
      #oo=lapply(seq_len(length(DDF)),                                     function(i)
ddf1[i,,]=t(DDF[[i]][as.character(subTr),subdurs]))
      for (i in 1:nstas){
        if (!is.null(DDF[[i]])) ddf1[i,,]=t(DDF[[i]][as.character(subTr),subdurs])
      } #end i

      mddf1[im,,,]=ddf1

    }

    if (!(lab2=="Fmpadj1" | lab2=="Fmpadj2")) {
      #load(paste("./",RCPs[r],"/model_",f,
      #                       "/",dataset2,"_",lab2,"_GEVpars_bysite_RegLmom_",syr2,"-
",eyr2,"_allyrs.RData",sep=""))
```

```
      load(paste("./",RCPs[r],"/model_",f,
                   "/",dataset2,"_",lab2,"_GEVpars_bysite_RegLmom_",syr2,"-
",eyr2,".RData",sep=""))

      #Populate the overall arrays
      mxi2[im,,]=xi
      malfa2[im,,]=alfa
      mk2[im,,]=k

      #Create matrix from DDF list
      ddf2=array(dim=c(nstas,nsubdurs,ntss),dimnames=list(staid2,subdurs,subTr))
      #oo=lapply(seq_len(length(DDF)),                                function(i)
ddf2[i,,]=t(DDF[[i]][as.character(subTr),subdurs]))
      for (i in 1:nstas){
        if (!is.null(DDF[[i]])) ddf2[i,,]=t(DDF[[i]][as.character(subTr),subdurs])
      } #end i

      mddf2[im,,,]=ddf2

    } else {
      ddf2=mddf2[im,,,]
    }


    if (r==1) colr="red"
    if (r==2) colr="blue"
    if (r==3) colr="green"

    if (r==1 & f==1) {

taylor.diagram(ddf1,ddf2,col=colr,sd.arcs=3,grad.corr.lines=c(0.2,0.4,0.6,0.7,0.8,0.9)
,
                    ngamma=5,normalize=TRUE)
    } else {
      taylor.diagram(ddf1,ddf2,add=TRUE,col=colr,normalize=TRUE)
    }

    #valid stations
    vs=(!is.na(ddf1[,1,1])&!is.na(ddf2[,1,1]))

    mDiff[im,1]=mean(na.omit(as.vector(ddf2[vs,,]))-na.omit(as.vector(ddf1[vs,,])))
    mRatio[im,1]=mean(na.omit(as.vector(ddf2[vs,,]))/na.omit(as.vector(ddf1[vs,,])))
    mSD[im,1]=sd(na.omit(as.vector(ddf2[vs,,])))
    mRMSD[im,1]=RMSE(na.omit(as.vector(ddf1[vs,,])),na.omit(as.vector(ddf2[vs,,])))
    mRMSDC[im,1]=RMSE(na.omit(as.vector(ddf1[vs,,]))-mean(ddf1[vs,,],na.rm=TRUE),
                      na.omit(as.vector(ddf2[vs,,]))-mean(ddf2[vs,,],na.rm=TRUE))
    mMAE[im,1]=MAE(na.omit(as.vector(ddf1[vs,,])),na.omit(as.vector(ddf2[vs,,])))
    mR[im,1]=cor(ddf1[vs,,],ddf2[vs,,],use="pairwise")
    mNS[im,1]=R2(na.omit(as.vector(ddf1[vs,,])),na.omit(as.vector(ddf2[vs,,])))
    if (lab1!="Foc") {
      oSD[im,1]=sd(na.omit(as.vector(ddf1[vs,,])))
      mSDRatio[im,1]=mSD[im,1]/oSD[im,1]
      mRMSDCRatio[im,1]=mRMSDC[im,1]/oSD[im,1]
    }

    icol=1
    for (u in 1:nsubdurs) {
      for (t in 1:ntss) {
        icol=icol+1
        mDiff[im,icol]=mean(na.omit(as.vector(ddf2[vs,u,t]))-
na.omit(as.vector(ddf1[vs,u,t])))
```

```
mRatio[im,icol]=mean(na.omit(as.vector(ddf2[vs,u,t]))/na.omit(as.vector(ddf1[vs,u,t]))
)
        mSD[im,icol]=sd(na.omit(as.vector(ddf2[vs,u,t])))

mRMSD[im,icol]=RMSE(na.omit(as.vector(ddf1[vs,u,t])),na.omit(as.vector(ddf2[vs,u,t])))
        mRMSDC[im,icol]=RMSE(na.omit(as.vector(ddf1[vs,u,t]))-
mean(ddf1[vs,u,t],na.rm=TRUE),
                            na.omit(as.vector(ddf2[vs,u,t]))-
mean(ddf2[vs,u,t],na.rm=TRUE))

mMAE[im,icol]=MAE(na.omit(as.vector(ddf1[vs,u,t])),na.omit(as.vector(ddf2[vs,u,t])))
        mR[im,icol]=cor(ddf1[vs,u,t],ddf2[vs,u,t],use="pairwise")

mNS[im,icol]=R2(na.omit(as.vector(ddf1[vs,u,t])),na.omit(as.vector(ddf2[vs,u,t])))
        if (lab1!="Foc") {
          oSD[im,icol]=sd(na.omit(as.vector(ddf1[vs,u,t])))
          mSDRatio[im,icol]=mSD[im,icol]/oSD[im,icol]
          mRMSDCRatio[im,icol]=mRMSDC[im,icol]/oSD[im,icol]
        }

      }#end t
    } #end u

  } #end f
} # end r

mR2=mR^2
if (lab1=="Foc") {
  mSDRatio=mSD/matrix(rep(oSD,im),nrow=im,ncol=nvals,byrow=TRUE)
  mRMSDCRatio=mRMSDC/matrix(rep(oSD,im),nrow=im,ncol=nvals,byrow=TRUE)
}

rownames(mDiff)=rownames(mRatio)=rownames(mSD)=rownames(mRMSD)=rownames(mRMSDC)=rownam
es(mMAE)=

rownames(mR)=rownames(mR2)=rownames(mNS)=rownames(mSDRatio)=rownames(mRMSDCRatio)=
                allprojs
colnames(mDiff)=colnames(mRatio)=colnames(mSD)=colnames(mRMSD)=colnames(mRMSDC)=colnam
es(mMAE)=

colnames(mR)=colnames(mR2)=colnames(mNS)=colnames(mSDRatio)=colnames(mRMSDCRatio)=
                c("all",paste(rep(subdurs,each=ntss),"_",rep(subTr,nsubdurs),"-
year",sep=""))

#substr(allprojs,1,nchar(allprojs)-6)
#unique(substr(allprojs,1,nchar(allprojs)-6))

#Taylor diagrams for all durations and return periods of interest
icol=1
png(paste("Taylor_diagram_",lab1,"_vs_",lab2,"_",colnames(mR)[icol],"_normalized.png",
sep=""))
taylor.diagram(ddf1,ddf1,sd.arcs=3,grad.corr.lines=seq(0.1,0.9,by=0.1),
               ngamma=5,normalize=TRUE,pch=1,col="black",main=c("Taylor
Diagram",colnames(mR)[icol]))
points(na.omit(mSDRatio[,icol])*na.omit((mR[,icol])),na.omit(mSDRatio[,icol])*sin(acos
(na.omit((mR[,icol])))),
       col=1:6,pch=2:21)
#lpos<-1.5
#legend(lpos,lpos,legend=allprojs,cex=0.5,ncol=2,col=1:6,pch=2:21)
dev.off()

png(paste("Taylor_diagram_",lab1,"_vs_",lab2,"_",colnames(mR)[icol],".png",sep=""))
```

```
taylor.diagram(ddf1,ddf1,sd.arcs=3,grad.corr.lines=seq(0.1,0.9,by=0.1),
               ngamma=5,pch=1,col="black",main=c("Taylor Diagram",colnames(mR)[icol]))
points(na.omit(mSD[,icol])*na.omit((mR[,icol])),na.omit(mSD[,icol])*sin(acos(na.omit((
mR[,icol])))),
       col=1:6,pch=2:21)
#lpos<-1.5
#legend(lpos,lpos,legend=allprojs,cex=0.5,ncol=2,col=1:6,pch=2:21)
dev.off()

for (u in 1:nsubdurs) {
  for (t in 1:ntss) {
    icol=icol+1

png(paste("Taylor_diagram_",lab1,"_vs_",lab2,"_",colnames(mR)[icol],"_normalized.png",
sep=""))

taylor.diagram(ddf1[,u,t],ddf1[,u,t],sd.arcs=3,grad.corr.lines=seq(0.1,0.9,by=0.1),
                   ngamma=5,normalize=TRUE,pch=1,col="black",main=c("Taylor
Diagram",colnames(mR)[icol]),pos.cor=FALSE)

points(na.omit(mSDRatio[,icol])*na.omit((mR[,icol])),na.omit(mSDRatio[,icol])*sin(acos
(na.omit((mR[,icol])))),
           col=1:6,pch=2:21)
    dev.off()


png(paste("Taylor_diagram_",lab1,"_vs_",lab2,"_",colnames(mR)[icol],".png",sep=""))

taylor.diagram(ddf1[,u,t],ddf1[,u,t],sd.arcs=3,grad.corr.lines=seq(0.1,0.9,by=0.1),
                   ngamma=5,pch=1,col="black",main=c("Taylor
Diagram",colnames(mR)[icol]),pos.cor=FALSE)

points(na.omit(mSD[,icol])*na.omit((mR[,icol])),na.omit(mSD[,icol])*sin(acos(na.omit((
mR[,icol])))),
           col=1:6,pch=2:21)
    dev.off()

