Use of Satellite Thermal Infrared Telemetry in Soil Moisture and Rainfall Mapping

Randall Lee Chilson

Use of Satellite Thermal Infrared Telemetry in Soil Moisture and Rainfall Mapping

by

Randall Lee Chilson

South Dakota State University
Brookings, SD 57007

A thesis
presented to South Dakota State University
in fulfillment of the
thesis requirement for the degree of
Masters of Science
in
Engineering
1986

USE OF SATELLITE THERMAL INFRARED TELEMETRY

IN SOIL MOISTURE AND RAINFALL MAPPING

This thesis is approved as a creditable and independent investigation by a candidate for the degree, Master of Science, and is acceptable for meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Dr. Warren Hein　　　　Date

Thesis Advisor

Dr. Warren Hein　　　　Date

Head, Physics Department

Acknowledgements

The author wishes to thank Dr. Warren Hein, Don Moore, and Dr. Jerald A. Tunheim for their help and guidance during this study. The author also wishes to thank the EROS Data Center and the Remote Sensing Institute staffs for their help, use of equipment and data for for this study.

Finally, thanks to the author's parents and brothers, for their support, encouragement, and understanding during the long process to complete this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Introduction

As the use of satellite telemetry becomes a more popular tool in management of natural resources, more tasks are being done from the satellite altitude to cover a larger expanse of land area. The telemetry in the infrared channels of the satellite, known as thermography, has become more important in many areas of resource management. One of the present applications of thermography is the detection of vegetation coverage. Other possible applications of thermography may include rainfall monitoring and soil moisture monitoring, both of which are the topics of concern in this study.

Current monitoring of rainfall and soil moistures is being done by ground collected data. This method is time consuming, expensive and impractical for remote regions. A method which could use data for a large area collected from an orbiting platform would greatly ease the task and assist in predicting oncoming droughts and desert expansions. This leads to one possible application of satellite collected thermography from orbiting satellites.

Since the soil temperature is affected by many factors such as vegetation coverage, slope, texture, composition, and soil moisture levels, measurement of soil temperature would then be an ideal method of determining some of these variables. Surface temperature in turn can be monitored by its thermal emitance, since thermal emitance is

proportional to the surface temperature raised to the fourth power. Thus, the thermography from a satellite should be proportional to surface temperature of the ground.

As stated earlier, a great number of variables affect the surface temperature of the ground. This study was restricted to rainfall, soil moisture, and other variables readily available for use in the model calculations. These other variables include the length of time the ground is exposed to the sun, cultivated vegetation coverage, air temperature, and normalized difference. It is true that more variables can be added to increase the accuracy of the model, but this would not be practical for a remote region where these parameters are not known.

Previous experimental and theoretical work on a model to predict surface temperature changes has been done at South Dakota State University. A theoretical heat flow model to predict the soil temperature for a profile from surface to a depth of 50 cm as a function of time was developed (Meyer, 1972). An experimental study to provide qualitative proof of the theoretical model using field collected data showed a good correlation (Beutler, 1980). Further enhancements were done to the original model to include soil characteristics and solar and plant canopy parameters. After a series of experimental calculations, a quadratic relationship between soil moisture and surface temperature resulted (Ness, 1982). Further experimental work was carried out to determine the effect of these parameters on the model and their significance. Results of this study indicated that these parameters added no significance to the model for the field collected data, but

resulted in a significant linear relationship between soil moisture and surface temperature (Roesler, 1984).

This study was an attempt to use the previous relationships developed between soil moisture and surface temperature and to apply these on a broader scale. Using experimental data collected by satellite and ground collected data for the entire state of South Dakota, a relationship between satellite gathered thermography and ground collected moisture and rainfall data was found. Significant differences between this study and previous studies result from the scope of the data being used (small field plots as compared to the entire state) and the attempt to reverse the model to predict soil moisture, instead of predicting surface temperatures.

Objectives

The specific objectives of this study were then as follows:

1. To gather together data collected from satellite thermography, and to put the data into an acceptable form for later analysis.
2. To experimentally produce state-wide data from sparsely collected ground data throughout the state.
3. To combine the data collected from the above objectives together with other significant parameters to form a data base for further work.
4. To use the data base of objective 3 for statistical analysis of relationships between satellite-collected data and ground-based collected data consisting of soil moisture and rainfall.
5. To compare the statistical relationships found to those produced in previous experimental and theoretical studies.

Background Literature

The use of satellite thermal imagery to track soil moisture conditions began with Moore (1975) who studied the relationship of soil moisture to infrared data gathered from Skylab. His study indicated a positive correlation and concluded that a potential existed for soil moisture tracking from satellite thermal imagery data. Thermocouple data and aircraft gathered infrared data were correlated by Schmugge et. al. (1976) and Reginato et. al. (1978), who found a positive relationship between both temperature monitoring sources. Tunheim (1977) found a relationship between aircraft thermal imagery patterns and ground temperature patterns caused by near surface moisture conditions influenced by saline seeps in Montana.

In 1978, NASA undertook the Heat Capacity Mapping Mission (HCMM) project utilizing a satellite with a 10.5 - 12.5 micron sensor. From data collected by this satellite, Heilman and Moore (1981) showed that high soil moisture areas could be detected from the HCMM satellite data with a significant correlation ($r = 0.74$) for soil moistures at 0 to 4 cm depths after empirical estimates and corrections were performed. The estimates were needed to account for the variety and cover density of crops in eastern South Dakota and a correction of the HCMM radiometric temperatures to standard known temperatures. Similar studies to compare the thermal infrared data to soil moisture conditions

were carried out and summarized in the a final report Evaluation of HCMM Data for Assessing Soil Moisture and Water Table Depth (1981). Short (1982) concluded that before thermographic mapping of soil moisture conditions can be used for operational purposes, more work is needed. Price (1984) concluded that apparent thermal inertia (ATI), the difference found between the early morning low temperature thermal images and afternoon high temperature thermal images, were of no value in highly used agricultural areas. This was due to the variation of surface evaporation rates caused by crop canopies. However, ATI was found useful for dry environments.

Various theoretical and empirical models have been developed to predict satellite telemetry data and conditions seen in small plot tests. Meyer (1972) developed a finite difference heat flow model to simulate heat flow through a soil profile during a diurnal period as a function of soil moisture. Tunheim (1977) modified the model to predict a soil temperature difference profile between irrigated and dry plots with identical soil profiles. Beutler (1980) modified the model to work with any soil type profile used for the irrigated and dry plots. He also verified that the model correctly predicted the functional temperature differences observed at the surface of the two plots. Heilman and Moore (1981) used an empirical model to show the relationship of soil moisture at the 0 to 4 cms depths to thermal infrared data, percent canopy cover present, and minimum air temperatures.

Price (1980) developed a model which used remotely sensed thermal data and predicted 24 hour evaporation rates and diurnal heat

capacity as a function of soil moisture. His model calculated mean surfaces temperatures from relationships of absorbed and emitted longwave and shortwave radiation, together with albedo and surface temperatures, meteorological data, and surface characteristics. Horton (1982) concluded that crop coefficients for evapotranspiration were affected by soil moisture conditions and canopy development. Horton also noted that pastures have a small evaporation component of evapotranspiration, and an insignificant evaporation effect on its coefficients.

Ness (1982) further modified the model of Beutler's (1980) to include crop and solar parameters to account for ground shading. He found that the computer model predicts a quadratic relationship between surface temperature differences and moisture differences for time independent soil moisture. Roesler (1984) found that a significant linear relationship existed for small plot data when comparing surface temperature differences to soil moisture differences, rather than the quadratic relationship found by Ness. Roesler also found that the addition of air temperature to the model significantly added to the model predictions.

Data Processing Procedure

Due to the scope of the data involved, a small mainframe computer was used to process the data. The primary data processing was performed on the IBM 370/3031AP mainframe located in the Computing Center at South Dakota State University (SDSU), using IBM's VS-Fortran and assembler and a statistical package written by Statistical Analysis Systems (SAS). Portions of the data were also processed by a HP3000 minicomputer at the EROS Data Center running an interactive digital image manipulation system called IDIMS, and a Prime 400 minicomputer at the Remote Sensing Institute on the SDSU campus with various image processing programs written at the institute. The data for each individual variable in the model equations was processed individually and later placed together into a large data base containing all of the needed variables. It should be noted that to process the quantity of data required, approximately 190,098 image sample sites within South Dakota, some of the later processing required large amounts of computer time and resources on the small mainframe.

Thermography

Thermography data was taken from an image produced by the NOAA-6 satellite, on July 6, 1981 at 9:38 AM central daylight time, obtained from the Remote Sensing Institute and EROS Data Center.

The data is produced by infrared and optical sensors located on the satellite and is stored in the satellite for later transmission to a ground receiving site. The resolution of the sensors is one kilometer by one kilometer. The data stream received is made up of a header label data block which contains satellite identification, location, time, and variables describing the imagery data to follow. Imagery data then follows which has various positioning and telemetry variables for calibration purposes followed by 2048 individual picture element (pixel) blocks containing separated sensor ranges (channels). This is trailed by additional positioning and telemetry variables. This composes two logical data records or one physical data block. Two physical data blocks are needed for each individual line in a picture image made of approximately 1500 lines, resulting in an image of about 3 million pixels. This type of image format is known as Local Area Coverage/High Resolution Picture Transmission (LAC/HRPT). The raw image from which the data for this project was extracted from is shown in Figure 1.

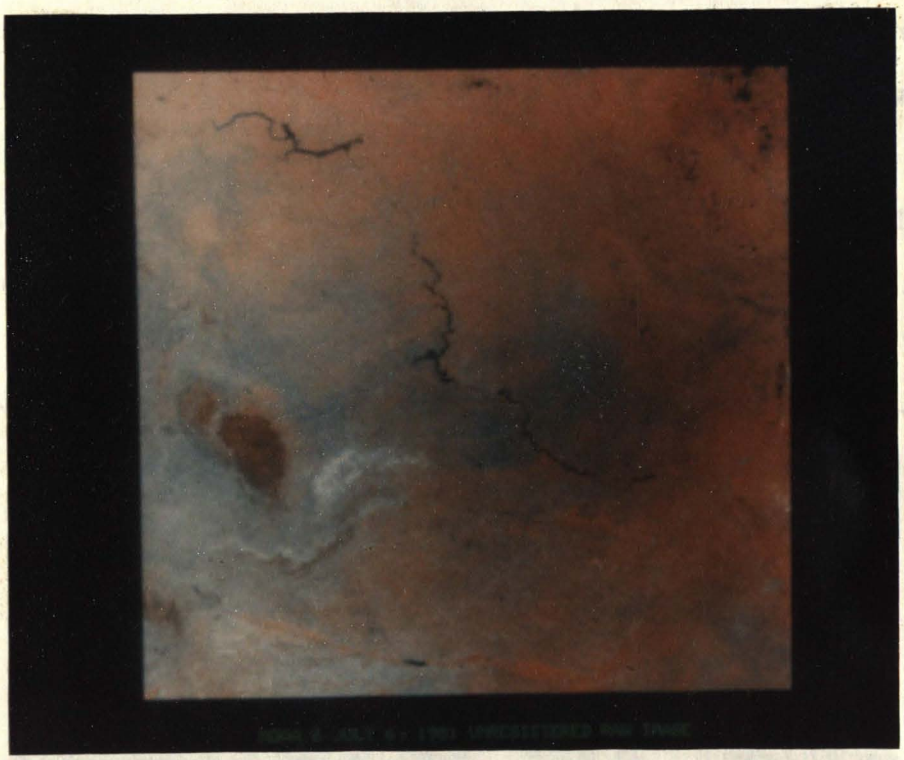

Figure 1: NOAA-6 All Channel Composite Raw Image of the Northern Plains.

The image was first processed by the IDMS system to extract the section of the image containing the state of South Dakota, and was then masked with null values to show only the state. This uses a process known as image registration which involves the use of the IDMS system to locate known sites within the area and to correct for any skewing and distortion caused by different speeds between the earth's rotation and satellite orbit speed. A mask is added to the registered image to show only the state after the positions have been verified. This

reduces the image to approximately 450 lines by 650 pixels, some of which are null mask values, leaving approximately 190,000 pixels of interest. The image was separated into individual sensor channels. Channel 4, which is the thermal infrared sensor range of 10.5 to 11.5 microns, is shown in Figure 2. The data was then transfered to the IBM mainframe to convert the values to the actual temperatures.

Figure 2: NOAA-6 Thermal Infrared Image of South Dakota.

To convert the image data to actual temperature data, both the raw image information and the extracted thermal image must be used.

The equations used for conversion to actual temperature readings are shown in Figure 3 and are described in NOAA Polar Orbiter Data Users Guide (1984). The values used for the slope and intercept of Equation 1 were extracted from the raw image data by the SAS program shown in Appendix A1. This program will extract the calibration for the thermal infrared sensor from the telemetry calibration variables heading each line of image data and report data set header information and statistical values for each of the parameters. The average value for the whole image is used for each parameter, since the variance in each parameter is not great enough to warrant the more involved approach of extracting the parameters for each line of the image. The value for the constant **W** in Equation 2 is the central wave number for the temperature range desired from image channel 4 for the satellite used. This is 911.41 for Figure 3 presented here. The equations are applied to each point within the image to produce the desired temperature readings. The combination assembler and VS-Fortran program shown in Appendix A2 processes the image created by the IDMS system and writes a new image of the temperatures read. This data was plotted by the SAS program listed in Appendix A3 which results in the temperatures shown in Figure 4.

$$E_i = S_i C + I_i \qquad (1)$$

E = energy value in milliwatts/m^2 -steradians-cm^{-1}

C = input satellite value, ranging from 0 - 1023

S = slope, read from the sensor calibration data

I = intercept, read from the sensor calibration data

$$T(E) = \frac{C_2 W}{\ln(1 + \frac{C_1 W^3}{E})} \qquad (2)$$

T = temperature (^oK)

E = energy value, from equation 1

W = central wave number for sensor temperature range

C_1 = 1.1910659 x 10^{-2} milliwatts/m^2 -steradians-m^{-4}

C_2 = 1.438833 cm-^oK

Figure 3: Equations Used for Image Value to Temperature Value Conversion. Equations from NOAA Polar Orbiter Data (TIROS-N, NOAA-6, NOAA-7, and NOAA-8) Users Guide (1984).

