CORRECTIONS TO RADAR-ESTIMATED PRECIPITATION USING OBSERVED RAIN GAUGE DATA

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

by

Eric Chay Ware

August 2005

ABSTRACT

A new method is presented for calculating daily rainfall amounts from radar. Radar data from two River Forecast Centers (RFC), and daily rain gauge data from stations around the Northeast U.S. are used to create a radar-level resolution grid of rainfall. The purpose for this method is to produce fields of precipitation estimates in the operational area of the Northeast Regional Climate Center (NRCC), to archive the high-resolution precipitation product, and to use the product as input into a crop modeling program. Considering rain gauge observations as the true values, radar errors are calculated at each rain gauge location every day. Using an interpolation method, the errors are estimated at each radar pixel and added back to the radar grid. Thirty cases were selected from different times of year and different weather types. Three interpolation methods, Inverse Distance Weighting, Multiquadric Interpolation, and Ordinary Kriging, are compared to the Multisensor Precipitation Estimation (MPE), used operationally by River Forecast Centers. Parameters associated with each interpolation method are adjusted daily using cross-validation to produce the best results for each case. By using daily rain gauge data, all three interpolation methods perform similarly and better than MPE, which uses hourly rain gauge data. All methods for estimating precipitation perform best at low values of precipitation and worst at high values of precipitation. Because of the similarity in results between interpolation methods, the simplest method computationally, Inverse Distance Weighting, has been chosen to be used operationally for the Northeast Regional Climate Center.

BIOGRAPHICAL SKETCH

Eric was born on December 8, 1980 in Alva, Oklahoma, where he later graduated with very highest honors from Alva High School in 1999. After high school, he attended the University of Oklahoma for four years where as a member of the Pride of Oklahoma Marching Band, he watched the Sooners win their 7th National Championship in football in 2000. He graduated magna cum laude with a degree in Meteorology and a minor in Mathematics in 2003.

Eric came to Cornell in 2003 to study Statistical Meteorology. After 2 years at Cornell, he realized that his true passion was teaching. After completing his degree at Cornell, Eric will be following that path by returning to the University of Oklahoma to begin work on a doctorate in Mathematics.

ACKNOWLEDGEMENTS

I would like to thank my advisor and committee chairman, Dr. Dan Wilks, for his guidance and instruction during the past two years. I would also like to thank Dr. Art Degaetano, Brian Belcher, and the Northeast Regional Climate Center for computer programming help, rain gauge data, and the opportunity to archive the results of this project. Thank you to my minor committee member, Dr. Pat Sullivan, for help with SPlus and expertise on this project.

A special thanks to D. J. Seo for his advice during this process. And thank you to the Mid-Atlantic and Northeast River Forecast Centers, especially Patricia Wnek, Joseph Ostrowski, Jeff Ouellet, and Robert Shedd, for their assistance in acquiring radar data.

This work was supported through the Cornell Initiative on Computational Agriculture, funded by a Special Grant of the USDA-CSREES.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Rainfall estimation has been an important part of radar research since radars were first implemented. Based on the Z-R relationship (1), rainfall rate can be estimated from radar reflectivity.

$$Z = AR^b , \tag{1}$$

where Z is the radar reflectivity, R is the rainfall rate, and A and b are constants (Wilson and Brandes 1979). Early research showed that while average optimal constants in (1) could be found, the more appropriate values for individual cases would change depending on location and weather type (Wilson and Brandes 1979, Austin 1987). More specifically, these variations in the test parameters are caused by microphysical and kinematical processes that affect the drop-size distribution and fall speeds (Wilson and Brandes 1979; Hunter 1996). Even if a perfect Z-R relationship could be found for any spatial or time scale, other errors such as calibration, attenuation, bright bands, anomalous propagation, and range degradation exist that affect the radar estimates (Fulton et al. 1998, Hunter 1996). Numerous methods have been proposed to correct for one or many of these errors (Seo et al. 2000, Anagnostou and Krajewski 1999, Fulton et al. 1998, Hunter 1996). Anagnostou and Krajewski (1999) created an algorithm which makes corrections for many of these errors by defining adjustment parameters for each type of error individually.

Even before the Z-R optimal constants were determined, combining rain gauges with the radar reflectivity was found to improve accuracy in precipitation estimation from radar (Wilson 1970, Brandes 1975). Rain gauges are typically considered the "ground truth", but some small errors still exist. In high rainfall rate events, rain gauges may underestimate the rainfall because of high winds or tipping bucket losses (Crum and Alberty 1993, Groisman and Legates 1994, Hunter 1996).

Non-automated rain gauges will often be inconsistent with radar estimates because of sampling time differences, and human observation error.

The most recent use of rain gauges in radar precipitation estimation is to calculate a mean field bias. The mean field bias is calculated by dividing the gauge amount by the radar amount (Smith and Krajewski 1991, Seo et al. 1999). The mean field bias calculates a bias over any desired spatial domain or time scale, but does not describe biases that may occur at individual rain gauge stations. NWS applies the mean field bias to individual radar umbrellas every hour. Seo and Breidenbach (2002) grouped their rainfall estimates into a number of categories based on season, gauge rainfall amount, number of gauges used, and hourly versus daily estimates. They compared estimates using raw uncorrected radar estimates and mean field bias-corrected estimates. Cases were divided by season (cool or warm), time of observation (hourly or daily), number of gauges used (one-half (230) or one-quarter (115) of all operational gauges within the Arkansas Basin RFC selected randomly), and amount of rainfall at gauges (all gauges, gauges with rainfall greater than 2.54 cm, and gauges with rainfall greater than 5.08 cm). When using uncorrected radar estimates, the resulting root mean squared errors (RMSEs) were as much as 4.05 cm for cool season, daily estimate, one-half gauge network, and greater than 5.08 cm rainfall cases. Their methods improved the radar RMSE estimates to at least 0.333 cm for cool season, hourly estimate, one-half gauge network, and greater than zero rainfall cases. The mean field bias corrections reduced the MSE by 27% in the cool season and 26% in the warm season.

Corrections made to radar precipitation totals by the National Weather Service have been divided into four sequential stages. For each individual radar umbrella, the data pass through the Precipitation Processing Subsystem (PPS), also known as Stage 1, to create a number of derived products, including hourly precipitation estimates.

The PPS algorithm contains 46 parameters characterizing rain system type, season, climatology, topography, etc. (Wilson and Brandes 1979; Fulton et al. 1998). The PPS makes numerous corrections including beam blockage due to terrain, ground clutter due to man-made features such as tall buildings, and range degradation which underestimates rainfall at long distances (Fulton et al. 1998). Stage 2 is performed hourly at River Forecast Centers (RFCs), where rain gauges are used to calculate a mean field bias and other corrections for individual radar umbrellas (Seo 1998a, Seo 1998b, Fulton et al. 1998; Seo et al. 1999, Seo and Breidenbach 2002). Stage 3 is also performed at River Forecast Centers, and uses multiple radars to expand to the coverage area of the RFC (Fulton et al. 1998). Since some points lie in more than one radar umbrella, a method based on the height of the radar beam is used to select which radar rainfall estimate radar will be used at that point. The height of the radar beam at the lowest radar tilt is determined for the entire radar umbrella and the radar with the lowest height at that point is selected. Human interaction is used during this step to remove bad data or other errors that the previous steps may have overlooked (Fulton et al. 1998). Stage 4 is performed at the National Center for Environmental Prediction (NCEP) and uses the same method to expand the area to the entire United States.

Over the past few years, several new areas of research have developed in rainfall estimation from radar. One area focuses on experimentation with X-band polarimetric radars (Anagnostou et al. 2004, Brandes et al. 2002). Polarimetric radars collect data with vertical and horizontal polarizations simultaneously to account for the oblate shape of raindrops.

Another new area for real-time correction of radar precipitation estimates has been developed and implemented at River Forecast Centers and some National Weather Service offices. The procedure is called Multisensor Precipitation Estimators (MPE) and has refined and merged Stages 2 and 3 (Seo and Breidenbach 2002).

Radar values are mapped on the Hydrologic Rainfall Analysis Project (HRAP) grid, which is based on a polar stereographic projection, and results in a rotated grid conforming to the curvature of the earth. The resolution (~4km x 4km) of the HRAP grid therefore changes slightly with latitude. Like in Stage 3, the radar beam height is used to find the closest radar to each point in overlap regions. Satellite-based rainfall estimates are used in areas of poor radar coverage. The final grid covers each RFCs coverage area and is rectangular in shape rather than radial. Every hour automated rain gauges are used along with a mean field bias correction and optimization methods outlined in Seo (1998a) and Seo (1998b) are used to create a field of hourly estimated precipitation. The mean field bias (once part of Stage 2) corrects for any large scale biases over then entire RFCs area from calibration error for that hour. Unfortunately, the final product is not available online in real time.

The methods presented in this paper use radar data, after passing through the PPS (Stage 1), and rain gauge data, to interpolate a daily product of estimated rainfall. Section 2 describes the data and how it was acquired. The three interpolation methods used to correct the initial radar estimates are discussed in section 3. Section 4 compares results from the different interpolation methods using cross-validation and groups the test cases by time of download, season, weather type, and rain gauge amount to look for differences in the mean squared errors. Section 5 will describe the product more fully by discussing a few characteristic cases in more depth. Some of the useful applications for this product as well as some of the shortcomings are discussed in section 6. Conclusions from this research are presented in section 7.

2. Data Description

This project builds upon the work being done at the River Forecast Centers to develop a daily product of high-resolution precipitation fields for use by the Northeast

Regional Climate Center (NRCC). As part of a larger project with other departments at Cornell University, this product will ultimately be used for real-time agricultural and hydrological modeling as well as for climatological archives. A daily estimate of precipitation at the resolution of the HRAP grid (~4km x 4km), is calculated using daily cooperative observer network rain gauges (www.nws.noaa.gov/om/coop/) and hourly radar estimated precipitation (PPS/Stage 1) from the River Forecast Centers. This procedure differs from the MPE method in two ways. Rather than using hourly automated rain gauges and hourly radar data, this method uses daily rain gauge amounts, together with the summation of hourly radar data over 24 hours. Instead of taking the MPE approach of calculating a multiplicative mean field bias or local bias, an additive error between a rain gauge and its corresponding radar pixel is used. These error values are interpolated to each radar pixel, and added back to the original radar estimate.

The radar data were obtained from the Northeast River Forecast Center (NERFC) and the Mid-Atlantic River Forecast Center (MARFC). The radar data used have been mapped to an HRAP grid and some quality control has been performed by the River Forecast Centers. Archived raw uncorrected radar estimated rainfall (after Stage 1) and MPE results for these RFC areas can be found at http://dipper.nws.noaa.gov/hdsb/data/nexrad/nerfc_mpe.php and http://dipper.nws.noaa.gov/hdsb/data/nexrad/marfc_mpe.php.

There are a number of relevant products that are available on these websites. The "HEIGHT" product shows a field of heights above ground level at the lowest tilt of the radar. If a pixel is located within more than one radar umbrella, the lowest height is used to determine which radar estimate of rainfall will be used. The "INDEX" field then indicates which radar has been selected for that point. The

"RMOSAIC" product contains only the radar estimated rainfall (Stage 1) and is used as input for this project in order for a comparison to the MPE procedure to be made. The "GAGEONLY" product displays a field of the hourly, automated rain gauge amounts. The "BMOSAIC" product contains a field of estimated rainfall after a mean field bias correction is applied. "MMOSAIC" is considered the best estimate since a mean field bias correction and rain gauge amounts have both been used to make estimates. The results from the methods proposed in this paper will be compared to the "MMOSAIC" product. More extensive descriptions as well as other available product descriptions are available at:
http://www.crh.noaa.gov/ncrfc/html_frames/rfcwide.htm

Thirty cases were downloaded from all seasons and a variety of synoptic situations (Table 1). Most cases were obtained from RFC archived data, and a few cases were downloaded daily from the RFC in real time. For the archived data, RFCs tend to trim off areas that are far from their operational area, resulting in more missing pixels than the real-time data. The full radar domain for each RFC is outlined in black in Figure 1. The rain gauge data were obtained through the NRCC. Cooperative Observer Network rain gauge stations (dots in Figure 1) were used from all states in the NRCC operational area (states in Figure 1). Since most but not all rain gauges take 7am (EST) observations, only the rain gauges with 7am observations were used. Likewise, the hourly radar data were taken from the 24 hours before the 7am rain gauge observation time. The 24 hourly estimates were summed to correspond with the rain gauge observations. The radar data (RMOSAIC products) from MARFC (bottom solid-lined box in Fig. 1) and NERFC (top solid-lined box in Fig. 1) were combined into one continuous dataset that covered all states in the Northeast Regional Climate Center's operational area. Since the MPE procedure is performed independently at each RFC, some radar pixels fell in an overlap region (rectangular box in middle of

Table 1. Characteristics and optimized parameters for each of 30 cases used and three interpolation methods

Case	Type of Event	Download time	No. of gauges	IDW			MQ			Kriging (exponential variogram)				MPE
				radius	exp	mse	radius	lambda	mse	sill	range	nugget	mse	mse
				(°)		(cm^2)	(°)		(cm^2)	(cm^2)	(°)	(cm^2)	(cm^2)	(cm^2)
11/01/97	Stratiform	Archived	367	2.4	2.5	1.0368	2.00	11.5	1.0275	1.57	1.26	0.33	1.0519	
01/31/00	Winter Storm	Archived	397	0.8	1.5	0.1964	5.50	4.5	0.2097	0.61	2.92	0.01	0.2315	
09/24/01	Convective	Archived	390	8.3	1.5	1.1251	11.50	12.5	1.1161	0.76	3.46	0.42	1.1221	
01/03/03	Winter Storm	Archived	355	1.2	1.5	0.4893	2.50	9.0	0.4701	0.82	5.49	0.18	0.4748	0.8163
02/03/03	Winter Storm	Archived	359	0.8	1.0	0.1226	14.00	3.0	0.1287	0.10	0.67	0.02	0.1135	0.1432
03/20/03	Stratiform	Archived	336	1.5	1.5	0.9571	2.50	5.5	0.9382	1.34	3.02	0.19	0.9495	0.8149
04/07/03	Winter Storm	Archived	338	1.4	2.0	0.1791	7.50	2.0	0.1810	0.20	2.30	0.08	0.1821	0.2627
04/21/03	Convective	Archived	366	0.9	1.5	0.0801	7.00	4.5	0.0797	0.05	1.98	0.05	0.0817	0.0831
05/11/03	Convective	Archived	366	0.9	1.5	0.3758	2.75	7.0	0.3707	0.44	2.62	0.04	0.3790	0.4889
06/11/03	Convective	Archived	372	5.6	1.0	0.2032	7.50	37.5	0.2044	0.09	0.56	0.03	0.2162	0.2114
07/08/03	Convective	Archived	369	0.5	0.5	1.8802	9.25	4.5	1.9037	865.89	53158.92	0.09	1.9150	3.7022
08/03/03	Convective	Archived	356	1.1	0.5	0.6814	13.00	10.0	0.6751	0.22	0.94	0.13	0.7125	0.6806
09/15/03	Convective	Archived	342	0.8	1.0	0.6966	4.00	13.5	0.7051	0.61	1.01	0.08	0.7672	0.7575
10/15/03	Frontal	Archived	371	1.2	1.0	0.8627	6.00	11.5	0.8604	0.68	0.95	0.29	0.8909	1.0388
10/28/03	Stratiform	Archived	344	1.1	1.0	0.5093	15.00	6.0	0.5150	2.56	16.97	0.04	0.5450	0.8150
11/19/03	Stratiform	Archived	357	1.2	2.0	1.1454	13.00	2.0	1.1123	1.36	0.55	0.22	1.1376	1.5705
12/14/03	Winter Storm	Archived	328	2.0	2.0	0.5962	15.00	0.5	0.5429	9892.77	77524.61	0.39	0.5973	0.9867
03/07/04	Frontal	Archived	366	1.4	1.5	0.3433	15.00	11.5	0.3444	112.19	18007.36	0.09	0.3501	0.3907
04/18/04	Small Scale	Archived	349	0.7	3.0	0.0438	1.50	3.5	0.0455	407.87	68487.11	0.00	0.0464	0.0854
05/08/04	Convective	Archived	357	0.5	0.5	0.1975	1.00	19.5	0.2039	0.04	2.54	0.00	0.2192	0.1782
06/28/04	Conv/Strat	Archived	327	0.7	1.0	0.1850	4.75	3.0	0.1792	0.12	1.12	0.02	0.1796	0.1703
07/26/04	Stratiform	Archived	361	0.8	1.5	0.7371	4.25	2.5	0.7271	205.05	1987.40	0.34	0.7906	1.2037
08/30/04	Front/Trop	Real time	309	16.9	1.5	1.0087	14.00	45.0	1.0427	0.10	1.42	0.27	1.0358	19.7215
09/08/04	Tropical	Real time	277	0.8	1.0	1.9570	5.00	4.5	1.8760	5.22	9.62	1.07	1.9612	3.1167
10/18/04	Conv/Strat	Archived	224	1.0	2.0	0.3933	5.25	3.0	0.3881	4178.43	105253.1	0.34	0.4230	0.4388
11/17/04	Small Scale	Real time	365	2.4	2.0	0.0273	11.00	4.5	0.0276	230.01	96789.29	0.00	0.0276	0.0596
12/07/04	Frontal	Real time	297	0.8	1.0	0.2251	6.50	7.5	0.2288	0.11	1.31	0.12	0.2230	0.4511
01/30/05	Small Scale	Real time	260	0.8	0.5	0.0022	0.75	0.5	0.0044	0.00	0.51	0.00	0.0023	0.0015
02/16/05	Winter Storm	Real time	265	3.2	2.0	0.0912	4.00	4.0	0.1009	84.38	19961.17	0.08	0.1032	0.2925
03/01/05	Winter Storm	Real time	253	1.7	2.5	0.1269	1.75	7.5	0.1143	0.11	0.55	0.02	0.1367	0.3883

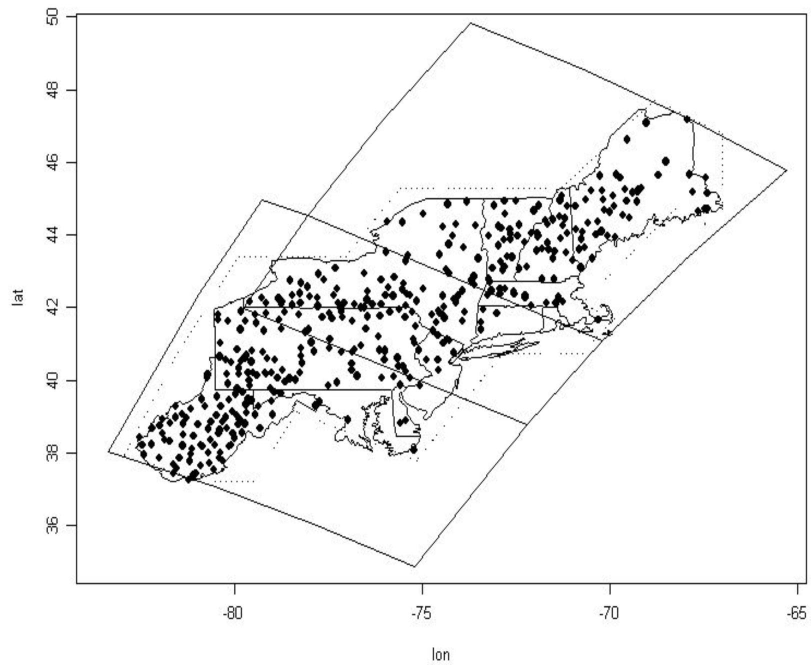


Figure 1. Base map of rain gauge stations used (dots), areas from which data were retrieved (solid lines), and area which will be interpolated for final product (dotted line).

Fig. 1) with two estimated values. Non-missing pixels in the overlap area were averaged.

Occasionally, an estimate at a pixel will be missing due to radar malfunction. This can occur on the scale of a few pixels, or throughout the entire radar umbrella. Missing data can last for one hour, or multiple hours. Therefore, for each pixel, if data is missing during any hour of the day, that pixel is designated as missing for the entire 24 hour period. Interpolation is not performed on these points, and will be designated as missing in the final product.

3. Interpolation Procedures

Each of the three interpolation methods follows the same basic premise for estimating rainfall. Using the radar pixel in which each rain gauge was located, an error was calculated

$$e_i = r_i - g_i \ , \tag{2}$$

where e_i is the error at the *i*th rain gauge, r_i is the radar estimated rainfall at the *i*th rain gauge, and g_i is the gauge value at the *i*th rain gauge. These error values were then interpolated over the entire radar grid using the following equation:

$$R_e = \sum_{i=1}^{n} e_i w_i \ , \tag{3}$$

where, R_e is the estimated radar error at the pixel being interpolated, e_i is once again the error at the *i*th rain gauge, and w_i is the weight assigned to the *i*th rain gauge. This basic equation is used for each method. What differs between methods is the process used to find the weights.

a. Inverse Distance Weighting

The first and simplest method is Inverse Distance Weighting (IDW). Also referred to as the Reciprocal Distance Squared Method, the estimated error value interpolated to each pixel can be estimated by Equation 3. For the IDW method, the weights are calculated with the following equation:

$$w_i = \frac{1/d_i^b}{\sum_{i=1}^{n} 1/d_i^b} \ , \tag{4}$$

where w_i is the weight on the *i*th rain gauge, d_i is the distance (in degrees latitude/longitude) between the radar pixel and the *i*th rain gauge, b is an exponent, and n is the number of rain gauges within a specified radius of the radar pixel (Simanton and Osborn 1980). The exponent and the radius can both be changed to minimize the mean squared error (MSE) using cross-validation (see section 3d). The optimal exponent values found by Simanton and Osborn (1980) range from 1.0 to 3.0, which is consistent with the results from this procedure (Table 1). The radius of influence can range from the shortest distance between any two gages (~0.1°) to the size of the domain (~20°). Equations 3 and 4 are used for each radar pixel, resulting in different weights on each rain gauge for each radar pixel. Rain gauges that are located far from the radar pixel will have large distances and small weights, and will have little effect on the error estimate. The weights will then increase as the distance decreases until approaching infinity at the radar pixel location. The error at radar pixels that contain a rain gauge will therefore just be the error (Eqn. 2) at that rain gauge. The Inverse Distance Weighting method is therefore known as an “exact interpolator” method. The Inverse Distance Weighting method is also simple to program and fast to use.

b. Multiquadric Interpolation

Multiquadric interpolation (MQ) is similar to IDW in that it is based on the distances between points. Nuss and Titley (1994) developed and demonstrated this procedure on pressure estimates, but stated that it could be applied to other spatially varying fields. MQ uses the same interpolation equation (3), but the weights are calculated by,

$$\mathbf{w} = \mathbf{Q}\mathbf{Q}_j^{-1}, \tag{5}$$

where **w** is a vector of weights on the n rain gauges, **Q** is a vector (n dimensional) of radial basis function values between the radar pixel and all rain gauges, that are nonzero for gauges within a radius of influence, and $\mathbf{Q}_j^{-1}$ is a matrix (n x n dimensional) of radial basis function values between each rain gauge and all others within a radius of influence. The radial basis functions use a hyperboloid radial basis function,

$$Q_i = -\left(\frac{d_i^{\,2}}{c^2} + 1\right)^{1/2}, \tag{6}$$

where d_i is the distance between the radar pixel and the ith rain gauge, and c is a small constant needed to make the basis function infinitely differentiable, and,

$$Q_{ij} = -\left(\frac{d_{ij}^{\,2}}{c^2} + 1\right)^{1/2} + (n\lambda\delta_{ij}), \tag{7}$$

where d_{ij} is the distance between the ith and jth rain gauges, and c is the same constant as in Eqn. 6. When $i = j$, the Kronecker delta δ_{ij}, equals 1 (otherwise, $\delta_{ij} = 0$), and $\mathbf{Q}_{ij}$ is corrected for observational uncertainty. Again n is the number of rain gauge observations, and λ is a smoothing parameter which includes a mean-squared observation error. To insure that $\mathbf{Q}_{ij}$ is invertible, c must be relatively small (Nuss and Titley 1994). The constant, c, is insensitive to the results and was set at 0.0008 for this procedure. A radius of influence that determines the number of observations to be used, and the smoothing parameter λ, can be adjusted through cross-validation (see section 3d) for each case to minimize the MSE. Like the IDW method, calculations are made for each radar pixel, but MQ is not an exact interpolator method because of the correction for observational uncertainty. The programming is also slightly more complicated than IDW since the inversion of a large matrix is required.

c. Ordinary Kriging

The final interpolation method is Ordinary Kriging, which looks at the variance between all points over the entire radar field. The equation used to find the estimated error is once again Equation 3. The weights are calculated based on a semivariogram function, also called the variogram, and a linear system of equations, which yields the "best linear unbiased estimator" (BLUE). The variogram gives the variance of differences between two points as a function of the distance between them (Kitanidis 1997, Cressie 1993). The equation for the variogram is:

$$\gamma(d) = \frac{1}{2} E[(e_i - e_j)^2] \ , \tag{8}$$

where $\gamma(d)$ is called the semivariogram, but is typically referred to as the variogram. The variogram, $\gamma(d)$, is a function of the expected value, E, of the squared difference between e_i and e_j, which are the errors at the ith and jth rain gauges, respectively. The weights are found by solving a system of linear equations,

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{b}, \tag{9}$$

where **w** is a vector the weights,

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \nu \end{bmatrix}, \tag{10}$$

in which ν is a Lagrange multiplier, and **b** contains the variogram functions between each gauge and the point to be solved,

$$\mathbf{b} = \begin{bmatrix} -\gamma(\|e_1 - e_0\|) \\ -\gamma(\|e_2 - e_0\|) \\ \vdots \\ -\gamma(\|e_n - e_0\|) \\ 1 \end{bmatrix}, \tag{11}$$

and **A** is a matrix of all possible variograms

$$\mathbf{A} = \begin{bmatrix} 0 & -\gamma(\|e_1 - e_2\|) & \cdots & -\gamma(\|e_1 - e_n\|) & 1 \\ -\gamma(\|e_2 - e_1\|) & 0 & \cdots & -\gamma(\|e_2 - e_n\|) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\gamma(\|e_n - e_1\|) & -\gamma(\|e_n - e_2\|) & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{12}$$

(Kitanidis 1997). By setting the diagonal values of (12) to zero, Ordinary Kriging becomes an exact interpolator method, but will result in a discontinuous map at most rain gauge locations since only the pixel containing the rain gauge will be affected. The computer program, Splus, and more specifically the "krige" function, was used to solve the Kriging equations for this project (Kaluzny et al. 1998).

In order to use the variogram in calculations, an empirical variogram model must be formulated. Three models were compared for this project, exponential (Eqn. 13), spherical (Eqn. 14), and Gaussian (Eqn. 15). Each model uses three parameters, the nugget, sill, and range, plus the distance between points. Typically, the variance at a distance of zero is zero, but in some cases there is a "nugget effect" or discontinuity of the origin (Cressie 1993). At a large enough distance, the variogram should level out to a limit called the sill. The distance at which the data are no longer autocorrelated is called the range (Kaluzny et at. 1998). The exponential model is,

$$\gamma(d) = \begin{cases} 0, & d = 0 \\ c_0 + c_s\left(1 - \exp(-d/3a)\right), & d > 0 \end{cases}, \tag{13}$$

the spherical model is,

$$\gamma(d) = \begin{cases} 0, & d = 0 \\ c_0 + c_s\left(\dfrac{3d}{2a} - \dfrac{d^3}{2a^3}\right), & 0 < d \le a_s \\ c_0 + c_s, & d > a_s \end{cases}, \tag{14}$$

and the Gaussian model is,

$$\gamma(d) = \begin{cases} 0, & d = 0 \\ c_0 + c_s\left(1 - \exp(-7d/4a)^2\right), & d > 0 \end{cases}, \tag{15}$$

where d is the distance between points, c_o is the nugget, ($c_o + c_s$) is the sill, and a is the range. For this project, the optimum nugget, sill, and range were estimated using the "variogram.fit" function in Splus (Kaluzny et al. 1998). It was found that for the 30 cases used, the exponential model performed the best overall, so the results from only the exponential model are hereafter presented.

Using variances and covariances in approaches similar to Kriging have been tried before on radar-rainfall calculation (Seo 1998a, Seo 1998b). A related method called co-Kriging has also been experimented with for rainfall estimation (Azimi-Zonooz et al. 1989, Krajewski 1987). Co-Kriging calculates not only the variances of differences between points for radar values and rain gauges, but also the covariance between both the radar and rain gauge values to calculate an estimate. Results from four different methods of co-Kriging in three climates were presented by Azimi-Zonooz et al. (1989) and are similar to the results of MPE in Seo and Breidenbach (2002). RMSE values range from 0.349 cm to 2.594 cm. For this project, by using ordinary Kriging on the errors, both the rain gauge and radar values are incorporated, and the calculations are simpler. Therefore, co-Kriging was not explored any further. The calculations for ordinary Kriging are also much more complicated than the other methods and a statistical program with Kriging capabilities must be used.

d. Cross-Validation

Cross-validation is used in this project to calculate the optimum parameter values for each interpolation method on a case-by-case basis. Cross-validation is performed by removing each rain gauge point and interpolating to that point using all remaining data. The error between the interpolated value and the actual value is then calculated. After cross-validating for all points, a cross-validated mean squared error (MSE) is calculated. The parameters are then adjusted to a new set of trial values and the process is repeated. For instance, the Inverse Distance Weighting method has two parameters, radius of influence and the exponent. The radius can range from 0.1° to 20° and the exponent can range from 0.5 to 3.0. The result will be a two-dimensional field of MSE values for every possible radius and exponent combination. One combination will have a minimum cross-validated MSE and that combination is considered the best. Cross-validation is performed in this way for each case, resulting in different optimum parameters each time.

e. Smoothing Methods

Because interpolation was performed on the errors, negative values of interpolated precipitation are possible. Since this result is unrealistic, any gauge measurement point with both a rain gauge and radar value of zero were automatically set to an error value of zero. Any interpolated points producing a negative value were set to an error value of the negative of the radar value (thus resulting in a precipitation estimate of zero).