  }
}


#Boxplots of statistics
stats=c("Diff","Ratio","SD","RMSD","RMSDC","MAE","R","R2","NS","SDRatio","RMSDCRatio")
substats=c("Diff","RMSD","MAE","Ratio","SDRatio","R2","NS")


png(paste("allstats_",lab1,"_vs_",lab2,"_alldurallltr.png",sep=""))
nf=layout(t(seq(1:length(substats))),width=rep(1,length(substats)))
for (s in 1:length(substats)) {
  par(mar=c(5,0,5,0),mgp=c(1,0,0))

bxp=boxplot(get(paste("m",substats[s],sep=""))[,1],xlab=substats[s],axes=FALSE,cex.lab
=1.5)
  text(y=as.vector((bxp$stats)),x=(rep(c(1),each=5))+0.35,labels=round(bxp$stats,2),)
}
dev.off()


for (s in 1:length(stats)) {

png(paste(stats[s],"_",lab1,"_vs_",lab2,"_boxplot.png",sep=""),height=1000,width=720,p
ointsize=20)
  nf=layout((c(1,2,3,4,5,6)),heights=c(2,2,2,2,2,2))
```

```
  #pars=par(mar=c(5.1,4.1,4.1,10),xpd=TRUE)
  par(mar=c(0,4.1,2,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(7,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[6],"-yr",sep=""),
      main=paste("Comparison of",stats[s],"across models"))
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:nsubdurs,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$st
ats[3,])),2),
      col="red",font=2)

  par(mar=c(1,4.1,1,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(6,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[5],"-yr",sep=""))
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:nsubdurs,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$st
ats[3,])),2),
      col="red",font=2)

  par(mar=c(1,4.1,0,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(5,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[4],"-yr",sep=""))
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:nsubdurs,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$st
ats[3,])),2),
      col="red",font=2)

  par(mar=c(1,4.1,0,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(4,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[3],"-yr",sep=""))
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:nsubdurs,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$st
ats[3,])),2),
      col="red",font=2)

  par(mar=c(1,4.1,0,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(3,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[2],"-yr",sep=""))
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:nsubdurs,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$st
ats[3,])),2),
      col="red",font=2)

  par(mar=c(2,4.1,0,2.1),mgp=c(2,1,0))

bxp=boxplot(get(paste("m",stats[s],sep=""))[,seq(2,31,by=6)],xaxt="n",ylab=paste("Tr="
,subTr[1],"-yr",sep=""))
  axis(1,at=1:nsubdurs,labels=subdurs,cex.axis=1)
  abline(h=axTicks(side=2),col="lightgray",lty="dotted")

text(x=rep(1:ntss,1),y=as.vector(t(bxp$stats[3,])),labels=round(as.vector(t(bxp$stats[
3,])),2),
       col="red",font=2)
```

```
  dev.off()
}

#Quantiles of GEV parameters and DDF
if              (lab1!="Foc")              quantsddf1=apply(mddf1,c(2,3,4),function(x)
quantile(x,probs=probs,na.rm=TRUE))
if (!(lab2=="Fmpadj1" | lab2=="Fmpadj2")) {
  dimnames(mk2)=dimnames(malfa2)=dimnames(mxi2)=list(allprojs,staid2,subdurs)
  quantsxi2=apply(mxi2,c(2,3),function(x) quantile(x,probs=probs,na.rm=TRUE))
  quantsalfa2=apply(malfa2,c(2,3),function(x) quantile(x,probs=probs,na.rm=TRUE))
  quantsk2=apply(mk2,c(2,3),function(x) quantile(x,probs=probs,na.rm=TRUE))
}
quantsddf2=apply(mddf2,c(2,3,4),function(x) quantile(x,probs=probs,na.rm=TRUE))
dimnames(quantsddf2)[[2]]=staid2
dimnames(quantsddf2)[[3]]=subdurs
dimnames(quantsddf2)[[4]]=subTr

if (lab1!="Foc") {
  diffddf=mddf2-mddf1
} else {
  diffddf=array(dim=c(nmodels,nstas,nsubdurs,ntss))
  for (im in 1:nmodels) {
    diffddf[im,,,]=mddf2[im,,,]-ddf1
  }
}

quantsdiffddf=apply(diffddf,c(2,3,4),function(x) quantile(x,probs=probs,na.rm=TRUE))
dimnames(quantsdiffddf)[[2]]=staid2
dimnames(quantsdiffddf)[[3]]=subdurs
dimnames(quantsdiffddf)[[4]]=subTr

#Contour maps of quantiles of GEV parameters and DDF
for (u in 1:nsubdurs) {
  for (p in 1:length(probs)) {#
    if (!(lab2=="Fmpadj1" | lab2=="Fmpadj2")) {
      contourmap(quantsxi2[p,,u],staid2,direc=getwd(),

main=paste(dataset2,"_",lab2,"_GEV_locpar_",subdurs[u],"_",probs[p],"_",syr2,"-
",eyr2,"",sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")
      contourmap(quantsalfa2[p,,u],staid2,direc=getwd(),

main=paste(dataset2,"_",lab2,"_GEV_scalepar_",subdurs[u],"_",probs[p],"_",syr2,"-
",eyr2,"",sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")
      contourmap(quantsk2[p,,u],staid2,direc=getwd(),

main=paste(dataset2,"_",lab2,"_GEV_shapepar_",subdurs[u],"_",probs[p],"_",syr2,"-
",eyr2,"",sep=""),
                 res=1000,idp=2,posonly=FALSE,pval=NULL,labs="none",
                 zlim=c(-
max(abs(quantsk2[p,,u]),na.rm=TRUE),max(abs(quantsk2[p,,u]),na.rm=TRUE)))
    }
    for (t in 1:ntss) {
      contourmap(quantsddf2[p,,u,t],staid2,direc=getwd(),
                 main=paste(dataset2,"_",lab2,"_DDF_",subdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",syr2,"-",eyr2,"",sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")

      #if (lab1!="Foc") ddfdiff=quantsddf2[p,,u,t]-quantsddf1[p,,u,t]
      #if (lab1=="Foc") ddfdiff=quantsddf2[p,,u,t]-ddf1[,u,t]
      contourmap(quantsdiffddf[p,,u,t],staid2,direc=getwd(),
```

```
                 main=paste(dataset2,"_",lab2,"-
",dataset1,"_",lab1,"_DDF_",subdurs[u],"_",subTr[t],
                 "-year_",probs[p],"_",syr2,"-",eyr2,"",sep=""),
                 res=1000,idp=2,posonly=TRUE,pval=NULL,labs="none")


    }#end t
  }#end p
}#end u

mdifnegall=apply(mDiff,2,function(x) sum(x<0))
mdifnegrcp45=apply(mDiff[1:30,],2,function(x) sum(x<0))
mdifnegrcp85=apply(mDiff[31:60,],2,function(x) sum(x<0))

png(paste("Perc_negchanges_",lab1,"_vs_",lab2,".png",sep=""))
plot(mdifnegrcp45/30*100,col="blue",type="b",pch=2,ylim=c(0,100),axes=FALSE,
     main=c("Percentage of models showing negative overall changes",
     paste("in extremes by RCP category for ",lab2," (",syr2,"-",eyr2,")",sep="")),
     xlab="",ylab="%",cex.main=0.9)
points(mdifnegrcp85/30*100,col="green",type="b",pch=3)
axis(1,at=1:31,labels=colnames(mDiff),las=3,cex.axis=0.5)
abline(v=c(1,seq(2,31,by=6)))
axis(2)
legend("topright",legend=c("RCP45(30)","RCP85(30)"),pch=2:3,col=c("blue","green"))
box()
dev.off()

png(paste("Number_negchanges_",lab1,"_vs_",lab2,".png",sep=""))
plot(mdifnegrcp45,col="blue",type="b",pch=2,ylim=c(0,65),axes=FALSE,
     main=c("Number of models showing negative overall changes",
     paste("in extremes by RCP category for ",lab2," (",syr2,"-",eyr2,")",sep="")),
     xlab="",ylab="#",cex.main=1.1)
points(mdifnegrcp85,col="green",type="b",pch=3)
axis(1,at=1:31,labels=colnames(mDiff),las=3,cex.axis=0.5)
abline(v=c(1,seq(2,31,by=6)))
axis(2)
legend("bottomright",legend=c("RCP45(30)","RCP85(30)"),pch=2:3,col=c("blue","green"))
box()
dev.off()


if (lab2=="Fmpadj1" | lab2=="Fmpadj2") {
     save(mDiff,mRatio,mSD,mRMSD,mRMSDC,mMAE,mR,mNS,mxi2,malfa2,mk2,mddf2,
     quantsxi2,quantsalfa2,quantsk2,quantsddf2,quantsdiffddf,staid2,
     file=paste("stats_quants_",lab1,"_vs_",lab2,"_allduralltr.RData",sep=""))
} else {
     save(mDiff,mRatio,mSD,mRMSD,mRMSDC,mMAE,mR,mNS,mddf2,
     quantsddf2,quantsdiffddf,staid2,
     file=paste("stats_quants_",lab1,"_vs_",lab2,"_allduralltr.RData",sep=""))