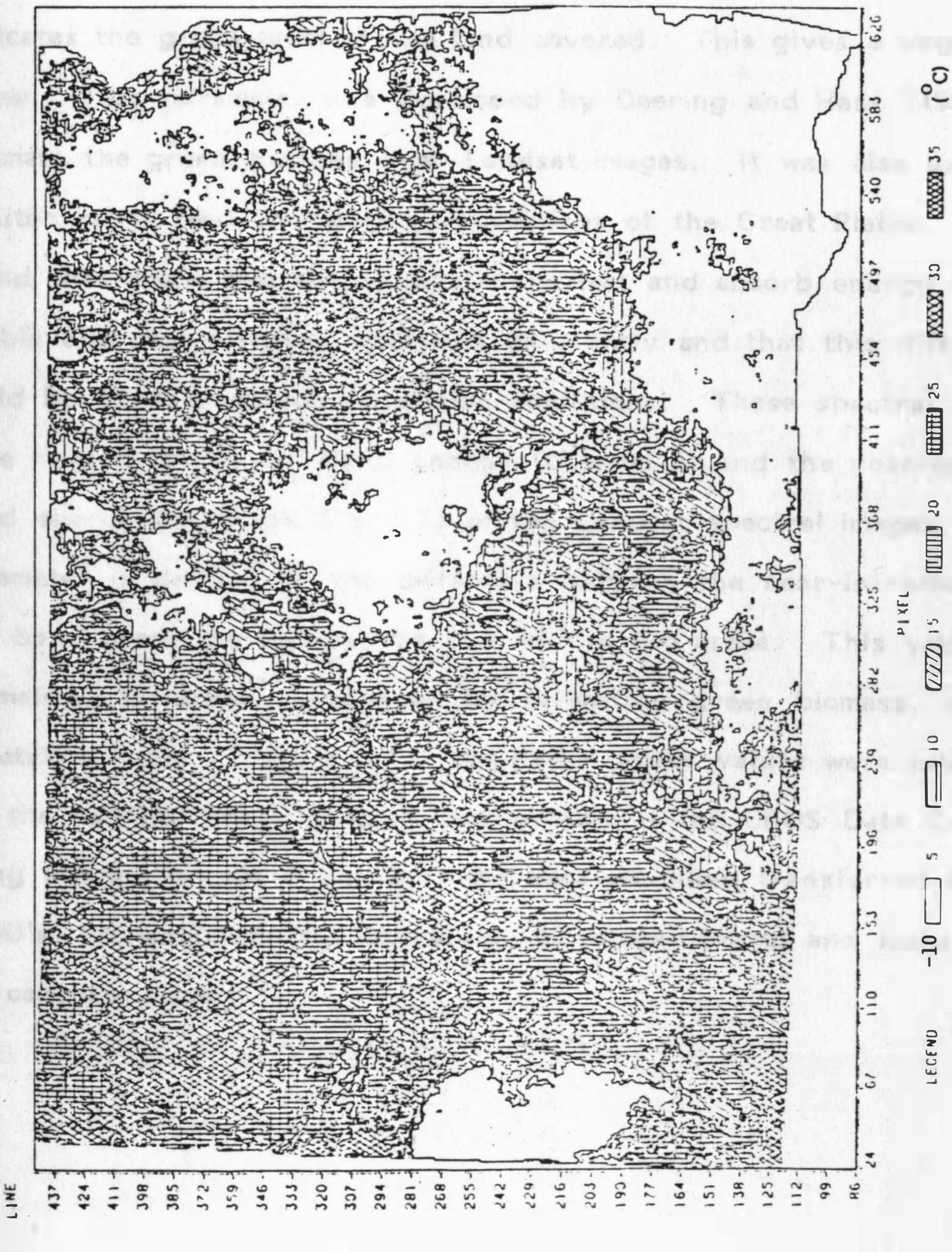

Figure 4: Thermal Infrared Temperature Map of South Dakota; at 9:38 CDT, July 6, 1981.

Normalized Difference

Normalized difference is a satellite gathered parameter which indicates the green level of the land covered. This gives a vegetation index. This parameter was developed by Deering and Hass (1980) to estimate the green biomass from Landsat images. It was also used to monitor range feed and drought conditions of the Great Plains. They found that green and dry vegetation reflect and absorb energy in the visible and near-infrared spectrum differently and that this difference could be used to detect vegetation conditions. These spectral areas were related to the red band energy (channel 5) and the near-infrared band energy (channels 6 and 7) of the Landsat spectral images. The parameter is defined as the difference between the near-infrared and red band areas divided by the sum of the two areas. This yields a normalized difference parameter for estimating green biomass, or a vegetation index (Figure 5). Vegetation index values were calculated for the satellite image used in this project at the EROS Data Center using image channels 1 and 2. The data was then transferred to the SDSU's IBM mainframe for conversion to integer format and inclusion in the complete model.

$$\text{Vegetation Index(VI)} = \frac{\text{Near IR} - \text{Red}}{\text{Near IR} + \text{Red}}$$

Near IR = Landsat MSS channels 6 and 7; NOAA channel 2; approximately. 0.7 - 1.1 microns.

Red = Landsat MSS channel 5; NOAA channel 1; approximately. 0.6 - 0.7 microns.

Vegetation Index(VI) = values between -1 and +1, which are scaled to numeric values of 0 to 255, estimating green biomass; values below 127 are relatively barren.

Figure 5: Normalized Difference Equation Used for Vegetation Coverage Estimation. Equation from Using Landsat Digital Data For Estimating Green Biomass, by Deering and Haas (1980).

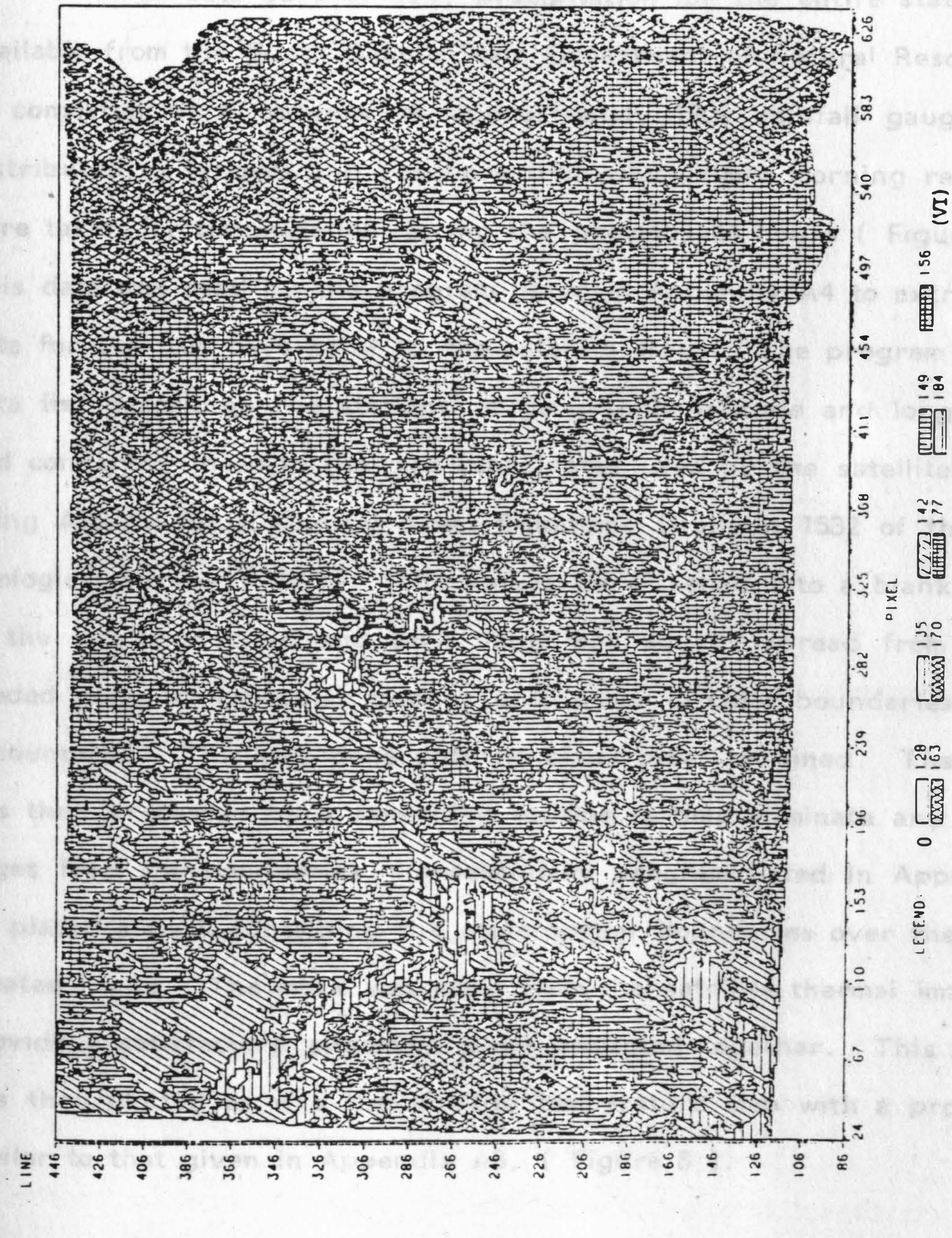

Figure 6: Normalized Difference Map of South Dakota; at 9:38 CDT, July 6, 1981.

Rainfall

A large data base of daily precipitation for the entire state was available from the South Dakota State Department of Natural Resources. It consisted of a network of approximately 1300 rainfall gauges distributed to farmers and ranchers from which daily morning readings were taken for precipitation during the previous 24 hours (Figure 7). This data was read by the program listed in Appendix A4 to extract the data for the five days prior to the satellite image. The program used a data list describing the stations in geographic latitude and longitude and converted these to the map coordinates used by the satellite image using Alber's Equal Conical Areas Equations (Bulletin 1532 of the US Geological Survey (1982)). The values were seeded into a blank image at the locations found and each value was equally spread from the seeded point in a circular manner until other circular boundaries were encountered. This was done until no blank pixel remained. The image was then processed by a smoothing routine to help eliminate any sharp edges from the boundaries. A secondary program listed in Appendix A5 placed a mask containing the state border boundaries over the newly created image. The mask was made from the satellite thermal image to provide a satisfactory alignment when projected together. This image was then plotted as was the satellite temperature map with a program similar to that given in Appendix A3, (Figure 8).

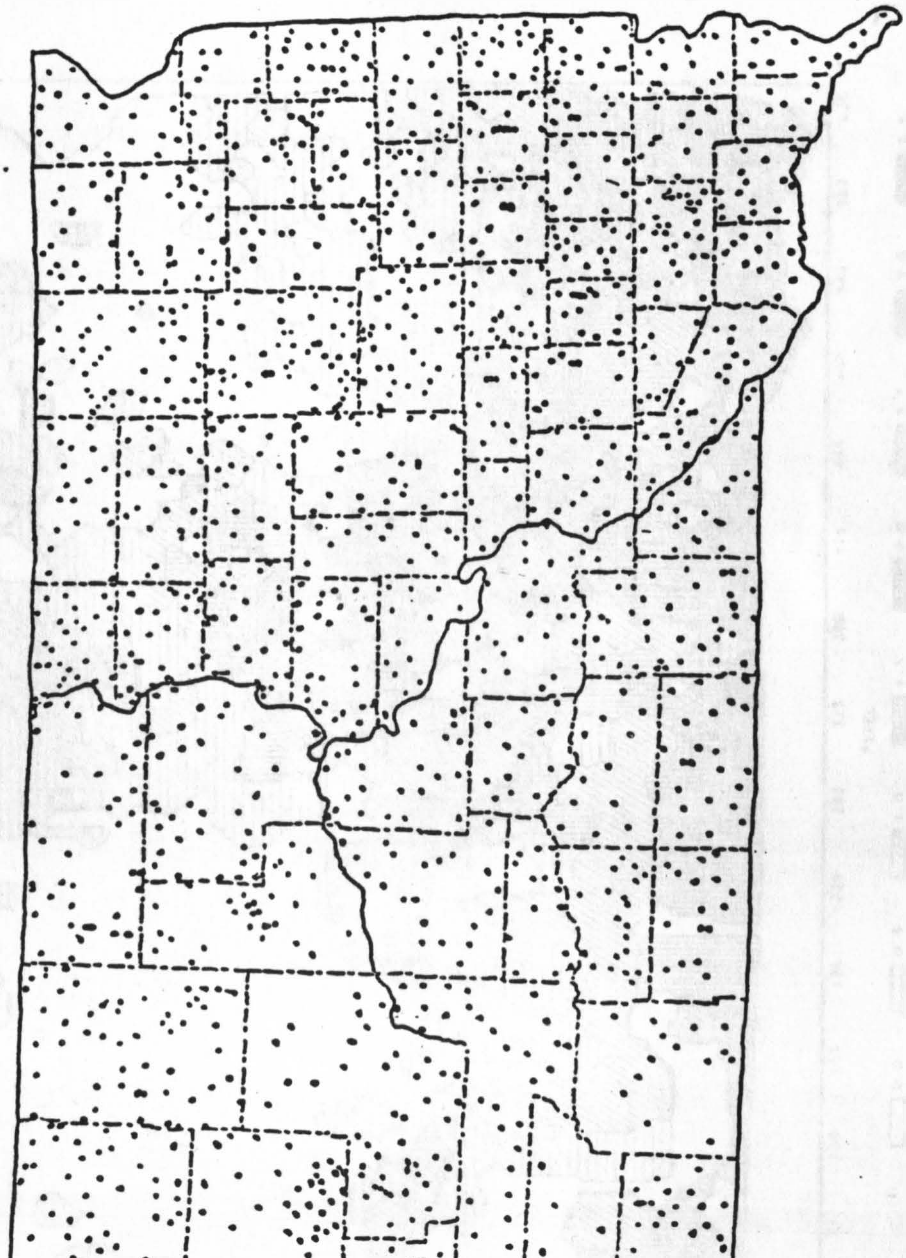

Figure 7: Rainfall Gauge Stations for South Dakota Between 1979 - 1981.