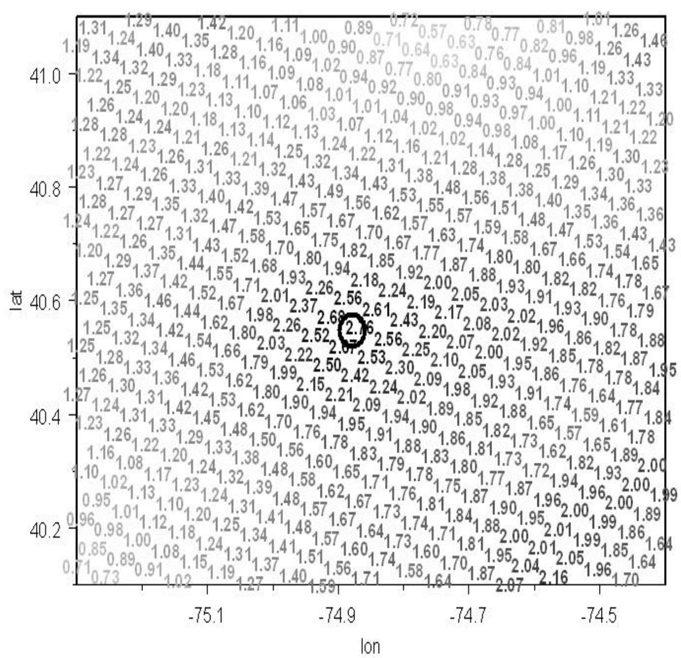


Figure 2: Estimated precipitation amounts around an anomalously high rain gauge amount

Because of the nature of IDW and MQ interpolation, anomalously high or low rain gauge values tend to influence adjacent points in the error field. Figure 2 shows a close up view of the estimated precipitation amounts around an anomalously high rain gauge amount. The rain gauge is specified by the circle and has a precipitation amount of 2.79 cm. The precipitation values decrease almost symmetrically as the distance from the rain gauge increases. To remedy this problem, an additional procedure was applied to the data field to remove rain gauges that disagree strongly enough with the rest of the data, in order to produce a smoother and more physically consistent final map. At each rain gauge location, all radar pixels within a radius of 0.1° were selected. If 75% or more of those radar pixels had a difference in error estimates relative to the gauge error higher or lower than 0.5 cm, that rain gauge was removed. The rain gauges that met these criteria were removed and cross-validation was rerun, which often resulted in new values for the parameters. The same procedure was repeated using a smaller difference of 0.4 cm between gauges and radar pixels. Two more iterations using differences values of 0.3 cm and 0.2 cm followed. To ensure that a large number of rain gauges were not eliminated, the procedure was terminated before the 0.2 cm iteration if more than 10% of the total number of gauges were omitted. The results in Table 1 do not reflect the reduction of cross-validated MSE resulting from this deletion of inconsistent rain gauges.

4. Results

Thirty cases were selected from all seasons, representing various types of weather conditions. Table 1 shows the thirty cases, and various characteristics including weather type, whether the data were downloaded from archives or in real-time, and the number of gauges used in the interpolation. All thirty cases were optimized individually by cross-validated MSE (cm^2) for each of the three interpolation methods. The optimal parameters for each interpolation method are also shown in Table 1, as well as the MSE for those methods and MPE. Three cases are missing in the MPE column because MPE was not operational for those cases. Even for these thirty cases, a wide range of parameters exist. The radius of influence for the IDW and MQ methods were selected from a range of 0.1° to 20° as stated before, and the exponent for the IDW method ranged from 0.5 to 3.0 as stated before. The smoothing parameter for the MQ method was selected from a range of 1 to 100. For Kriging, the values for the nugget tend to be near zero, but the sill and range can approach infinity if the variogram never completely levels off. The values for the nugget, sill, and range, were therefore between zero and infinity.

Table 2. Mean Squared Errors of Three Interpolation Methods and MPE procedure.

	MSE (cm^2)	RMSE (cm)
Multiquadric Interpolation	0.5174	0.7193
Inverse Distance Weighting	0.5229	0.7231
Ordinary Kriging	0.5356	0.7318
Mulitsensor Precipitation Estimation	1.4396	1.1998
	* 0.7365	* 0.8582
Uncorrected Radar	0.8701	0.9328

* After one abnormal case was taken out

Table 2 shows the average MSE of all cases in which MPE was operational for each interpolation method, the MPE method output “MMOSAIC” produced by the River Forecast Centers, and the uncorrected radar. These data only contain cases in which MPE was available. MPE performed very poorly for one case, which will be discussed in more detail in section 5b. Therefore, the MPE results are presented with and without this case. Results for all three methods appear significantly different from MPE and the uncorrected radar. The MQ method seems to perform slightly better than IDW or Kriging, although this difference is on the order of 0.01-0.02 cm.

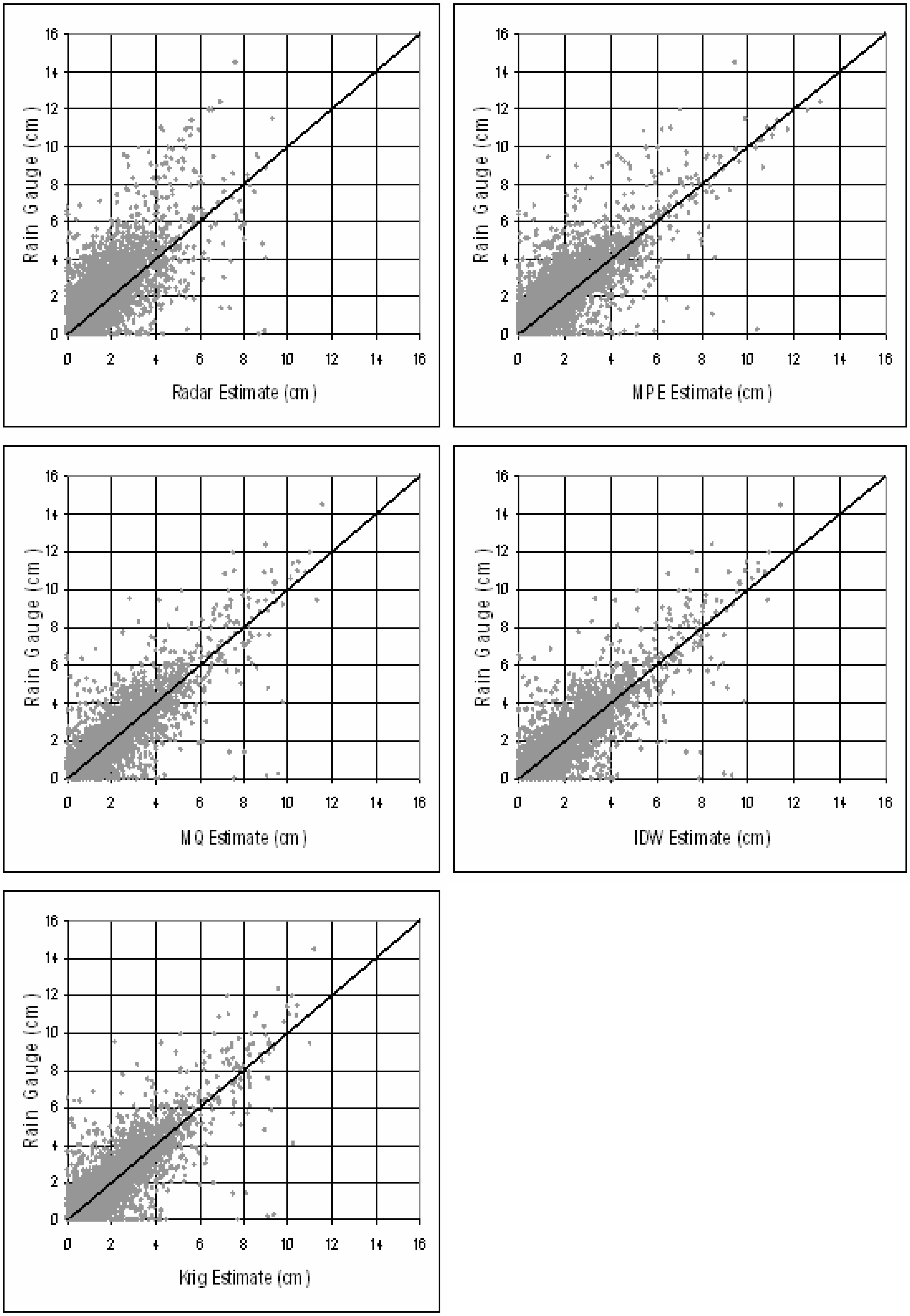


Figure 3. Rain gauge estimates versus the estimates at those rain gauge locations by radar, the MPE method, and three interpolation methods.

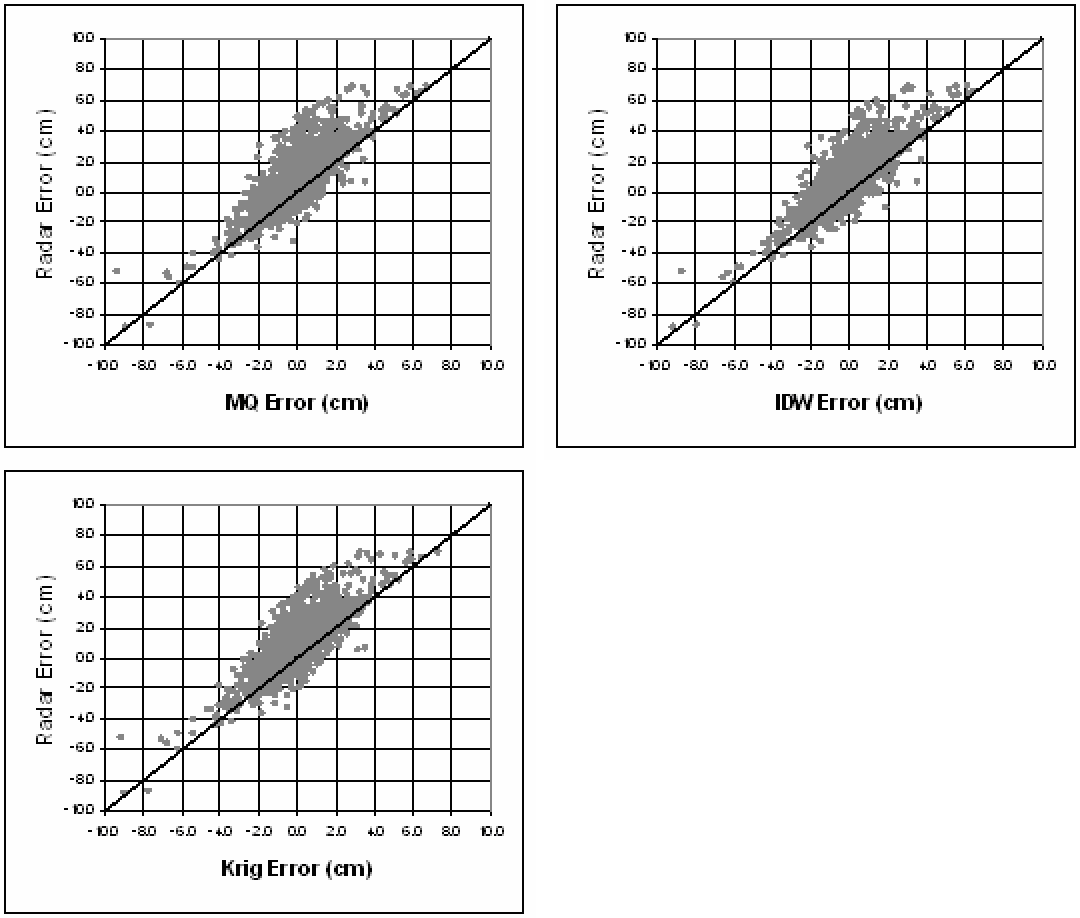


Figure 4. One-to-one graphs comparing radar errors at each rain gauge point to the errors in each interpolation method.

This similarity can also be confirmed graphically by comparing the methods on scatter plots. Figure 3 shows that there is a wider scatter of points when rain gauge amounts are compared to radar estimates and MPE method estimates, and the three interpolation methods. Comparisons can also be made using one-to-one graphs, where graphs that contain points close to the one-to-one line indicate that those methods are similar. Comparisons of each method to the radar estimates (Figure 4), the MPE estimates (Figure 5), and each other (Figure 6) confirm the results from Table 2. The point clouds in Figures 4 and 5 are much larger than in Figure 6, indicating a difference between the interpolation methods and the radar estimates and MPE

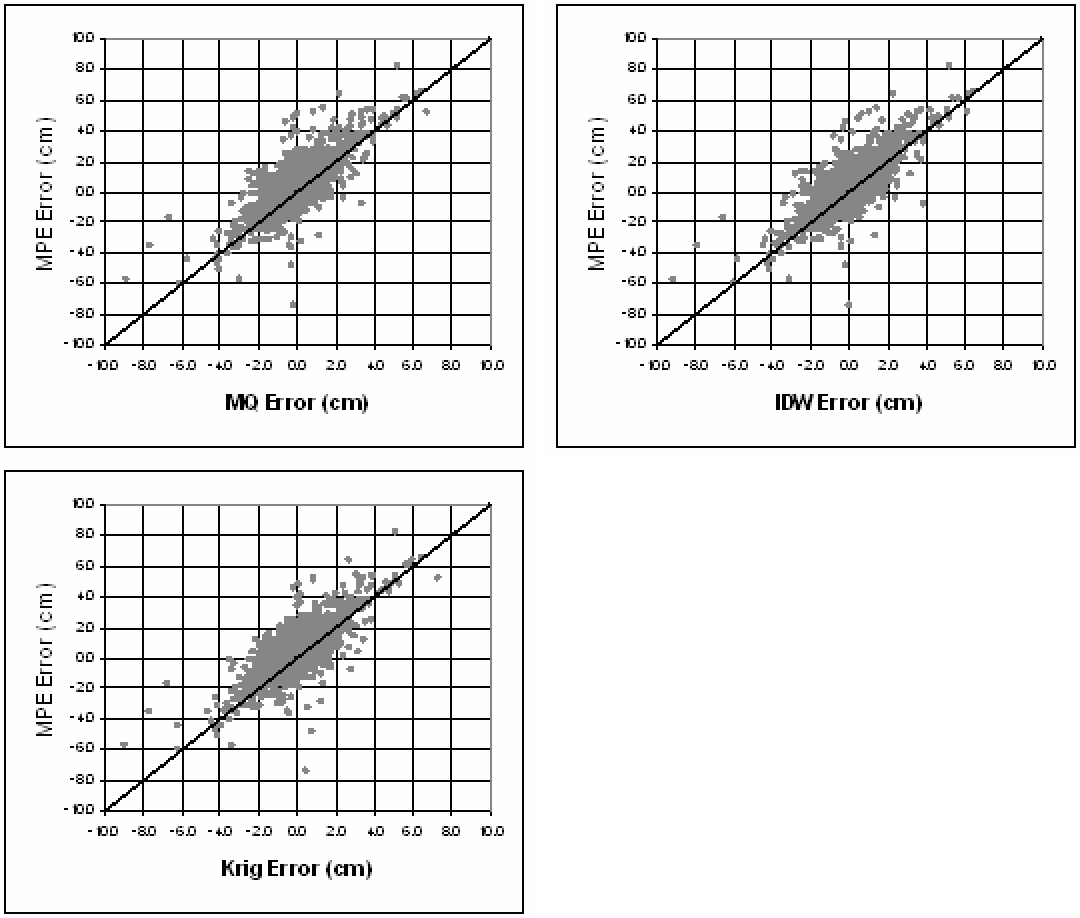


Figure 5. One-to-one graphs comparing MPE method errors at each rain gauge point to the errors in each interpolation method.

estimate. Even though the MQ method performs slightly better than the other two methods, Figure 6 indicates that the difference is minimal.

To explore these differences more quantitatively, permutation tests for each method comparison were performed. The purpose of a permutation test is to compare two groups of data, in this case the average cross-validated MSEs for each pair of interpolation methods (Table 1), by calculating a large number of permutations of the two groups. For each permutation, a test statistic, in this case the ratio of the average cross-validated MSEs for each method is calculated. This is repeated for a large number of permutations, in this case 10,000, creating a distribution of test statistics

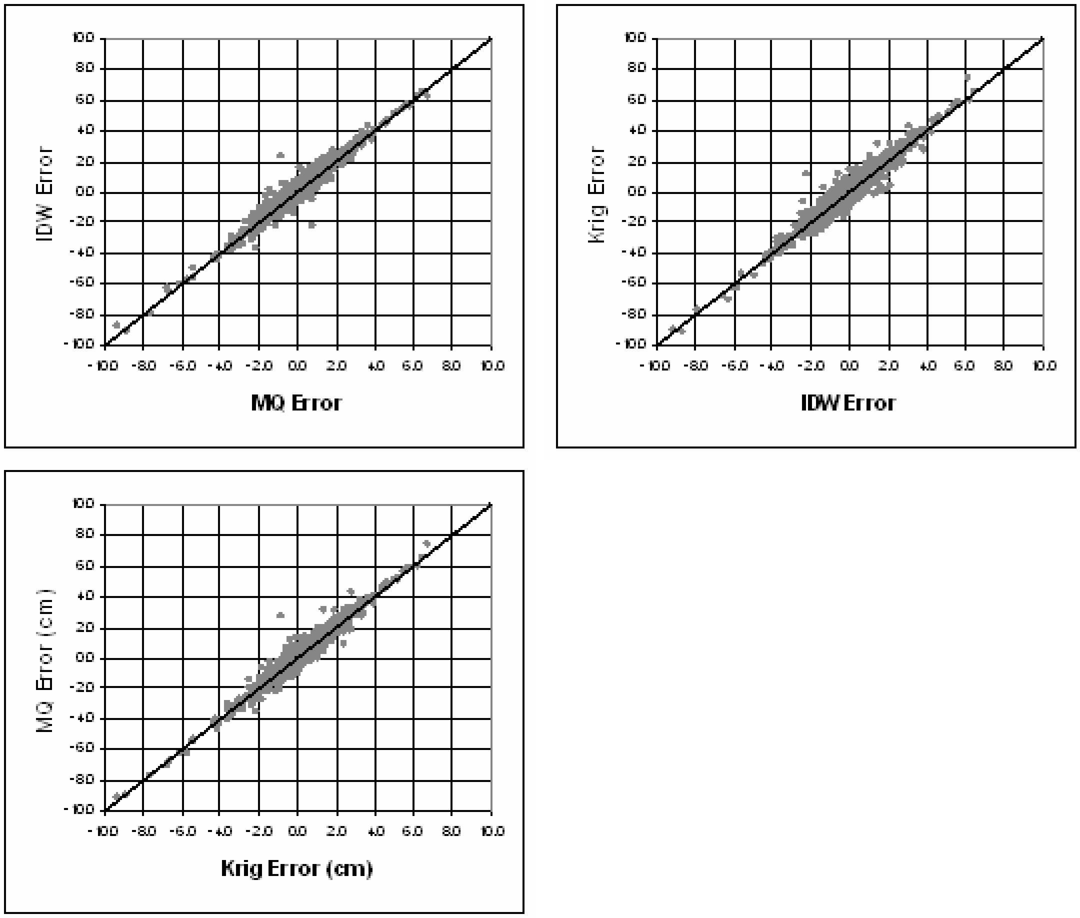


Figure 6. One-to-one graphs comparing the errors at each rain gauge point for each interpolation method to the errors of each other interpolation method.

consistent with the null hypothesis of no differences between interpolation methods. By observing where the sample test statistic lies in this distribution, one can evaluate the plausibility of that null hypothesis. The permutations were selected by a uniform random number generator. The data are paired by case day, and each case is considered independent. The random number determines into which group the data will be placed for each of the 10,000 permutations (i.e. if number is < 0.5, data stays in observed groups, if number is ≥ 0.5, data are moved to opposite groups).

Three permutation tests comparing MQ to IDW, IDW to Kriging, and Kriging to MQ were calculated. When comparing MQ to IDW, the ratio of the MSEs of all

MQ cases to the MSEs of all IDW cases corresponded to the 3.27 percentile of the null distribution. For the IDW to Kriging comparison, the sample ratio was in the 0.13 percentile. For the Kriging to MQ comparison, the sample ratio was in the 100 percentile. In the comparisons of IDW to Kriging and Kriging to MQ, the null hypotheses involving Kriging were rejected in favor of the alternative that Kriging results are different from the others. Given that this permutation test is two-tailed, the comparison of MQ to IDW does not reject the null hypothesis at the 5% level of significance. Despite the permutation test results that indicate that MQ and IDW perform similarly, only one interpolation method could be chosen to be used operationally. The IDW method was chosen since the calculations are simpler.

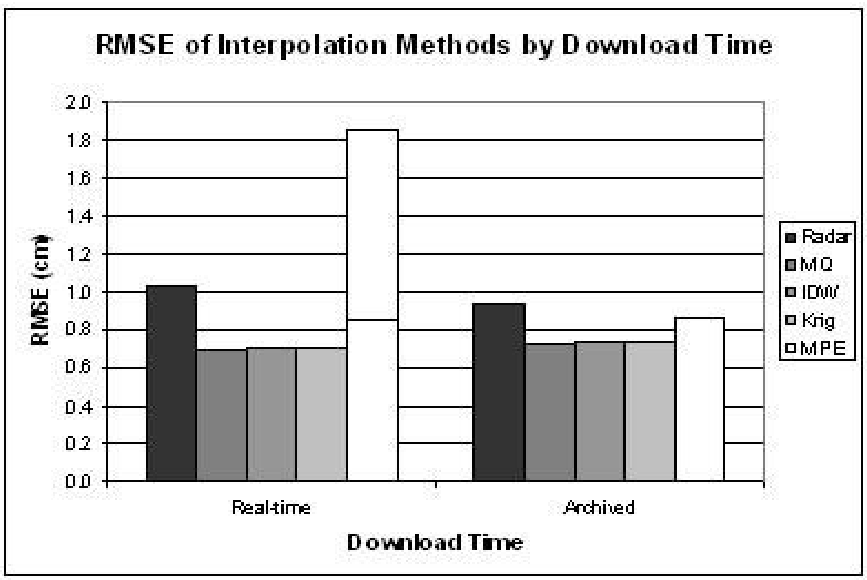


Figure 7. Cross-validated RMSE grouped by time of download.

Cases in which MPE data were available were also grouped in a few ways to look for similarities. First, the cases were divided based on when they were downloaded (real-time or archived) to determine whether the quality control steps performed for archiving improves the estimates. The results (Figure 7) indicate that there is very little difference in real-time and archived cases after the interpolation methods are performed. The decrease in RMSE in radar may be a factor of the quality control of the archiving process. The large spike in the MPE method for real-time cases can be attributed to a few observations in one case which will be explained in more detail in Case 2 (see section 5b). This spike can also be observed in Figures 8-11. The line on the MPE bar indicates the MSE value excluding this case.

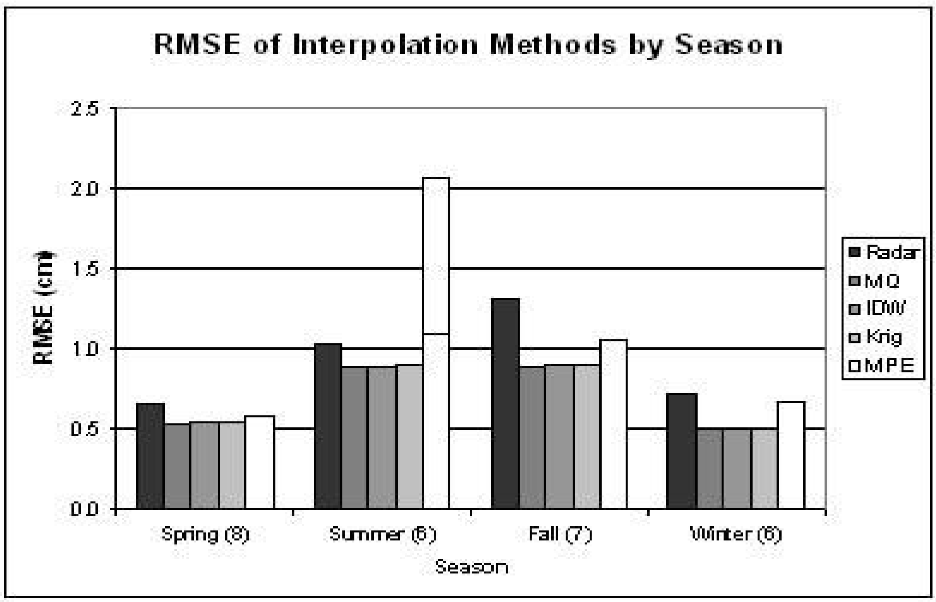


Figure 8. Cross-validated RMSE of cases grouped by season.

The cases were next grouped by season (Figure 8). The numbers in parentheses indicate how many cases are in each category. Each season contains cases from three months, e.g. spring contains cases in March, April, and May. Cases in summer and fall tend to have a higher RMSE than those in spring and winter, but the difference is only about 0.3 cm. Cases in the summer and fall also have a higher average rain gauge amounts which may be the reason for the higher RMSE.

Consistent with Figure 5, all three interpolation methods perform similarly, and better than MPE. The large spike in the MPE estimate in the summer is once again the errors from Case 2 (see section 5b).

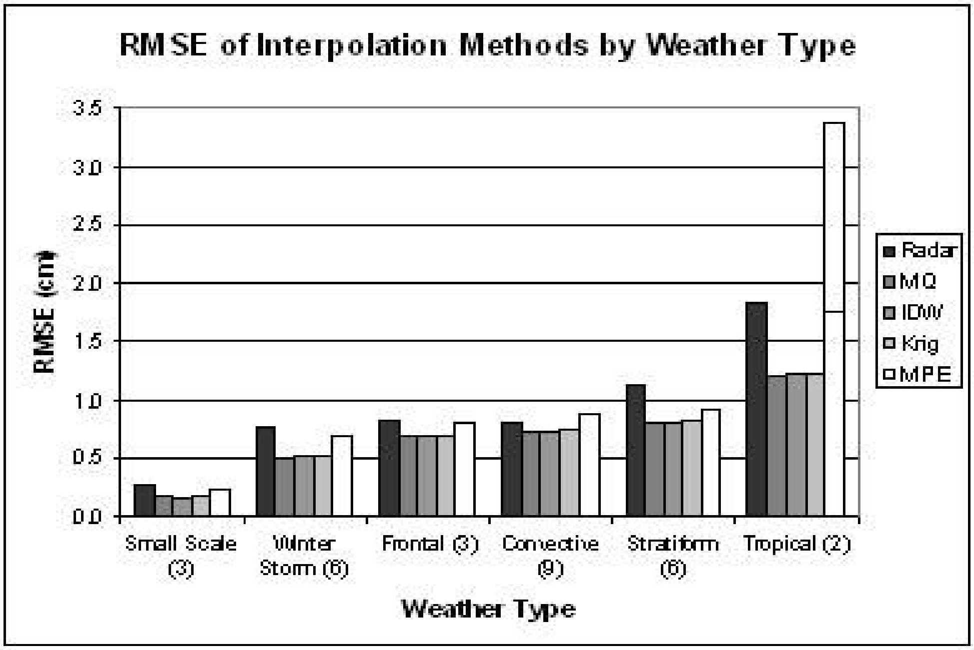


Figure 9. Cross-validated RMSE of cases grouped by type of weather.

Cases were also separated based on weather types (Figure 9). Because of the large area covered, in a few cases different weather types affected different parts of the region (i.e. frontal in the north, tropical in the south). These were included in more than one category. Some categories took precedence over the others and are explained below in order of precedence. The small scale category includes cases in which only a small area (less than 1/3) of the coverage area received precipitation. The tropical category contains cases in which a remnant of a tropical system passed over the coverage area. Winter storm cases included systems capable of producing ice and/or snow, which was determined by surface temperature observations. The frontal category indicates cases containing fronts and/or an extratropical system. If the case did not fall into the previous categories and contained convective precipitation, it was classified as convective. Likewise, if the case did fall into the previous categories and contained stratiform precipitation, it was classified as stratiform.

All three interpolation methods perform similarly for each weather type and better than MPE. Although, there seems be three different levels of performance with small scale cases performing the best, tropical cases performing worst, and intermediate performance for the other groups. The average rain gauge amounts reflect this trend as well. All three interpolation methods perform better than the radar estimates in every type of weather.