}

quantsmdiff=apply(mDiff,2,quantile,probs=c(0.05,.1,.5,.9,.95))
quantsmratio=100*(apply(mRatio,2,quantile, probs=c(0.05,.1,.5,.9,.95))-1)

library(rtf)
output=paste(lab1,"_vs_",lab2,"_diff_ratio_tables.doc",sep="")
rtf=RTF(output,width=8.5,height=11,font.size=10,omi=c(1,1,1,1))

for (u in 1:nsubdurs) {
  mat1=matrix(paste(round(quantsmdiff[,(2+(u-1)*ntss):(1+(u*ntss))],2),"
(",round(quantsmratio,1)[,(2+(u-1)*ntss):(1+(u*ntss))],"%)",sep=""),ncol=ntss)
  colnames(mat1)=colnames(quantsmdiff[,(2+(u-1)*ntss):(1+(u*ntss))])
```

```
  rownames(mat1)=rownames(quantsmdiff[,(2+(u-1)*ntss):(1+(u*ntss))])
  #copytable(mat1,align=rep("r",ntss+1))
  addParagraph(rtf,paste("Table ",u,". Differences in ",subdurs[u]," DDF precipitation
depths in inches (%) for various return periods for ",
                         lab2, " - ", lab1, ". 5-95th percentiles across models
shown.\n",sep=""))

addTable(rtf,mat1,font.size=9,col.justify=rep("R",ntss+1),header.col.justify=rep("C",n
tss+1),
           col.widths=c(0.42,rep(1.0,ntss)),row.names=TRUE)
  addNewLine(rtf)
  addNewLine(rtf)
}

done(rtf)

}

copytable <- function(x, cap=NULL,align...) {
  library(xtable)
  f <- tempfile(fileext=".html")
  print(xtable(x, caption=cap, align=align...), "html", file = f)
  browseURL(f)
}


###########################################################################
```

contourmap_Tps <- function(statis,ids,direc="./",main,res=1000,posonly=TRUE,pval=NULL,labs="none",zlim=NULL){

```
# Uses Tps to smooth the data over FL and then contour it
# statis: data to contour
# ids: IDs of the stations
# direc: Directory where to save the png file
# main: title for plot and file name
# res: resolution of grid for interpolation prior to contouring
# posonly: whether variable only has positive values
# pval: pval associated with the data (default is NULL, i.e. none)
#       If given it must be the same length as statis and stations with a
#       significant pval (<0.05) are labeled with an '*'
# labs: "none" so only station locations are plotted (default)
#       "names" so stations are plotted and labeled with their names
#       "data" so stations are plotted and labeled with the data values
#       "dn" so stations are plotted and labeled with their names and data values
# zlim: Limits for colormap use NULL to have code compute them automatically from data
ranges
#       Enter a pair of values otherwise (e.g. c(-0.5,0.5))
###########################################################################

library(maps)
library(akima)
library(sp)
library(ggplot2)
library(maptools)
library(gstat)
library(colorRamps)
library(raster)
library(geospt)
library(fields)

# First eliminate stations with missing (NA) data.
if (!is.null(pval)) {
```

```
  tokeep=is.finite(statis)&is.finite(pval)
  statis=statis[tokeep]
  ids=ids[tokeep]
  pval=pval[tokeep]
} else {
  tokeep=is.finite(statis)
  statis=statis[tokeep]
  ids=ids[tokeep]
  pval=pval[tokeep]
}
# Number of stations left
nstas=length(ids)
print(names(statis))
print (length(statis))

datadir="Z:/miriza/Work/FIU/FL_Building_Code/Data/Rainfall/"

# Load FL boundary
FL_Boundary=read.csv(paste(datadir,"/Code/FL_Boundarydetailed.csv",sep=""))

#Load canals
cnls=shapefile("Z:/miriza/Work/FIU/FL_Building_Code/Data/USGS_MODFLOW/ancillary/ancill
ary/gis/umd_swr_hydrography.shp")
cnlslatlon=spTransform(cnls,CRS="+proj=longlat        +datum=WGS84        +ellps=WGS84
+towgs84=0,0,0")

## Get station locations
# Read weather station file
stas=read.csv(paste(datadir,"/ATLAS14/noaa_atlas14_included_stations.csv",sep=""),fill
=FALSE,stringsAsFactors=FALSE)
stas2=read.csv(paste(datadir,"/SFWMD/sfwmd_hourly_included_stations.csv",sep=""),fill=
FALSE,stringsAsFactors=FALSE)
stas3=read.csv(paste(datadir,"/SFWMD/sfwmd_included_stations.csv",sep=""),fill=FALSE,s
tringsAsFactors=FALSE)

# Weather station lats and lons
unordered_mystasNames=c(stas$STATION.ID,stas2$STATION,stas3$DBKEY)
unordered_stasLat=c(stas$LAT..degrees,stas2$LAT..degrees,stas3$LAT..degrees)
unordered_stasLon=c(stas$LONG..degrees,stas2$LONG..degrees,stas3$LONG..degrees)


#Get lat and lon for stations in the order they're listed in mds_amsunc30
stasLat=unordered_stasLat[match(names(statis),unordered_mystasNames)]
stasLon=unordered_stasLon[match(names(statis),unordered_mystasNames)]

print(paste(length(stasLat),length(stasLon)))
mydata=data.frame(cbind(statis,stasLon,stasLat))
names(mydata)=c("statis","x","y")
coordinates(mydata) = ~x + y

#Define labels for plot
a=character(nstas)
pch=rep(16,nstas)
if (!is.null(pval)) {
  a[pval<0.05]="*"
  pch[pval<0.05]=15
}

if (labs=="none") lab=rep(NULL,nstas)
if (labs=="names") lab=paste(ids,sep="")
if (labs=="data") lab=paste(round(statis,1),a,sep="")
if (labs=="dn") lab=paste(ids,": ",round(statis,1),a,sep="")
```

```
#Determine colormap and z-limits
if (posonly) {
  colorpal=matlab.like
  if (is.null(zlim)) zlim=range(statis, finite=TRUE)
} else {
  zabmax=max(abs(statis))
  colorpal=blue2red
  if (is.null(zlim)) zlim=c(-zabmax,zabmax)
}
#Make a grid unto which to interpolate data
grd=expand.grid(x=seq(min(stasLon),max(stasLon),length=res),
                y=seq(min(stasLat),max(stasLat),length=res))
coordinates(grd) = ~x + y
gridded(grd) = TRUE

histogram(statis)
histogram(log(statis))

#Try Tps with lambda=0.02 for smoothing (lambda=0.0 gives exact interpolation)
grid.list=list(x=seq(min(stasLon),max(stasLon),length=res),
                y=seq(min(stasLat),max(stasLat),length=res))
t<-Tps(cbind(stasLon,stasLat),statis,lambda=0.02)
u<-predictSurface(t,grid.list,extrap=TRUE)