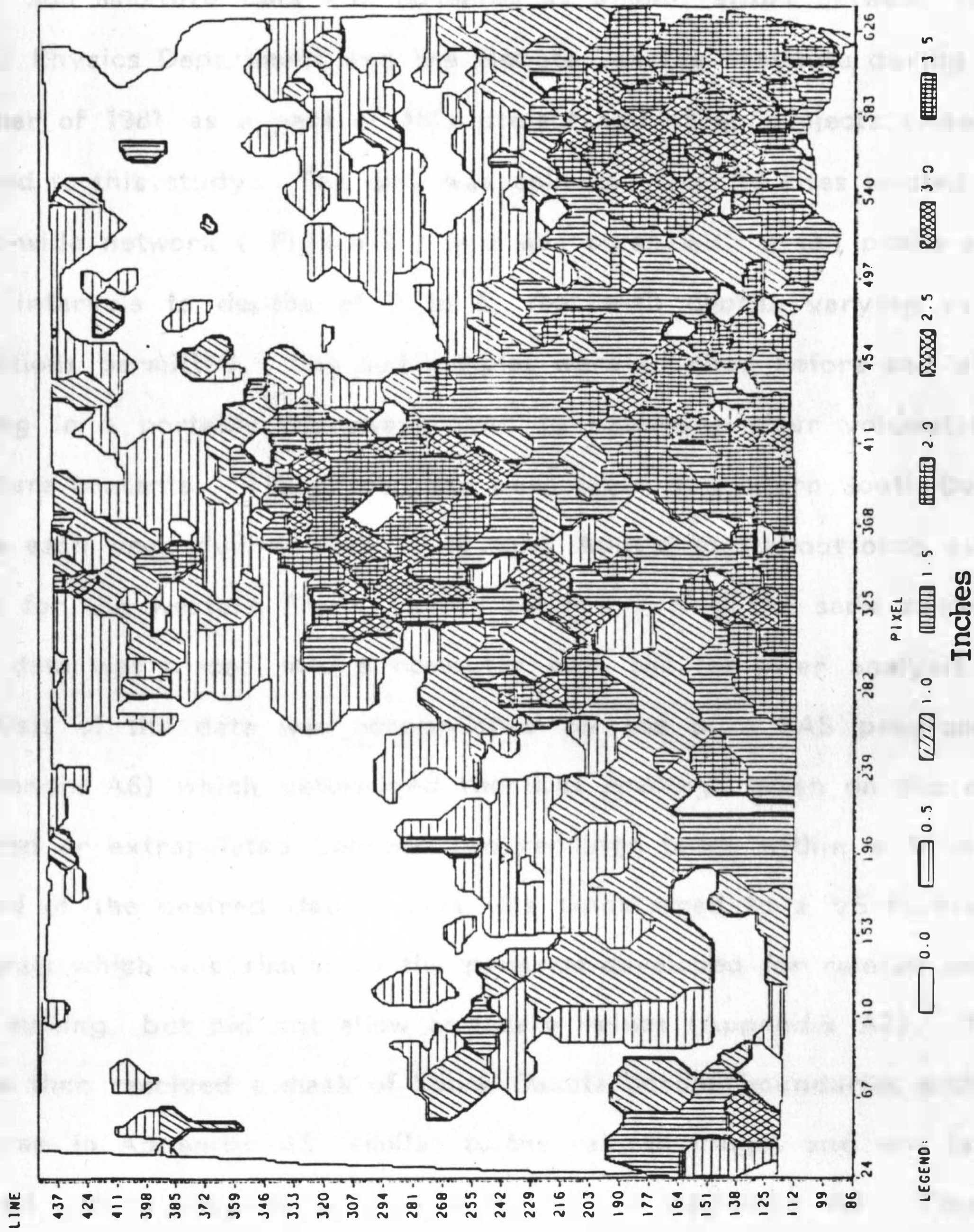

Figure 8: Rainfall Map of South Dakota July 2 - 6, 1981.

Soil Moisture

Soil moisture data was collected as a joint effort between the SDSU Physics Department and the Remote Sensing Institute during the summer of 1981 as a part of this project and other projects closely related to this study. The data was collected from 99 sites located in a state-wide network (Figure 9) and was taken with a soil probe at 6 inch intervals to depths of 3 to 4, feet with depths varying as conditions permitted. The soil samples were weighed before and after drying in a portable microwave oven to determine their volumetric moisture contents. These readings were taken in Eastern South Dakota twice each week during June, July, and August and about once every week for the Western South Dakota regions during the same period. The data was placed into a computer data file for later analysis. Analysis of the data was accomplished by use of a SAS program (Appendix A6) which determined the data readings taken on the day desired or extrapolated between two readings taken within a 10 day period of the desired data. This was transferred to a VS-Fortran program which was similar to the program developed for rainfall image map making, but did not allow any zero values (Appendix A7). The image then received a mask of South Dakota border boundaries with the program in Appendix A5, similar to the rainfall image, and was later plotted with a program similar to the one in Appendix A3. This process was done for the surface soil moisture (Figure 10) and for the subsurface soil moisture (one to six inch depth) (Figure 11).

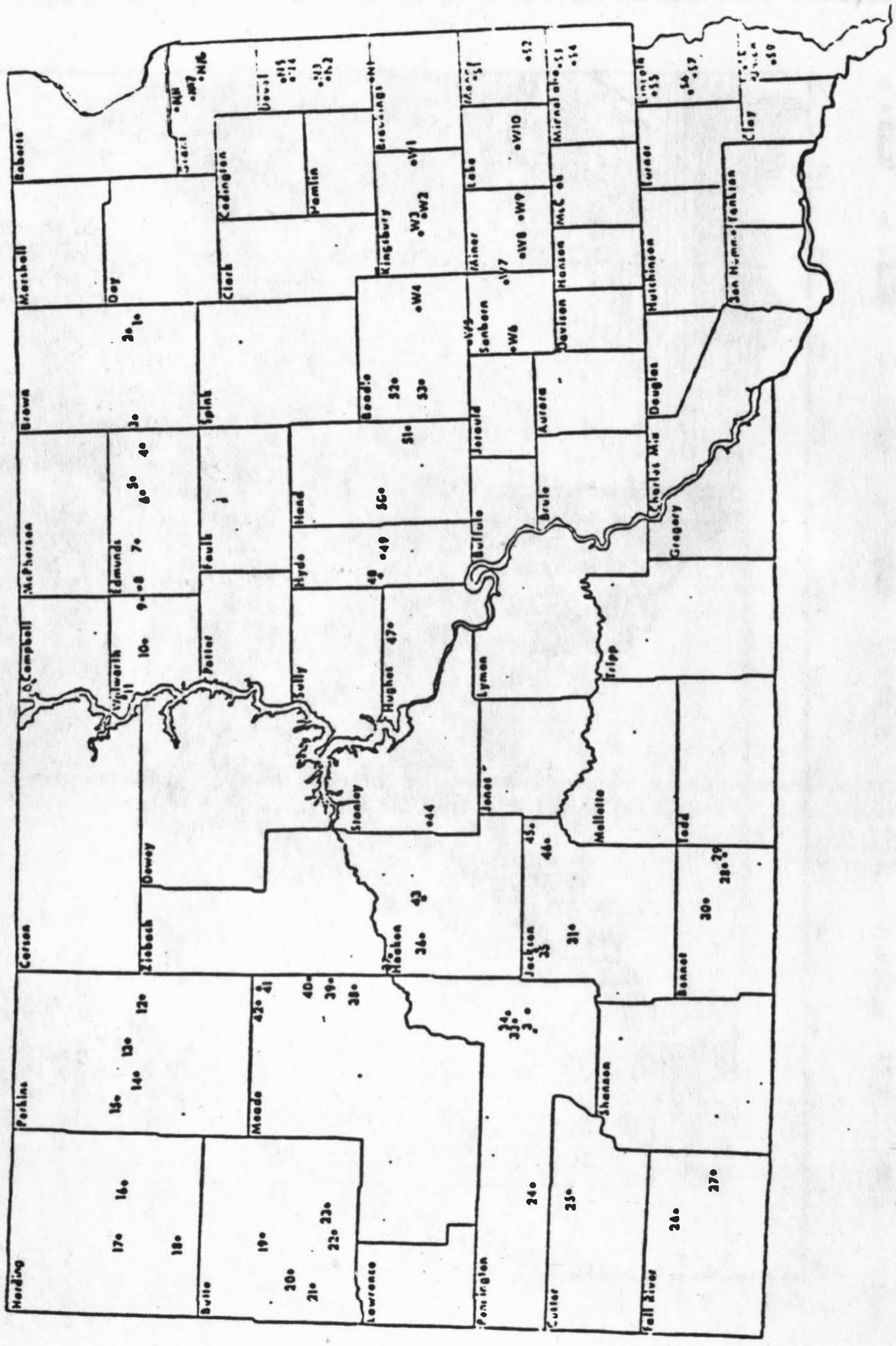

Figure 9: Soil Moisture Collection Sites for South Dakota Between 1979 - 1981.

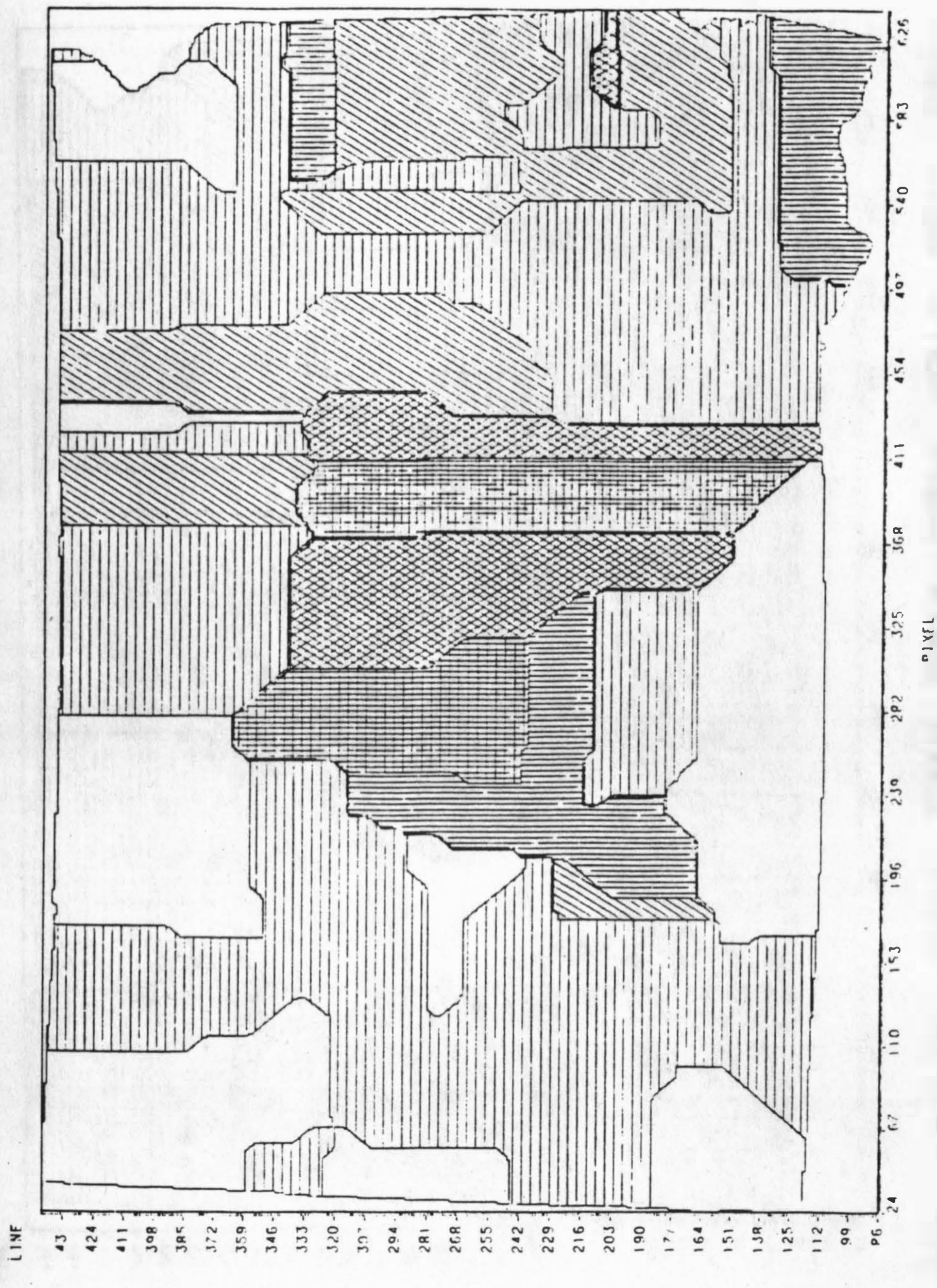

Figure 10: Surface Soil Moisture Map of South Dakota on July 6, 1981.

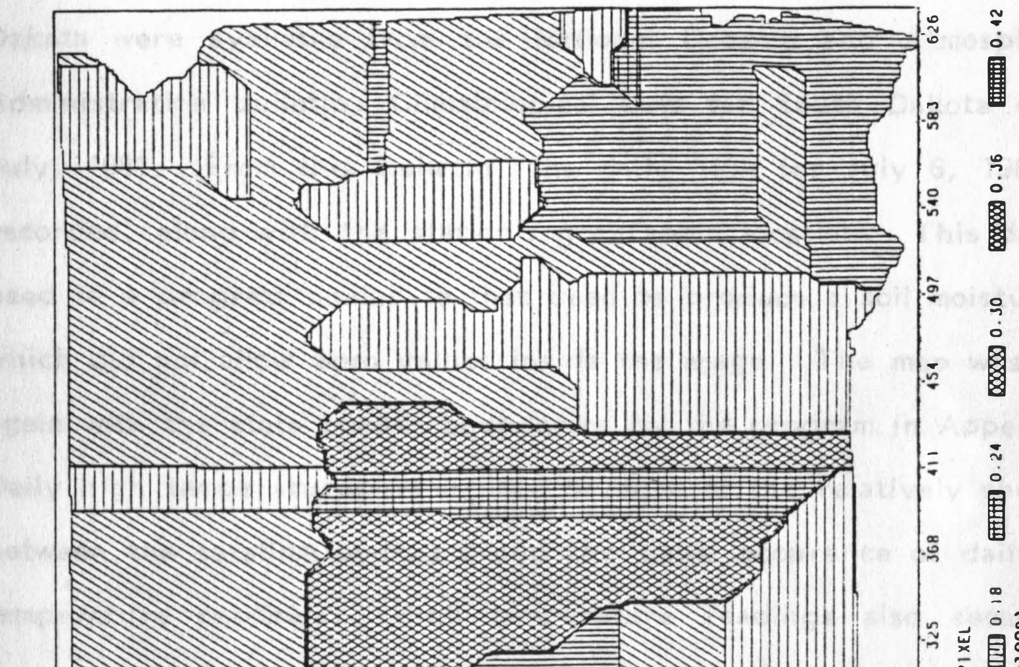

Figure 11: Subsurface Soil Moisture Map of South Dakota on July 6, 1981.

Air Temperature

Air temperature readings for weather reporting stations in South Dakota were available from the National Oceanic and Atmospheric Administration bulletin, Climatological Data for South Dakota during July, 1981. From this bulletin, the daily high for July 6, 1981 was recorded, along with the station's geographic location. This data was used in a program similar to that used to produce a soil moisture map which did not allow zero values inside the image. The map was masked again with the state border boundaries by the program in Appendix A5. Daily high temperatures were used because of the relatively short time between the satellite overpass and the usual occurance of daily high temperature readings. High temperature readings also seem to represent the soil moisture and atmospheric conditions present for the surrounding area, which may affect satellite readings. The resulting image (Figure 12) was plotted by a program similar to that of Appendix A3.