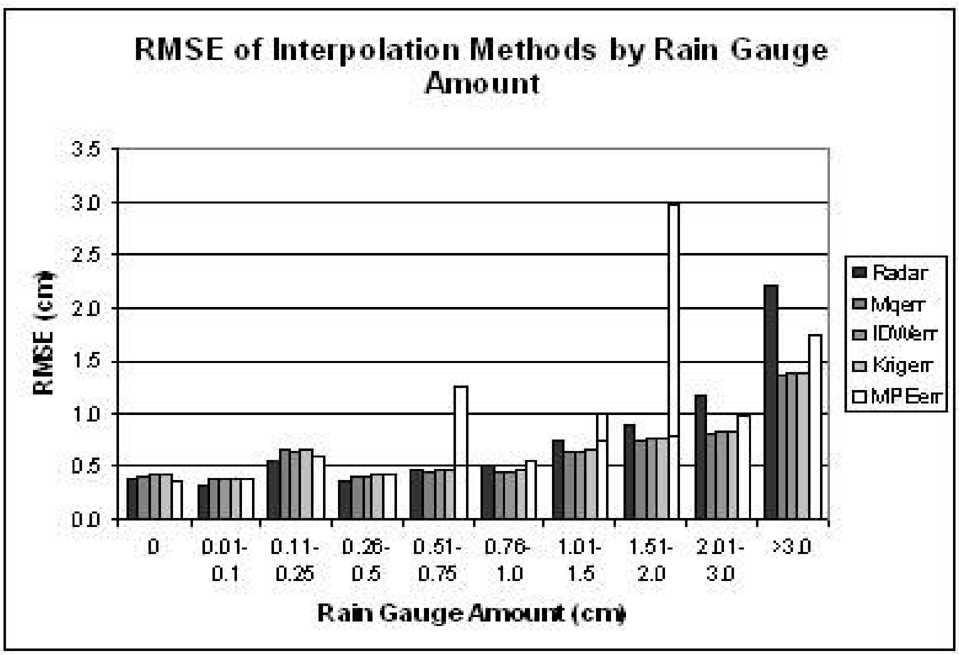


Figure10. Cross-validated RMSE of rain gauges grouped by rainfall amount.

Figure 10 stratifies individual observations by rain gauge amounts. The basic trend indicates that errors increase as the rainfall amounts increase. The three interpolation methods once again perform similarly. An interesting difference though, is that they improve the radar estimates at high rain gauge values, but do slightly worse than the radar estimates at low rain gauge values. This difference in RMSE at low values is less than 0.1 cm, compared to the improvements of 0.2 cm to almost 1.0 cm at higher values. Another way to look at this is shown in Figure 11 where the RMSE is divided by the rain gauge amount, resulting in a percentage. From this perspective, all methods perform poorly at low rainfall amounts. The radar estimates tend to perform best in the middle ranges and decrease in accuracy with smaller and larger amounts. With the exception of MPE, all interpolation methods perform well at rain gauge values of over 0.25 cm.

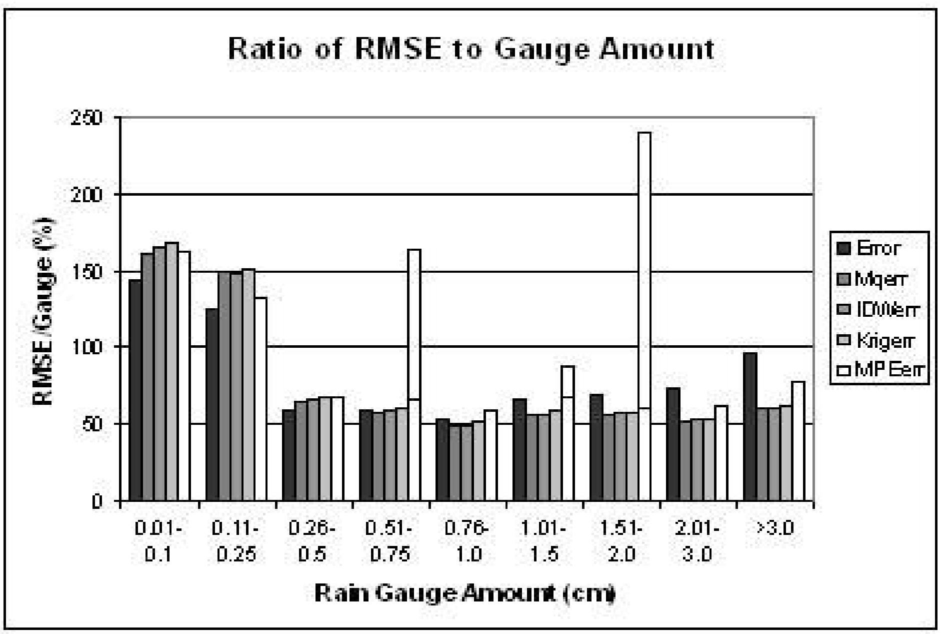


Figure 11. Absolute error at rain gauges divided by rain gauge amounts and grouped by rainfall amount.

The trend found in Figures 10 and 11 most likely accounts for the differences found for the categories in Figures 8 and 9. The summer and fall cases have an average rain gauge amount of 0.94 cm and 2.06 cm, respectively, while the spring and winter cases have an average rain gauge amount of 0.62 cm and 0.83 cm, respectively. The small scale category has an average rain gauge amount of 0.10 cm, the tropical category has an average rain gauge amount of 2.84 cm, and the other categories lie in the middle (Winter: 0.87 cm, Frontal: 1.29 cm, Convective: 0.79 cm, Stratiform: 1.67 cm)

5. Case Studies

a. Case 1 – April 7, 2003

To demonstrate the procedure, two cases will be examined step by step. Figure 12a-b shows the rain gauge and radar fields for the April 7, 2003 winter storm case. The domain is outlined with a black dotted line. Radar values of 0.0 to 0.19 cm are shaded in the lightest gray, while any white pixels within the dotted area are missing

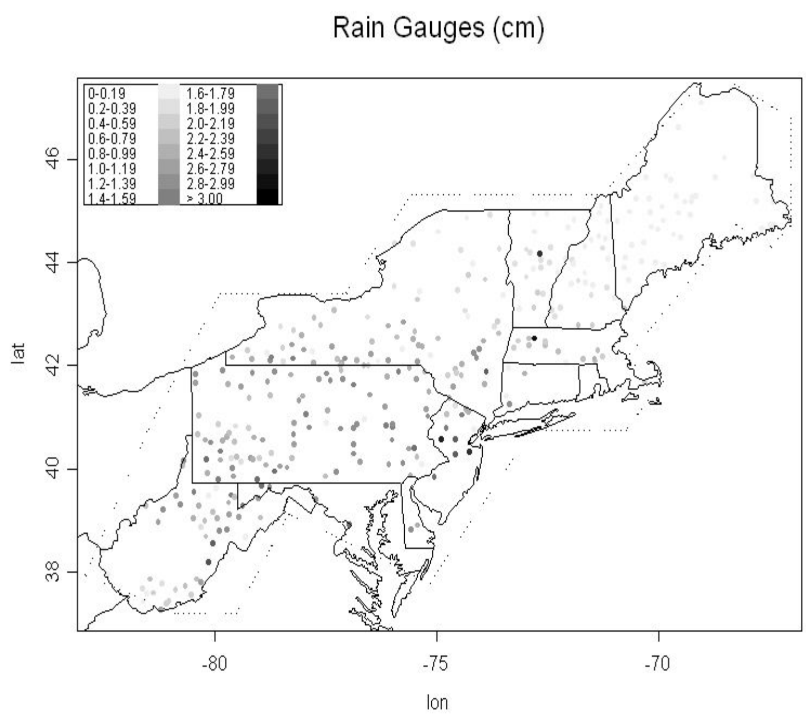


Figure 12a. Rain gauge amounts for Apr. 7, 2003 case.

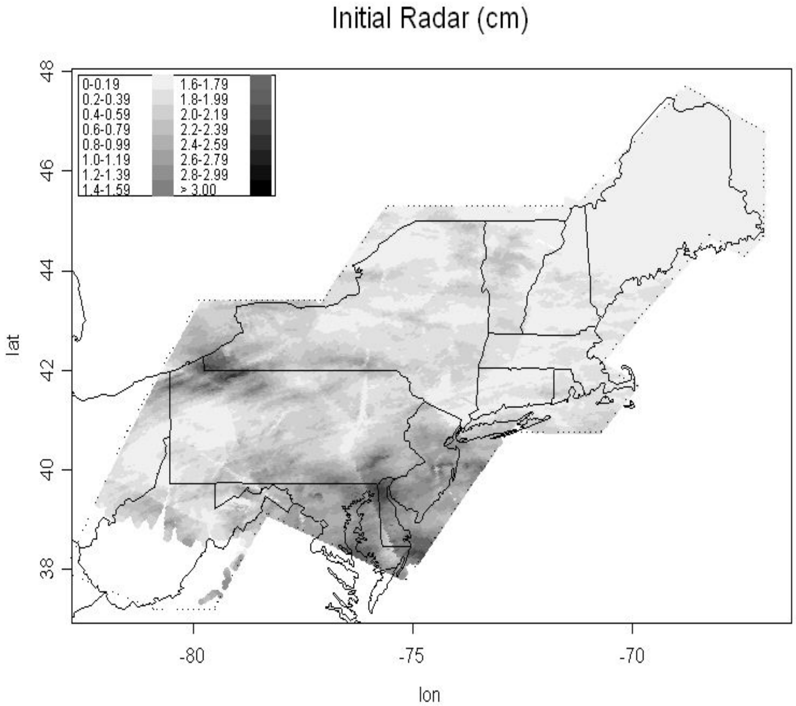


Figure 12b. Uncorrected radar estimates for Apr. 7, 2003 case.

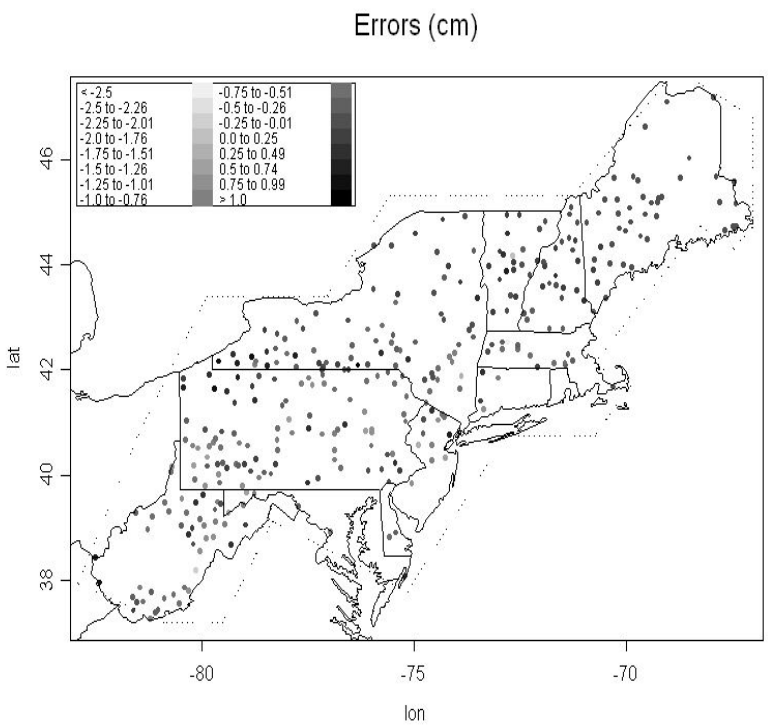


Figure 12c. Radar errors at each rain gauge for Apr. 7, 2003 case.

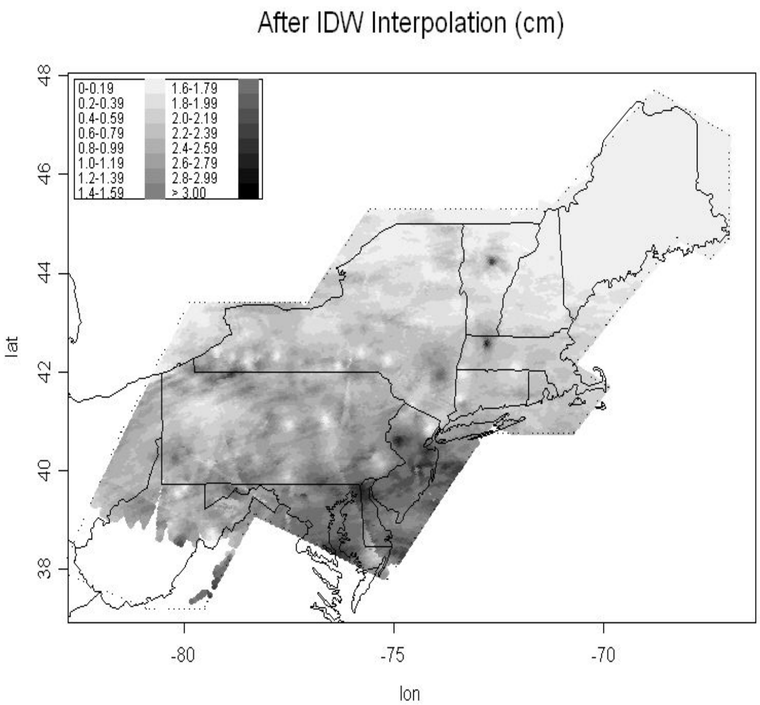


Figure 12d. IDW corrected radar for Apr. 7, 2003 case.

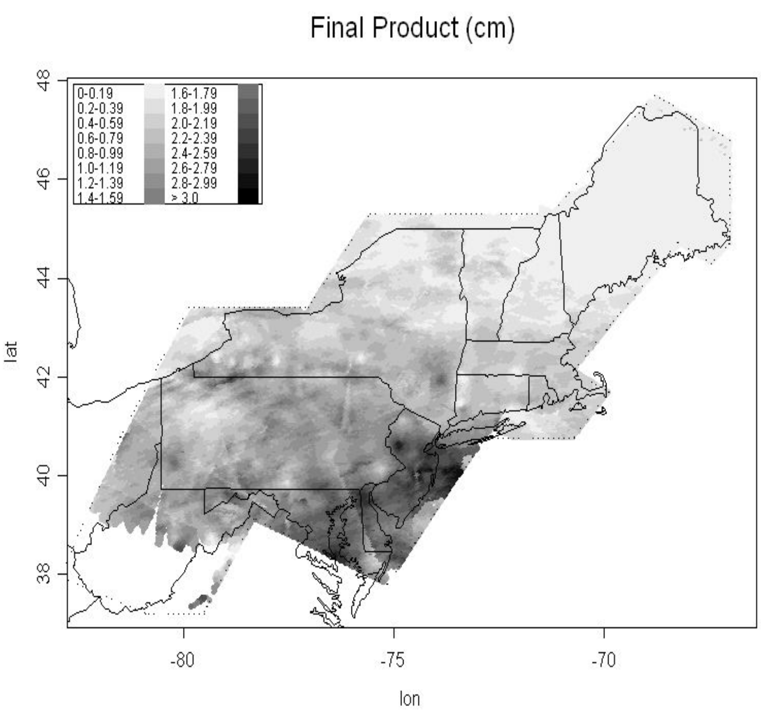


Figure 12e. Final IDW corrected radar after bad points were removed for Apr. 7, 2003 case.

data. The radar pixel that corresponds to a rain gauge is used to find the errors at each rain gauge (Figure 12c) by subtracting the rain gauge observations from the radar estimates (Eqn. 2). The original MSE for the initial radar field is 0.3418 cm^2 and the MPE method improves the MSE to 0.2627 cm^2 (Table 1). Cross-validation was run on this case and the optimum radius and exponent for the IDW method was 1.4° and 2.0, respectively (Table 1). The IDW method improved the precipitation estimate even more to an MSE value of 0.1791 cm^2. IDW performed better than MQ (0.1810 cm^2) and Kriging (0.1821 cm^2) for this case (Table 1). Figure 12d shows the resulting estimated precipitation field after performing an IDW interpolation on the errors (Figure 12c) and subtracting these values from the initial radar estimates (Figure 12b).

The next step removes the numerous local minima and maxima that appear in Figure 12d. For IDW, after the final iteration, the number of rain gauges was reduced from 338 to 319 and the MSE decreased from 0.1791 cm^2 to 0.1086 cm^2. The decrease in MSE results from removing the gauges that are inconsistent with the rest of the map and therefore their removal reduces the cross-validated MSE. When the same type of procedure is applied to MQ, with the change of radius mentioned in section 3e, the results are similar. The number of gauges decreases from 338 to 327, and the MSE decreases from 0.181 cm^2 to 0.1108 cm^2. Figure 12e shows the resulting radar field for April 7, 2003 after IDW interpolation and the local min/max corrections.

b. Case 2 – August 30, 2004

The synoptic situation for Aug. 30, 2004 was a frontal system approaching from the north, and a tropical system from the south. The result was two distinct areas of precipitation (Figures 13a-b). With two areas of precipitation and large precipitation amounts, the resulting MSE for the radar estimate was large (1.0825 cm^2). What makes this case even more unique is that the MPE method performed poorly with an MSE of 19.7215 cm^2 (Table 1). Further investigation found that the MPE method produced an area of extreme precipitation in part of Vermont with a maximum of 229 cm over one day. Three rain gauges with reasonable observations (0.58 cm, 1.45 cm, 1.52 cm) lie in this area. This bad value was not found in the original radar estimate. Since the MPE method involves some human interaction, it is a possibility that this was a human error, but there is probably no way to know for sure. Nevertheless, these rain gauges were used in the MSE calculation for MPE, and an anomalously high value of MSE resulted.

The same procedure was used on the errors in this case as outlined in the previous case to make estimates of the precipitation amounts (Figures 13c-d). An interesting outcome of this case was the high values for the optimal radius of influence. For IDW, the radius was 16.9° and for MQ the radius was 14° (Table 1).

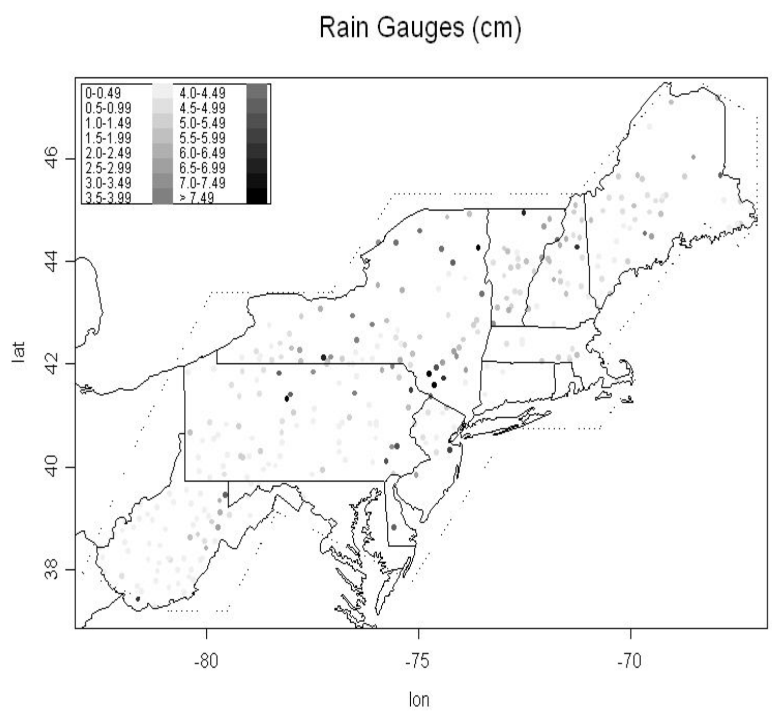


Figure 13a. Rain gauge amounts for Aug. 30, 2004 case.

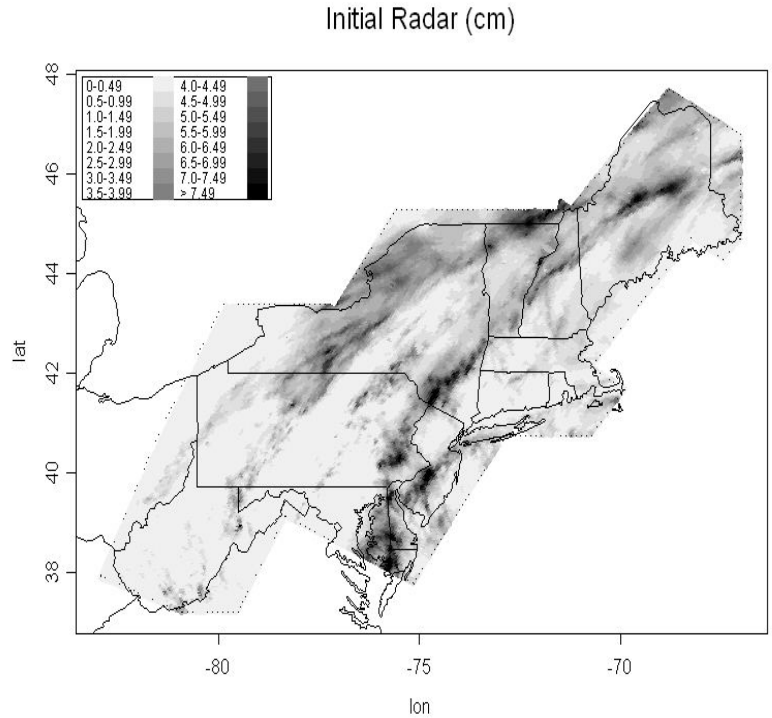


Figure 13b. Uncorrected radar estimates for Aug. 30, 2004 case.

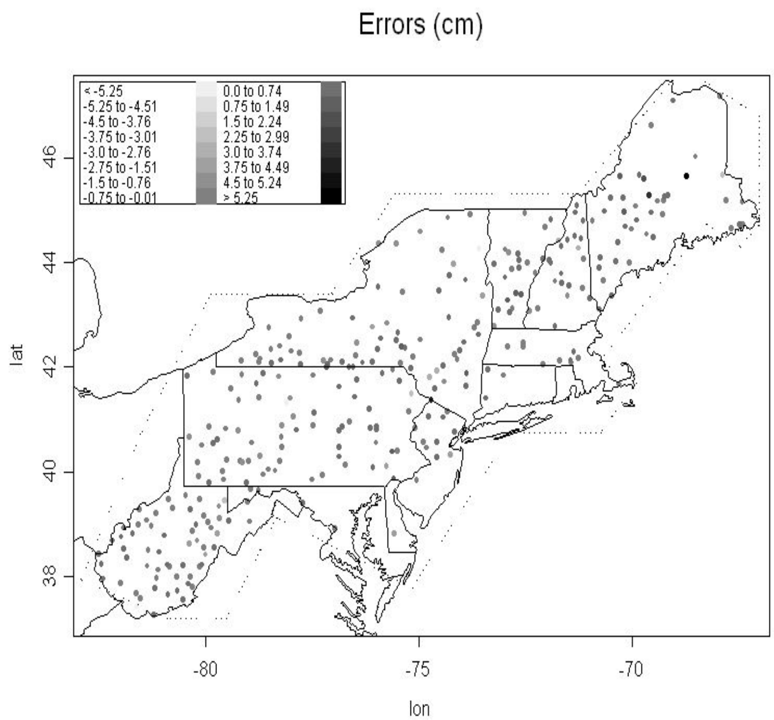


Figure 13c. Radar errors at each rain gauge for Aug. 30, 2004 case.

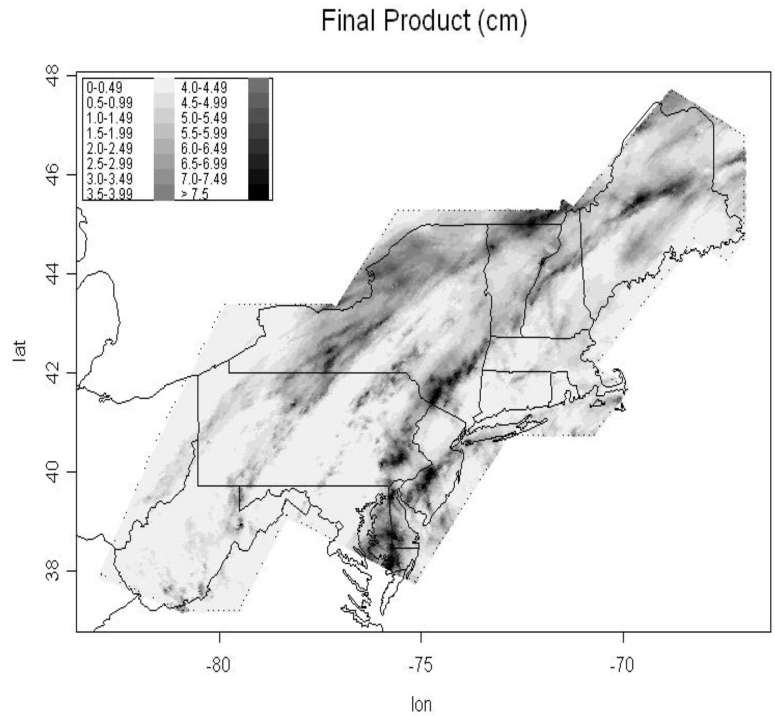


Figure 13d. IDW corrected radar for Aug. 30, 2004 case.

At these values, a large percentage of the rain gauges were used in the interpolation of each radar pixel. It is possible that since there are two distinct areas of precipitation, by adding more rain gauges, more information is added, slightly improving the outcome. The IDW method performed the best with an MSE of 1.0087 cm^2, in comparison to MQ with 1.0427 cm^2 and Kriging with 1.0358 cm^2. All methods still improved on the radar estimates and easily beat the MPE estimate. In this case, no local minima or maxima appear in the final precipitation product, but for consistency, the procedure to identify bad rain gauges was run. No rain gauges were removed in this case and the final produce did not change from Figure 13d.

c. Cases 3 and 4 – March 20, 2003 and May 8, 2004

Something that cannot be analyzed using MSE values is the small scale radar errors that can only be seen visually. Since the cross-validated MSEs are calculated using rain gauges and since these errors are often localized, they are sometimes overlooked. These radar-associated errors should probably be fixed in the future, but for the purposes of this project, the errors will simply be addressed so users can be aware. An error field can be constructed using the rain gauges, but when that error field is subtracted from the radar, any problems associated with the radar will still exist. Figure 14 shows some of these errors in the radar for the March 20, 2003 case after the IDW interpolation correction. The most apparent problems are the radial lines extending from the two radar tower locations, denoted by arrows. These lines can range from the size of a few pixels wide, to encircling the radar. This error is

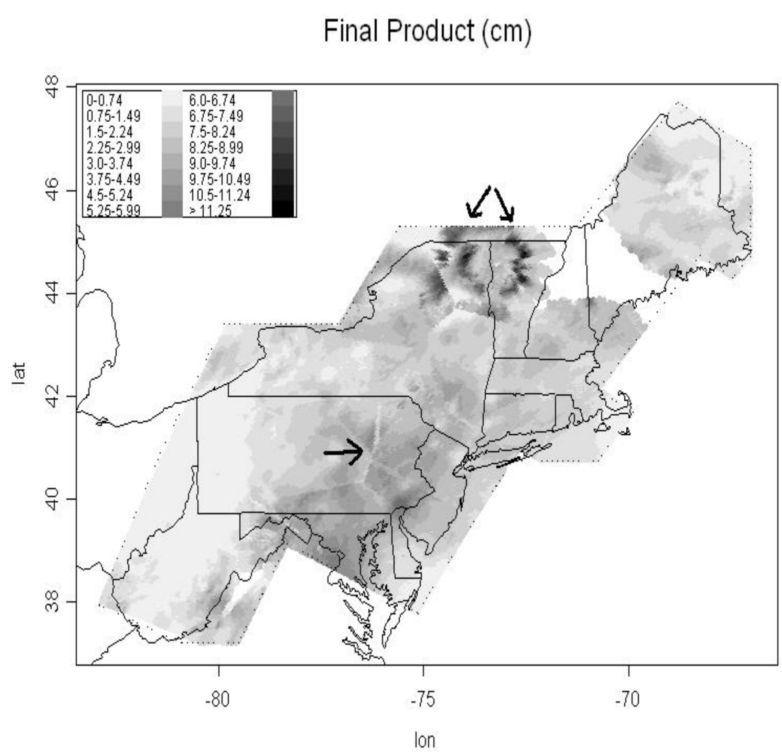


Figure 14. Final IDW corrected radar precipitation estimates for March 20, 2003.

most likely causes by internal errors within the radar. This case is the most extreme of all 30 cases, but 6 cases had this problem to some extent. The other problem in Figure 14 is the missing radar data in southern Maine, northern New Hampshire, and northern Vermont. This can be observed in 8 of the 30 cases, and seems to happen in the same area most of those times. This problem cannot be remedied using this method, but perhaps the area could be filled in using satellite or rain gauge data and other statistical techniques.