#Reformat the output for mapping
xcoord=grd$x
ycoord=grd$y
zcoord=u$z
mycoords=list(x=xcoord,y=ycoord)
ind=point.in.polygon(xcoord,ycoord,FL_Boundary[,1],FL_Boundary[,2])
zcoord[!ind]=NA
u$z[!ind]=NA

png(paste(direc,"/contourmap_lines",main,"_TPS.png",sep=""),height=720,width=720,point
size=15)
surface(u,axes=FALSE,xlim=c(-81.2,-80),ylim=c(25,26.2),
        xlab="",ylab="",asp=1,labcex=1,ps=18,legend.shrink=0.6)#zlim=c(zmin,zmax)
map('county',"Florida",type="l",xlim=c(-81.2,-
80),ylim=c(25,26.2),asp=1,col="black",add=TRUE)
axis(1)
axis(2)
box()
grid()
lines(cnlslatlon,col="light blue")
points(stasLon,stasLat,pch=pch,cex=0.6)
text(stasLon,stasLat,labels=lab,cex=0.6,pos=4)
title(main=main,xlab="Lon",ylab="Lat")
mtext(paste("TPS interpolation with lambda of 0.02 ",sep=""),side=3,line=0,cex=0.6)
if (!is.null(pval)) mtext("* Significant at the 0.05 level",side=1,adj=1,line=3,cex=0.6)
dev.off()
}
```

Appendix III. Evaluation of FBC Related Requirements

Task 3. An assessment of the potential changes to the code for incorporating sea level rise and update extreme rainfall using Miami-Dade area as a case study

Rain Loads

Objective 3.1: Evaluate the current Florida Building Code requirements to recommend what additional steps will be necessary to incorporate results of the study into relevant sections of the Codes. Specifically, the changes to the rain loads and their implications for **Rain Loads** as applied to Figure 1611.1 and figure 1106.1 of the FBC, Plumbing shall be recommended.

Context for evaluation: Rain loads contribute to the design specifications of a structure through weight of water and drainage of water from the structure's roof. Rain loads applied to building and plumbing are interconnected, as the size of the drainage system determines how fast water can drain from a roof, reducing the potential for structural failures. But also, structural considerations for rain loads extend to the combination of loads that must be computed by adding rain load to other loads of the structure.

FBC – Plumbing

Chapter 11, Storm Drainage

Figure 1106.1

Current code: Roofs shall be designed for the maximum possible depth of water that will pond. The published roof drain flow rate, based on the head of water above the roof drain, shall be used to size the storm drainage system in accordance with Section 1106. The maximum possible depth of water includes the height of the water required above the inlet of the secondary roof drainage to achieve the required flow rate of the secondary drainage to accommodate the design rainfall rate, and assuming all primary roof drainage is blocked (FBC 2017). Fundamentally, the code implies use of a flow rate for sizing the storm drainage piping that is based on the maximum anticipated ponding at the roof drain (Section 1105.2, FBC 2017).

The size of the vertical conductors and leaders, building *storm drains*, building storm sewers and any horizontal branches of such drains or sewers shall be based on the 100-year hourly rainfall rate indicated in Figure 1106.1 or on other rainfall rates determined from *approved* local weather data (FBC 2017).

The 100-yr, hourly rainfall (i) and the roof area serviced by a single drainage system is used to determine flow rate for a single drainage system by Q = 0.0104Ai (ASCE 7-05). Static head (ds) is the depth of water on the undeflected roof up the inlet of the secondary drainage system when the primary drainage is blocked, provided Q and Table 1106.2. Hydraulic head (dh) is the additional depth of water on the undeflected roof above the inlet of the secondary drainage system at is design flow, and can be determined from the minimum required flow for the secondary drain, referencing ASCE/SEI 7-16 (in Patterson and Mehta (2018)). Computing the total depth of water on the roof when the primary system is blocked ($d_s + d_h$) * 5.2 gives the design rain load in psf.

Results of data analyses: The updated 100-yr, hourly rainfall rate determined for the Miami-Dade County region was both higher and more spatially-variable than indicated in Figure 1106.1/1611.1 (see Figure 98 or Figure 7 in the main report) Further, a recent paper used historical data to found increased rainfall in most wet season months (Abiy et al., 2019), however, they did not analyze 100-yr return intervals for 15-minute events.

Additional literature research: In a paper presented to the 33rd RCI International Convention and Trade Show in 2018, Patterson and Mehta noted some limitations of using 100-yr, hourly rainfall. One, that 100-yr, hourly rainfall is often not a constant rainfall rate over the 60-minute period. Two, the secondary or overflow drainage system is intended as a safety provision against failures (e.g., roof collapse, pipe-fitting separation, pulled hanger from pre-stressed concrete floor/ceiling, flood of upper-balcony decks, fitting component failure, flooding in upper building floors due to pipe failure, Ballanco 2012) in the case that the primary drainage system is compromised. Patterson and Mehta (2018) noted that past codes had used higher rainfall rates for the secondary drainage system. In 1991, the SPC required overflow drainage to be designed to 100-yr, 15-minute rainfall rate. The first International Plumbing Code (IPC) published in 1995 divided in half the drainage capacity of the secondary system, effectively doubling the design rainfall rate for overflow drainage. Among Ballanco (2012) recommendations for code changes were new sizing requirements to be based on two rainfall rates: 100-yr, hourly rainfall and 10-yr, 5-minute rainfall rates, and applying the rate that accommodates the greatest amount of ponding expected. The National Standard Plumbing Code of the Plumbing- Heating-Cooling Contractors National Association continues to use 100-yr, 15-minute rainfall rate for the secondary drainage system. In fact, ASCE 7-16 apparently also recommends using 100-yr, 15-minute rainfall rates to accommodate those heavy, short duration storms.

FBC - Building

Chapter 16, Structural Design

Figure 1611.1

Current Code: Similarly, design rain loads (R) are determined for each portion of a roof to sustain the load of rainwater that will accumulate on it if the primary drainage system for that portion is blocked (static head = d_s) plus the uniform load caused by water that rises above the inlet of the secondary drainage system (hydraulic head = d_h) at its design flow ($R = 5.2\ (d_s + d_h)$). The design rainfall is based on the 100-year hourly rainfall rate indicated in Figure 1611.1 or on other rainfall rates determined from *approved* local weather data (FBC 2017).

Results of data analyses: As described above, the updated 100-yr, hourly rainfall rate determined for the Miami-Dade County region was both higher and more spatially-variable than indicated in Figure 1106.1/1611.1. See Figure 98. Further, a recent paper used historical data to found increased rainfall in most wet season months (Abiy et al 2019), however, they did not analyze 100-yr return intervals for 15-minute events.

Key Recommendation: Two recommendations are proposed related to Rain Loads for Storm Drainage in the Plumbing volume and Structural Design in the Building volume of the FBC.

1. Currently, the FBC allows Figure 1106.1/1611.1 to be used to determine 100-yr, hourly rainfall to determine flows and rain loads for structural and plumbing design. Updated data (provided in the main report) and guidance in relevant international and national codes suggest that the 100-yr, hourly rainfall maps for the State should be based on updated data. Further, 100-yr, 15-minute rainfall rate data should also be reviewed, and updated as needed, to facilitate consideration of new code language that the higher of the 100-yr, hourly rainfall rate or 100-yr, 15-minute rainfall rate be applied for the secondary drainage system.
2. Large roof areas may result in the exceedance of the flow capacities provided in Tables 1106.2 and 1106.3.

Flood Loads

Objective 3.2: Evaluate how the groundwater table maps and the revised rainfall maps should be used to update the **Flood Loads** as applied to Structural Design (Chapter 16, including Table 1612.1), Flood Resistant Construction (Chapter 3, Section R322, Residential) and the structures seaward of the coastal construction line (Chapter 31, Section 3109, Building) of the FBC. In the list of codes identified, this objective also included review of Chapter 18, Soils and Foundations, of the Building volume and Chapter 11, Storm Drainage, of the Plumbing volume.

Context of Evaluation: Loads are **"**forces or other actions that result from the weight of building materials, occupants and their possessions, environmental effects, differential movement and restrained dimensional changes. Permanent loads are those loads in which variations over time are rare or of small magnitude, such as dead loads. All other loads are variable loads" (FBC 2017). Buildings are other structures and portions thereof shall be designed to resist Load Combinations (dead, earthquake, fluid, flood, lateral earth pressure, roof and floor live, rain, snow, self-straining, wind speed and pressure loads, Section 1605). Foundation walls and retaining walls shall be designed to resist lateral soil loads (Section 1610). Flood loads apply to buildings and other structures located in areas prone to flooding, as defined on a flood hazard map (Section 1612; ASCE 7-05, Chapter 5). Flood loads for structural systems of buildings or other structures are designed, constructed, connected, and anchored to resist floatation, collapse, and permanent lateral displacement due to action of loads due to flooding associated with design flood and other loads in accordance with load combinations (ASCE 7-05, Chapter 5). Design and construction of structures seaward of a coastal construction control line (CCCL) or seaward of the 50-foot setback line, Flood resistant construction and Storm Drainage for plumbing are also covered. The FBC residential, adopts with amendments, the International Residential Code (2015), with provisions for flood-resistant construction.

FBC - Building

Chapter 16, Structural Design

Section 1605 - Load Combinations

Section 1610 - Soil Lateral Loads

Section 1612 - Flood Loads of Building