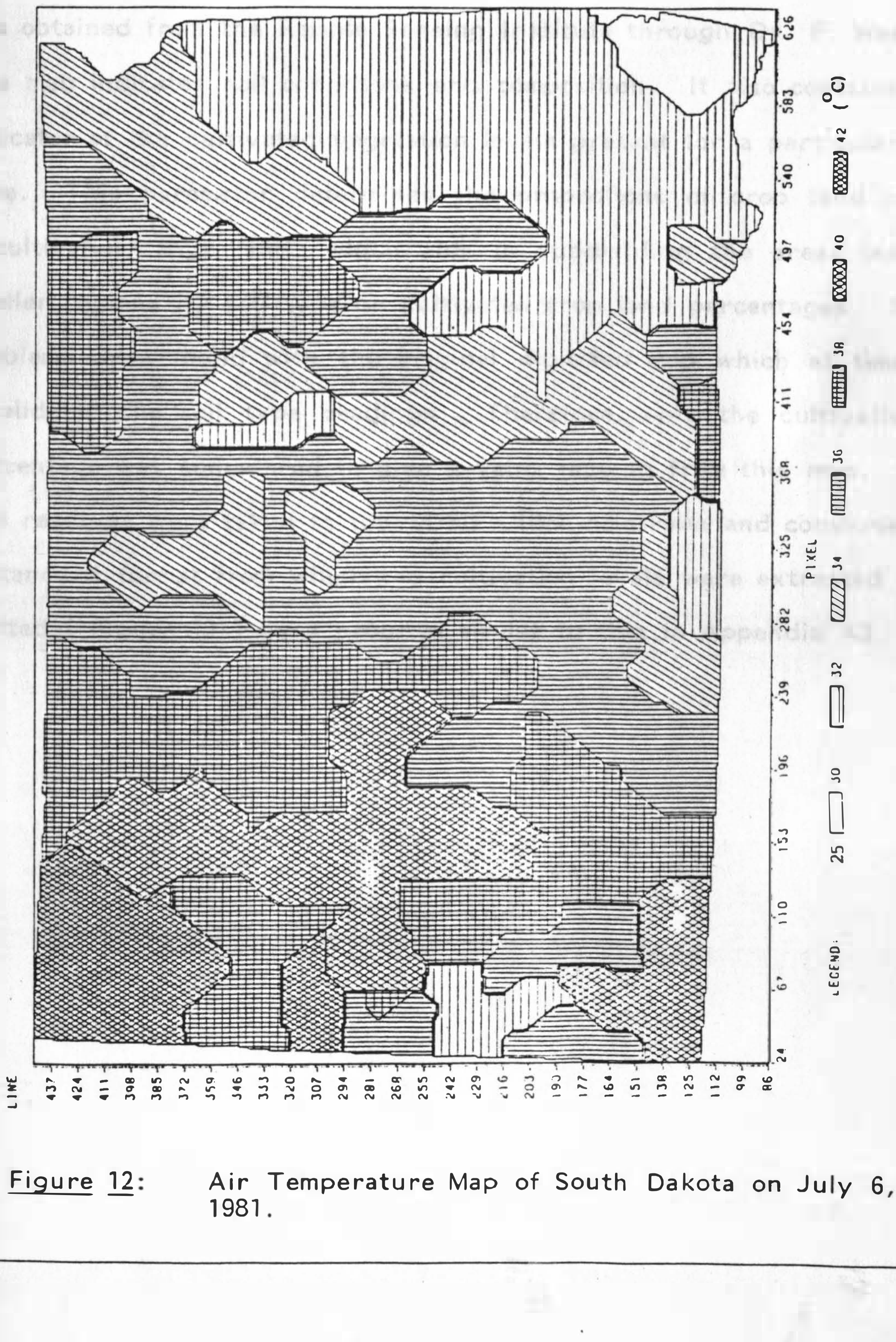

Figure 12: Air Temperature Map of South Dakota on July 6, 1981.

Crop Land Use Intensity

A digitized map of South Dakota indicating land cultivation percentages was obtained from the Remote Sensing Institute through Dr. F. Weston. The map indicates soil conditions and composition. It also contains an indicator of the cultivated vegetation levels present for a particular soil type. This parameter determines the proprotions of crop land or uncultivated range land, which aids in subdividing the areas into smaller regions by soil type or cultivated crop land percentages. Some problems were found with the original digitized map which at times invalidated the soil type readings. Therefore, only the cultivation percentage was considered to give a valid reading from this map. The map readings were taken to the SDSU's IBM mainframe and converted to a standard format from which the cultivation levels were extracted and plotted (Figure 13) by a program similar to that in Appendix A3.

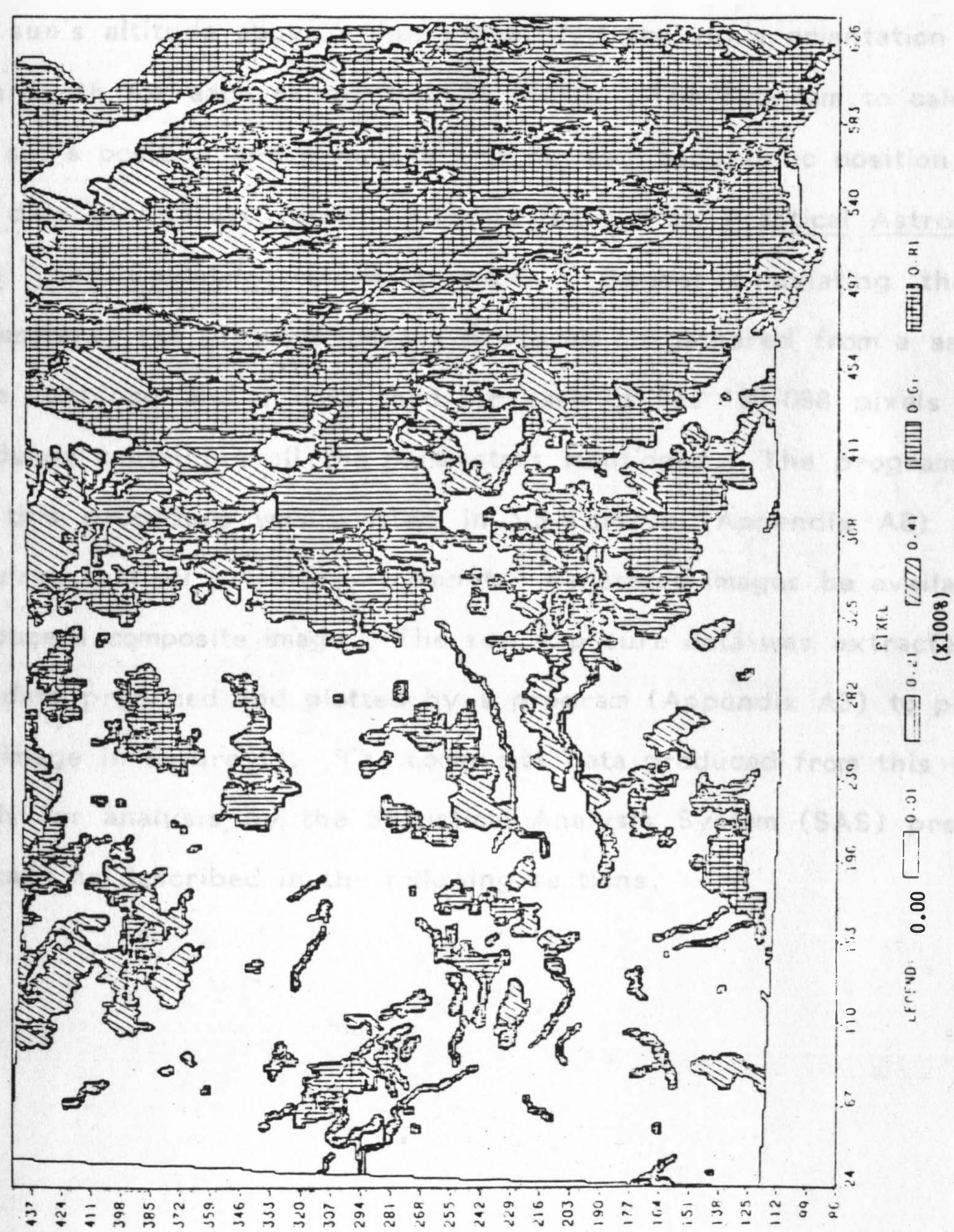

Figure 13: Cultivated Vegetation Map of South Dakota. The map indicates the percentage of land under cultivation.

Solar Parameters

A final calculation was done to determine the solar exposure time, the sun's altitude above a flat plane, and the sun's orientation from true South for each of the 190,098 pixels. The program to calculate the sun's position and exposure time for any geographic position, given the date of interest, used equations taken from Practical Astronomy with Your Calculator (Duffett-Smith). Besides calculating these parameters, the previous images produced or gathered from a satellite were combined and a data card for each of the 190,098 pixels was produced containing all the parameters mentioned. The program used for this procedure was written in VS-Fortran (Appendix A8) and requires that all satellite or computer generated images be available to produce a composite image. The sun exposure data was extracted from the data produced and plotted by a program (Appendix A3) to produce the image in Figure 14. The composite data produced from this step is ready for analysis by the Statistical Analysis System (SAS) program package as described in the following sections.

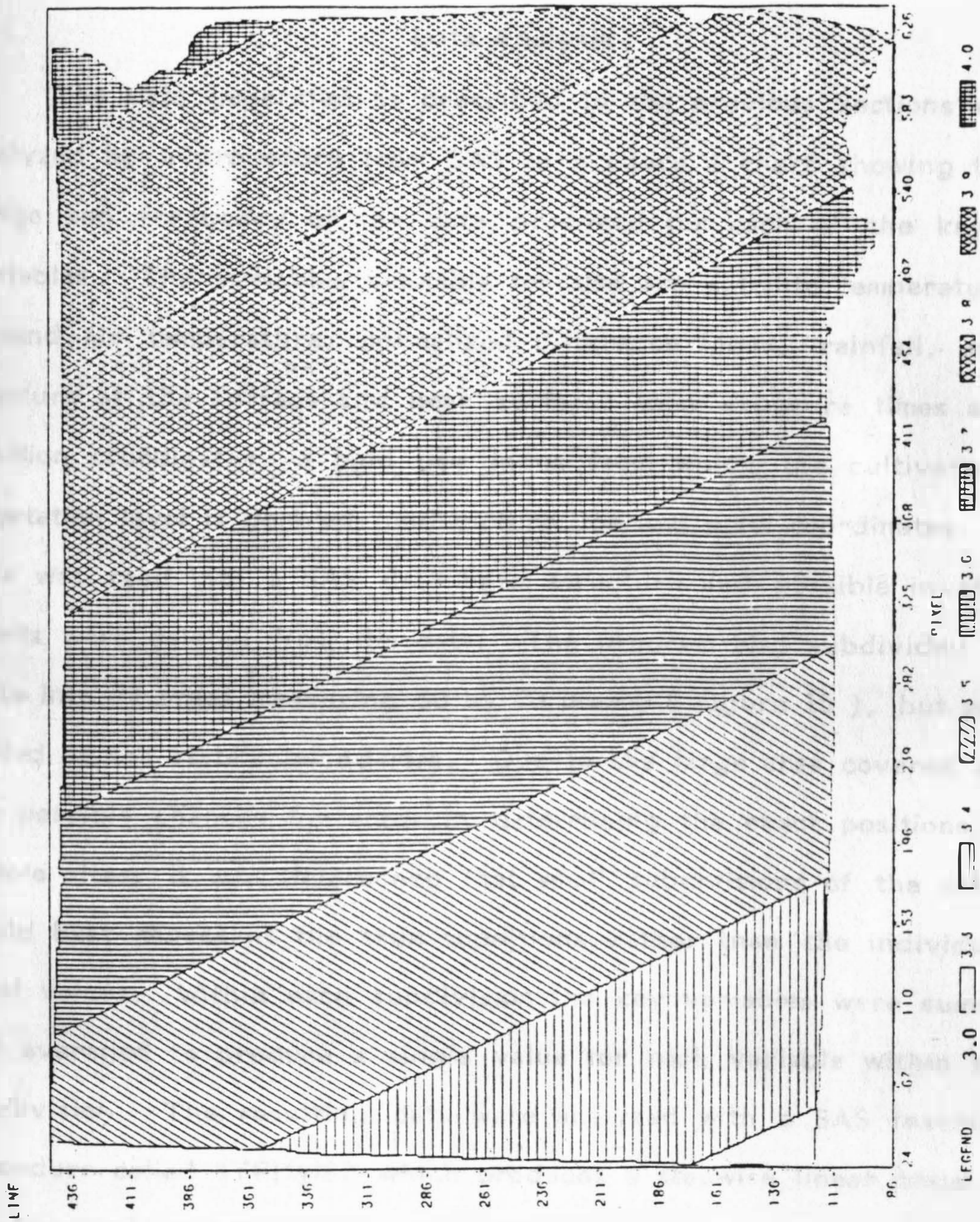

Figure 14: Solar Exposure Lengths for South Dakota Between Sunrise and 9:38 CDT; July 6, 1981.

Data Analysis

The data generated as described in the previous sections was analyzed by a procedure from SAS to produce a chart showing the range and frequency of the data produced for each of the key variables. The variables available for use are satellite temperature, ground air temperature, normalized difference index, rainfall, soil moisture at the surface and near surface, solar exposure times and position coordinates, a land use index from which the cultivated vegetation level is derived, and picture line and pixel coordinates. The data was read into a SAS data base structure and possible invalid points were deleted from the data. The program also subdivided the state into 71 areas measuring 50 by 70 pixels (Figure 15), but which varied at the state's boundaries. Due to the large area covered and the possible chances for error in determining the exact positions of sample sites, it was determined that small subdivisions of the state would best represent the true conditions rather than the individual pixel values. Within each subdivision the key variables were summed and averaged to produce a single value for each variable within the subdivision. The resulting data were analyzed with a SAS Institute procedure called STEPWISE which produces a stepwise linear model for the given dependent variable with the available independent variables.