The last radar error can be found in the May 8, 2004 case (Figure 15). Very small areas of very high of precipitation are indicated in Maine. Seven of the 30 cases contained this same problem to some extent. This case shows numerous dots of extreme values over a small area clustered around the Portland, ME radar. Other cases contained single dots scattered around the coverage area. The cause of this problem

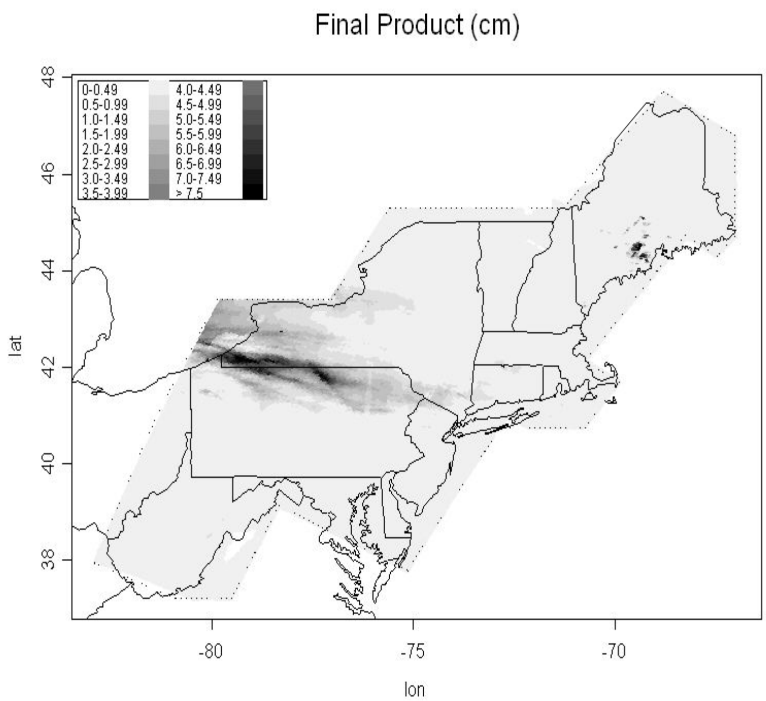


Figure 15. Final IDW corrected radar precipitation estimates for May 8, 2004.

could be a number of things ranging from anomalous propagation to bird migration. After an inspection of a national radar loop for this date, similar phenomena appeared at many other radar sites during the same time. Therefore, in this case these values seem to be errors. Although, with many other cases it may be difficult to distinguish between radar errors and an actual small scale event in order to correct the error. Unless a rain gauge is located at that point, there is no way to determine the correctness of these localized anomalous values. This could be a case where human interaction is needed to correctly identify the problem.

6. Discussion

This simple interpolation method (inverse distance weighting) produces an estimated precipitation product for the northeastern United States with radar-level resolution (~ 4km x 4km). Since this method performs better than Kriging and similarly to MQ, and since the calculations are simpler than MQ, IDW was chosen over the other two methods to be used operationally. The trend of rain gauge amounts and method performance (Figures 10 and 11) indicates that all of these candidate interpolation methods perform best for moderate rainfall amounts (0.25 cm to 1.5 cm). RMSE values become large at rain gauge values above ~1.5 cm, and percentages of RMSE divided by rain gauge amounts are large at rain gauge values below ~0.25 cm. Cases in which large amounts of rain falls, such as tropical systems, perform poorly, but still better than the uncorrected radar and MPE estimates. Multiplicative rather than additive corrections were tested to account for this trend, but the MSE values increased. This result could partially explain why the interpolation methods, which use an additive error, performed better than the MPE method, which uses a multiplicative bias. However, most of the improvement likely comes from using a large number of daily rain gauge observations rather than a few hourly rain gauge observations.

Some problems with this procedure still exist because of the quality of the radar input. As can be seen in some of the cases, radial lines, splotches of high values, and nonfunctioning radars result in unreliable output. These problems cannot be fixed with rain gauge verification since a rain gauge rarely aligns with a radar problem. Even then, the problem is usually so severe that an interpolation method would still not produce a value consistent with the rest of the map. Fixing some of these types of radar errors could be done as a next step to clean up the product. Like the procedures used by RFCs, satellite estimates could be implemented in areas where the radar does not produce any data. Algorithms could also be created that filter out radar splotches and radial lines. Some caution should be used though, to prevent results that produce even worse estimates like the MPE method for the Aug. 30, 2004 case (Table 1).

In real time, this product will work well for applications that need only daily data, such as crop modeling or long term flood forecasting or monitoring. By using a high resolution precipitation field in crop modeling, there is increased accuracy in the initialization of each day's simulation. Observation of high resolution rainfall over a few days can give forecasters a better idea of where flooding may or may not occur. The archival product will provide a database of these high resolution precipitation estimates for climatological analysis. The final product is a text file containing latitude, longitude, and the interpolated rainfall amount. Those data can then be used to either graph the entire domain, or just pick out points of interest. The data will not appear for points in which the corresponding radar has gone down for any part of that day. The final products will be soon be available online daily and archived.

7. Conclusion

Radar is probably the best available source of data for precipitation estimation at high spatial resolution. Because of the variability of the Z-R relationship, other information must be used to correct the resulting errors. Using daily radar estimated rainfall totals together with daily rain gauge amounts, produces substantial improvements. Synoptic and seasonal differences in precipitation characteristics can be accommodated separately for each day by performing cross-validation on the errors at each rain gauge location. The interpolation methods perform best in cases with medium amounts of precipitation (between 0.25 cm and 1.5 cm), and perform the worse in cases, such as tropical systems, with large amounts of precipitation. In most cases though, there is improvement over the radar estimates and the MPE estimates.

Improvement can still be made to this procedure. Since radar malfunction errors are not addressed in this procedure, some improvement could still be made in the accuracy of the product. For cases when one or two radar umbrellas fail to output any data, other methods such as satellite estimates must be used to fill in these holes. But future procedures must be careful not to decrease the accuracy already obtained, which occurred with the MPE method in the Aug. 30, 2004 case.

Even at this stage, the product can have many applications to scientists and non-scientists in the areas of crop modeling and long-term flood forecasting and monitoring. The IDW interpolation method is also simple to program. With the availability of radar data from local RFC offices, this procedure could easily be expanded to the entire country.

APPENDICES

Appendix A: Inverse Distance Weighting Computer Programming Codes

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!                                                                !!!
!!! Program:     inversedist                                       !!!
!!! Author:      Eric Ware                                         !!!
!!! Description: Interpolates rain gauge amounts to radar grid.    !!!
!!!                                                                !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
PROGRAM inversedist
    IMPLICIT NONE
!
! *******************************************************************
! * Declaration Section
! *******************************************************************
!
! *************
! * Variables *
! *************
!
! * I, J, K -- Iteration Values
! * length -- number of rain gauges
! * out, out2 -- 1 or 0
! * badgauge -- number of gauges being removed
! * radius -- radius of influence
! * expon -- exponent in idw
! * min_exp -- cross-validated exponent
! * diffval -- difference btwn radar pixels in min/max cleanup
! * stout -- number of stations removed
! * top, bottom -- used in idw
! * min_mse -- cross-validated mse
! * min_rad -- cross-validated radius
! * mse -- mean squared error
! * cnt -- counts total number of surrounding pixels in min/max
!          cleanup
! * hit -- counts surrounding pixels as incr or decr in min/max
!          cleanup
! * diff -- difference in pixel values in min/max cleanup
! * avg -- hit/cnt in min/max cleanup
!
    INTEGER :: I, J, K, length, out, out2
    INTEGER :: radius, expon
    REAL :: diffval, stout, badgauge
    REAL :: top, bottom, min_mse, min_rad, min_exp, mse
    REAL :: cnt, hit, diff, avg
!
! *************
! * Constants *
! *************
!
! * radit -- upper limit of radius*10
!
    INTEGER, PARAMETER :: radit = 200
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
!
! **********
! * Arrays *
! **********
!
! * stn -- just station id from precip file
! * hrap -- hrap values for station
! * lat, lon -- from precip file
! * gauge -- rain gauge value
! * radar -- radar value at gauge
! * err -- error at gauge
! * fullrad, fulllat, fulllon -- radar values, lats/lons for entire
!                                 grid
! * rad, radlat, radlon -- radar values, lats/lons for final grid
! * esterr -- estimated error
! * est_p -- estimated precipitation
! * nerad -- radar values in Northeast
! * marad -- radar values in Mid-Atlantic
! * dist -- distance between radar points
!
    INTEGER, DIMENSION (500) :: stn
    INTEGER, DIMENSION (2,500) :: hrap
    REAL, DIMENSION (500) :: lat, lon, gauge, radar, err
    REAL, DIMENSION (66600) :: fullrad, fulllat, fulllon
    REAL, DIMENSION (36610) :: radlat, radlon, rad, esterr, est_p
    REAL, DIMENSION (152,175) :: nerad
    REAL, DIMENSION (200,200) :: marad
    REAL, DIMENSION (36610,500) :: dist
!
! * Data statements
!
    DATA top, bottom/0.0,0.0/
    DATA diffval/0.5/
    DATA mse/0.0/
    DATA min_mse/500.0/
    DATA out/0/
!
! *******************************************************************
! * Execution Section
! *******************************************************************
!
! *****************
! * Input Section *
! *****************
!
! * Open files
!
   OPEN (unit=1, file='/home/ecw25/precip_totals/Apr0703/tot7.txt',
         status="unknown")
   OPEN (unit=2, file='/home/ecw25/precip_totals/latlonfile.txt',
         status="unknown")
   OPEN (unit=3, file='/home/ecw25/precip_totals/fulllatlonfile.txt',
         status="unknown")
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
   OPEN (unit=4,
        file='/home/ecw25/precip_totals/Apr0703/Apr0703radar.txt',
        status="unknown")
   OPEN (unit=5, file='/home/ecw25/statout.txt', status="unknown")
   OPEN (unit=6, file='/home/ecw25/finalradar.txt', status="unknown")
 10 FORMAT (I6, 1x, 2(I4, 1x), 2(F7.3, 1x), 3(F6.2))
 11 FORMAT (2(F10.4, 1x))
 12 FORMAT (2(F8.4, 1x), F6.3)
 13 FORMAT (I6, 1x, 2(F7.3, 1x), 3(F6.2))
!
! * Read files
!
    length = 0
    DO I = 1, 500
        READ (1,10,END=20), stn(I), hrap(1,I), hrap(2,I), lat(I),
                            lon(I), gauge(I), radar(I), err(I)
      length = length + 1
    END DO
 20 CONTINUE
    DO I = 1, 36610
        READ (2,11) radlat(I), radlon(I)
    END DO
        DO I = 1, 66600
        READ (3,11) fulllat(I), fulllon(I)
    END DO
    DO J = 1, 152
        READ (4,*) (nerad(J,K), K = 1, 175)
    END DO
    DO J = 1, 200
        READ (4,*) (marad(J,K), K = 1, 200)
    END DO
!
! * Read in full radar
!
    J = 1
    K = 1
    out = 0
    DO I = 1, 66600
        IF (out == 0) THEN
            fullrad(I) = nerad(J,K)
          IF (K == 175) THEN
            IF (J == 152) THEN
                  out = 1
                J = 1
                K = 1
            ELSE
                  J = J + 1
                  K = 1
            END IF
          ELSE
              K = K + 1
          END IF
      ELSE IF (out == 1) THEN
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
          fullrad(I) = marad(J,K)
          IF (K == 200) THEN
              J = J + 1
              K = 1
          ELSE
              K = K + 1
          END IF
      END IF
    END DO
!
! * Take out edges of radar
!
    J = 1
    DO I = 1, 66600
        IF ((fulllat(I) == radlat(J)) .AND. (fulllon(I) ==
            radlon(J))) THEN
          rad(J) = fullrad(I)
          J = J + 1
      END IF
    END DO
!
! * Calculate distance between points
!
    out2 = 0
 21 DO I = 1, length
        DO J = 1, length
            dist(I,J) = ((lat(I) - lat(J))**2 + (lon(I) -
                        lon(J))**2)**0.5
      END DO
    END DO
!
! ******************
! * Crossvalidation *
! ******************
!
    DO radius = 1, radit
        DO expon = 1, 6
          mse = 0.0
          stout = 0.0
!
! * Interpolate Error *
!
            DO I = 1, length
                DO J = 1, length
                IF (gauge(I) == 9999.0) THEN
                    stout = stout + 1.0
                    GO TO 22
                END IF
                IF ((dist(I,J) <= (radius/10.0)) .AND. (J /= I) .AND.
                   (gauge(J) /= 9999.0)) THEN
                        top = top + (err(J)/dist(I,J)**(expon/2.0))
                        bottom = bottom + (1/dist(I,J)**(expon/2.0))
                        hit = 1.0
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
                    END IF
                END DO
                IF (hit == 0.0) THEN
                    bottom = 1
                END IF
              esterr(I) = top/bottom
!
! * Zero filter
!
              IF ((radar(I) == 0.0) .AND. (gauge(I) == 0.0)) THEN
                  esterr(I) = 0.0
              ELSE IF ((radar(I) + esterr(I)) < 0.0) THEN
                  esterr(I) = -radar(I)
              END IF
!
! * Calculate MSE for Error Only
!
              mse = mse + (err(I) - esterr(I))**2
              top = 0.0
              bottom = 0.0
              hit = 0.0
 22             CONTINUE
          END DO
          mse = mse / (length - stout)
!
! * Find Minimum MSE
!
            IF (mse < min_mse) THEN
              min_mse = mse
            min_rad = radius/10.0
            min_exp = expon/2.0
          END IF
       END DO
    END DO
!
! *****************
! * Interpolation *
! *****************
!
! * Calculate distances
!
    DO I = 1, 36610
        DO J = 1, length
            dist(I,J) = ((radlat(I) - lat(J))**2 + (radlon(I) -
                        lon(J))**2)**0.5
      END DO
    END DO
!
! * Interpolate point
!
    DO I = 1, 36610
      top = 0.0
      bottom = 0.0
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
      hit = 0.0
        DO J = 1, length
            IF ((dist(I,J) <= min_rad) .AND. (gauge(J) /= 9999.0))
                THEN
                top = top + (err(J)/dist(I,J)**min_exp)
                bottom = bottom + (1/dist(I,J)**min_exp)
                hit = 1.0
            END IF
        END DO
        IF (hit == 0.0) THEN
            bottom = 1
        END IF
        esterr(I) = top/bottom
!
! * Calculate rainfall at each radar point
!
        IF (rad(I) < 0.0) THEN
          est_p(I) = rad(I)
          GO TO 23
        ELSE IF ((rad(I) + esterr(I)) < 0.0) THEN
            esterr(I) = -rad(I)
        END IF
        est_p(I) = rad(I) + esterr(I)
 23     CONTINUE
    END DO
!
! * Pos/Neg method
!
    DO J = 1, length
        cnt = 0.0
      hit = 0.0
      DO I = 1, 36610
          IF ((dist(I,J) <= 0.1) .AND. (dist(I,J) /= 0.0)) THEN
              cnt = cnt + 1.0
              diff = err(J) - esterr(I)
              IF (diff > diffval) THEN
                  hit = hit + 1.0
              ELSE IF (diff < -diffval) THEN
                  hit = hit - 1.0
              END IF
          END IF
      END DO
      avg = hit/cnt
          IF (abs(avg) >= 0.75) THEN
            gauge(J) = 9999.0
          END IF
    END DO
    badgauge = 0.0
    DO I = 1, length
        IF (gauge(I) == 9999.0) THEN
          badgauge = badgauge + 1.0
        END IF
    END DO
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
!
! * Iterate difference value
!
    IF (out2 == 3) THEN
        GO TO 24
    ELSE
        IF ((badgauge/length) >= 0.1) THEN
            GO TO 24
        END IF
        diffval = diffval - 0.1
        min_mse = 500.0
        out2 = out2 + 1
        GO TO 21
    END IF
!
! * Interpolate point without bad points
!
 24 DO I = 1, length
        IF (gauge(I) /= 9999.0) THEN
            WRITE (5,13), stn(I), lat(I), lon(I), gauge(I), radar(I),
                              err(I)
        END IF
    END DO
    DO I = 1, 36610
        hit = 0.0
        top = 0.0
        bottom = 0.0
        DO J = 1, length
            IF ((dist(I,J) <= min_rad) .AND. (gauge(J) /= 9999.0))
                  THEN
                   top = top + (err(J)/dist(I,J)**min_exp)
                   bottom = bottom + (1/dist(I,J)**min_exp)
                   hit = 1.0
            END IF
        END DO
        IF (hit == 0.0) THEN
            bottom = 1
        END IF
        esterr(I) = top/bottom
!
! * Calculate rainfall at each radar point
!
        IF (rad(I) < 0.0) THEN
          est_p(I) = rad(I)
          GO TO 25
        ELSE IF ((rad(I) + esterr(I)) < 0.0) THEN
            esterr(I) = -rad(I)
        END IF
        est_p(I) = rad(I) + esterr(I)
 25     CONTINUE
    END DO
```

Appendix A (Continued): Inverse Distance Weighting Computer Programming Codes

```
!
! ******************
! * Output Section *
! ******************
!
    DO I = 1, 36610
        IF (est_p(I) /= -9999.0) THEN
            WRITE (6,12), radlat(I), radlon(I), est_p(I)
      END IF
    END DO
!
! * Close files
!
    DO I = 1, 6
        CLOSE (I)
    END DO
END PROGRAM inversedist
```

Appendix B: Multiquadric Interpolation Computer Programming Codes in Python

```
#! /usr/bin/env python

import sys
import struct
import string
from MQroutinesLoop import *

### MQxval : Program to perform cross validation on Multiquadric
###          Interpolation
### Calls routines: InterpMQxval (imported from MQroutines.py)
### Written by Brian Belcher
### Revised by Eric Ware
###

##################################################
###### User Specs ################################
##################################################
miss1=-9999.00      # Specifies missing value for grid
miss2=-99.99 # Specifies missing value for station
wTxt=1       # Write interpolated output to text file (1=yes, 0=no)
#radius=1.25 # Uses points within radius of influence (degrees)
radiusList=[13.0]
lambdaList=[50.0,55.0,60.0,65.0,70.0,75.0,80.0]

### Specify area where interpolation will take place ###
runArea=[miss1,miss1,miss1,miss1] # interpolate to all available points

### Specify filenames and path to files ###
# NOTE: grid latitude, longitude and val files are assumed
#      written to files in same order.
filePath='/home/ecw25/'
xFile='newtot.txt'  # name of file containing lat/lon/vals
                    # each record:  lat lon val  (space delimited)
##################################################
#### End User Specs ##############################
##################################################

############## Open lat/lon/val file
ifile4 = open(filePath+xFile,'r')

############### read lat/lon/val
xLonList=[]; xLatList=[]; gauge=[]; radar=[]; xValList=[]
while 1:
  line1=ifile4.readline()        # read data from text file
  if len(line1)==0:break
  info=string.split(line1)
  xLatList.append(float(info[3]))
  xLonList.append(float(info[4]))
  gauge.append(float(info[5]))
  radar.append(float(info[6]))
  xValList.append(float(info[7]))

if (len(xValList)!=len(xLatList)) or (len(xValList)!=len(xLonList)) or \
  (len(xLatList)!=len(xLonList)): print 'Warning: Lists not equal in length'
ifile4.close()
############### end read Stn

mse={}
for radius in radiusList:
```

Appendix B (continued): Multiquadric Interpolation Computer Programming Codes in Python

```
  mse[radius]={}
  for lam in lambdaList:
    print 'Perform MQ interpolation ... radius=',radius,'   lambda=',lam
    ToValList=InterpMQxval(xLatList,xLonList,\
      xLatList,xLonList,xValList,radius,lam,runArea,miss1,miss2)
#
#  Prevent Negative Values
#
    for vals in range(len(ToValList)):
      if (ToValList[vals][2]>50) or (ToValList[vals][2]<-50):
        ToValList[vals]=(ToValList[vals][0],ToValList[vals][1],0.0)
      elif (radar[vals]==0) and (gauge[vals]==0):
        ToValList[vals]=(ToValList[vals][0],ToValList[vals][1],0.0)
      elif ((radar[vals]+ToValList[vals][2])<0):
        ToValList[vals]=(ToValList[vals][0],ToValList[vals][1], \
          -radar[vals])
#
# Calculate MSE
#
    mse_list=[]
      for vals in range(len(ToValList)):
        mse_list.append((ToValList[vals][2]-xValList[vals])**2)
      mse[radius][lam]=sum(mse_list)/len(mse_list)
#
# Write MSEs to file
#
if wTxt:
  oFilename='MQxval.txt'
  ofile1 = open(oFilename,'w')
  print 'Writing to output file ... ', oFilename

  headers = []
  for radius in mse.keys():
    headers.append(radius)
  headers.sort()
  ofile1.write ('       ')
  for lam in lambdaList:
    ofile1.write ('%4.1f    '  % (lam))
  ofile1.write ('\n')
  for radius in radiusList:
    ofile1.write ('%4.2f  ' % (radius))
    for lam in lambdaList:
      ofile1.write('%6.4f  ' % (mse[radius][lam]))
    ofile1.write('\n')
  ofile1.close()

### MQroutinesLoop.py ###

import sys
import os
import string
import struct
import math
import Numeric
import LinearAlgebra

def MQroutine(ltUnknown,lnUnknown,ltKnownList,lnKnownList,\
  ValKnownList,LAMBDA):
```

Appendix B (continued): Multiquadric Interpolation Computer Programming Codes in Python

```
### The following code follows the methods of
### Multiquadric Interpolation given by Nuss and
### Titley (MWR 1994).
###
### input: lat/lon of station with missing data
###       lat/lon/val of points used for interpolation to estimate missing
###       ltUnknown: latitude of point interpolating to
###       lnUnknown: longitude of point interpolating to
###       ltKnownList:     list of latitudes of points with known values
###                 used for interpolation
###       lnKnownList:     list of longitudes of points with known values
###                 used for interpolation
###       ValKnownList:list of known values at points used for interpolation
###
### output: interpolated value at station
###       ValUnknown:      Estimated value at point of interest
###           (value interpolated to point (ltUnknown,lnUnknown))
###
### USAGE in python:
###    from MQ_interp_routine import *
###    val = MQ_INTERP(lat,lon,latList,lonList,valList)
###
###
### Example python session:
### Python 2.3 (#1, Sep 13 2003, 00:49:11)
### [GCC 3.3 20030304 (Apple Computer, Inc. build 1495)] on darwin
### Type "help", "copyright", "credits" or "license" for more information.
### >>> from MQ_interp_routine import *
### >>> latK=[42.0,43.0,44.0]; lonK=[-78.0,-77.0,-76.0]
### >>> valK=[1,2,3]
### >>> lat=42.6
### >>> lon=-77.3
### >>> val=MQ_INTERP(lat,lon,latK,lonK,valK)
### >>> val
### 1.6530458150801886

# ERRM: the mean error value for the variable being
#      analyzed. Results are not too sensitive to
#      this parameter, but results must be in the ballpark
# C:   Multiquadric Parameter (arbitrary small constant
#      making basis function infinitely differentiable)
# IX,JY:Dimensions of grid
# nobs:       number of observations (stations)
# XC:  x-coordinate of the observation (XC[nobs])
# YC:  y-coordinate of the observation (YC[nobs])
# Hj:  Vector of observations

IX=1; JY=1    # Interpolate to one point at a time
nobs=len(ValKnownList)
ERRM=0.5      # ERRM for temperature
max_dim=max(IX,JY)
C = 0.0008 * max_dim
# Assign coordinates and values
XC=lnKnownList; YC=ltKnownList; dat=ValKnownList

# Fill the Qij matrix
Qij={}
for j in range(nobs):
```

Appendix B (continued): Multiquadric Interpolation Computer Programming Codes in Python

```
  Qij[j]=[]
  for i in range(nobs):
    Qij[j].append(-1.0*math.sqrt(((math.pow(math.fabs(XC[j]-XC[i]),2) +\
      pow(math.fabs(YC[j]-YC[i]),2))/(C*C))+1.0))

# Account for observational uncertainty
for j in range(nobs):
  i=j
  Qij[j][i]=Qij[j][i] + nobs*LAMBDA*ERRM

# manipulate into matrix form
for j in range(nobs):
  if j==0:
    Qij_matrix=Numeric.array([Qij[j]])
  else:
    temp=Numeric.array([Qij[j]])
    Qij_matrix=Numeric.concatenate([Qij_matrix,temp])

# Find the inverse of Qij (Qij_inv)
Qij_inv=LinearAlgebra.inverse(Qij_matrix)

# Fill the Qgi matrix
temp=[]
for i in range(nobs):
  temp.append(-1.0*math.sqrt(((math.pow(math.fabs(lnUnknown-XC[i]),2) +\
    pow(math.fabs(ltUnknown-YC[i]),2))/(C*C))+1.0))
Qgi_matrix=Numeric.array([temp])

# Multiply Qij_inv and Hj (determine ALPHAi)
Hj=Numeric.array([dat])
HjT=Numeric.transpose(Hj)
ALPHAi=Numeric.dot(Qij_inv,HjT)

# Multiply Qgi and ALPHAi (determine Hg)
Hg=Numeric.dot(Qgi_matrix,ALPHAi)

ValUnknown = Numeric.reshape(Hg,(IX,JY))[0][0]

return ValUnknown

def InterpMQ(ToLatList,ToLonList,FromLatList,FromLonList,FromValList,\
rad,lam,rArea,miss1,miss2):
# Horizontal interpolation using Multiquadric Interpolation
# Calls routine:    MQroutine (imported from MQroutines.py)
# INPUT ---
# ToLatList:    List of latitudes interpolating to
# ToLonList:    List of longitudes interpolating to
# FromLatList:  List of latitudes interpolating from
# FromLonList:  List of longitudes interpolating from
# FromValList:  List of values interpolating from
# rad:          Radius of influence
# lam:       LAMBDA used in MQroutine
# rArea:     Interpolate only to the points in this area
# miss1:     Missing value specified for grid
# miss2:     Missing value specified for station
# OUTPUT ---
# ToValList:    List of values resulting from interpolation
```

Appendix B (continued): Multiquadric Interpolation Computer Programming Codes in Python

```
from MQroutinesLoop import MQroutine

ToValList=[]
for i in range(len(ToLatList)):
  FromLat=[]; FromLon=[]; FromVal=[]
  if ToLatList[i]==miss1: continue
  if ToLonList[i]==miss1: continue
  if ToLatList[i]==miss2: continue
  if ToLonList[i]==miss2: continue

# See if this location is within area of interest
  if (miss1 in rArea) or (miss1 in rArea):
  # Interpolate to all available points
    pass
  else:
  # Interpolate to points within area of interest
    if (ToLatList[i]>=rArea[0] and ToLatList[i]<=rArea[1]) and \
      (ToLonList[i]>=rArea[2] and ToLonList[i]<=rArea[3]):
        pass
    else:
        continue

  # Find locations within radius of influence
    for ii in range(len(FromValList)):
      if FromLatList[ii]==miss1: continue
      if FromLonList[ii]==miss1: continue
      if FromValList[ii]==miss1: continue
      if FromLatList[ii]==miss2: continue
      if FromLonList[ii]==miss2: continue
      if FromValList[ii]==miss2: continue
      if math.fabs(FromLatList[ii]-ToLatList[i])>rad: continue
      if math.fabs(FromLonList[ii]-ToLonList[i])>rad: continue
      FromLat.append(FromLatList[ii])
      FromLon.append(FromLonList[ii])
      FromVal.append(FromValList[ii])

  # Interpolate
    ToLat=ToLatList[i]; ToLon=ToLonList[i]
    try:
      ToVal=MQroutine(ToLat,ToLon,FromLat,FromLon,FromVal,lam)
    except:
      ToVal=miss1
    print '         Interpolate to  %10.3f %10.3f %8.2f' % \
      (ToLat,ToLon,ToVal)
    ToValList.append((ToLat,ToLon,ToVal))

  return ToValList

def InterpMQxval(ToLatList,ToLonList,FromLatList,FromLonList,FromValList,\
  rad,lam,rArea,miss1,miss2):
# Horizontal interpolation using Multiquadric Interpolation
# ###### For cross-validation only #############
# Calls routine:        MQroutine (imported from MQroutines.py)
# INPUT ---
# ToLatList:    List of latitudes interpolating to
# ToLonList:    List of longitudes interpolating to
# FromLatList:  List of latitudes interpolating from
# FromLonList:  List of longitudes interpolating from
```

Appendix B (continued): Multiquadric Interpolation Computer Programming Codes in Python

```
# FromValList:  List of values interpolating from
# rad:          Radius of influence
# rArea:        Interpolate only to the points in this area
# miss1:                Missing value specified for grid
# miss2:                Missing value specified for station
# OUTPUT ---
# ToValList:    List of values resulting from interpolation

  from MQroutinesLoop import MQroutine

  ToValList=[]
  for i in range(len(ToLatList)):
    FromLat=[]; FromLon=[]; FromVal=[]
    if ToLatList[i]==miss1: continue
    if ToLonList[i]==miss1: continue
    if ToLatList[i]==miss2: continue
    if ToLonList[i]==miss2: continue

    # See if this location is within area of interest
    if (miss1 in rArea) or (miss1 in rArea):
      # Interpolate to all available points
      pass
    else:
      # Interpolate to points within area of interest
      if (ToLatList[i]>=rArea[0] and ToLatList[i]<=rArea[1]) and \
         (ToLonList[i]>=rArea[2] and ToLonList[i]<=rArea[3]):
          pass
      else:
          continue

    # Find locations within radius of influence
    for ii in range(len(FromValList)):
      if FromLatList[ii]==miss1: continue
      if FromLonList[ii]==miss1: continue
      if FromValList[ii]==miss1: continue
      if FromLatList[ii]==miss2: continue
      if FromLonList[ii]==miss2: continue
      if FromValList[ii]==miss2: continue
      if (FromLatList[ii]==ToLatList[i]) and \
         (FromLonList[ii]==ToLonList[i]): continue
      if math.fabs(FromLatList[ii]-ToLatList[i])>rad: continue
      if math.fabs(FromLonList[ii]-ToLonList[i])>rad: continue
      FromLat.append(FromLatList[ii])
      FromLon.append(FromLonList[ii])
      FromVal.append(FromValList[ii])