Current Code: The flood hazard area is the area subject to flooding during the design flood. The design flood is the greater of the following 2 events: 1) the Base Flood, affecting those areas on the community's Flood Insurance Rate Map, or 2) the flood corresponding to the area designated as a Flood Hazard Area on a community's Flood Hazard Map or otherwise legally designated. The Coastal High Hazard Area (V-Zone) and Coastal A-Zone are areas within a Special Flood Hazard Area (SFHA, land in a floodplain subject to a 1% or greater chance of flooding in a given year). The V-Zone, extends from offshore to the inland limit of a primary frontal due along an open coast, and any other area subject to high-velocity wave action from storms or seismic action. The coastal A-zone is landward of a V-Zone or landward of an open coast without mapped V-Zones. The principal source of flooding must be astronomical tides, storm surges, not riverine flooding, and potential for breaking waves greater than or equal to 1.5ft during the base flood (Chapter 5, ASCE 7-05).

Design and construction of structures located in flood hazard areas shall consider all flood-related loads and conditions, including the following: hydrostatic loads, hydrodynamic loads, wave action; debris impact; rapid rise and rapid drawdown of floodwaters; prolonged inundation; alluvial fan flooding; wave-induced and flood-related erosion and local scour; deposition of sediments; ice flows and ice jams; and mudslides in accordance with requirements of this standard if specified, or if not specified in this standard then in accordance with requirements approved by the authority having jurisdiction. Design considerations shall be documented and shall take into account the applicable flood-related loads and conditions, and load combinations that will act on the foundation and the structure (Chapter 1, ASCE 24).

Where flood loads, F_a, are to be considered in the design, the load combinations of Section 2.3.3 of ASCE 7 shall be used (FBC 2017). When a structure is located in a flood zone (e.g., Flood Hazard Area, Section 5.3.1, Chapter 5, ASCE 7-05), the following load combinations shall be used, applying load combinations 4 and 6 for strength design (Section 2.3.3, Chapter 2, ASCE 7), below. For allowable stress design, see Section 2.4.2., Chapter 2, ASCE 7-05:

4) 1.2D + 1.6W + L + 0.5 (Lr or S or R)

6) 0.9D + 1.6W + 1.6H

1. In V-Zones or Coastal A-Zones, 1.6W in combinations (4) and (6) shall be replaced by 1.6W + 2.0Fa.
2. In noncoastal A-Zones, 1.6W in combinations (4) and (6) shall be replaced by 0.8W + 1.0Fa.

The nominal flood load, Fa, is based on the 100-year flood (Chapter 5, ASCE 7-05), although design flood elevation should be used if flooding in the area designated as a flood hazard area on a community's flood hazard map or otherwise legally designated area is greater (Chapter 1, ASCE 24). The recommended flood load factor of 2.0 in V Zones and Coastal A-Zones is based on a statistical analysis of flood loads associated with hydrostatic pressures, pressures due to steady overland flow, and hydrodynamic pressures due to waves, as specified in Section 5.3.3 (Chapter 5, ASCE 7-05). The flood load criteria were derived from an analysis of hurricane-generated storm tides produced along the United States East and Gulf coasts, where storm tide is defined as the water level above mean sea level resulting from wind-generated storm surge added to randomly phased astronomical tides (Mehta et al. 1998 in C2.3.3, ASCE 7). Also, D = dead load, or the actual weights of materials of construction and fixed service equipment; L = live loads are roof (>20 psf) and floor live loads uniformly distributed (psf), or concentrated (lbs.) based on occupancy or use; R = rain load and W = load due to wind pressure.

In the design of structures below grade, provision shall be made for the lateral pressure of adjacent soil. This is determined by geotechnical investigation, or if not given, soil loads specified in Table 1610.1 (Chapter 16, FBC). In ASCE 24 (Section 1.5.3.), it is stated that foundations of structures shall be designed and constructed to provide the required support to prevent flotation, collapse, or permanent lateral movement under the load combinations specified in Section 1.6.2 during design flood conditions in flood hazard areas. Any part of the foundation that is below the minimum elevations specified by Table 2-1 (SFHA, non-coastal) or Table 4-1 (SFHA, coastal), as applicable, and that provides structural support shall meet applicable foundation requirements in this standard. (Section 1.5.3, Chapter 1, ASCE 24). FBC cites “below grade” whereas ASCE 24 references “below minimum elevations”. In doing so, ASCE 24 implicitly includes reference to free surface water, which kicks in an additional provision put forth in ASCE 7-05 when computing loads during flood “for surfaces exposed to free water, the design depth shall be increased by 1 ft (0.30 m)”.

FEMA provides significant technical guidance on determining hydrostatic and hydrodynamic loads for residential and non-residential buildings in flood hazard areas that are not currently referenced in the FBC with reference to Section 1610, Soil Lateral Loads, but are referenced in ASCE 24. For new construction (U.S. Army Corps of Engineers publication, Flood Proofing Regulations (USACE 1995) and two publications, FEMA P-936, Floodproofing Non- Residential Buildings (FEMA 2013a) and FEMA P-259, Engineering Principles and Practices for Retrofitting Flood Prone Residential Buildings (FEMA 2012a) are referenced. For existing, residential structures FEMA P-312, Homeowner’s Guide to Retrofitting: Six Ways to Protect Your House from Flooding (FEMA 2009a) are referenced. Documents FEMA P-936, P-259 and P-312 provide standard calculations for hydrostatic and hydrodynamic loads not provided in ASCE 7-05. Further, these documents provide technical guidance with reference to Section 1612, Flood Loads. While documentation therein with reference to Section 1605, Load Combinations, are provided, Section 1612 is the most direct guidance provided in FBC, Chapter 16 of the Building volume. This section directs the code provisions to Chapter 5 of ASCE 7 and ASCE 24, as indicated above. The cross-references to flood-resistant provisions of the Florida Building Code as provided in Table 1612.1 are useful, but those sections only sometimes reference back to ASCE 7 and ASCE 24.

With regard to impacts of sea-level rise, ASCE 24 cautions the designer that the minimum elevation requirements provided by FEMA as the BFE do not provide for uncertainties in flood frequency nor take into account changes in flooding because of watershed development, sea-level rise, or changes in precipitation patterns. ASCE 24 provides for minimum elevation requirements with a factor of safety above the BFE, dependent on the critical or essential nature of the structure occupancy and use.

Results of data analyses: Updated data associated with this objective included wet season groundwater table maps and depth to water maps under low and high scenarios of sea-level rise. In the proposed work, we were not tasked with reviewing code with respect to storm surge per se. However, changes in sea level will also have an influence on the design flood elevation in the SFHA, particularly in the V-zone and Coastal A-Zone. In the 2009 M-D county FIS, it was reported that flood elevations for multiple canal basins and areas were determined using XP-SWMM (Miami-Dade County DERM, August 2003-March 2007) and that a hypothetical tidal wide of 2-ft height was used as the downstream boundary condition (FEMA 2009). We do not know to what extent groundwater elevation was considered in these runs to determine BFE.

New groundwater table maps and rainfall data will be useful in updated determinations of BFE and DFE (see sections I and II of the main report). Depth to groundwater table maps provide the building code officer with a quick reference to evaluate whether a geotechnical investigation should accompany the building permit. See Main modeling results section and Section I of the main report for further recommendations referencing updated data on depth to groundwater.

Additional literature research: In ASCE 24, it is noted the Design Flood Elevation (DFE), the higher of the BFE on FIRMs or flood elevation shown on a community's map, DFE often = BFE. However, communities may elect to adopt flood elevations that are higher than those determined by FEMA, for example, to *incorporate recent change* or show future conditions (assuming predicted upland development, subsidence, or sea-level rise), to reflect the flood or record or other flood events that exceeded the 1% annual chance flood, or to incorporate freeboard as an additional safety factor to reflect local conditions. The DFE also depends on the Flood Design Class, a classification of buildings and other structures for determination of flood loads and conditions, and determination of minimum elevation requirements on the basis of risk associated with unacceptable performance (ASCE 24). The FBC recently adopted DFE = BFE + 1ft, as recommended in ASCE 24 that applies to residential structures (R322.3.2).

Key Recommendations: Six areas of recommendations are proposed related to Flood Loads for Structural Design in the Building volume of the FBC.

1. Currently, the code does not take into consideration changing sea level as part of the flood load calculations. This is recommended for 3 primary reasons: 1) sea level has increased 4-5 inches since 1992 in south Florida (see Future ocean boundary condition section), Southeast Florida has experienced an uptick in the rate of rise in the 2006-2016 period of record (Wdowinski et al. 2017), and a recent paper suggests we are on a high scenario projection for future changes in sea level (PNAS paper 2019); 2) changes in sea level influence the surface exposed to free water, increasing both the hydrostatic and hydrodynamic load in flood hazard areas; ASCE 7-05 provides that BFE + 1ft should be mandated where water level exceeds the ground surface as "free water". This is the common result of regular, extreme flooding due to astronomical tides.; 3) the lifetime for residential and non-residential structures can exceed 50 years, by which time an approximate 2 ft increase in sea level may occur; and 4) the cumulative influence of astronomical tides, uncertainties in flood frequency analyses and hydraulic modeling, changes between wet season and dry season groundwater tables, and changes in flooding from watershed development, sea-level rise and changing precipitation patterns are not accounted in BFE.