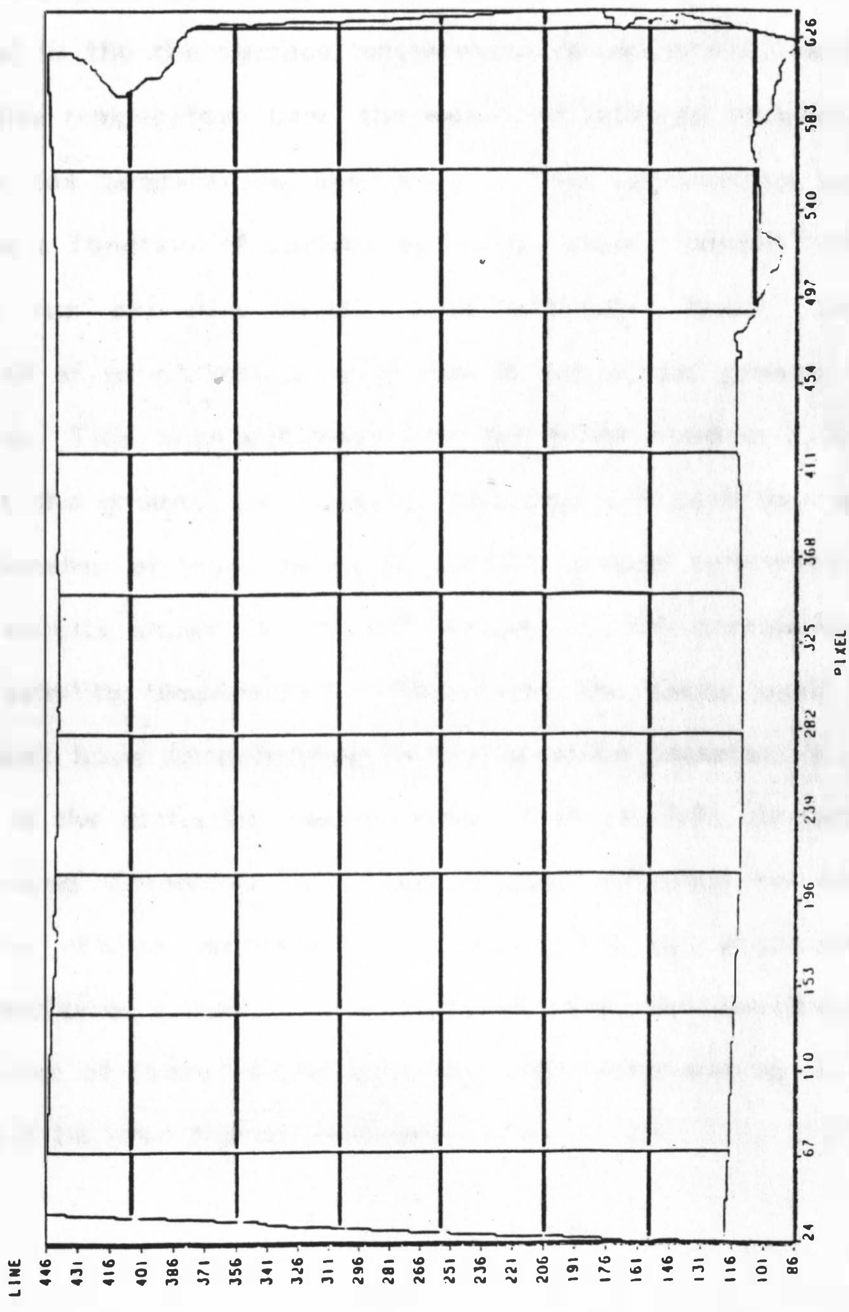

Figure 15: South Dakota Subdivided Into 71 Areas.

Satellite Temperature Modelling

According to the Stefan-Boltzman law, the thermal emittance is proportional to the the surface temperature raised to the fourth power. The satellite temperature used the measured infrared emittance, which resulted in the temperature used here. This temperature was then analyzed as a function of surface air temperature, rainfall, normalized difference, sun exposure length, land cultivation levels, and soil moisture; all of which may play a role in the actual ground surface temperature. This does not mean that the terms used in this equation will predict the ground temperature, but they are used because of the close relationship of these terms to surface ground temperature. The statistical results shown in Table 1 indicate a high correlation to the measured satellite temperature. Therefore, the terms used in the equation must have some bearing on the satellite temperature. A close inspection of the statistical results shows that rainfall, air temperature, and normalized difference have the biggest influence on satellite temperature. These parameters are followed by sun exposure length and soil moisture values. This indicates that the possibility of predicting one of these values from an equation consisting of the other terms should be investigated further.

Table 1: Satellite Temperature Modelling R^2 Statistic Results

Observations are divided into 71 regions and the mean of each parameter within the region is used.

Satellite temperature=surface air temperature*(a) + rainfall*(b) + normalized difference*(c) + Sun exposure length*(d) + Land cultivation level*(e) + soil moisture at surface*(f) + soil moisture at near surface*(g) + Constant

Observation count	R^2 result
71	0.902

Ground Air Temperature Modelling

Using the air temperature as the dependent variable in the linear equation process in place of satellite temperature also results in a high degree of correlation. This is somewhat surprising, since the air temperature measurements used are the daily high temperature readings, which do not normally take place until the afternoon, while the satellite readings used are from a morning overpass. The statistical analysis shows a significant correlation of the afternoon air temperature to the satellite temperature term. Other terms of lesser significance are the solar exposure length, vegetation cultivation level and normalized difference. The vegetation cultivation level, soil moisture at surface, rainfall, and sun exposure length parameters have negative correlations, while the soil moisture at near surface, satellite temperature and normalized difference, have positive correlations. The

negative correlation for solar exposure length is caused by the solar exposure times at the eastern side of the state being greater than the western side during morning hours, while the western side of the state has higher afternoon temperature readings. This indicates that the afternoon temperature readings can be predicted from morning satellite temperature readings because of patterns seen in the morning satellite readings, if other factors remain constant.

Table 2: Air Temperature Modelling R^2 Statistic Results

Air temperature=Soil moisture at surface*(a) + Soil moisture at near surface*(b) + Rainfall*(c) + Normalized difference*(d) + Sun exposure length*(e) + Land cultivation level*(f) + Satellite temperature*(g) + constant

Observation count	R^2 result
71	0.874

Rainfall Modelling

Because of the significance of the rainfall parameter in both of the previous model calculations, the next step was to replace air temperature with rainfall as the dependent variable. The air temperature is placed back into the independent variable list and the soil moisture variables deleted due to the large weighting effect that they cause on the model. When the statistics are computed, the results

show a very good correlation for the model using the subdivided region values, while the model using all valid pixel locations yields a R-squared statistic which is not as good as the regional model. This indicates that some problems exist with positioning of overlaying images and data that does not give an accurate value for small areas compared to that for the larger regions. The significance levels for the variables in the model indicates that a high correlation exists with satellite temperature and normalized difference, while the sun exposure length, cultivation levels, and air temperature all have progressively less significance. The negative correlations for normalized difference, satellite and air temperature parameters are acceptable due to the drying effect temperatures have on moisture readings, while high greenness levels will indicate possible inaccurate cool readings. A positive correlation for the vegetation cultivation level is thought to be accurate when farming practices in heavier rainfall areas are considered. The negative correlation for sun exposure lengths are due to the drier conditions that normally would exist in the western areas of the state, together with shorter sun exposure lengths in the morning hours.

Due to the possible weighting of the equation for sun exposure lengths, the parameter was removed from the independent variable list. The resulting R-squared value is about the same, with only slight variations in the estimated parameters for the end equation.

Table 3: Rainfall Modelling R^2 Statistic Results

Rainfall = surface air temperature*(a) + normalized difference*(b) + Land cultivation level*(c) + Sun exposure length*(d) + satellite temperature*(e) + Constant

Observation count	Includes sun exposure lengths R^2 result	Excludes sun exposure lengths R^2 result
71	0.617	0.612

Since it is obvious that rainfall does affect soil moisture levels, the next step is to determine if soil moisture can be determined as well as rainfall. The rainfall dependent variable is now replaced by the soil moisture measurements and the rainfall variable is placed back into the model, while all soil moisture variables are removed. The resulting model yields a R-squared statistic of 0.362 for surface soil moisture and 0.419 for near surface moisture. The equations for both models are somewhat similar with rainfall, sun exposure time and normalized difference all being significant.

Table 4: Soil Moisture Modelling R^2 Statistic Results; Step 1.

Soil moisture=surface air temperature*(a) + rainfall*(b) + normalized difference*(c) + Sun exposure length*(d) + Land cultivation level*(e) + satellite temperature*(f) + Constant

Observation count	R^2 for surface	R^2 for near surface
71	0.362	0.419

Soil Moisture Modelling

Because rainfall does weight the statistics, and it is a variable that would also be determined from the satellite, it was removed from the model. The new model shows the significance of satellite temperatures and normalized difference on soil moisture. The model still indicates the significance of sun exposure periods, which also is thought to be a weighted term, due to moist conditions that exist in the eastern part of the state and the greater amount of exposure time in the morning hours. This term is then eliminated from the modelling equation. The resulting statistics then indicate the significance of normalized difference, vegetation cultivation level and satellite temperature. This gives a lower R-squared correlation statistic of 0.190 for surface and 0.295 for near surface moisture.

Table 5: Soil Moisture Modelling R^2 Statistic Results; Step 2.

Soil moisture=surface air temperature*(a) + normalized difference*(b) + Sun exposure length*(c) + Land cultivation level*(d) + satellite temperature*(e) + Constant

Observation count	R^2 for surface	R^2 for near surface
71	0.237	0.332

Table 6: Soil Moisture Modelling R^2 Statistic Results; Step 3.

Soil moisture=surface air temperature*(a) + normalized difference*(b) + Land cultivation level*(c) + satellite temperature*(d) + Constant

Observation count	R^2 for surface	R^2 for near surface
71	0.190	0.295

In the model of Heilman and Moore (1981), an empirical correction to the satellite temperatures and empirical estimates were used to account for variety and cover density of crops appearing in the small areas under study. Since the estimate for crop density and cover is

already accounted for in the previous models by the normalized difference and vegetation cultivation level parameters, correction factors for satellite temperatures could be added to the independent variables. This would then result in a model similar to Heilman and Moore's, but would use calculated values instead of empirical values.

A statistical procedure from the SAS Institute called GLM (generalized linear modelling) was used which determines the model from a list of possible independent variables. The data was prepared for the procedure by including variables that where found to be significant in the previous modelling of satellite temperature to the satellite temperature term. This resulted in a large number of parameters, but only a few significantly affected the satellite temperature. This process was run several times to eliminate the insignificant terms from the independent variable list. The final result included the addition of four interaction terms which may correct for any deviation of the true satellite temperature. These terms are the satellite temperature divided by the product of air temperature and normalized difference, satellite temperature multiplied by air temperature, satellite temperature divided by the vegetation cultivation level and the product of satellite temperature and vegetation cultivation level divided by the normalized difference.

Results

For the final processing of the collected data, a new procedure called LEAPS provided to the SAS Institute by G. Furnival and R. Wilson (1974) was used. This procedure finds a specified number of models having the largest R^2 statistic for a subset of models with the same number of independent variables, where the number of independent variables increases from 1 to the total number of given independent variables. This procedure is similar to the stepwise modelling procedure, but performs each step on a number of possible models, letting the user choose the most valid model. Another change was introduced to test the restrictions placed on the model by the intercept term. This was done by declaring no intercept in the modelling procedure. The independent variables used included rainfall, soil moistures, air temperature, normalized difference, vegetation cultivation levels, satellite temperature, and the interaction terms found from the previous step, with the dependent variables excluded in each respective modelling.

This procedure is applied to the rainfall, surface moisture, and near surface moisture as described above. The results of these models indicated a significant gain in correlation for all of the models involved. A possible problem was seen in the models displayed for rainfall. This was a random sign change for the independent terms, a behavior which usually indicates some collinarity or influences by adjoining terms. The model was processed by the REG procedure from SAS, requesting collinarity and influence data for the variables listed. The results

indicated that the fourth interaction term exhibited collinarity and the other interaction terms also demonstrated some collinarity. This was as expected but not as the fourth term was doing. This term was eliminated from the rainfall model and the models were reprocessed with the estimated parameters being printed for both the reduced set of data and all 190,098 points.

For rainfall modelling, the results indicated a high correlation for only 2 independent variables and only slight improvement with additional terms. The model with the terms determined from the satellite were chosen to eliminate the need for ground collected data. Since it is desirable to use only satellite gathered terms for independent variables, as well as not including invalid interaction terms, a slightly lower r^2 value is acceptable. The resulting model involved the satellite temperature and normalized difference as indicated in Table 7. The plot shown in Figure 16 represents the reduced data, while the plot shown in Figure 17 shows the resulting plane from the equation determined by the model.

The modelling equation resulting from the same procedure applied to surface soil moistures yields some surprising results. The results from the procedure are shown in Table 8 , where rainfall and satellite temperature give the best correlation, and additional terms do not significantly influence the correlation. Plots shown in Figure 18 and Figure 19 show the actual data and the resulting equations plane. The data was then reprocessed with the normalized difference term included in all modelling equations. The results are shown in Table 9 and the

Table 7: Rainfall Modelling Equation Results

Rainfall=Satellite temperature*(a) + Normalized difference*(B)

Independent R^2 value is the value given for the model if only the respective parameter is used.

	State subdivided into 71 areas		All 190,098 pixels analyzed	
		R^2 result		R^2 result
		0.73		0.57
Parameter	Estimated value	Independent R^2 value	Estimated value	Independent R^2 value
a	-0.0525	0.15	-0.0414	0.08
b	0.0106	0.54	0.00950	0.43

plots exhibiting the actual data and the predicted equation plane are shown in Figure 20 and Figure 21. A significant similarity is found when the final model (Table 9) is compared to that shown in Table 8 with the model shown in Table 7 replacing the rainfall term of the equation. The resulting equations (Table 10) are alike in signs used for each term and the parameters are very nearly the same except for the model including all valid pixels, which is explained by the possible inaccuracy of surface moisture readings. These results indicate that rainfall and soil moistures are linked in the statistical model, just as they are in real world situations, demonstrating the validity of the modelling results.