    # Interpolate
    ToLat=ToLatList[i]; ToLon=ToLonList[i]
    try:
      ToVal=MQroutine(ToLat,ToLon,FromLat,FromLon,FromVal,lam)
    except:
      ToVal=miss1
    print '            Interpolate to  %10.3f %10.3f %8.2f' % \
      (ToLat,ToLon,ToVal)
    ToValList.append((ToLat,ToLon,ToVal))

  return ToValList
```

Appendix C: Kriging Computer Programming Codes in SPlus

```
#
# Kriging from precip data set
# Written by Pat Sullivan
# Revised by Eric Ware
#
attach(Apr0703tot)
module(spatial)
options(object.size=5e8)
varmodel = exp.vgram
covarmodel = exp.cov
#
# Plot the locations and values
#
scaled.plot(lon,lat,main="Locations",cex=0.75,pch=16)
#
# Use plot.geo to examine the data visually
#
plot.geo(data.frame(lon,lat,error),geotitle="Observed Error",dd=.05)
#
# Kriging data for Error
#
error.vario<-variogram(error~loc(lon,lat),method="robust")
#
# Plot empirical variogram for all data
#
plot(error.vario,pch=16,col=1,cex=1.2)
#
# Fit parameters
#
error.fit <- variogram.fit(
   error.vario,
   param=c(
      sill=0.1,
      range=0.5,
      nugget = 0.0),
   fun=varmodel)
#
# Save estimated parameters
#
err.sill <- error.fit$parameters["sill"]
err.range <- error.fit$parameters["range"]
err.nugget <- error.fit$parameters["nugget"]
#
# Show model estimated from the data in blue
#
lines(error.vario$distance,
   varmodel(
      error.vario$distance,
      sill=err.sill,
      range=err.range,
      nugget=err.nugget),
      lwd=4,col=2)
#
# Show parameter estimates on plot
#
text.x<-0.75*max(error.vario$distance)
text.y<-0.25*max(error.vario$gamma)
text(text.x,text.y,
   paste(
```

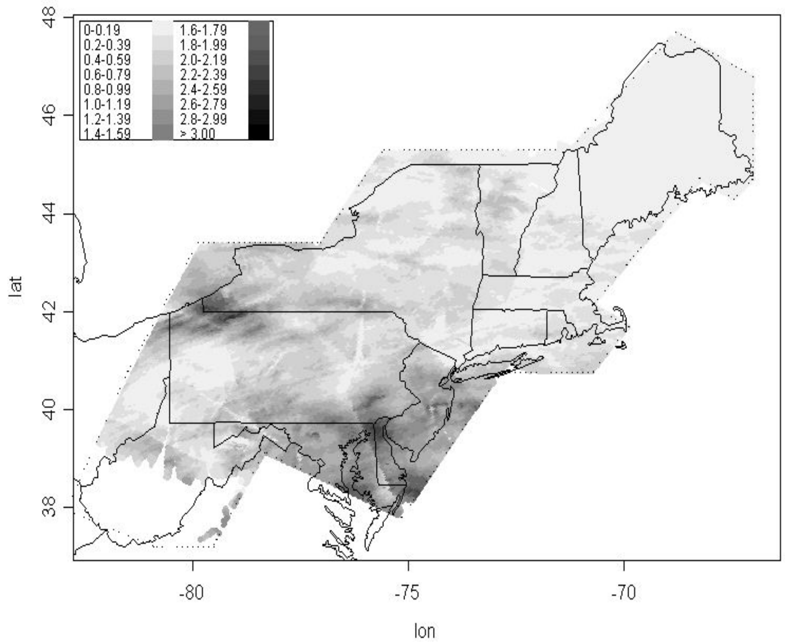
Appendix C (continued): Kriging Computer Programming Codes in SPlus