Thus, at a minimum, Flood Design Classes should be applied for structures that meet criteria 3 and 4 (Chapters 2 – 4, ASCE 24), and BFE = 1ft be designated for non-residential structures. However, a more robust, scientifically-backed methodology for computing elevation requirements that take into account these cumulative uncertainties and sources of flooding is warranted. Further, to ensure the most up-to-date sea-level rise projections are being taken into consideration for the design of flood elevations, it is recommended that there be a harmonized procedure for developing a unified projection for each region of the State, that is updated every 5 years and mandated for use in the FBC. Finally, coastal A-zones can be considered for use to determine and accommodate the increasing influence of astronomical tides (inland extent of tide) and additional area of flood inundation extent with surge + tide. In absence of other data, the limit of moderate wave action (LiMWA) could be used to determine the limit of influence of astronomical tide, and evaluated for what conditions it is appropriate to apply (applying, for instance, a table like 1610.1).

2. Currently, load combinations apply Fa use hydrostatic and hydrodynamic calculations provided in Chapter 5, ASCE 7-05, and load combinations including flood load referencing analyses and publications from the 90s and earlier. Advancements in experimental facilities and modeling warrant review, and possible update, of load combinations that include flood and the recommended flood load factor applied in V- and coastal-A zones (see p.256, C2.3.3. for a discussion of determination of flood load criteria). In particular, recurring tidal flooding meets the conditions set forth for higher flood load factors (e.g., Mehta et al. 1998). Because in these situations, flood load is generally small, a flood load factor of 2.0 is deemed sufficient based on the fact that the most important structural design conditions are for floods of greater stillwater depths (flood level above ground) than 4ft.
3. ASCE 24 provides a more in depth discussion of flood resistant standards in most areas of the FBC, including with regard to soil lateral loads and building to flood elevations (DFE). It is recommended that: 1) Section 1605 and 1610.1 reference ASCE 24, including Chapter C6, when building in flood hazard areas, including reference to Flood Design Class in Section 1604.5. 2) A footnote be added to Table 1610.1 referencing ASCE 24 and substantial improvement/damage provisions, so that foundation walls are designed to support "the weight of the full hydrostatic pressure of undrained backfill, unless a drainage system is installed in accordance with Sections 1805.4.2 and 1805.4.3" in a flood hazard area (Section 1610.1). It is recommended that the FBC provide the standardized approaches or make reference to the standard approaches it recommends for groundwater control.
4. With reference to Section 1612 and Table 1612.1, it is recommended that language be modified in the code in Section 1612.4 with text in red font as "The design and construction of buildings and structures located in flood hazard areas, including coastal high hazard areas and Coastal A Zones, and those flood-resistant provisions of the FBC cross-referenced in Table 1612.1, shall be in accordance with Chapter 5 of ASCE 7 and with ASCE 24.
5. Currently, many structures within the flood hazard area are not up to flood code provisions because they are pre-FIRM buildings and/or have been grandfathered. It is recommended that the FBC consider a study on the number, type, location and flood risk of pre-FIRM buildings to determine specific, standardized guidance to inform the basis for any potential code changes consistent with the intent of the FBC and helps transition both residential and non-residential pre-FIRM structures to "minimum requirements for reasonable safety, public health and general welfare" as provided in Section 101.3.

6. Table 1612.1 cross-references flood-resistant provisions in the Florida Building Code. Section 107.2.5 and 107.3.5 (Submitted documents for site plan and minimum plan review criteria for buildings) should reference ASCE 24, section 1.5 for flood hazard areas. Section 117.1 provides that the variance procedures adopted in the local floodplain management ordinance shall apply to requests to the building official for variances. A clause could be added, such as: including but not limited to the floodplain manager on staff, to ensure that someone certified in floodplain management provides input on the variance requested. Chapter 2, Section 202, should include definitions for "return period" and "combined total storm tide elevation". Chapter 4 provides for the Flood Design Class Criteria specific in Table 2-1 and 4-1 of ASCE 24. It is recommended that a study be conducted on the cost-benefit of reducing the substantial improvement/damage percentage for Flood Design Class 4 buildings and structures.
7. As referenced in Section 453.2, public schools and Florida colleges are exempt from local requirements, as provided in Section 1013.371(1)(a), *Florida Statutes, with specific provisions for how construction documents are reviewed and inspected. Two recommendations are provided with regard to Section 453.2 given their Flood Design Class of 3: 1)* Add: Exception: Educational facilities in flood hazard areas must comply with must comply with this code or the floodplain management ordinance of the municipality having jurisdiction in accordance with 44 CFR Parts 59, 60, 65, and 70. 2) Add after "Section 1013.38, *Florida Statutes*.": Consistent with 105.14, permit issued on basis of a sworn affidavit shall not extend to flood load and flood resistance requirements of the Florida Building Code, as per **44 CFR** Parts 59, 60, 65, and 70.

FBC - Building

Chapter 18, Soil & Foundations

Section 1803 Geotechnical Investigations

section 1804 Excavation, Grading & Filling

Section 1805 Damp proofing & Waterproofing

Section 1806 Presumptive Load-Bearing Values of soils

Section 1807 Foundation Walls, Retaining Walls & embedded Posts & Poles

Section 1808 Foundations

Section 1809 Shallow Foundations

Section 1810 Deep Foundations

Current Code: The basis for soil and foundation code requirements is to design for allowable bearing pressures, and allowable stresses, and to determine the allowable stress design load combinations specified in Section 1605.3 and the quality and design of materials used structurally in excavations and foundations. Currently, the FBC Section 1804.5 allows fill in coastal high hazard areas and coastal A zones (contrary to ASCE 24, Section 4.5.4) "unless the fill is conducted and/or placed to avoid diversion of water and waves toward any building or structure". The following statement proceeds: "that cumulative effect of encroachment into a floodway, when combined with all other existing and anticipated flood hazard area encroachment, will not increase the design flood elevation more than 1 ft at any point".

Results of data analyses: Depth to groundwater table maps (see Main modeling results section and Section I in the main report) provide an estimation of variation (minimum dry season and maximum wet season) depth to groundwater. As noted, the maps provide the building code officer with a quick reference to evaluate whether a geotechnical investigation should accompany the building permit. In areas of shallow water table, like those near the coast, our analyses illustrate that the water table varies significantly depending on the time of year (e.g., dry season or wet season) and is projected to change from increasing sea level. Currently, the code provides no regulatory guidance on design and construction under these conditions, except with recommendations in ASCE 24 to provide additional safety with freeboard. In FBC Residential, FBC requires BFE + 1ft. Further, saltwater intrusion maps illustrate the 1996 saltwater-freshwater interface and the potential rate at which saltwater will intrude with changing sea-level rise. There are currently no provisions in the code for using salt-corrosion resistant materials for the design and construction of foundations, nor guidance on where these materials should be used.

Additional literature research: N/A

Key Recommendations: Five areas of recommendations are proposed related to Flood Loads for Soil and Foundations in the Building volume of the FBC.

1. Geotechnical investigations are the report of record for structures. The report elements need not be limited to those listed in 1803.6. It is recommended the following elements be listed among the report elements: 1) date of last geotechnical investigation, 2) if water table is not encountered, location of nearest well and water table depth at time of geotechnical investigation, to a cross-referenced benchmark, 3) whether the fill materials may be exposed to shrinking/swelling, and included in special design and construction provisions, 4) in foundation recommendations, type and design considerations for shrinking/swelling and salinity, and 5) document municipal regulations on setback and clearance and alternate design criteria recommendations.
2. 1804.5 is contradictory to ASCE 24, as written. Fill is prohibited in coastal high hazard areas and coastal A-zones.
3. Ground-water control (Section 1805.1.3) should be designed and constructed in accordance with shallow-water table conditions and where saltwater corrosion may occur. A standardized system

design for groundwater control outlined in the FBC is recommended, including for the purpose of subsoil drainage (1805.4).

4. Drainage discharge (Section 1805.4) Add to Exception: ", unless in a flood hazard zone – refers to "approved drainage system that complies with plumbing" – check that first
5. Shallow and Deep Foundations (Sections 1809 & 1810) – steel footings and soil conditions/changing water levels. Add; in flood hazard areas, comply with ASCE 24.

FBC - Building

Chapter 31, Special Construction
Section 3109 Structures seaward of a coastal construction control line