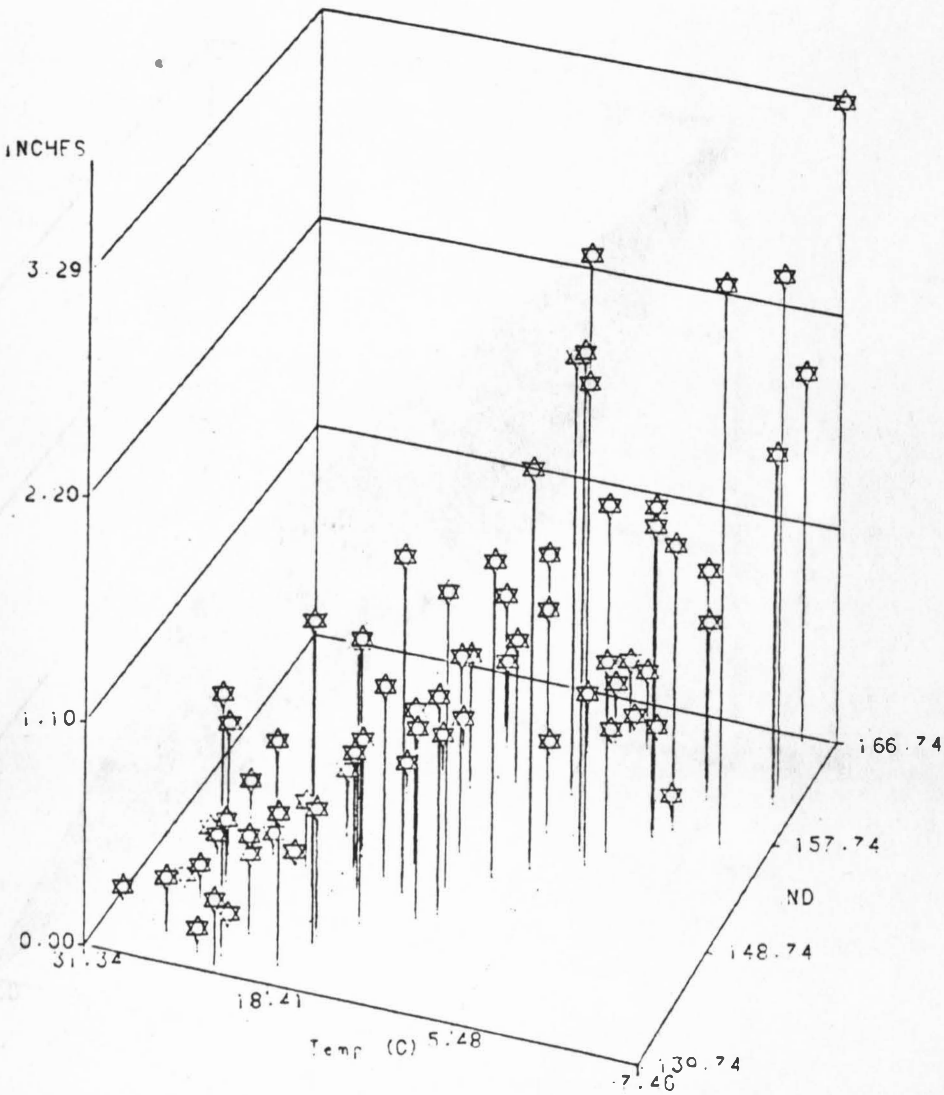

Figure 16: Scatter diagram of actual rainfall data by satellite temperature and normalized difference readings, from subdivided area data.

The final modelling equation is found for the near surface moisture which is processed similar to the previous models. The

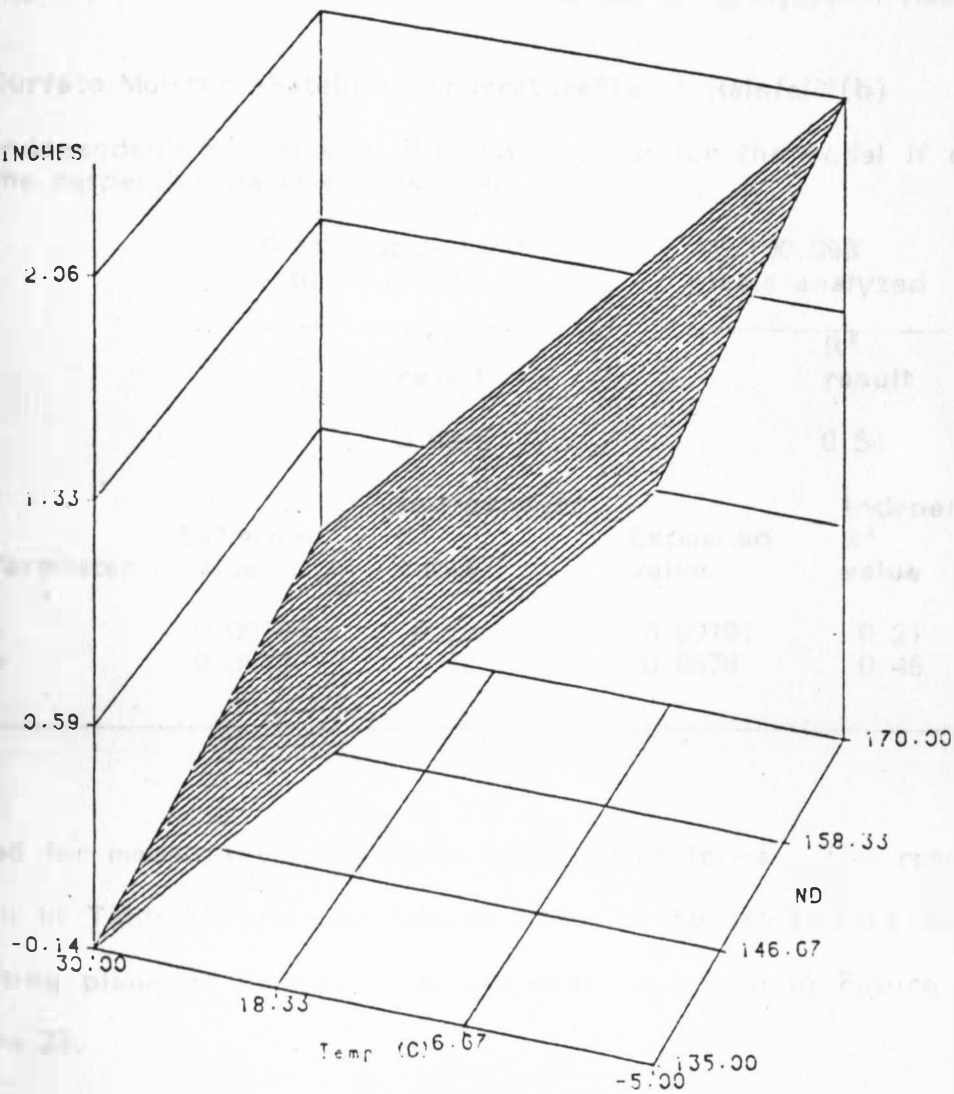

Figure 17: Predicted rainfall plane produced by the modelling equation using satellite temperature and normalized difference, equation from Table 7

results, as for the previous models, indicate that two independent terms will provide a significant correlation and only slight improvement is

Table 8: Primary Surface Moisture Modelling Equation Results

Surface Moisture=Satellite temperature*(a) + Rainfall*(b)

Independent R^2 value is the value given for the model if only the respective parameter is used.

	State subdivided into 71 areas		All 190,098 pixels analyzed	
		R^2 result		**R^2 result**
		0.68		0.54
Parameter	**Estimated value**	**Independent R^2 value**	**Estimated value**	**Independent R^2 value**
a	0.00184	0.29	0.00191	0.21
b	0.0683	0.62	0.0576	0.46

gained for models involving more independent terms. The results are shown in Table 11 and the related plots of the actual data and the resulting plane as derived from the equation shown in Figure 22 and Figure 23.

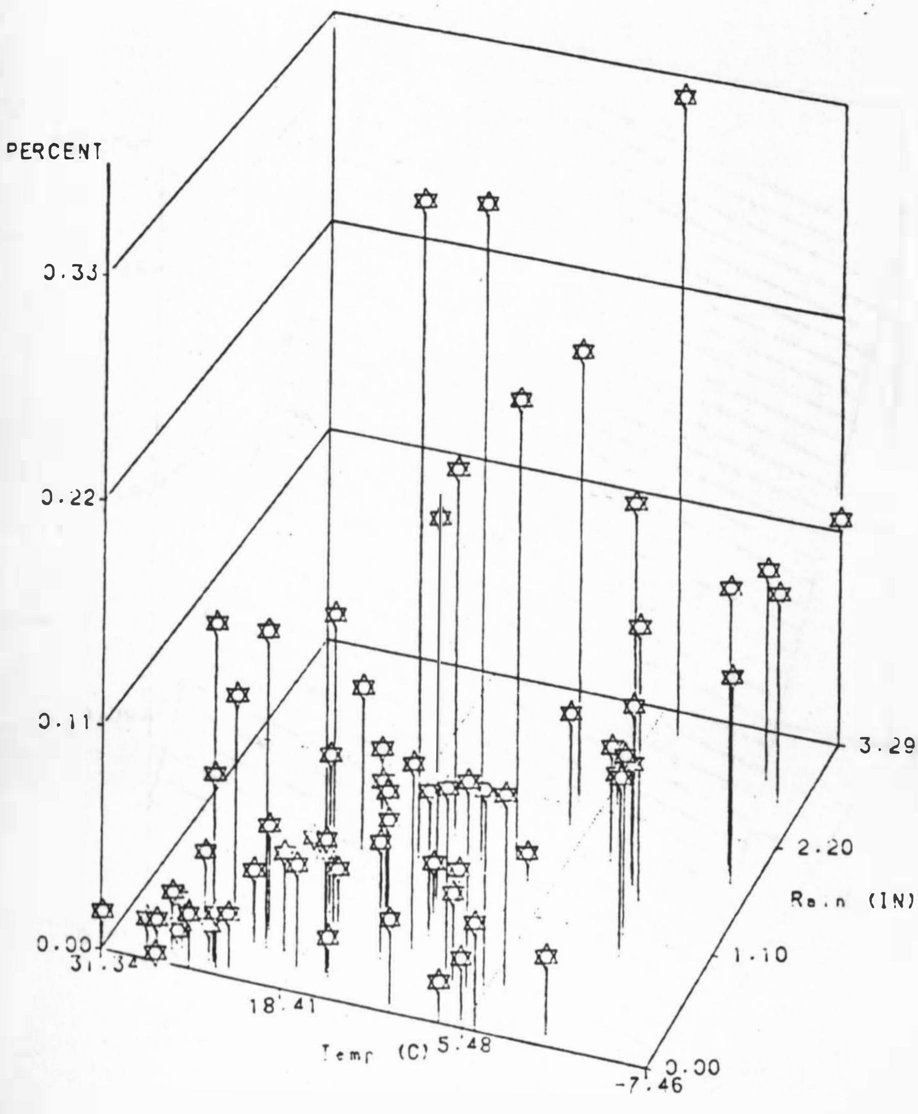

Figure 18: Scatter diagram of surface moisture data by satellite temperature and rainfall readings, from subdivided area data.

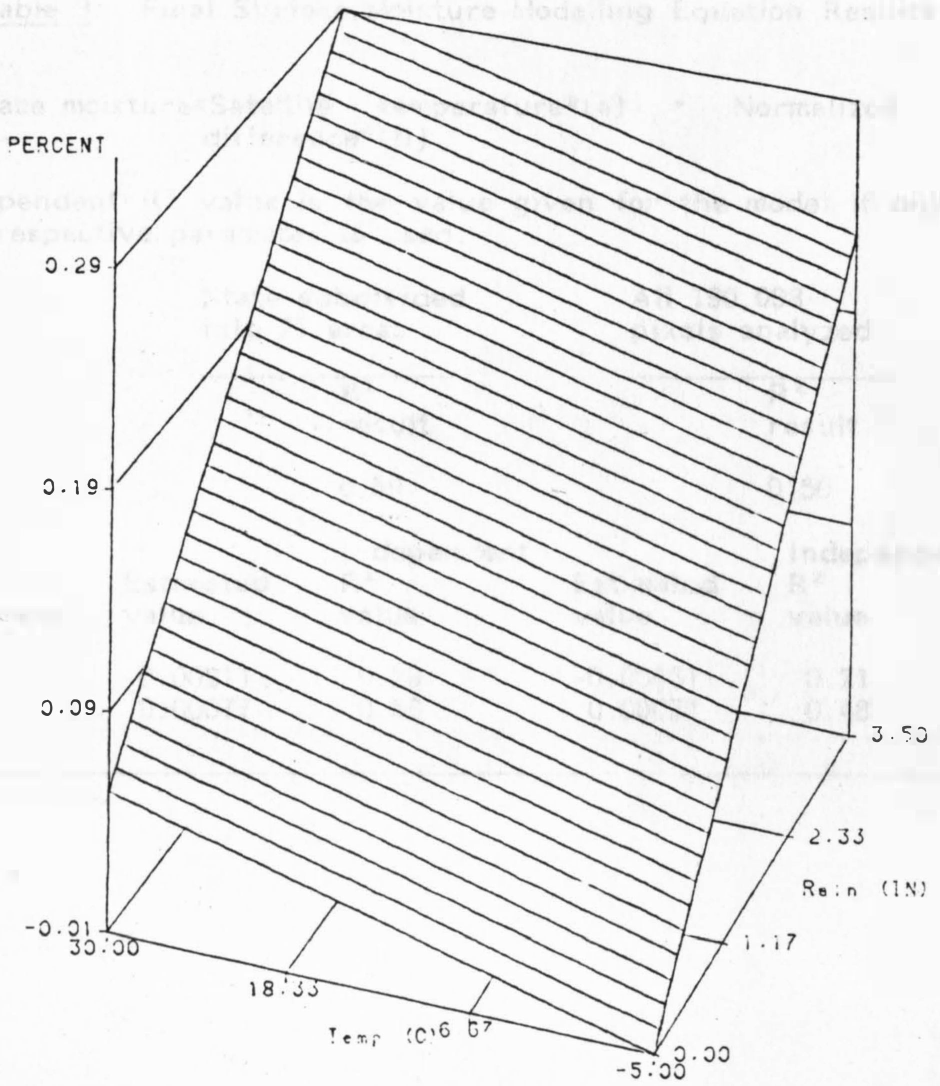

Figure 19: Predicted surface moisture plane from the primary modelling equation using satellite temperature and rainfall, equation from Table 8

Table 9: Final Surface Moisture Modelling Equation Results

Surface moisture=Satellite temperature*(a) + Normalized difference*(B)

Independent R^2 value is the value given for the model if only the respective parameter is used.