```
      "Sill        = ",round(err.sill,2),"\n",
      "Range  = ",round(err.range,2),"\n",
      "Nugget = ",round(err.nugget,2),
      sep=""),
   cex=1.2,adj=-1)
#
# Create a kriging object for prediction using estimated
# parameters.
#
error.krig<-krige(error~loc(lon,lat),
   covfun=covarmodel,
   sill=err.sill,
   range=err.range,
   nugget=err.nugget)
#
# cross-validation
#
errcross = NULL
id = seq(length(lon))
for(i in id)
{
   errcross.krig<-krige(error~loc(lon,lat),subset=id[-i],
   covfun=covarmodel,
   sill=err.sill,
   range=err.range,
   nugget=err.nugget)
   errcross.pred<-predict(errcross.krig,
      newdata=data.frame(lon=lon[i],lat=lat[i]))
#
# Zero filter
#
   if (gauge[i] == 0 && radar[i] == 0)
   {
      errcross.pred$fit = 0
   }
   else if ((radar[i] + errcross.pred$fit) < 0)
   {
      errcross.pred$fit = -radar[i]
   }
   errcross = rbind(errcross,errcross.pred)
}
#
# calculate mse
#
sqerr = (error - errcross$fit)^2
mse = sum(sqerr)/length(error)
#
# Plot estimated errors and estimated rainfall
#
plot.geo(data.frame(errcross$lon,errcross$lat,errcross$fit),
   geotitle="Error Fit",dd=0.05)
plot.geo(data.frame(errcross$lon,errcross$lat,(radar+errcross$fit)),
   geotitle="Rain Fit",dd=0.05)
#
# Print mse
#
mse
detach(Apr0703tot)
```

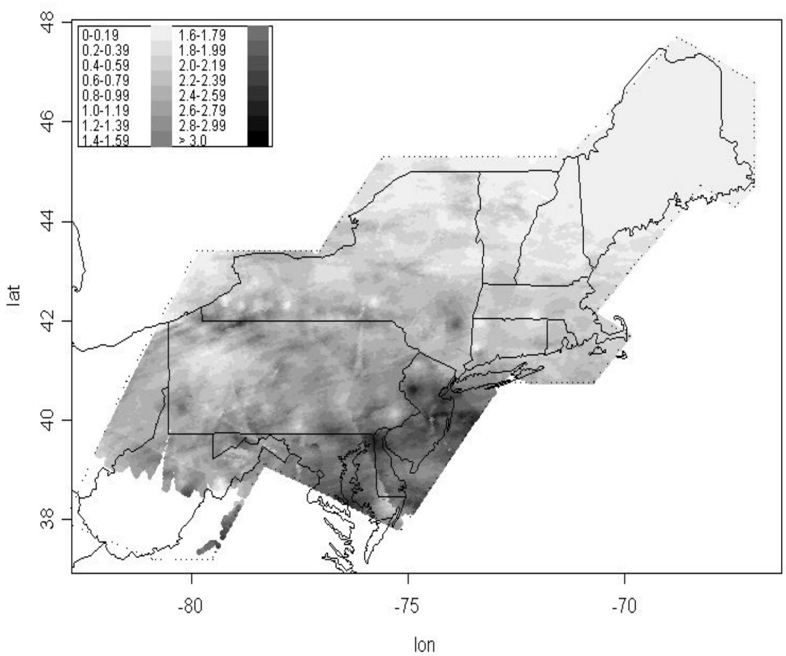
Appendix D: Before and After Graphs of All 30 Cases

Apr. 7, 2003

Initial Radar (cm)

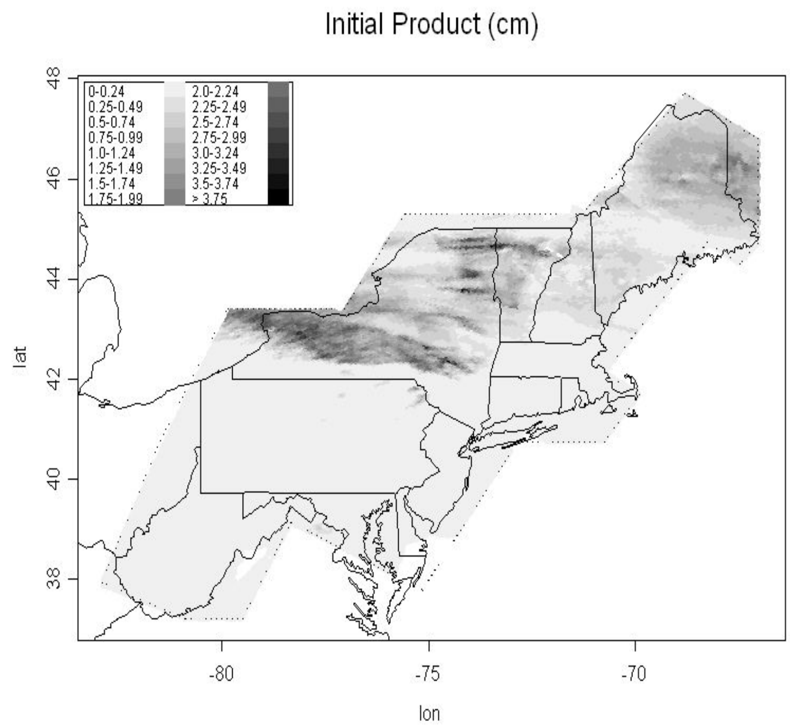


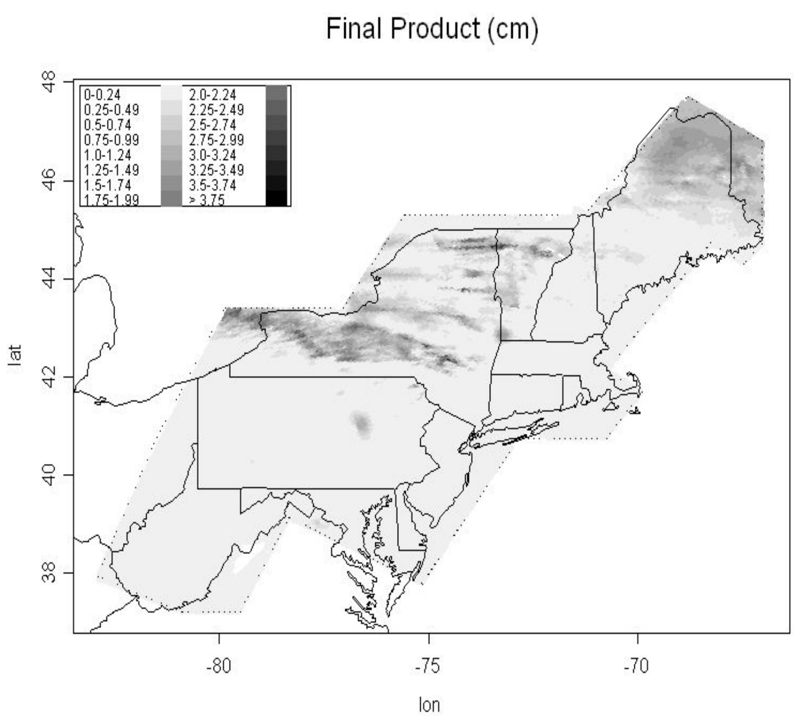
Final Product (cm)



Appendix D (continued): Before and After Graphs of All 30 Cases

Apr. 18, 2004

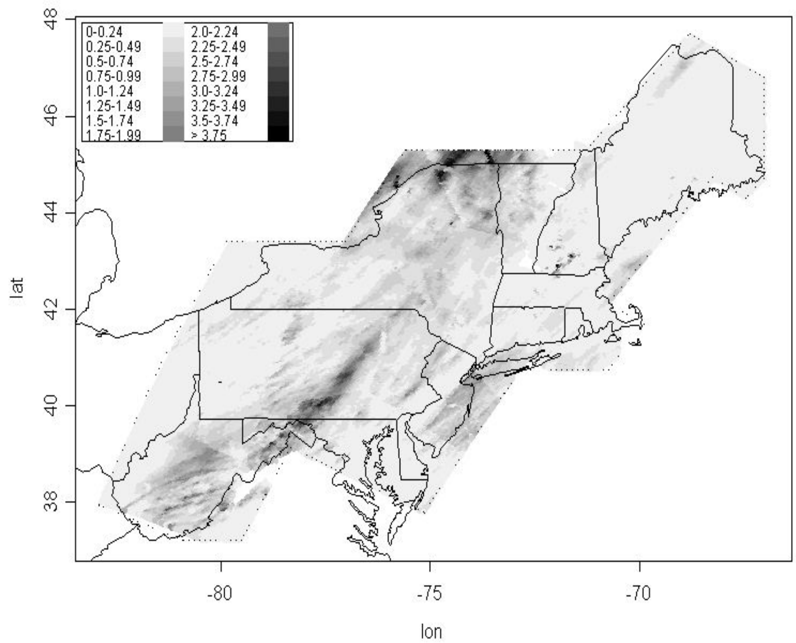




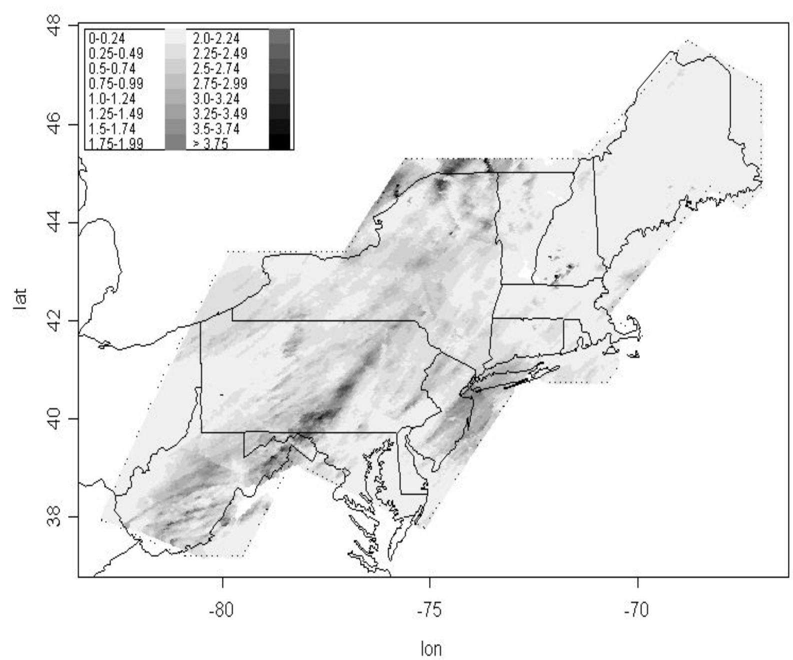
Appendix D (continued): Before and After Graphs of All 30 Cases

Apr. 21, 2003

Initial Product (cm)

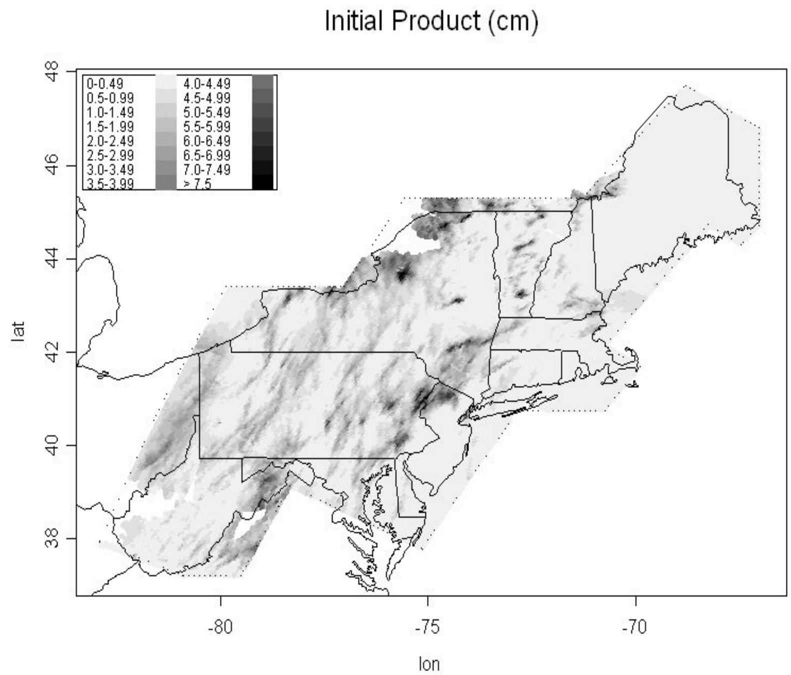


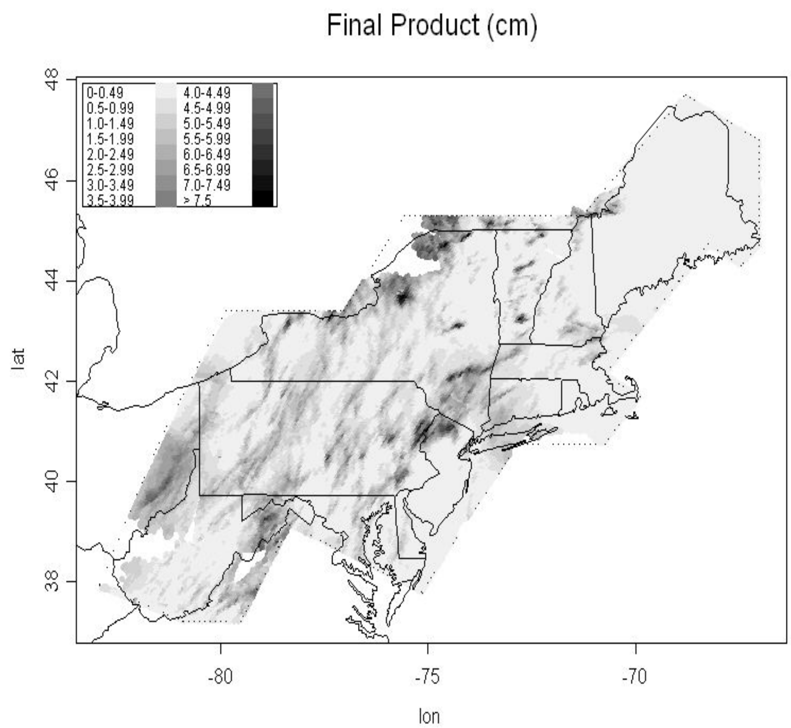
Final Product (cm)



Appendix D (continued): Before and After Graphs of All 30 Cases

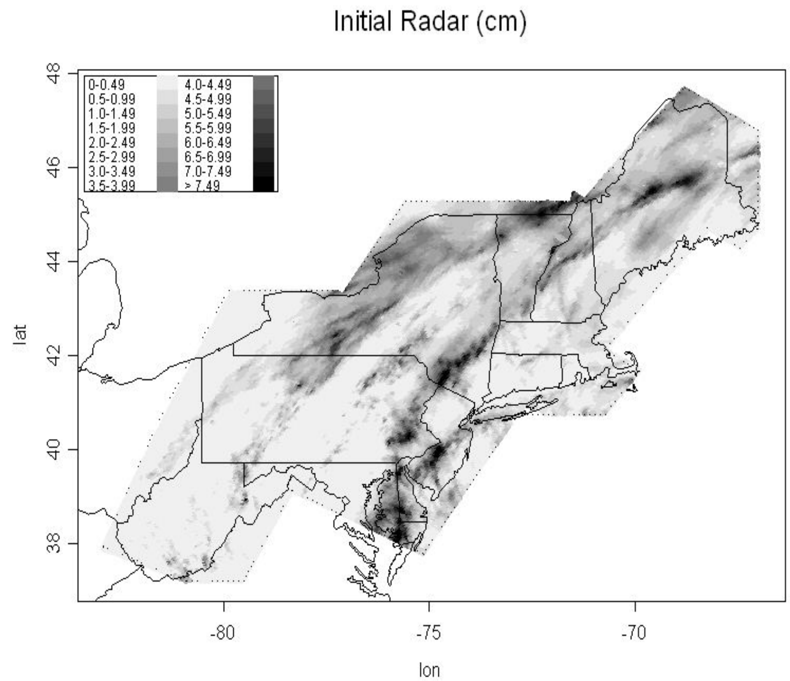
Aug. 3, 2003

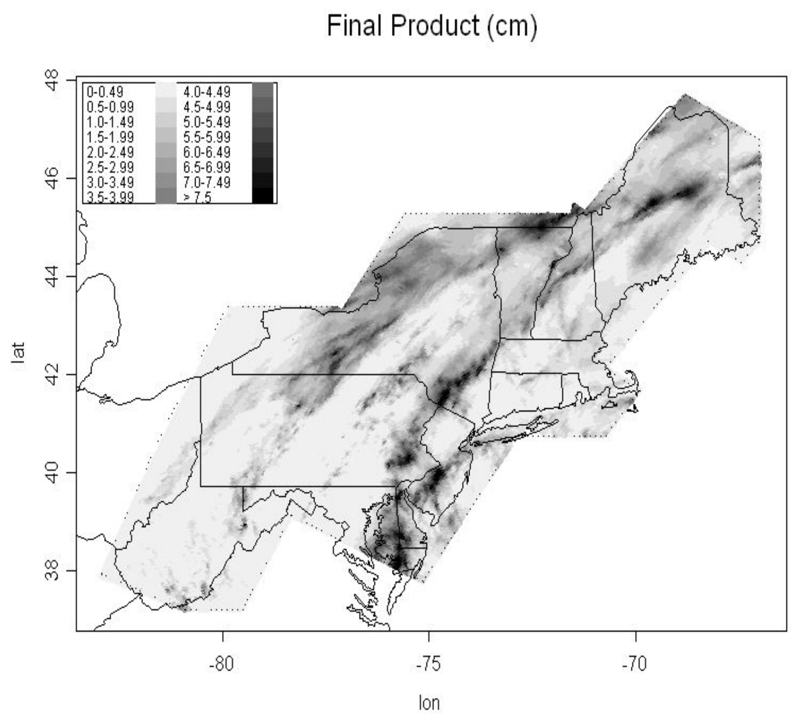




Appendix D (continued): Before and After Graphs of All 30 Cases

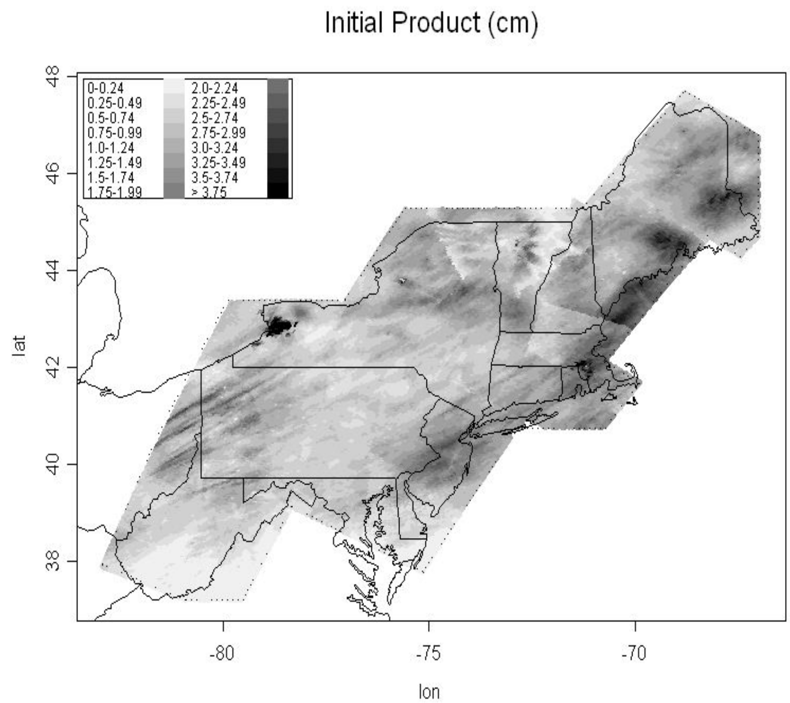
Aug. 30, 2004

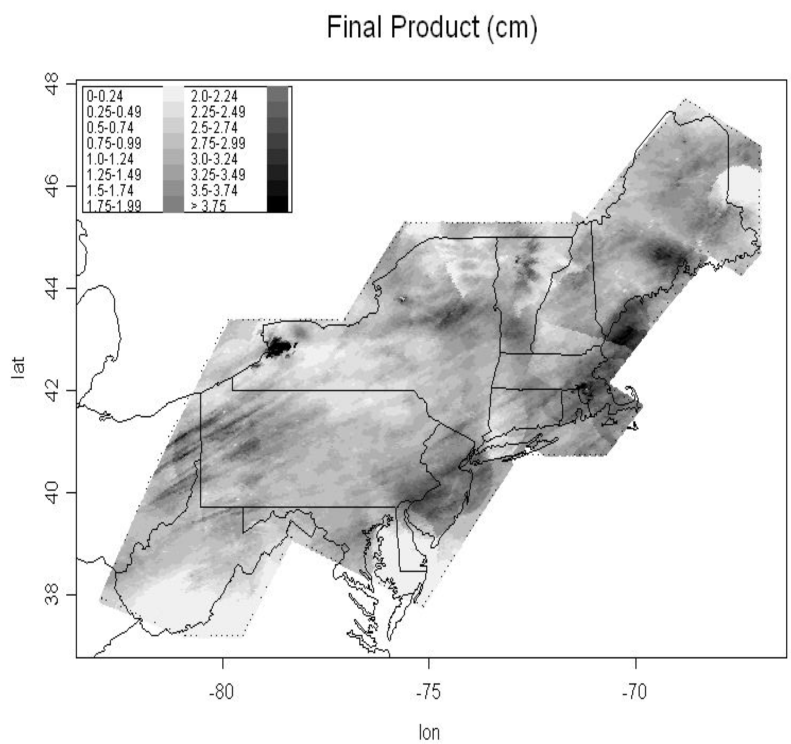




Appendix D (continued): Before and After Graphs of All 30 Cases

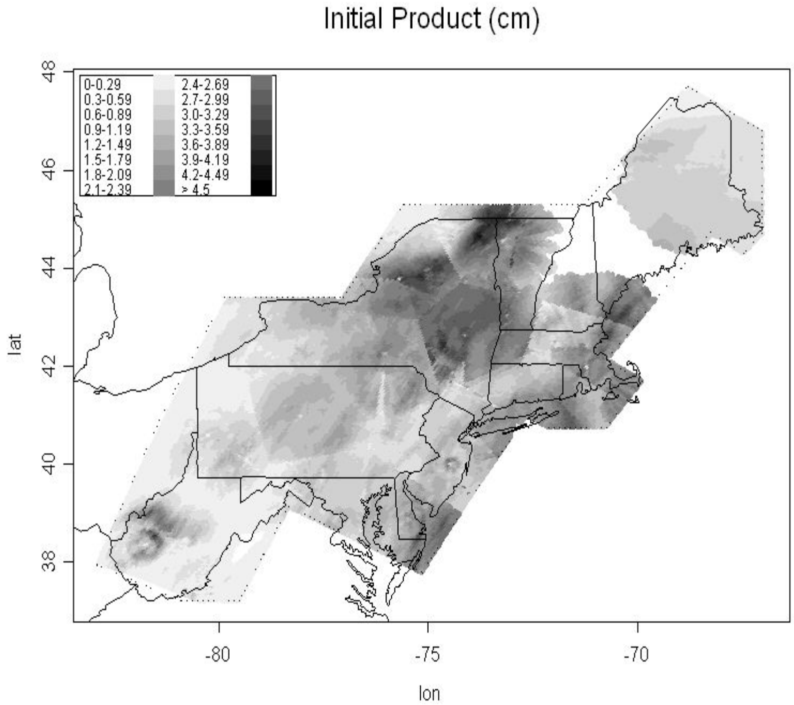
Dec. 7, 2004

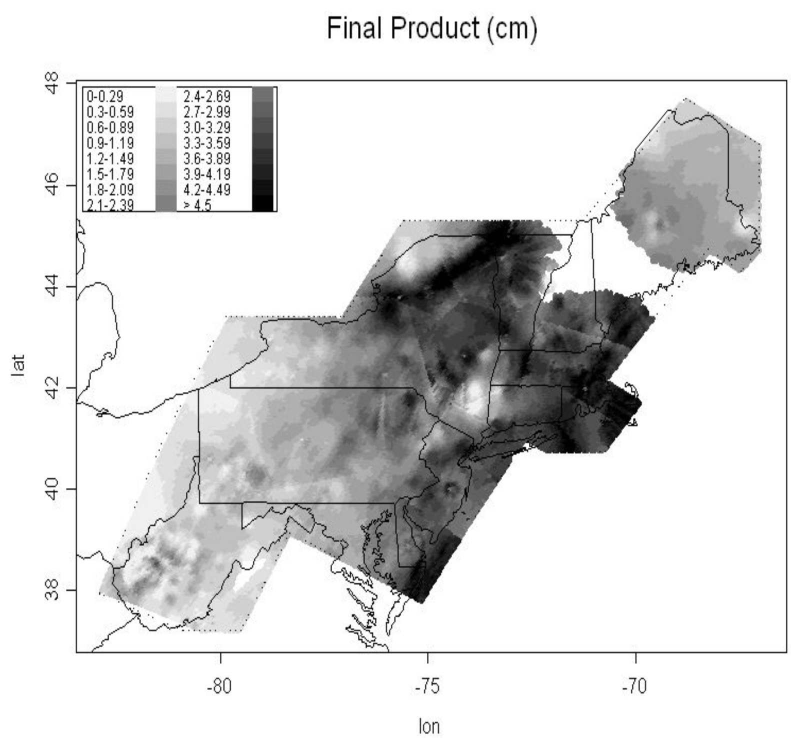




Appendix D (continued): Before and After Graphs of All 30 Cases

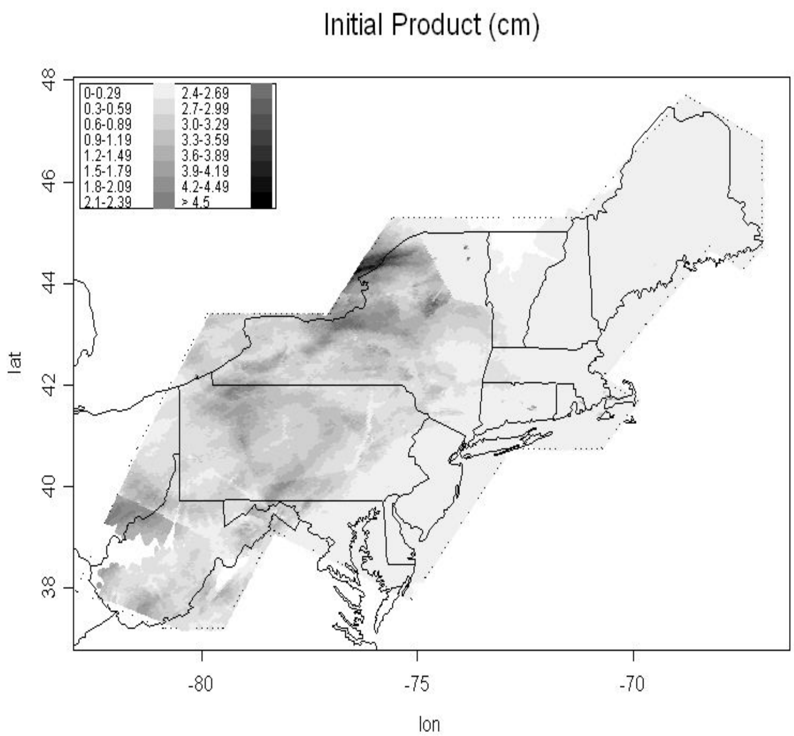
Dec. 14, 2003

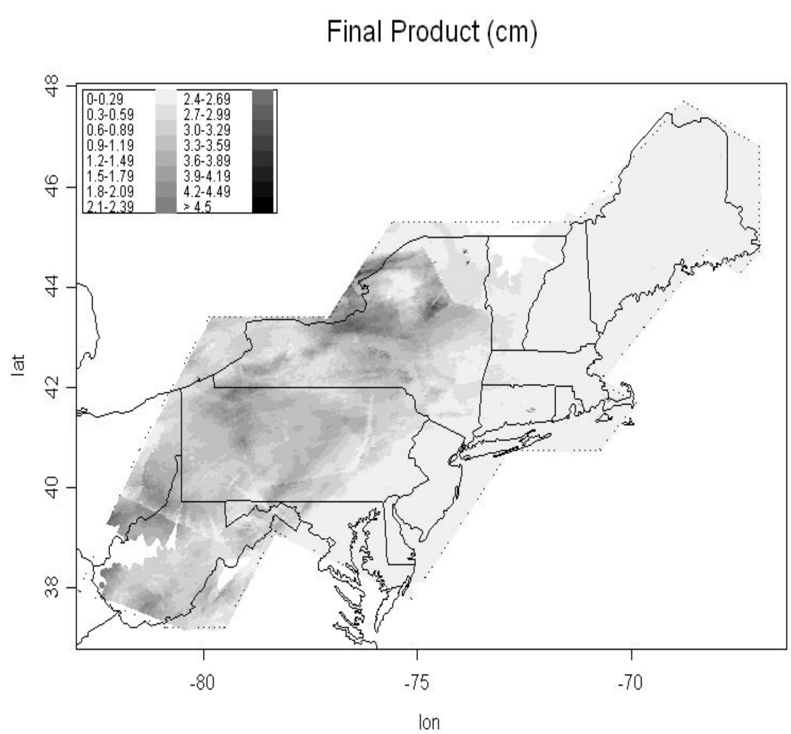




Appendix D (continued): Before and After Graphs of All 30 Cases

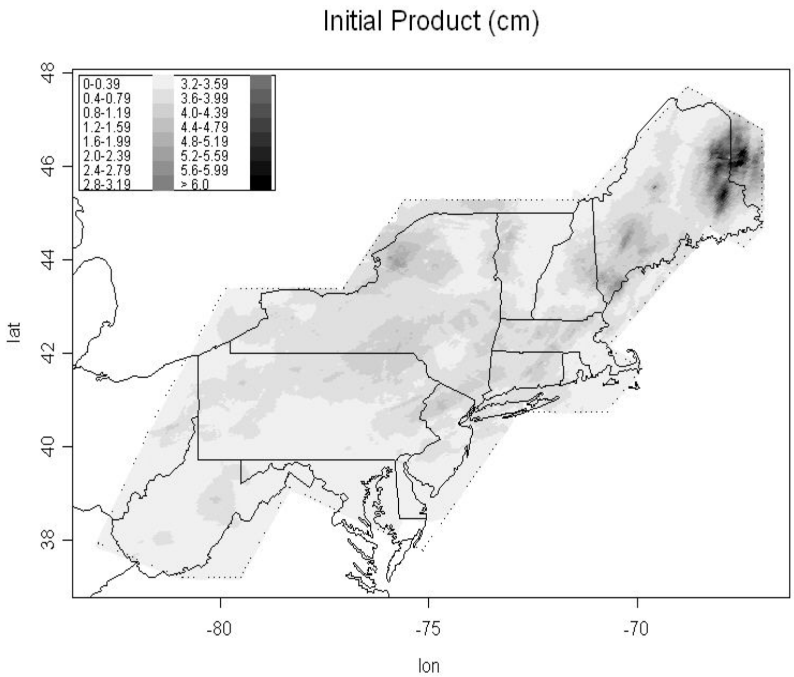
Feb. 3, 2003

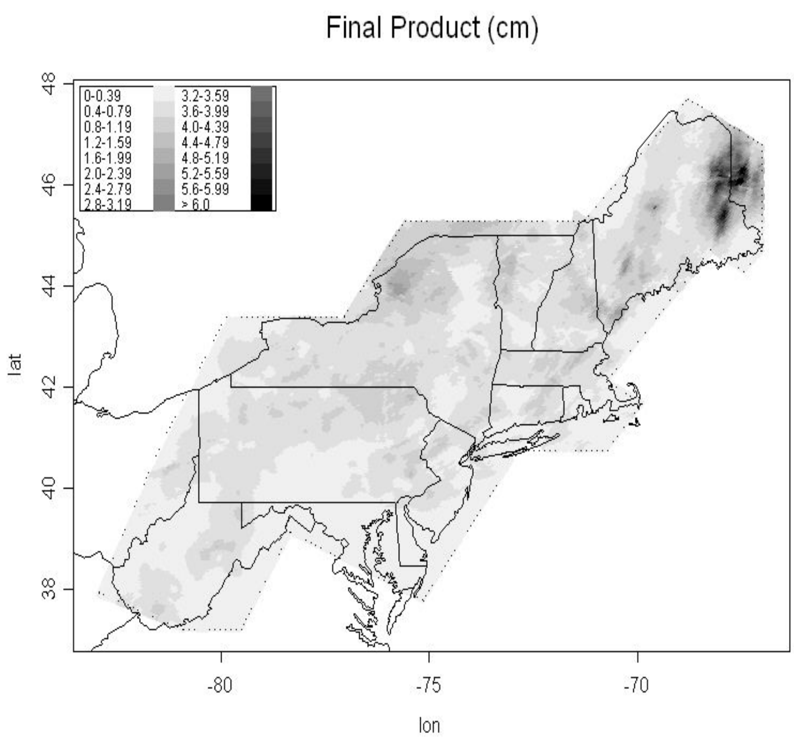




Appendix D (continued): Before and After Graphs of All 30 Cases

Feb. 16, 2005

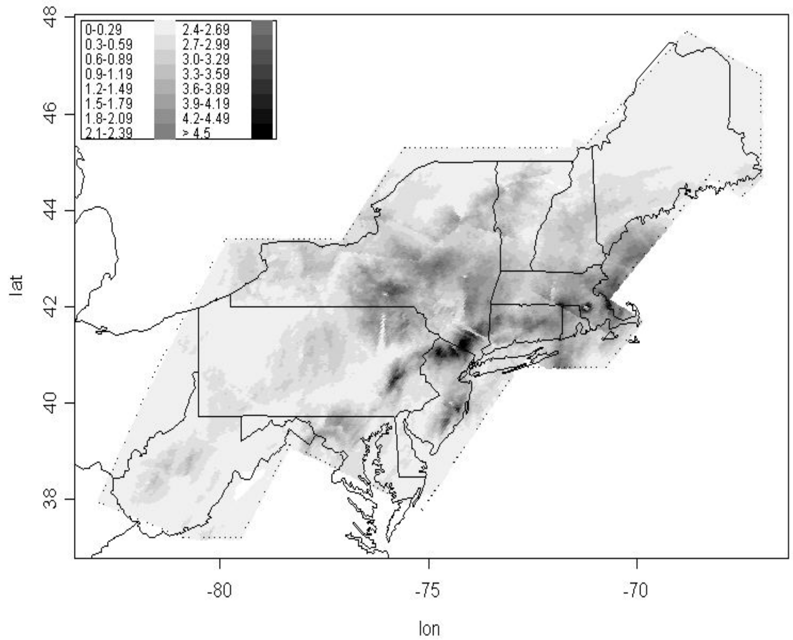




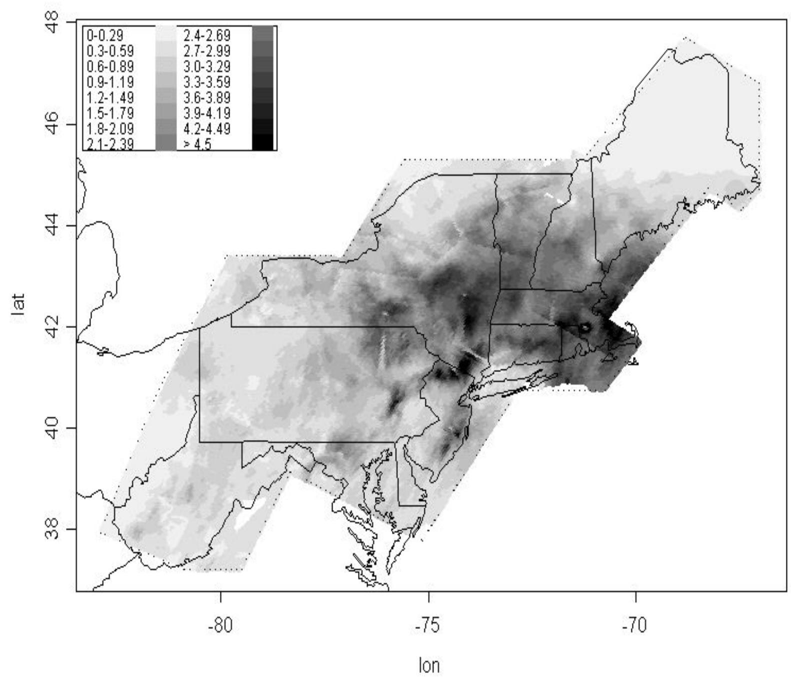
Appendix D (continued): Before and After Graphs of All 30 Cases

Jan. 3, 2003

Initial Product (cm)



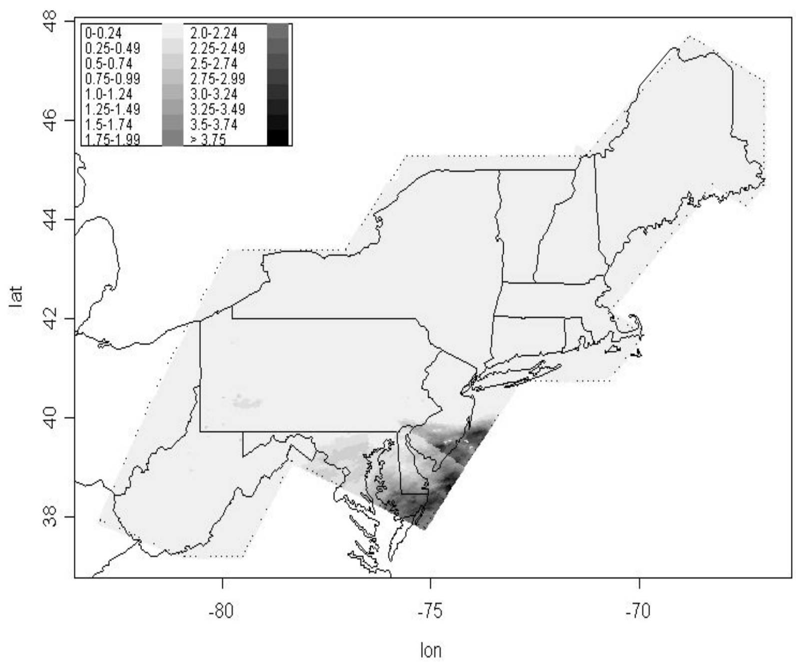
Final Product (cm)



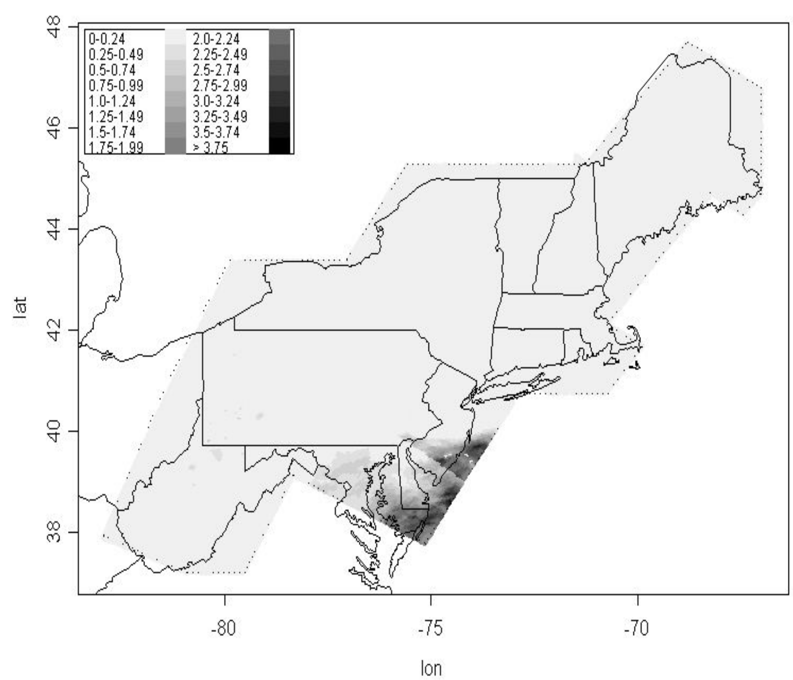
Appendix D (continued): Before and After Graphs of All 30 Cases

Jan. 30, 2005

Initial Product (cm)



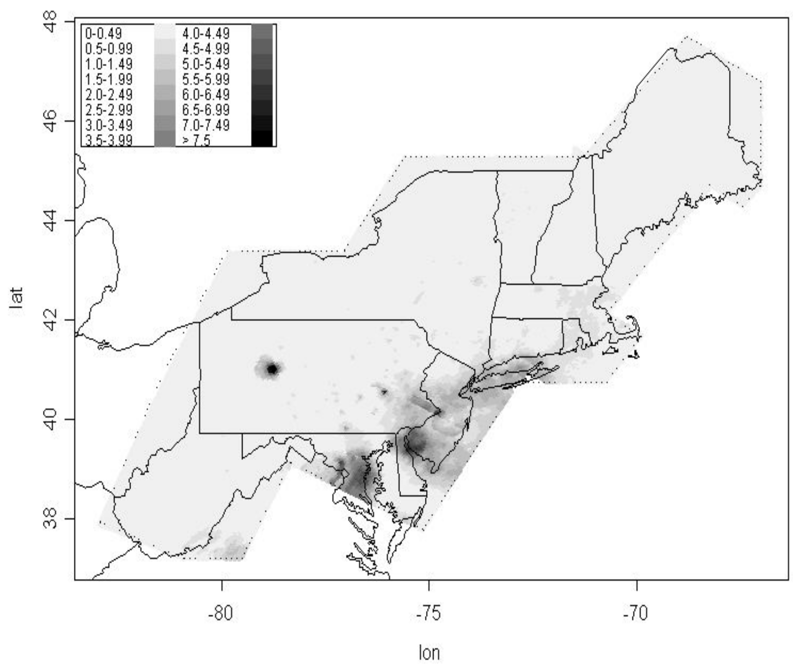
Final Product (cm)



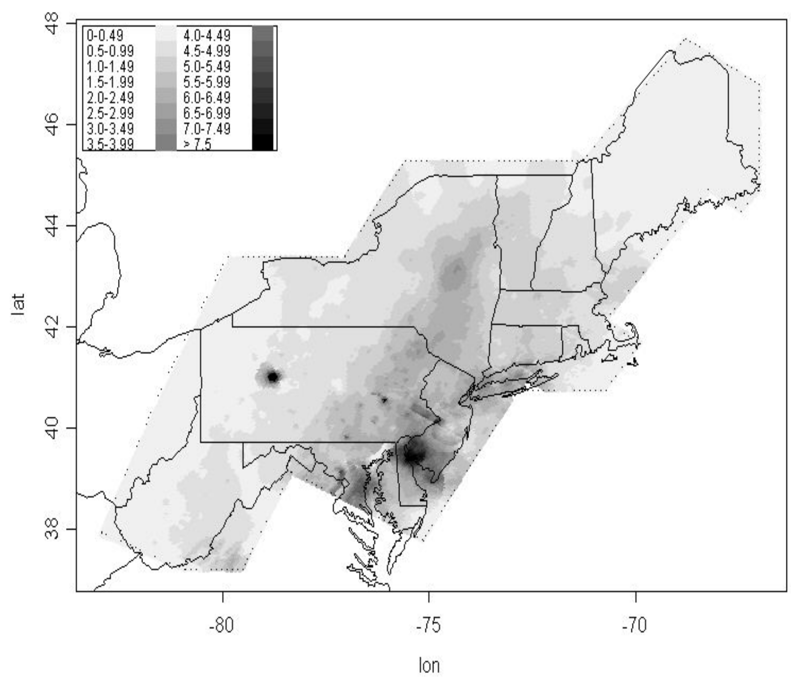
Appendix D (continued): Before and After Graphs of All 30 Cases