Current Code: The provisions of this section shall apply to the design and construction of habitable structures, and substantial improvement or repair of substantial damage of such structures, that are entirely seaward of, and portions of such structures that extend seaward of, the coastal construction control line (CCCL) or seaward of the 50-foot setback line, whichever is applicable. This section does not apply to structures that are not habitable structures, as defined in this section. Section 1612 shall apply to habitable structures and structures that are not habitable structures if located in whole or in part in special flood hazard areas established in Section 1612.3. It is specifically noted, If the modification or repair is determined to be substantial improvement or substantial damage, and if the building is located in a special flood hazard area (Zone A and Zone V) established in Section 1612.3, the requirements of Florida Building Code, Existing Building applicable to flood hazard areas shall apply.

Results of data analyses: New groundwater table maps with low and high scenarios for sea-level rise will be useful in updated determinations of BFE and DFE. <<summary of results here>> Depth to groundwater table maps will be useful to determine whether an existing foundation may need modification to comply with substantial improvement/damage provisions of 1612.

Additional literature research: All of Miami-Dade County seaward of CCCL is in SFHA, generally, but map revisions may have removed some structures

(https://ca.dep.state.fl.us/mapdirect/?webmap=a8c9e92fbad5446d987a8dd4ee5dc5cc).

Key Recommendations: Given the projected influence of sea-level rise on combined storm tide elevations, the following recommendations are proposed related to Flood Loads for Special Construction in the Building volume of the FBC. 1) The combined total storm tide elevation (value) for the 100-yr return period identified by the FDEP should be evaluated against those using other, approved methods of determining that value. 2) The 500-yr combined total storm tide elevation should be evaluated for consideration and use for Flood Design Class 2 - 4 structures, or the DFE, whichever is greater. 3) Given the extensive and valuable development along the Florida coastline, we recommend a study that evaluates how increasing the inland extent of the CCCL, and extending the CCCL to V-zones not currently within the CCCL, would reduce building damage, with further consideration given to those proposals after the study is complete. 4) There may be some inconsistencies, for example, when the CCCL is also in a coastal high hazard area (e.g., 3109.3.5) – need to remind myself if structural slabs are permitted here.

FBC – Residential

Chapter 3

Section R322 - Flood Resistant Construction

Current Code: Buildings and structures constructed in whole or in part in flood hazard areas, including A or V Zones and Coastal A Zones, as established in Table R301.2(1), and substantial improvement and restoration of substantial damage of buildings and structures in flood hazard areas, shall be designed and constructed in accordance with the provisions contained in this section, and those located within flood hazard areas, be designed and constructed in accordance with ASCE 24. Table R301.2 provides climatic and geographic design criteria for floods as "The applicable governing body shall, by local floodplain management ordinance, specify (a) the date of the jurisdiction's entry into the National Flood Insurance Program (date of adoption of the first code or ordinance for management of flood hazard areas), (b) the date(s) of the Flood Insurance Study and (c) the panel numbers and dates of the currently effective FIRMs and FBFMs or other flood hazard map adopted by the authority having jurisdiction, as amended." In FBC Residential, FBC requires BFE + 1ft. While the safety factor provided helps to address the critical nature of residential structures, it does not take into account other sources of flooding nor uncertainty.

Results of data analyses: New groundwater table maps and rainfall data will be useful in updated determinations of BFE and DFE (See Main modeling results section and Section I of the main report)

Additional literature research: N/A

Key Recommendations: The new, relevant FEMA publications should be referenced throughout. The following specific code language updates are also recommended related to Flood Loads for Flood Resistant Construction in the Residential volume of the FBC.

1. R322.1.4.1 #2 – Add: as provided by the local floodplain management ordinance e.g., documentation of flood-resistant design and construction (Table R301.2)

2. R322.1.4.2 – Add: Exception: when the proposed buildings and structures are in a coastal high hazard area, then Chapter 4, ASCE 24 should be followed. NOTE: R322.1.4.2 as written contradicts Chapter 4, ASCE 24, which states that use of fill for structural support should be prohibited in coastal high hazard areas and coastal A-zones. Importantly, riverine flood hazard areas can occur in coastal high hazard areas and coastal A-zones. **NOTE:** It is presented correctly in R322.3.2 Elevation requirements (for coastal high hazard areas and Coastal A-zones).

3. R322.1.6 Protection of mechanical, plumbing and electrical systems – requires systems to be elevated to BFE + 1ft or DFE, whichever is higher, for substantial improvements.

4. R322.1.7 – new and replacement water supply and sanitary sewage systems must be designed to "minimize or eliminate infiltration of floodwaters into systems and discharges from systems into floodwaters accordance with Chapter 64E-6 onsite sewage and treatment and disposal systems". Although the FBC refers to FDEP jurisdiction with Chapter 64E-6, FBC should mandate use of depth to groundwater maps to specify where installation of septic tanks should be prohibited, to comply with Section 101.3. where FBC provides for "minimum requirements for reasonable safety, public health and general welfare". This should be coordinated with FDEP and/or Bureau of Health.

5. R322.1.8 – A FEMA technical publication (TB-2) on flood-resistant materials is specifically referenced. The new, relevant FEMA publications should be referenced throughout.

6. R322.2.1 Pull out Coastal A zone from bullet since it is directing to R322.3 The existing bullet 3 is not clear.

7. R322.3.2 – Add to bullet 1: To account for SLR and recurring influence of astronomical tide (free water on surfaces), … is elevated to or above the base flood elevation plus 2 feet (610 mm), or the design flood elevation, whichever is higher.

8. R322.3.1/R322.3.2 – There are precedents for: 1) using "landward of the reach of mean high tide" to locate new buildings and buildings that are substantially improved and 2) using the Flood Design Class (2) to set the DFE in coastal high hazard areas and coastal A zones following Table 4.1 of ASCE 24. It is recommended that precedent 1 be used to develop code with regard to SLR. It is recommended that precedent 2 be used to extend setting DFE using Table 4.1 for higher risk Flood Design Class 3, and to other facilities in Class 4.

9. R322.3.3 Foundations – There is an exception that allows stem wall foundations be backfilled to the underside of the flood system provided the foundations are designed to account for wave action, etc. Under SLR and storm surge in V zone, coastal A zones seem like the worst place to allow fill because the flood heights can be high on both sides of the coastal A zone. Recommend that the exception to the exception also include that it only be allowed under conditions where it is demonstrated that combined inland flooding and tidal flooding will not increase flood levels above the DFE in the coastal A zone.

FBC – Plumbing

Chapter 11, Storm Drainage

Section 1101 General

Section 1102 Materials

Section 1103 Traps

Section 1105 Roof Drains

Section 1106 Size of Conductors, Leaders and Storm Drains

Section 1107 Siphonic Roof Drainage Systems

Section 1108 Secondary (Emergency) Roof Drains

Section 1109 Combined Sanitary and Storm Public Sewer

Section 1110 Controlled Flow Roof Drain Systems

Section 1111 Subsoil Drains

Section 1112 Building Subdrains

Section 1113 Sumps and Pumping Systems

Appendix B

Current Code: These provisions shall govern the materials, design, construction and installation of storm drainage. Subsoil drainage in Chapter 18 refers to Chapter 11, but limited guidance provided. There is no guidance on building in coastal flood hazard areas where saltwater corrosion and changing water levels can affect subsoil drain pipe, materials used for building storm sewer pipe, and fittings. Section 1106 provides that the size of conductors, leaders and storm drains shall be based on the 100-yr hourly rainfall or other rates determine from approved local weather data. Section 1111 provides also for subsoil drains with no reference to saltwater corrosion or changing water levels. Building subdrains below the public sewer shall discharge into a sump then automatically lifted as required for sumps and comply with Section 1113.

Results of data analyses: Primary aspects that were considered were related to sea-level rise. Similarly, with Chapter 18, Soil and Foundations in the Building Volume, Chapter 11, Plumbing Volume, considerations were primarily with corrosion due to exposure to saltwater and changing water levels.

Additional literature research: N/A

Key Recommendations: Use saltwater-freshwater interface or LimWA to determine the inland extent of saltwater corrosion, in the absence of other data. Recommend that the size of conductors, leaders and storm drains shall be based on the 100-yr hourly rainfall, 100-yr 15-min rainfall or other rates determine from approved local weather data applying whichever is highest.

Summary of Key Recommendations

Objective 3.3: Provide specific recommendations for Code modifications to incorporate the updated information on groundwater elevation due to sea level rise and rainfall.

Rain Loads

1. Recompute the flow capacities provided in Tables 1106.2 and 1106.3 with large roof areas using the new rain load data.
2. Add language pertaining to design of secondary drainage system e.g., higher of the 100-yr, hourly rainfall rate, 100-yr, 15-minute rainfall rate or local approved weather data, be applied for the secondary drainage system.

Flood Loads

1. It is recommended that the V-zone and coastal A-zones be used to delimit the areas where code should regulate the use of saltwater corrosion-resistant materials, following ASCE 24.