	State subdivided into 71 areas		All 190,098 pixels analyzed	
		R^2 result		R^2 result
		0.59		0.50
Parameter	Estimated value	Independent R^2 value	Estimated value	Independent R^2 value
a	-0.00211	0.29	-0.00151	0.21
b	0.00077	0.56	0.00071	0.48

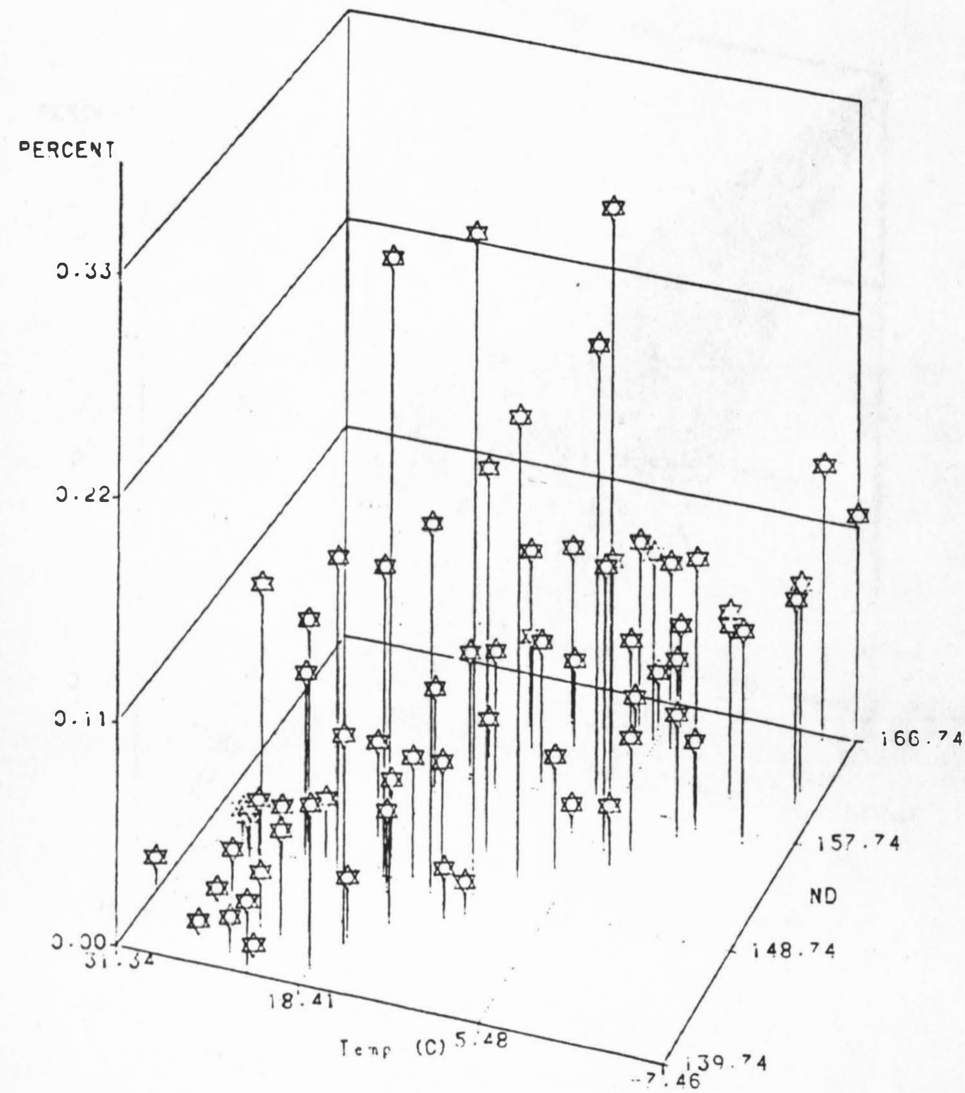

Figure 20: Scatter diagram of actual surface moisture data by satellite temperature and rainfall readings, from subdivided area data.

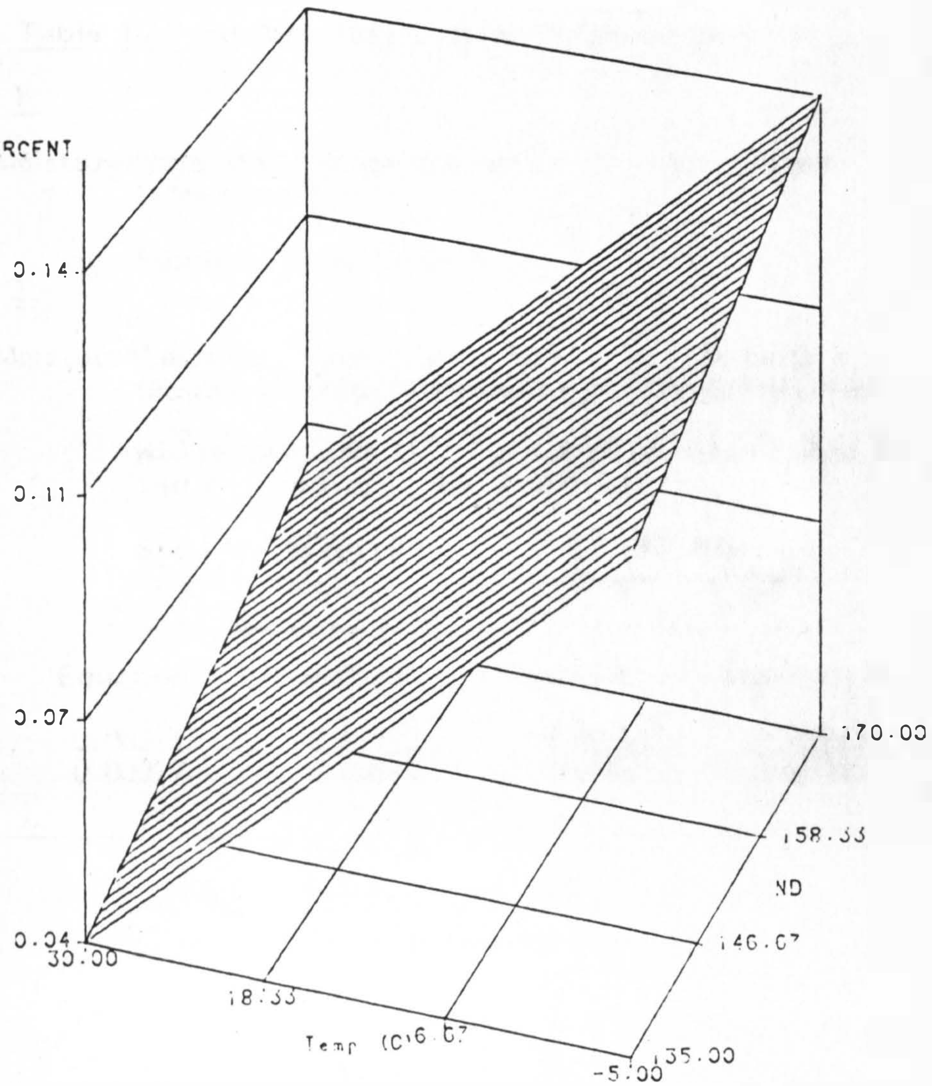

Figure 21: Predicted surface moisture plane from the final modelling equation using satellite temperature and normalized difference, equation from Table 9

Table 10: Surface Moisture Model Comparison

Equation 1:

Surface Moisture=Satellite temperature*(a) + Normalized Difference*(b)

Supplied from Table 9

Equation 2:

Surface Moisture=Satellite temperature*(a) + (satellite temperature*(c)+Normalized difference*(d))*(b)

Where (a) and (b) are supplied from Table 8 and (c) and (d) from Table 7

	State subdivided into 71 areas		All 190,098 pixels analyzed	
	Equation 1	Equation 2	Equation 1	Equation 2
a	-0.00211	-0.00175	-0.00151	-0.00048
b	0.00077	0.00072	0.00071	0.00055

Table 11: Near Surface Moisture Modelling Equation Results

Surface moisture=Satellite temperature*(a) + Normalized difference*(B)

Independent R^2 value is the value given for the model if only the respective parameter is used.

	State subdivided into 71 areas		All 190,098 pixels analyzed	
		R^2 result		R^2 result
		0.86		0.79
Parameter	Estimated value	Independent R^2 value	Estimated value	Independent R^2 value
a	-0.00175	0.49	-0.00127	0.39
b	0.00097	0.84	0.00091	0.78

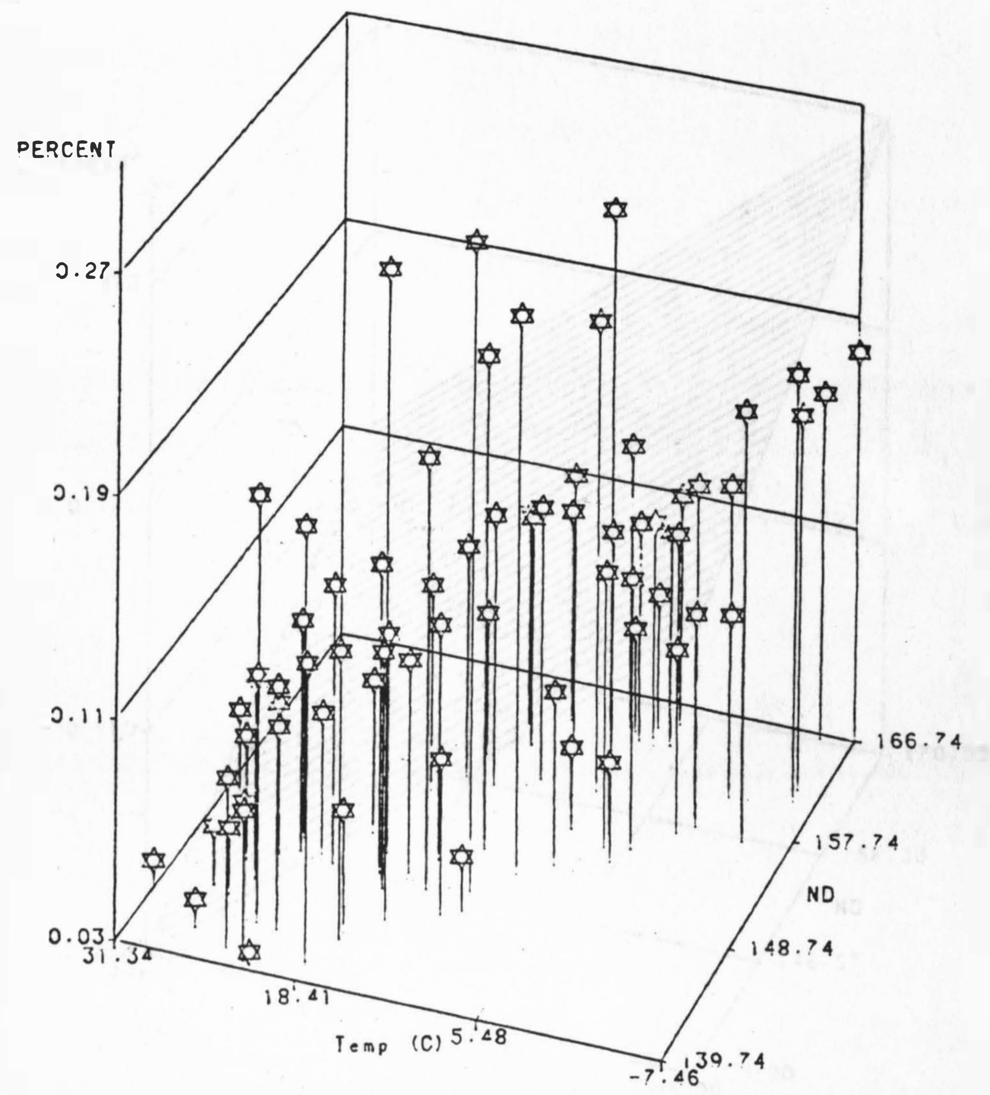

Figure 22: Scatter diagram of actual near surface moisture data by satellite temperature and normalized difference readings, from subdivided area data.

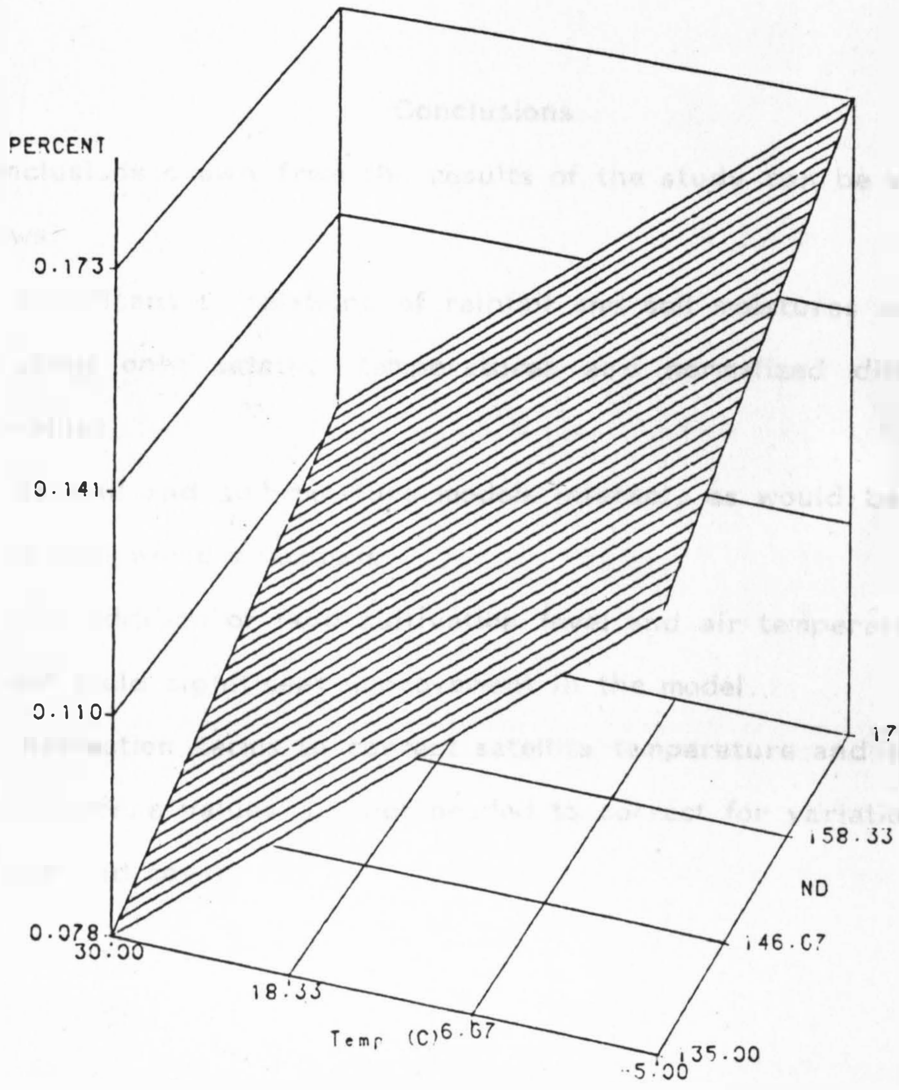

Figure 23: Predicted near surface moisture plane produced by the modelling equation using satellite temperature and normalized difference, equation from Table 11

Conclusions

The conclusions drawn from the results of the study can be summarized as follows:

1. Significant correlations of rainfall and soil moistures were found using only satellite temperatures and normalized difference values.
2. Rainfall and soil moisture models interact, as would be expected in real world situations.
3. The addition of land cultivation level and air temperatures does not yield significant improvement in the model.
4. Interaction terms to correct satellite temperature and normalized difference values are not needed to correct for variation of land use and cover.