Jan. 31, 2000

Initial Product (cm)



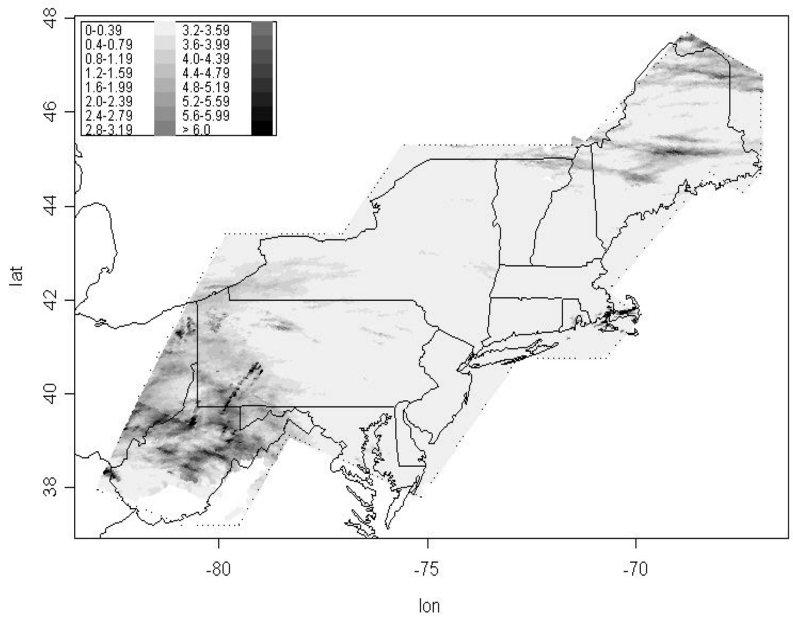
Final Product (cm)



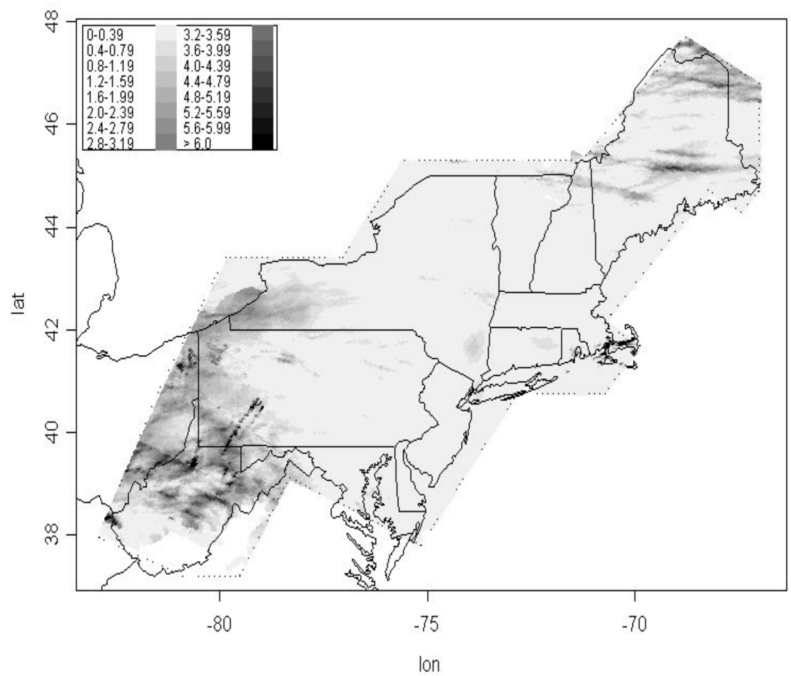
Appendix D (continued): Before and After Graphs of All 30 Cases

Jul. 8, 2003

Initial Product (cm)



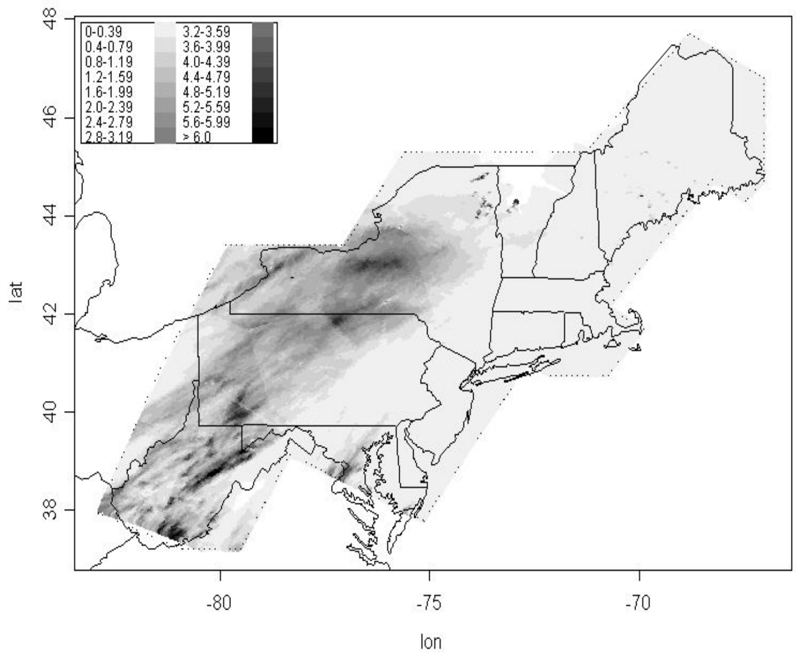
Final Product (cm)



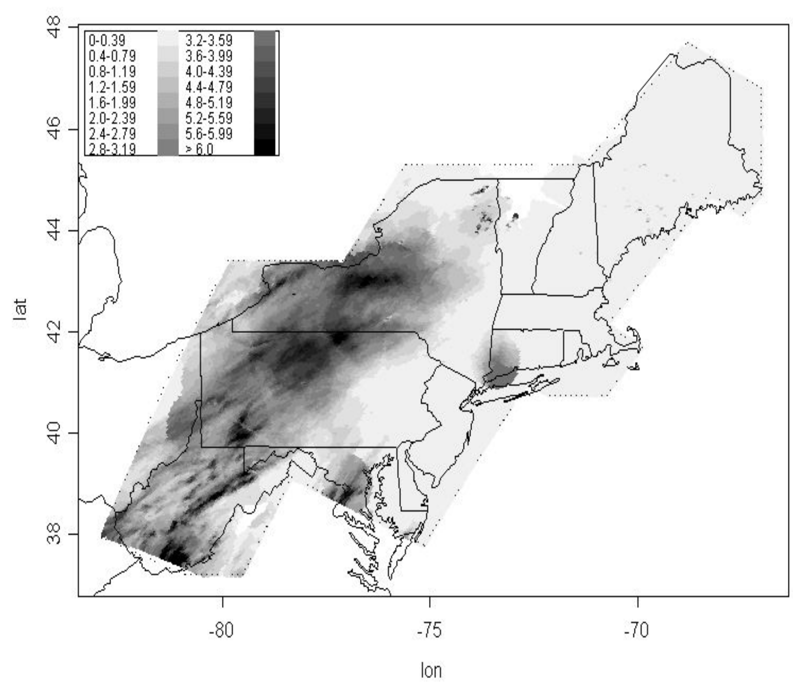
Appendix D (continued): Before and After Graphs of All 30 Cases

Jul. 26, 2004

Initial Product (cm)

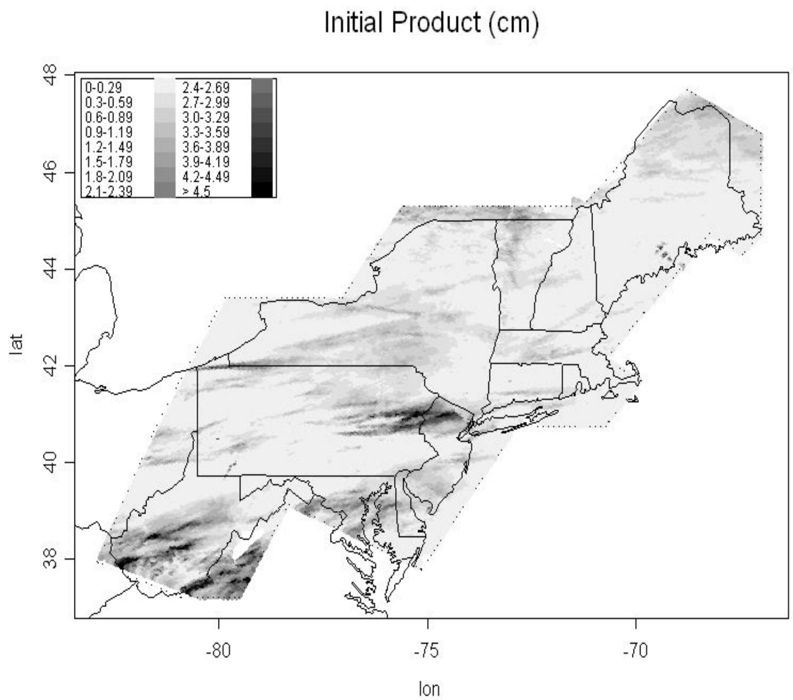


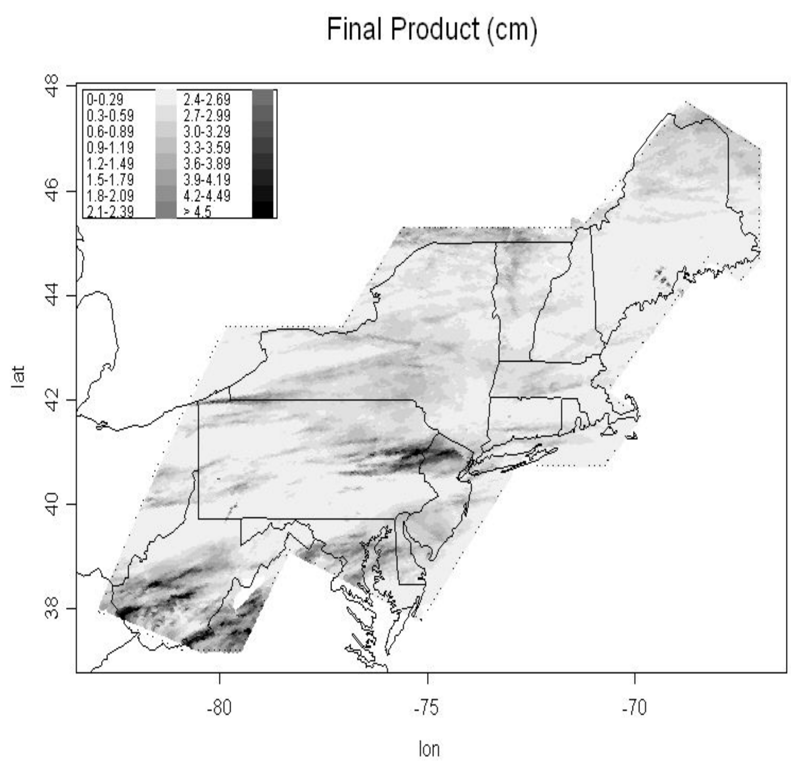
Final Product (cm)



Appendix D (continued): Before and After Graphs of All 30 Cases

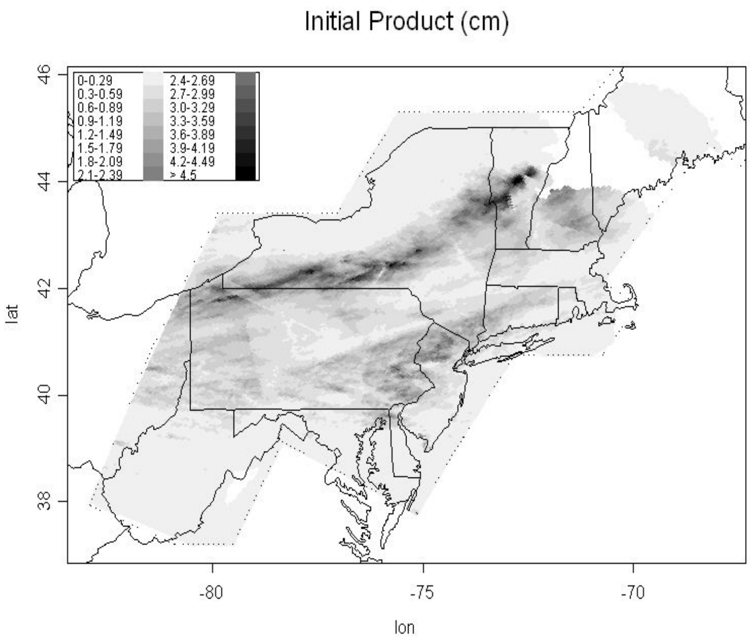
Jun. 11, 2003

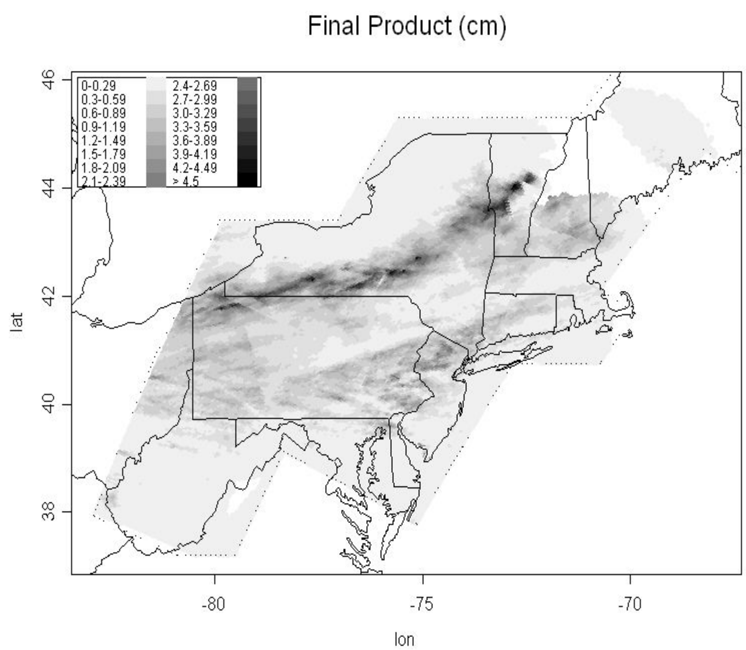




Appendix D (continued): Before and After Graphs of All 30 Cases

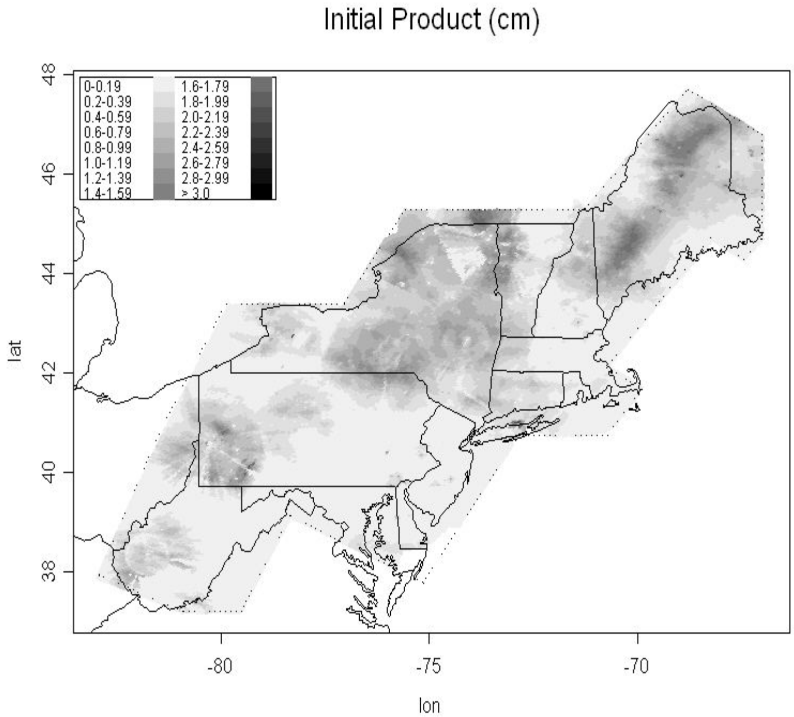
Jun. 28, 2004

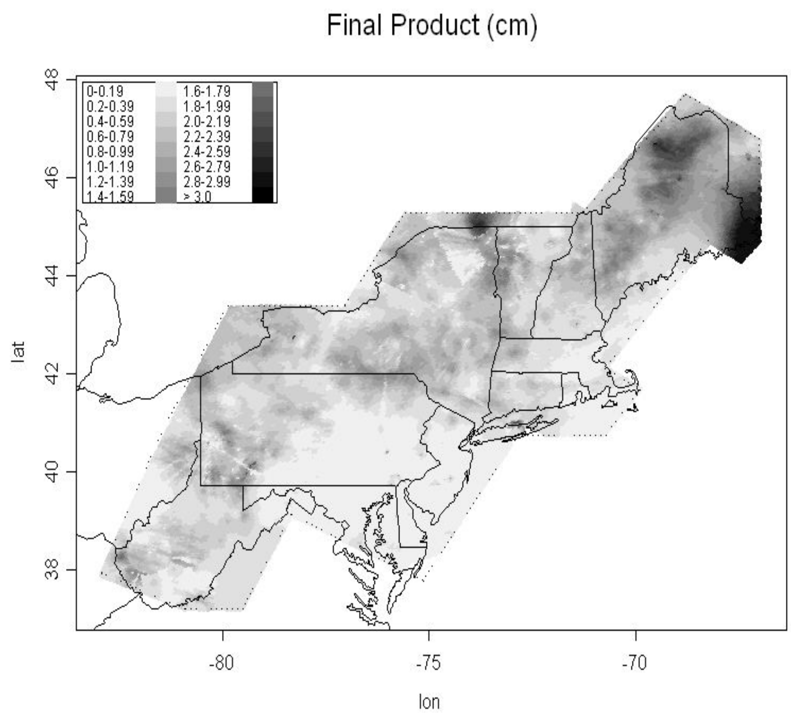




Appendix D (continued): Before and After Graphs of All 30 Cases

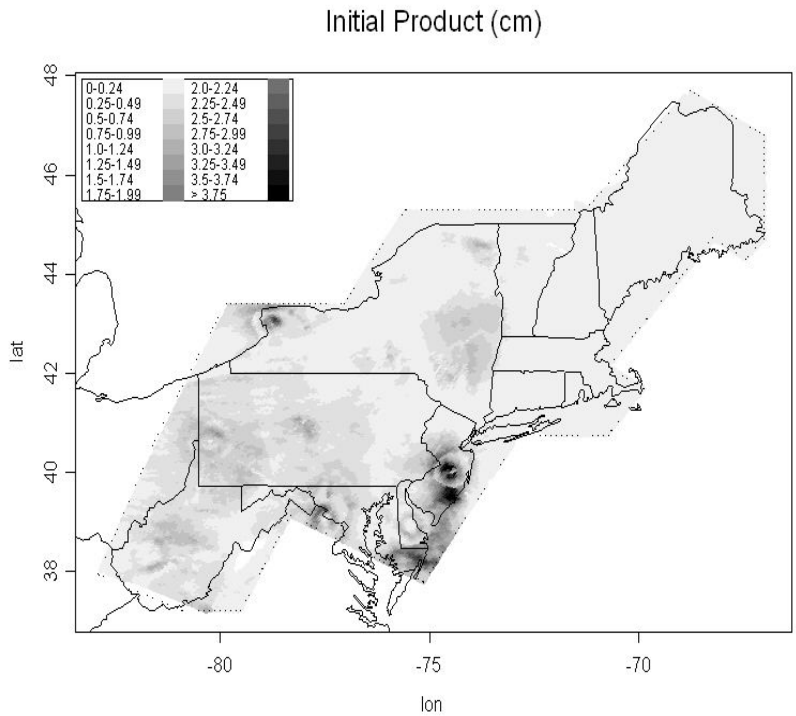
Mar. 1, 2005

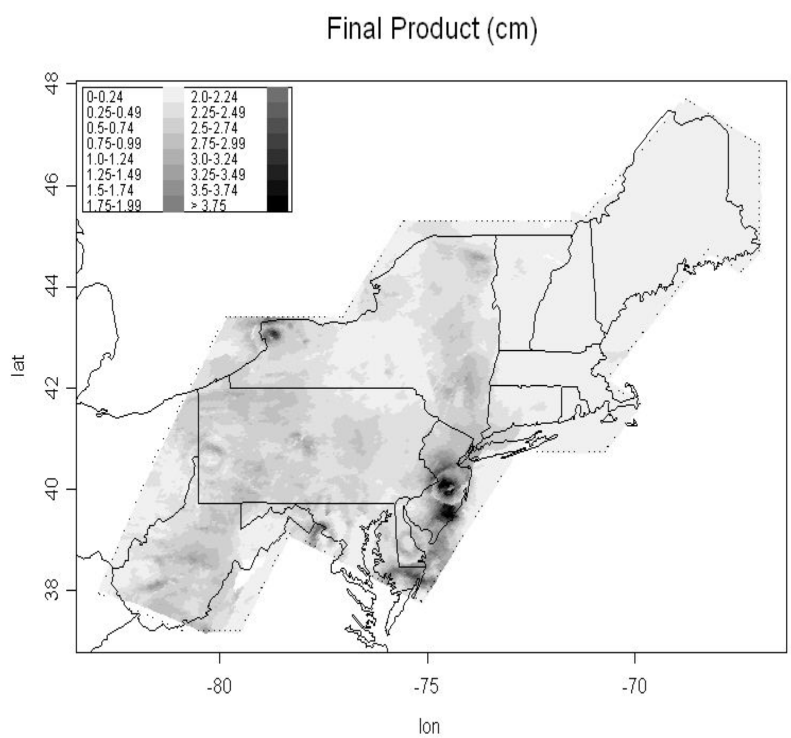




Appendix D (continued): Before and After Graphs of All 30 Cases

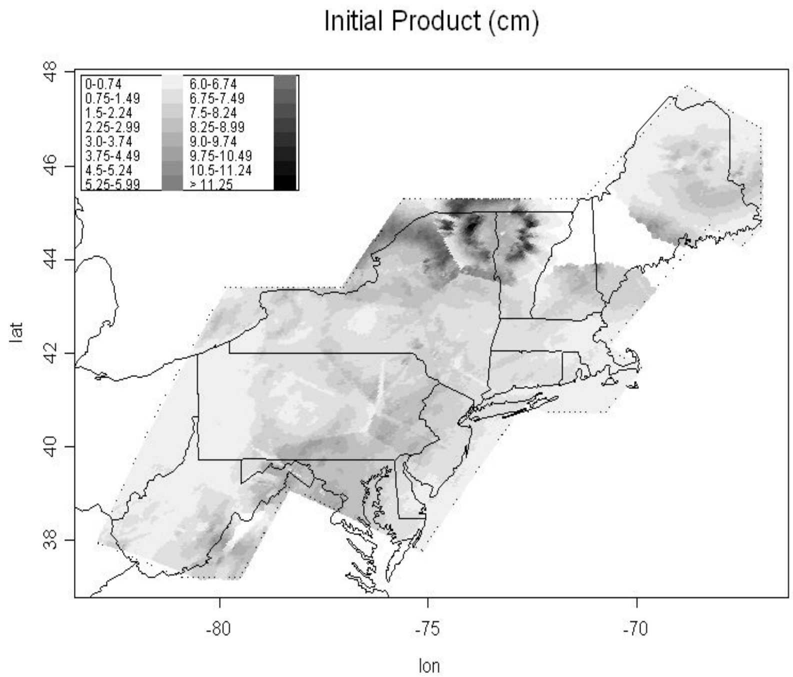
Mar. 7, 2004

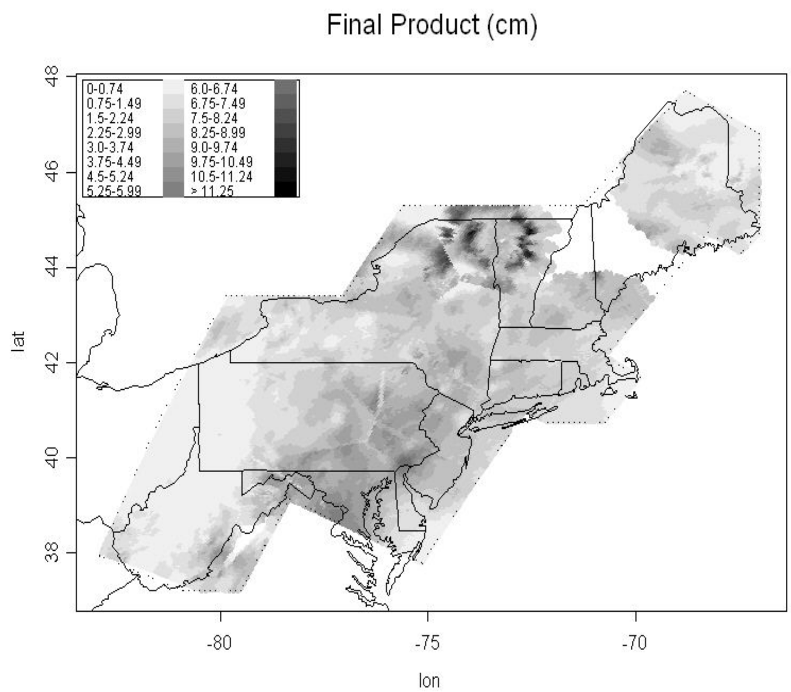




Appendix D (continued): Before and After Graphs of All 30 Cases

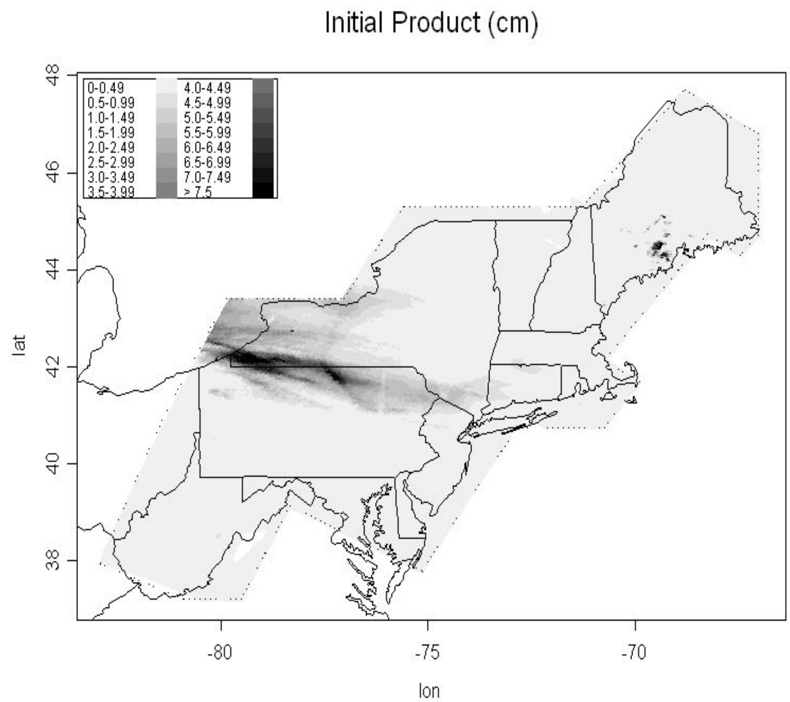
Mar. 20, 2003

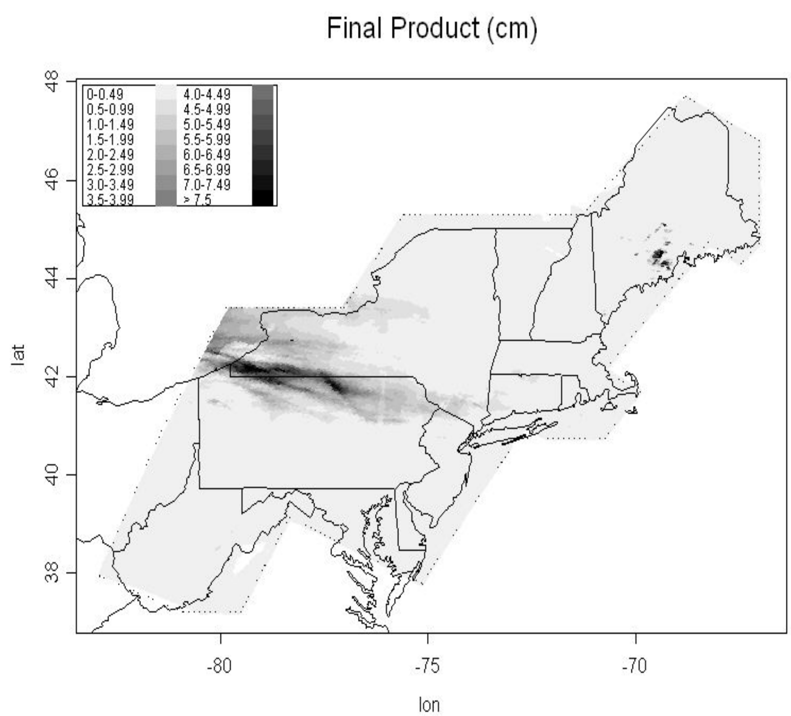




Appendix D (continued): Before and After Graphs of All 30 Cases

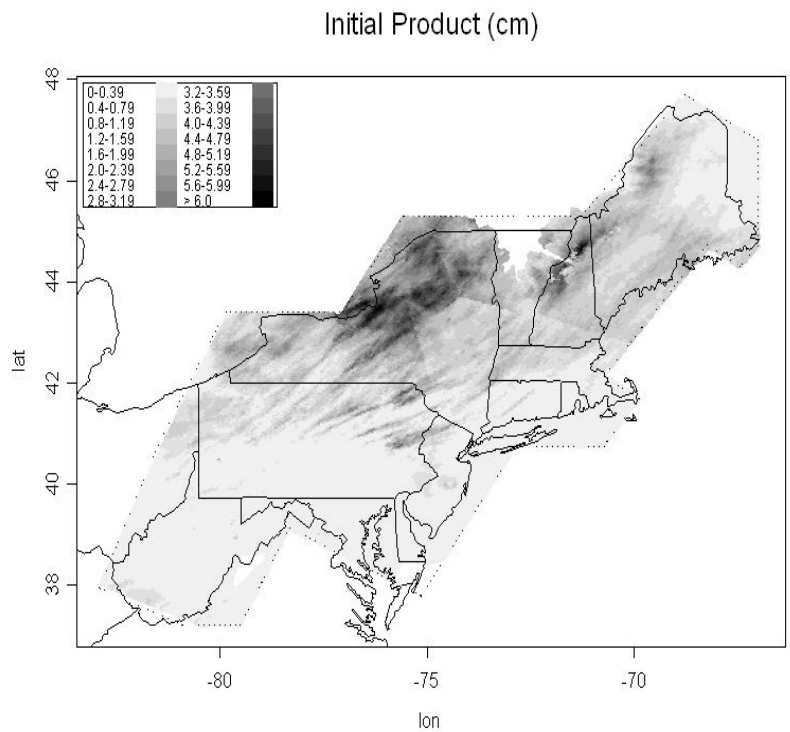
May 8, 2004

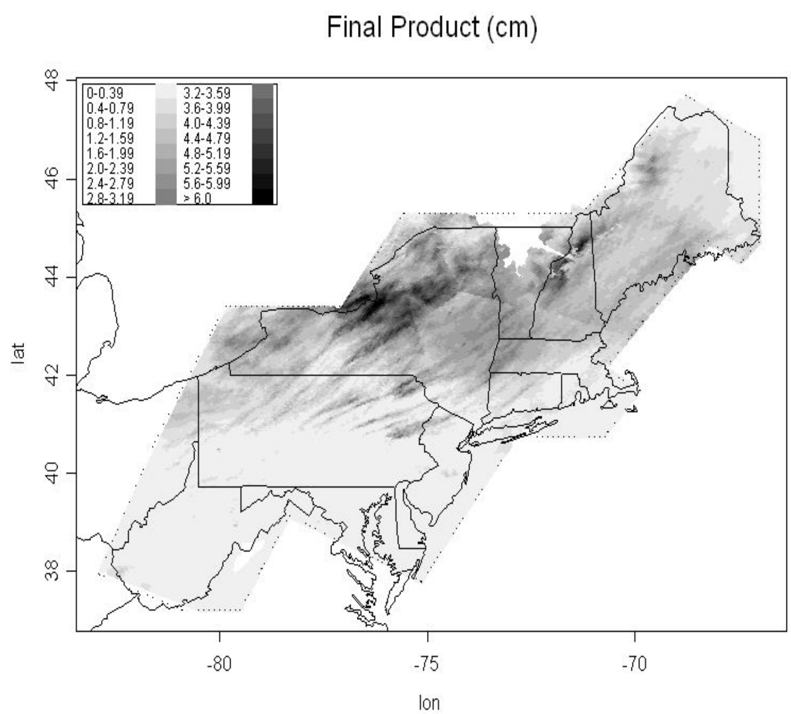




Appendix D (continued): Before and After Graphs of All 30 Cases

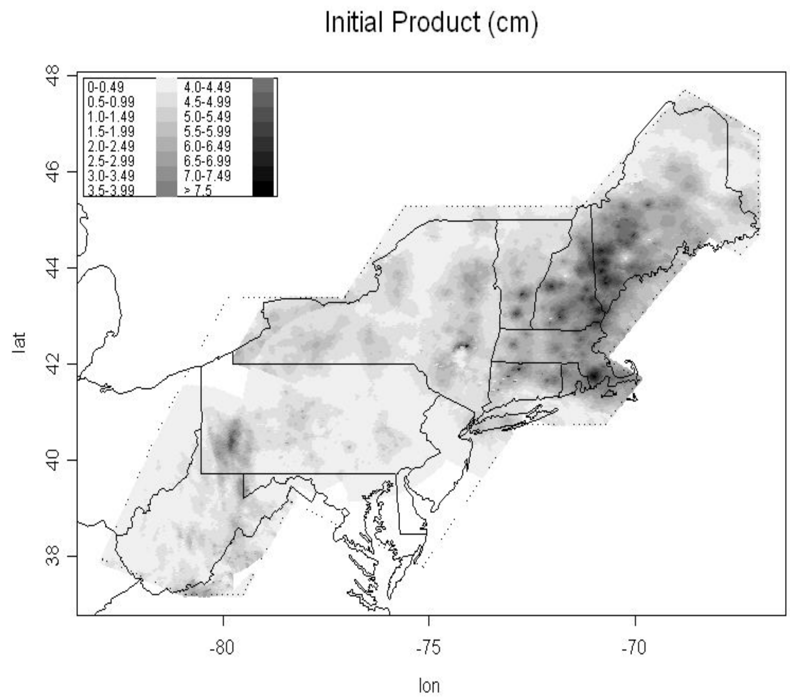
May 11, 2003

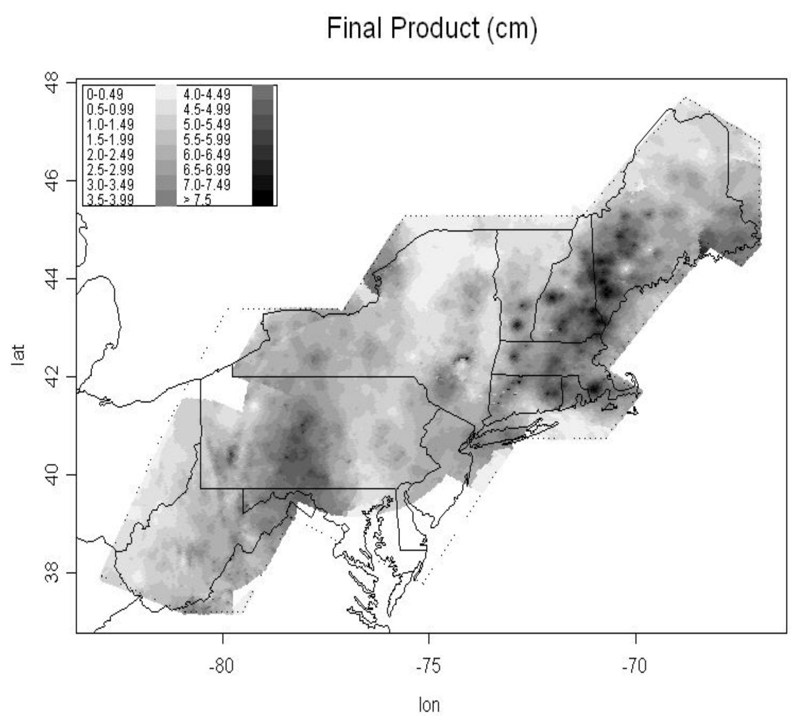




Appendix D (continued): Before and After Graphs of All 30 Cases

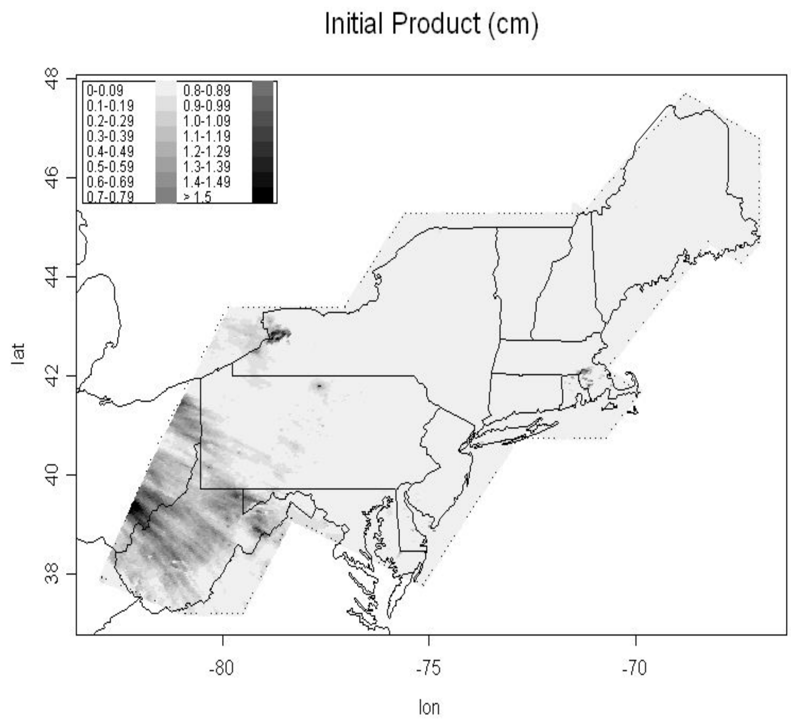
Nov. 1, 1997

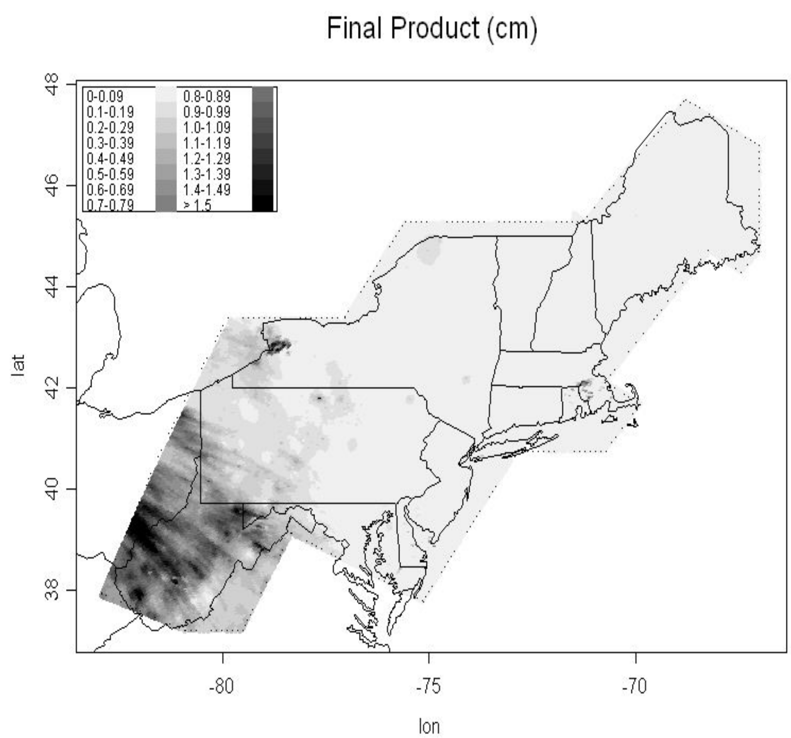




Appendix D (continued): Before and After Graphs of All 30 Cases

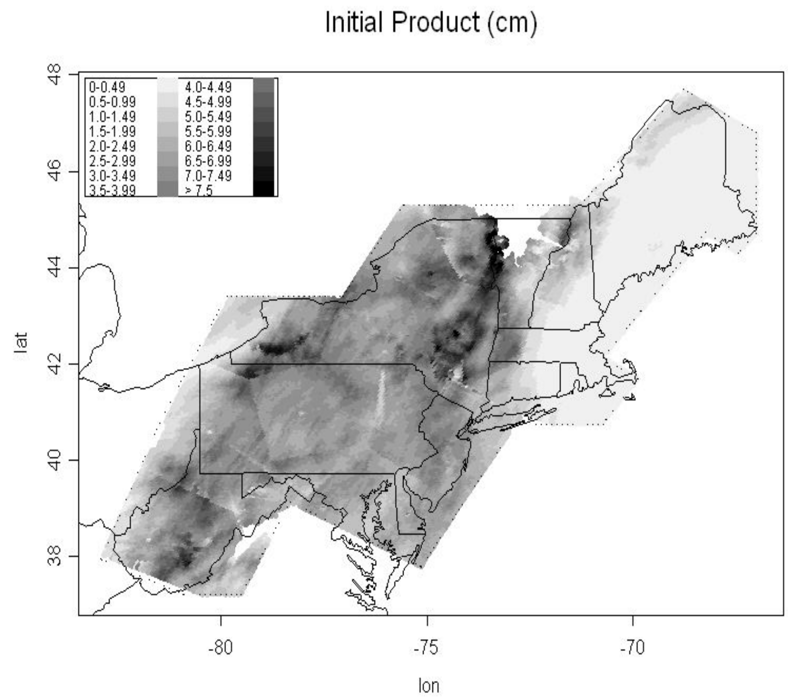
Nov. 17, 2004

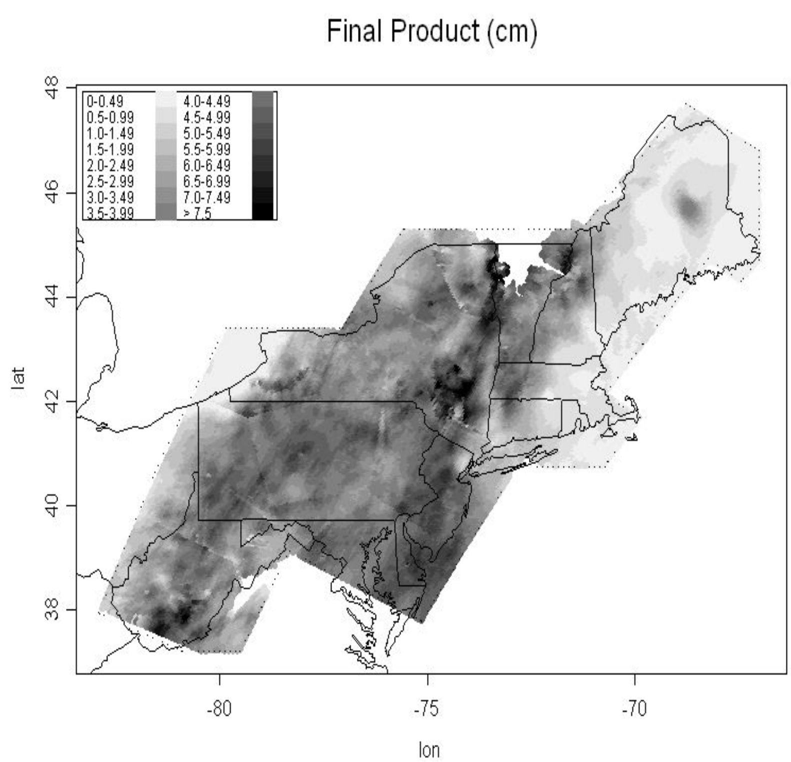




Appendix D (continued): Before and After Graphs of All 30 Cases

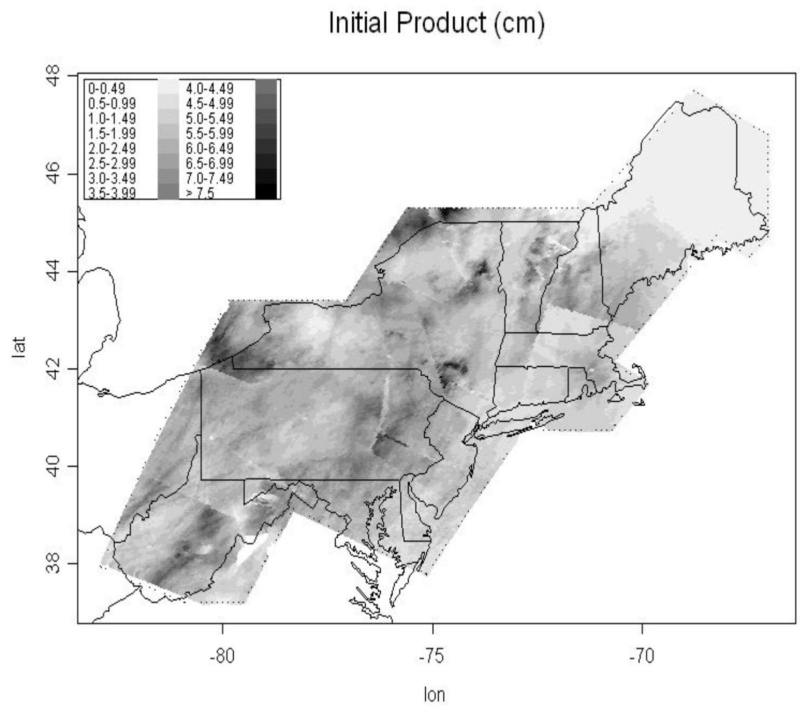
Nov. 19, 2003

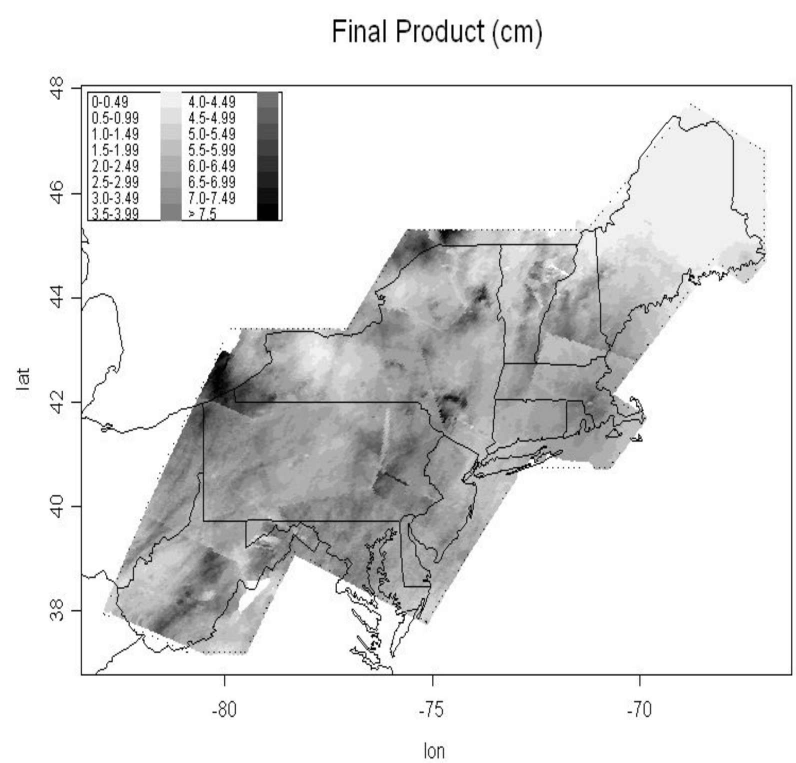




Appendix D (continued): Before and After Graphs of All 30 Cases

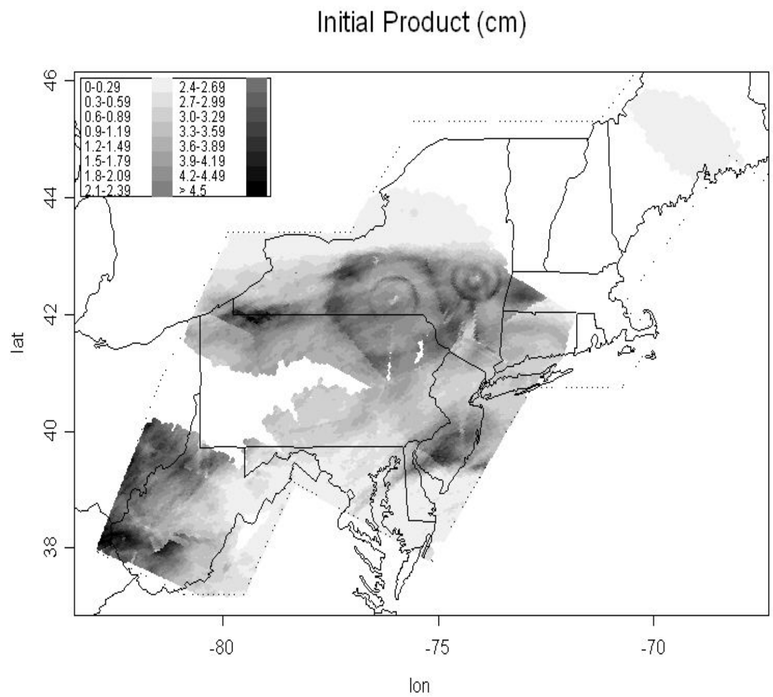
Oct. 15, 2003

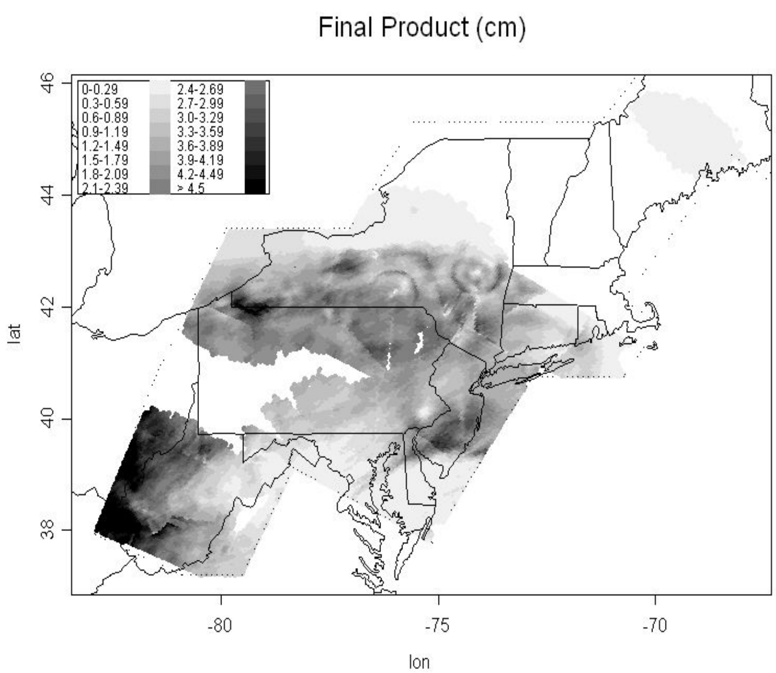




Appendix D (continued): Before and After Graphs of All 30 Cases

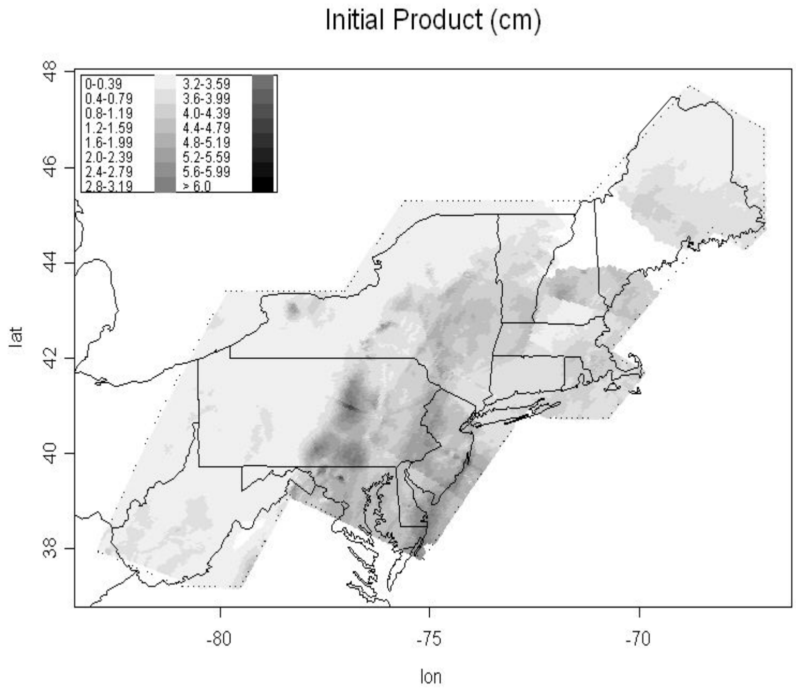
Oct. 18, 2004

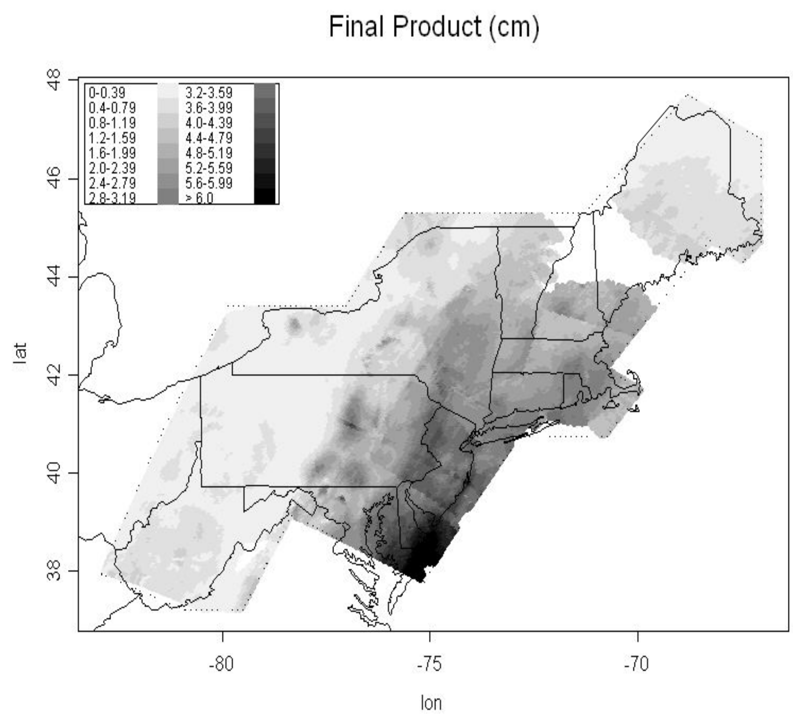




Appendix D (continued): Before and After Graphs of All 30 Cases

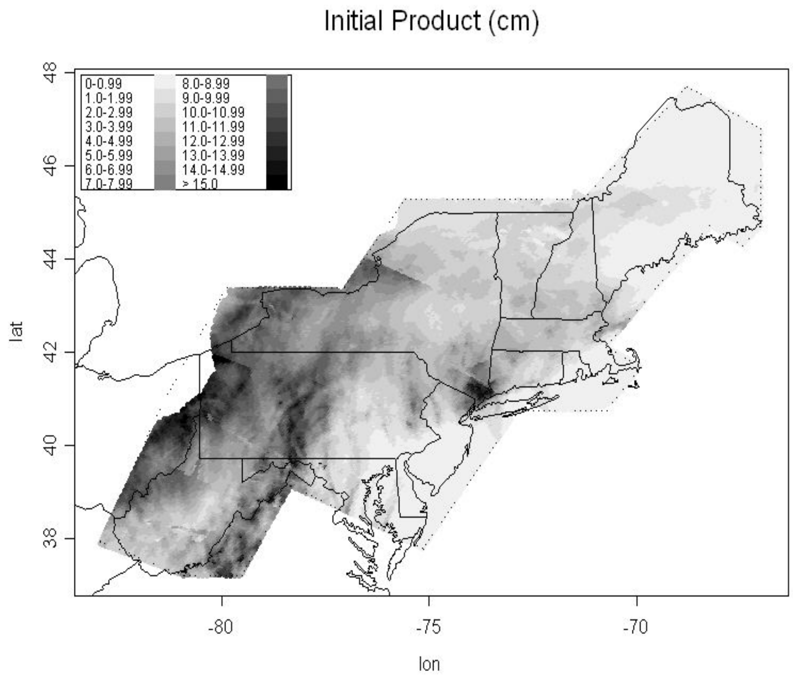
Oct. 28, 2003

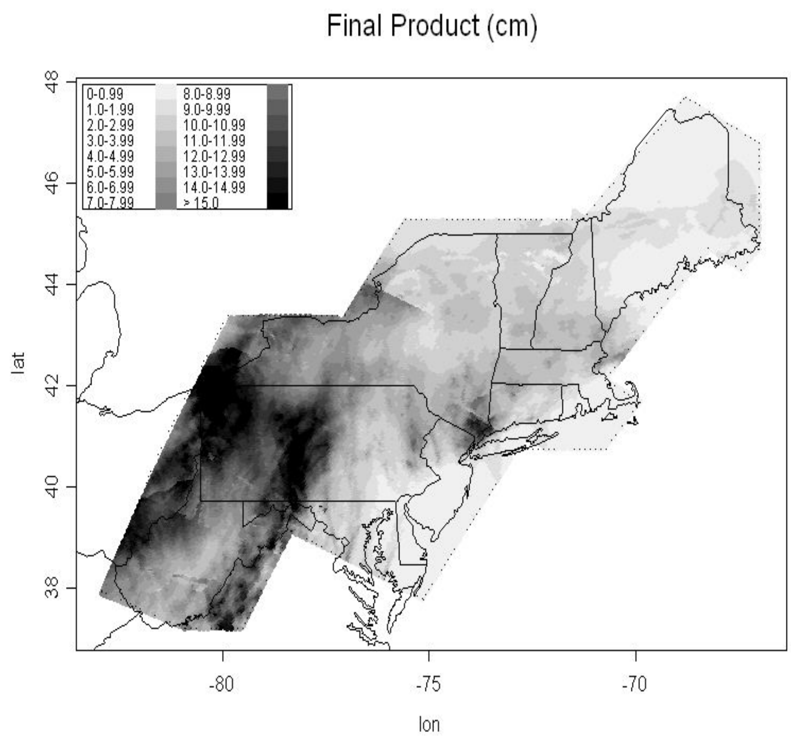




Appendix D (continued): Before and After Graphs of All 30 Cases

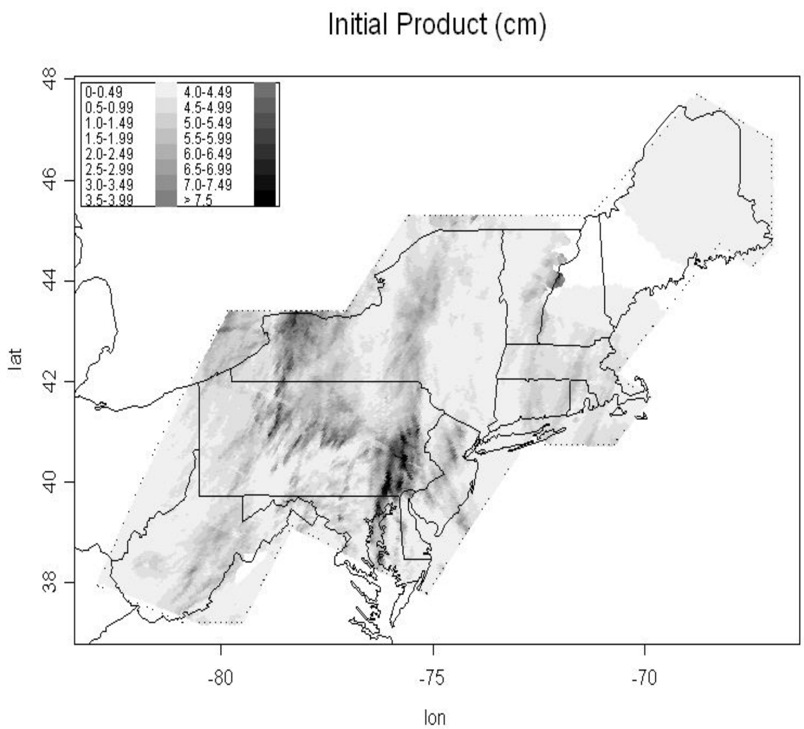
Sep. 8, 2004

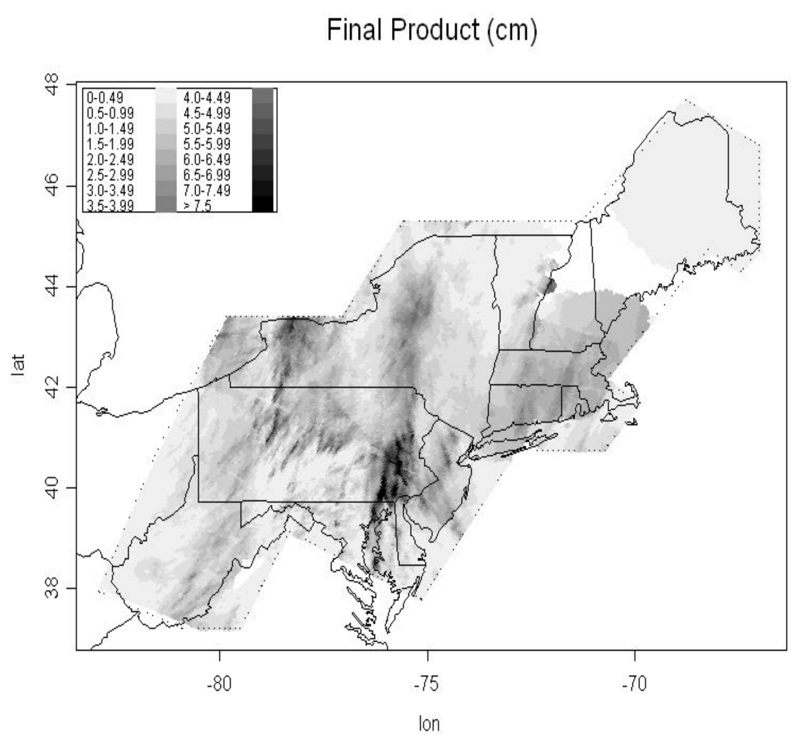




Appendix D (continued): Before and After Graphs of All 30 Cases

Sep. 15, 2003

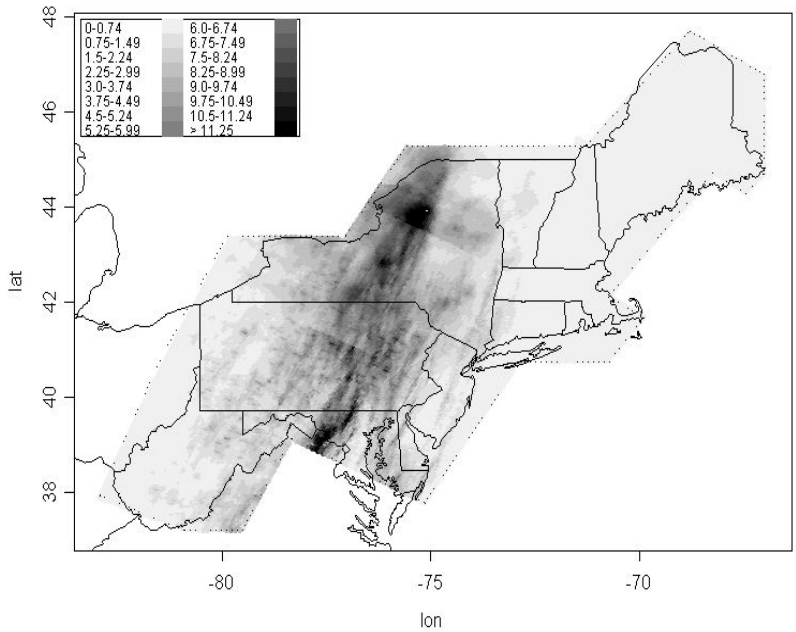




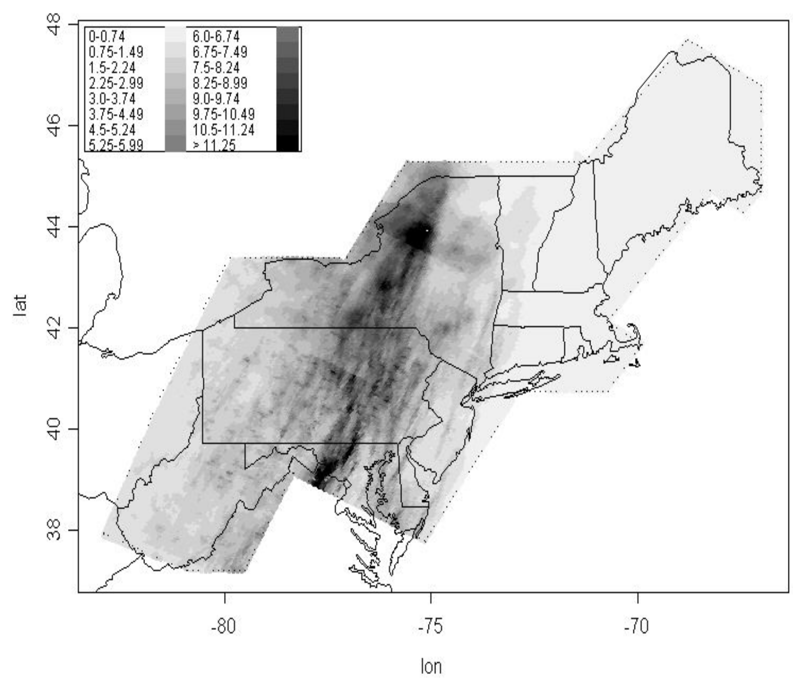
Appendix D (continued): Before and After Graphs of All 30 Cases

Sep. 24, 2001

Initial Product (cm)



Final Product (cm)



REFERENCES

Anagnostou, E. N., and W. F. Krajewski, 1999: Real-Time Radar Rainfall Estimation. Part 1: Algorithm Formulation. *J. Atmos. Oceanic Technol.* **16**, 189-197.