2. It is recommended that the LiMWA of coastal A-zones be used to delimit the inland extent of the influence of astronomical tide on free surface, tidal flooding by adding 1ft to the AE BFE as a safety factor, in the absence of other approved data, if it has been over 30 years since the last FIRM was updated and approved. This is to accommodate the analytical uncertainties and multiple sources of flooding not accounted for in the FEMA FIRM, notably in the coastal A-zone.
3. Currently, the FBC Section 1804.5 allows fill in coastal high hazard areas and coastal A zones "unless the fill is conducted and/or placed to avoid diversion of water and waves toward any building or structure". R322.3.3 allows stem wall foundations be backfilled to the underside of the flood system provided the foundations are designed to account for wave action, etc. However, the following statement proceeds: "that cumulative effect of encroachment into a floodway, when combined with all other existing and anticipated flood hazard area encroachment, will not increase the design flood elevation more than 1 ft at any point". The FBC should recommend tools for computing the cumulative flood hazard area encroachment using different storm tide elevations as the coastal boundary condition. New research may be needed.
4. It is recommended that the FBC provide the standardized approaches or make reference to the standard approaches it recommends for use for groundwater control (Section 1804.5).

5. At a minimum, Flood Design Classes should be applied for structures that meet criteria 3 and 4 (Chapters 2 – 4, ASCE 24), following, among other rationale provided, precedents set in R322.3.1/R322.3.2. Flood Design Class 2 should also be applied to non-residential structures given the significant economic hardship that could be caused by flood damage.
6. To ensure the most up-to-date sea-level rise projections are being taken into consideration for the design of flood elevations, it is recommended that there be a harmonized procedure for developing a unified projection for each region of the State, that is updated every 5 years and mandated for use in the FBC.
7. Mandate use of depth to groundwater maps, updated every 5 years, to specify where installation of septic tanks should be prohibited (*cf.* R322.1.7), to comply with Section 101.3. where FBC provides for "minimum requirements for reasonable safety, public health and general welfare". Coordinate with FDEP.
8. Additional recommendations for specific text edits are provided in the table below:

FBC Section	**Specific text edit (in red font)**	**Report Section**
1612.4	"The design and construction of buildings and structures located in flood hazard areas, including coastal high hazard areas and Coastal A Zones, and those flood-resistant provisions of the FBC cross-referenced in Table 1612.1, shall be in accordance with Chapter 5 of ASCE 7 and with ASCE 24".	FBC – Building, Chapter 16, Structural Design
1605	reference "ASCE 24, including Chapter C6, when building in flood hazard areas,"	FBC – Building, Chapter 16, Structural Design
1610.1	reference "ASCE 24, including Chapter C6, when building in flood hazard areas,"	FBC – Building, Chapter 16, Structural Design
1604.5. 2	reference to "Flood Design Class"	FBC – Building, Chapter 16, Structural Design
1610.1	Add a footnote to Table 1610.1 referencing ASCE 24 and substantial improvement/damage provisions in flood hazard areas, so that foundation walls are designed to support "the weight of the full hydrostatic pressure of undrained backfill, unless a drainage system is installed in accordance with Sections 1805.4.2 and 1805.4.3"	FBC – Building, Chapter 16, Structural Design
107.2.5 in Table 1612.1	reference "ASCE 24, section 1.5 for flood hazard areas"	
107.3.5 in Table 1612.1	reference "ASCE 24, section 1.5 for flood hazard areas"	FBC – Building, Chapter 16, Structural Design
202 in Table 1612.1	Include definitions for "return period" and "combined total storm tide elevation"	FBC – Building, Chapter 16, Structural Design
453.2 in Table 1612.1	Add: Exception: Educational facilities in flood hazard areas must comply with must comply with this code or the floodplain	FBC – Building, Chapter 16, Structural Design

	management ordinance of the municipality having jurisdiction in accordance with 44 CFR Parts 59, 60, 65, and 70.	
453.2 in Table 1612.1	Add after "Section 1013.38, *Florida Statutes*.": Consistent with 105.14, permit issued on basis of a sworn affidavit shall not extend to flood load and flood resistance requirements of the Florida Building Code, as per 44 CFR Parts 59, 60, 65, and 70.	FBC – Building, Chapter 16, Structural Design
1803.6	Add to list of elements: 1) date of last geotechnical investigation, 2) if water table is not encountered, location of nearest well and water table depth at time of geotechnical investigation, to a cross-referenced benchmark, 3) whether the fill materials may be exposed to shrinking/swelling, and included in special design and construction provisions, 4) in foundation recommendations, type and design considerations for shrinking/swelling and salinity, and 5) document municipal regulations on setback and clearance and alternate design criteria recommendations.	FBC - Building Chapter 18, Soil & Foundations
1805.4	Add to Exception: , unless in a flood hazard zone, then comply with ASCE 24	FBC - Building Chapter 18, Soil & Foundations
1809 & 1810	Add: in flood hazard areas, comply with ASCE 24	FBC - Building Chapter 18, Soil & Foundations
R322.1.4.1	Add: as provided by the local floodplain management ordinance e.g., documentation of flood-resistant design and construction (Table R301.2)	FBC – Residential Chapter 3
R322.1.4.2	Add: Exception: when the proposed buildings and structures are in a coastal high hazard area, then Chapter 4, ASCE 24 should be followed.	FBC – Residential Chapter 3
R322	The new, relevant FEMA publications on flood-resistant materials should be referenced throughout, like in R322.1.8 where FEMA technical publication TB-2 is referenced.	FBC – Residential Chapter 3
R322.2.1	Delete Coastal A zone from bullet since it is directing to R322.3 The existing bullet 3 is not clear.	FBC – Residential Chapter 3
R322.3.2	Add to bullet 1: To account for SLR and recurring influence of astronomical tide (free water on surfaces), … is elevated to or above the base flood elevation plus 2 feet (610 mm), or the design flood elevation, whichever is higher.	FBC – Residential Chapter 3

Summary of Priority Research Areas

Rain Loads

1. Determine the rainfall rate maps for different return intervals, at least 15-min, 100-yr, and compare with 1-hr, 100-yr for the State, for both historical and recent.

Flood Loads

1. Determine and apply a method to provide a scientific-basis for design flood elevations, based on uncertainties in flood frequency analyses, hydraulic modeling, increasing sea level, expected watershed development, changing rainfall patterns, and sources of flooding unaccounted for by FEMA BFE (e.g., sea level rise).
2. Evaluate whether and under what conditions the coastal A-zone designation of flooding due to astronomical tide and subsurface soil salinity, and whether LiMWA is a suitable proxy for the inland extent of tidal flooding and saltwater intrusion.
3. Advancements in experimental facilities and modeling warrant review, and possible update, of load combinations that include flood and the recommended flood load factor applied in V- and coastal-A zones (see p.256, C2.3.3. for a discussion of determination of flood load criteria).
4. New research may be needed to compute and evaluate the cumulative flood hazard area encroachment using different storm tide elevations as the coastal boundary condition (*cf.* 1804.5).
5. It is recommended that a study be conducted on the cost-benefit of reducing the substantial improvement/damage percentage criteria (<50%) for Flood Design Class 4 buildings and structures.
6. For the combined total storm tide elevation value, we do not know to what extent the uncertainties in analyses and modeling and sources of flooding are determined (*cf*. Section 3109). It is recommended that a study be conducted to evaluate: a) how the combined total storm tide elevation for the 100-yr return period be evaluated against those using other, approved methods of determining that value, and b) the 500-yr combined total storm tide elevation for consideration and use for Flood Design Class 2 - 4 structures (compared with BFE, DFE and cost-benefit). We also recommend an assessment of how increasing the inland extent of the CCCL to include V-zones reduces potential structural damage. Based on the results of these studies, further code changes may be warranted.

References

Abiy, A.Z., Melesse, A.M., Abtew, W., Whitman, D. 2019. Rainfall trend and variability in Southeast Florida: Implications for freshwater availability in the Everglades. PLoSONE 14(2):e0212008.https://doi.org/10.1371/journal.pone.0212008

Mehta, K.C., et al. 1998. "An investigation of load factors for flood and combined wind and flood." Report prepared for Federal Emergency Management Agency, Washington, D.C.

FEMA (Federal Emergency Management Agency). 2009. Flood Insurance Study, Miami-Dade County, Florida, and incorporated areas. Flood Insurance Study Number 12086CV000A. Federal Emergency Management Agency.

Appendix A. Relevant Sections of the Code

(a) Chapter 11, Storm Drainage, also Appendix B of FBC-Plumbing.
(b) Chapter 16, Structural Design, of the 6th Edition (2017) Florida Building Code (FBC), Building; Sections 1605 Load Combinations, 1610 Soil Lateral Loads
(c) Section 1611, Rain Loads (Figure 1611.1), of the FBC, Plumbing;
(d) Section 1612, Flood Loads, of the FBC, Building;
(e) Chapter 18, Soil & Foundations, sections 1803 Geotechnical Investigations, 1804 Excavation, Grading & Filling, 1805 Damp proofing & Waterproofing, 1806 Presumptive Load-Bearing Values of soils, 1807 Foundation Walls, Retaining Walls & embedded Posts & Poles, 1808 Foundations, 1809 Shallow Foundations, 1810 Deep Foundations
(f) Chapter 3, Section R322 Flood Resistant Construction, of the FBC, Residential;
(g) Chapter 31, Section 3109 Structures seaward of a coastal construction control line, of the FBC, Building; and
(h) Any other Chapters of the Florida Building Code that may be affected by sea-level rise and changes to extreme rainfall.