Anagnostou, E. N, M. N. Anagnostou, W. F. Krajewski, A. Kruger, and B. J. Miriovsky, 2004: High-Resolution Rainfall Estimation from X-Band Polarimetric Radar Measurements. *J. Hydrometeor.*, **5**, 110–128.

Austin, P. M., 1987: Relation between measured radar reflectivity and surface rainfall. *Mon. Wea. Rev.,* **115**, 1053-1070.

Azimi-Zonooz, A., W. F. Krajewski, D. S. Bowles, and D. J. Seo, 1989: Spatial rainfall estimation by linear and non-linear cokriging of radar-rainfall and rain gauge data. *Stoch. Hydrol. Hydraul.,* **3**, 51-67.

Brandes, E. A., 1975: Optimizing rainfall estimates with the aid of radar. *J. Appl. Meteor.*, **14**, 1339-1345.

Brandes, E. A., G. Zhang, and J. Vivekanandan, 2002: Experiments in Rainfall Estimation with a Polarimetric Radar in a Subtropical Environment. *J. Appl. Meteor.*, **41**, 674–685.

Cressie, N. A. C., 1993: Statistics for Spatial Data. John Wiley & Sons Inc., 900 pp.

Crum, T. D., and R. L. Alberty, 1993: The WSR-88D and the WSR-88D Operational Support Facility. *Bull. Amer. Meteor. Soc.*, **74**, 1669-1687.

Fulton, R. A., J. P. Breidenbach, D. J. Seo, D. A. Miller, and T. O'Bannon, 1998: The WSR-88D rainfall algorithm. *Wea. And Fore.*, **13**, 377-395.

Groisman, P. Y., and D. R. Legates, 1994: The Accuracy of United States Precipitation Data. *Bull. Amer. Meteor. Soc.*, **75**, 215-227.

Hunter, S. M., 1996: WSR-88D Radar rainfall estimation: Capabilities, limitations and potential improvements. *Natl. Wea. Dig.*, **20**, 26-38.

Kaluzny, Stephen P., S. C. Vega, T. P. Cardoso, and A. A. Shelly, 1998: S+ Spatial Stats. Math Soft, Inc.

Kitanidis, Peter K., 1997: Introduction to Geostatistics: Applications in Hydrogeology. Cambridge University Press.

Krajewski, W. F., 1987: Cokriging of radar-rainfall and rain gauge data, *J. Geophys. Res.,* **92**, 9571-9580.

Nuss, Wendell A., and David W. Titley, 1994: Use of Multiquadric Interpolation for Meteorological Objective Analysis. *Mon. Wea. Rev.*, **122**, 1611-1631.

Seo, D. J., 1998a: Real-time estimation of rainfall fields using rain gage data under fractional coverage. *J. Hydrol.*, **208**, 25-36.

Seo, D. J., 1998b: Real-time estimation of rainfall fields using radar rainfall and rain gage data. *J. Hydrol.*, **208**, 37-52.

Seo, D. J., and J. P. Breidenbach, 2002: Real-time correction of spatially nonuniform bias in radar rainfall data using rain gage measurements. *J. Hydrometeor.*, **2**, 93-111.

Seo, D. J., J. P. Breidenbach, and E. R. Johnson, 1999: Real-time estimation of mean field bias in radar rainfall data. *J. of Hydrol.*, **23**, 131-147.

Seo, D. J., J. P. Breidenbach, R. A. Fulton, D. A. Miller, and T. O'Bannon, 2000: Real-time adjustment of range-dependent biases in WSR-88D rainfall data due to nonuniform vertical profile of reflectivity. *J. Hydrometeor.*, **1**, 222-240.

Simanton, J. R., and H. B. Osborn, 1980: Reciprocal Distance Estimate of Point Rainfall. *J. of the Hydraulics Division, ASCE*, **106**, 1242-1246.

Smith, J. A., and W. F. Krajewski, 1991: Estimation of mean field bias of radar rainfall estimates. *J. Appl. Meteor.*, **30,** 397-412.

Wilson, J. W., 1970: Integration of Radar and Raingage Data for Improved Rainfall Measurement. *J. Appl. Meteor.*, **9**, 489-497.

Wilson, J. W., and E. Brandes, 1979: Radar measurement of rainfall—a summary. *Bull. Amer. Meteor. Soc.*, **60**, 1048-1058.