Suggestions For Further Study

From the experiences gained with this study, the following ideas might be used to repeat this study for further verification of the findings. It was found that the sparse network of soil moisture sampling sites produced very crude estimations of soil moisture throughout the state when compared to the rainfall generated map. Another factor to be considered in future studies would be the availablity of clear, unobscured satellite imagery. In this study, six images were found corresponding to dates of interest, but only one image was found suitable for use.

A method to measure soil moistures electrically, like that employed for measuring house plant moisture conditions, should be investigated. This method should be compared to the microwave oven method used in this study and to a third method known to be accurate to test the accuracy of both.

If an accurate method is found, a network should be set up similar to that used for rainfall measurements. A suggested network might be one station per county township. Each station operator would read the soil moisture for a fixed probe location, air temperature and a rain gauge reading each day at a fixed time. This data could then be sent to a central location where images for each day could be generated from the collected data. These images could then be compared to

satellite images available over a larger number of days, resulting in a more accurate model.

Literature Cited

Beutler, G.A., 1980, Model Development for Predicting Soil Moisture By Thermography, South Dakota State University, M.S. Thesis, 77 pp.

Deering, D.W. and R.H. Hass, 1980, Using Landsat Digital Data For Estimating Green Biomass, Greenbelt, Maryland, Goddard Space Flight Center, NAS TM-80727, 21 pp.

Duffett-Smith, P., Practical Astronomy with Your Calculator, Cambridge, Cambridge University Press, 1979.

Heilman, J.L. and D.G. Moore, 1981, "Evaluating Near-Surface Soil Moisture Using Heat Capacity Mapping Mission (HCMM) Data," Evaluation of HCMM Data For Assessing Soil Moisture and Ground Water Table Depths, Final Report for Goddard Space Flight Center, Contract NAS 5-24206 Greenbelt, Maryland, pp. 96-107.

Kidwell, K.B., NOAA Polar Orbiter Data (TIROS-N, NOAA-6, NOAA-7, and NOAA-8) Users Guide, Washington, D.C., National Oceanic and Atmospheric Administration, 1984.

Meyer, C.R., 1972, Simulation of a Surface Thermal Anomaly, South Dakota State University, M.S. Thesis, (Unpublished).

Moore, D.G., M.L. Horton, M.J. Russel and V.I. Myers, 1975, "Evaluation of Thermal X/5 Detector Skylab S-192 Data for Estimating Evaporation and Thermal Properties of Soils for Irrigation Management," Proceedings of the NASA Earth Resources Survey Symposium, Houston, Texas, NAS TM-X58168:2561-2583.

Ness, S.D., 1982, <u>Relationship Between Soil Moisture and Soil Surface Temperature,</u> South Dakota State University, M.S. Thesis, 86 pp.

Price, J.C., 1980, "The Potential of Remotely Sensed Thermal Infrared Data to Infer Surface Soil Moisture and Evaporation," <u>Water Resources Research,</u> Vol. 16, No. 4, Pages 787-795, August 1980.

Price, J.C., 1984, "On the Analysis of Thermal Infrared Imagery: The Limited Utility of Apparent Thermal Inertia," <u>Remote Sensing of Environment,</u> Vol. 18, Pages 59-73.

Roesler, D.P., 1984, <u>Correlation of Soil Moisture With Soil Surface Temperature Obtained By Thermography,</u> South Dakota State University, M.S. Thesis, 64 pp.

Short, N.M., and L.M. Stuart Jr., <u>The Heat Capacity Mapping Mission (HCMM) Anthology,</u> NASA SP-465, 1982.

Snyder, J.P., <u>Map Projections Used by the U.S. Geological Survey,</u> Geological Survey Bulletin 1532, Washington, D.C., U.S. Government Printing Office, 1982.

Tunheim, J.A., 1977, <u>"Application of Heat-Flow Temperature Model for Remotely Assessing Near Surface Soil Moisture by Thermography,"</u> OWRT Completion Report, Project A-053-SDAK, Agreement No. 14-31-0001-6043.

Tunheim, J.A. and W.W. Hein, 1985, <u>"Utilization of Thermal Emittance Data to Monitor Soil Moisture and Subsurface Groundwater,"</u> Completion Report, Project B-068-SDAK, Agreement No. 14-34-0001-1263.

Appendix A

PROGRAMS

Configuration of System and Software Products Used

System

CPU: IBM 3031AP 8 Megabytes

Disk Storage: IBM 3340, 3344, 3350, 3380
Some programs require as much as 50 megabytes of scratch area.

Operating System: IBM MVS 1.3.3/JES2 1.3.4

Software

IBM Level F Assembler
IBM VS Fortran 1.4.1
SAS Version 5

Special Considerations

The following programs have been developed for the NOAA series of satellites and for coverage of a masked and registered image of South Dakota measuring 454 lines by 686 pixels. Other satellite series or areas used may require changes to the programs. To aid in these changes, all areas which need changing are marked as dependent areas, with the exception of storage array dimensions, do loop limits, etc., all of which involve the image size measurements.

Satellite Label and Calibration Reporting

```
//A1 JOB acctnumber,'READ LABEL',CLASS=G,TIME=2
//********************************************************************
//*                                                                  *
//*     READ FIRST 120 BYTES OF 1ST RECORD OF IMAGE TO GATHER        *
//*                   IMAGE DESCRIPTION LABEL                        *
//*                                                                  *
//********************************************************************
//LABEL EXEC PGM=IEBPTPCH
//SYSPRINT DD SYSOUT=*                  UTILITY MESSAGES DATA SET
//SYSPUNCH DD SYSOUT=*                  PUNCH OUTPUT DATA SET IF USED
//SYSUT2 DD SYSOUT=*                    PRINTED TAPE LABEL DATA SET
//SYSUT1 DD VOL=SER=IMAGE,LABEL=(1,NL),UNIT=TAPE,
// DCB=(RECFM=U,BLKSIZE=3700,OPTCD=Q),DISP=(OLD,PASS)      INPUT TAPE
//SYSIN DD *
  PRINT MAXFLDS=1,STOPAFT=1
  RECORD FIELD=(120,1,,5)
/*
//********************************************************************
//*                                                                  *
//*     CHANNEL 4 CALIBRATION INFORMATION FOR ENTIRE IMAGE           *
//*     IS READ IN, AND THE AVERAGE CALIBRATION READINGS             *
//*                       ARE PRINTED                                *
//*                                                                  *
//********************************************************************
//AVERAGE EXEC SAS,OUT='*'
//TAPEIN DD DISP=(OLD,KEEP),DCB=(RECFM=U,BLKSIZE=7400),
// UNIT=TAPE,VOL=SER=IMAGE,LABEL=(1,NL)
//FT12F001 DD SYSOUT=*                   PROC MEANS OUTPUT FOR IMAGE
//SAS.SYSIN DD *
DATA NEW(KEEP=NUM1 NUM2);
  INFILE TAPEIN FIRSTOBS=4;
        INPUT @37 NUM1 IB4. @41 NUM2 IB4.;
        INPUT @1 JUNK $CHAR1.;
PROC PRINTTO UNIT=12;
PROC MEANS DATA=NEW;
VAR NUM1 NUM2;
OUTPUT OUT=OLD MEAN=C1 C2;
PROC PRINT;
/*
```

```
//A2 JOB acctnumber,'CONVERT SATELLITE',CLASS=G,TIME=10,
// MSGCLASS=D,REGION=7000K
//CONVERT EXEC ASMFCLG,OUT='*'
//ASM.SYSIN DD *
********************************************************************************
************* STORING SYSTEM REGISTERS IN SYSTEM SAVE AREA ************
********************************************************************************
MAIN     CSECT
           PRINT NOGEN
           USING MAIN,15         USE REG 15 FOR TEMPORARY BASE REGISTER
           STM   14,12,12(13)    STORING SYTEMS REGS IN SYSTEM SAVE AREA
           LA    12,SAVEAREA     ADDR OF PROGRAM SAVE AREA FOR SYS REGS
           ST    13,4(12)        ADDR OF SYSTEM SAVE INTO PROGRAM SAVE
           ST    12,8(13)        ADDR OF PROGRAM SAVE INTO SYSTEM SAVE
           LR    13,12           POINT TO PROGRAM SAVE AREA
           B     PROGRAM         BRANCH TO PROGRAM AREA
           DROP  15              STOP USING 15 AS BASE REGISTER
           USING SAVEAREA,13     USE 13 FOR NEW BASE REGISTER
SAVEAREA DS    18F
********************************************************************************
******************** DATA CONTROL BLOCK DEFINITIONS ********************
********************************************************************************
DISKIO   DCB   DSORG=PS,MACRF=(GM,PM),RECFM=FB,LRECL=2744,                    X
               BLKSIZE=2744,EODAD=ENDIO,DDNAME=DISK
TAPEIN   DCB   DSORG=PS,MACRF=GM,RECFM=FB,BLKSIZE=30870,EODAD=ENDIO,          X
               DDNAME=INPUT,LRECL=686
PRINTOUT DCB   DSORG=PS,MACRF=PM,RECFM=FA,BLKSIZE=133,DDNAME=OUTPUT
PROGRAM  EQU   *
********************************************************************************
           EJECT             SKIP TO TOP OF PAGE
********************************************************************************
********************************************************************************
*    I * M * A * G * E    C * O * N * V * E * R * S * I * O * N               *
*          WRITTEN BY RANDY CHILSON......JULY 19, 1983                        *
*  PROGRAM TO READ A TAPE FILE OF IMAGE DATA AND CONVERT THE DATA             *
*  FROM THE TRANSMITTED BIT FORM INTEGERS IN BYTES TO CHARACTER               *
*  INTEGERS FOR FURTHER PROCESSING.  CONVERTED IMAGE IS THEN STORED           *
*  AS A TEMPORARY FILE ON SCRATCH DISK AS DATA SET NAME OF:                   *
*                          &&IMAGE.                                           *
********************************************************************************
*************** REGISTERS IN USE IN PROGRAM LOGIC ********************
********************************************************************************
*     REG 1       USED FOR PARMLIST TRANSFER TO SUBROUTINE FOR PRINT          *
*     REG 2       USED IN SUBROUTINE FOR PRINTOUT DCB                         *
*     REG 5       USED FOR INCREMENTED COUNTER IN CONVERSION                  *
*     REG 6       USED FOR BINARY BYTE STORAGE AREA                           *
*     REG 8       USED FOR LINEOUT VIRTUAL STORAGE AREA REFERENCE             *
*     REG 10      USED FOR INCREMENTING BYTES CONVERTED                       *
*     REG 11      USED FOR INCREMENTING NUMBER OF FULLWORDS FORMED            *
```

```
*     REG 12        USED FOR LINEIN VIRTUAL STORAGE AREA REFERENCE       *
*      --           USED AS BASE REGISTER IN SUBROUTINE                  *
*     REG 13        USED AS BASE REGISTER IN MAIN PROGRAM                *
*    REGS 14 AND 15   RESERVED FOR BRANCH AND LINK                       *
*************************************************************************
*             STORAGE AREAS FOR CONVERSIONS                              *
*************************************************************************
          B    START
DUMMY    DS   0D                 FORCING DOUBLE WORD ALINEMENT
INTEGER  DS   1D                 DOUBLE WORD FOR PACKED INTEGER
NUMBER   DS   CL4                MISC NUMBER STORAGE AREA
TEMP     DS   CL4                TEMPORARY INTEGER STORAGE
PATTERN  DC   X'40202120'        UNPACKING PATTERN
IMAGE    DS   CL2744             IMAGE LENGTH TIMES 4 BYTES
ONE      DC   F'1'               INCREMENTING VALUE
FOUR     DC   F'4'               INCREMENTING VALUE
*************************************************************************
*   THIS AREA PASSES THE ADDRESS OF THE PRINTOUT AND LINEOUT AREA       *
*    TO THE PRINT SUBROUTINE.                                           *
*************************************************************************
PARMS    DC     A(PRINTOUT)     AREA FOR ADDRESS OF PRINTOUT DCB
*************************************************************************
*************************************************************************
*            AREA FOR RESERVING VIRTUAL STORAGE AREA TO BE USED         *
*         FOR INPUT AND OUTPUT AREAS BY PROGRAM AND REFERENCING THEM    *
*         BY STANDARD NAMES BASED AT A SPECIFIED BASE REGISTER          *
*************************************************************************
START    EQU  *
****************************** LINEOUT **********************************
         GETMAIN R,LV=136   GET 136 BYTES OF VIRT. STORAGE FOR LINEOUT
         USING  LINEOUT,8   LOADS ADDRESS OF LINEOUT IN BUFFER BASED
         LR     8,1         STORES VIRTUAL STORAGE AREA AT REG 8
****************************** LINEIN ***********************************
         GETMAIN R,LV=686    GETS 686 BYTES OF VIRT. STORAGE FOR LINEIN
         USING   LINEIN,12   BASES LINEIN IMAGE AT REG 12
         LR    12,1          STORES VIRTUAL STORAGE AREA AT REG 12
         EJECT
*************************************************************************
*************************************************************************
*  PRODUCING A FORMATTED INTEGER DISK FILE FROM BINARY BYTE TAPE DATA *
*************************************************************************
         OPEN   (TAPEIN,(INPUT),DISKIO,(OUTPUT))              OPENS FILES
         OPEN   (PRINTOUT,(OUTPUT))               OPENS PRINTER OUTPUT FILES
MORE     LA     11,IMAGE         LOADS IN BEGINNING ADDRESS OF IMAGE AREA
         LA     10,LINEIN          LOADS IN BEGINNING ADDRESS OF INPUT LINE
         L      5,=F'686'          INITIALIZE TO TOTAL NO. OF INPUT BYTES
         GET    TAPEIN,LINEIN      GETS DATA FROM TAPEIN TO LINEIN
REPEAT   SR     6,6              ZEROING OUT R6
         IC     6,0(0,10)          MOVE IN 8 BITS TO RIGHT MOST BIT OF R6
         CVD    6,INTEGER          CONVERTING BINARY TO PACK NUMBER
         MVC    TEMP(4),PATTERN          MOVING IN BLANK PATTERN AREA
```

```
         EDMK  TEMP(4),INTEGER+6       UNPACKING PACKED NUMBER
         MVC   0(4,11),TEMP            MOVING INTEGER NUMBER TO CORRECT
*                                    POSITION OF IMAGE (REG 10)
         A     10,ONE         STEPPING R10 BY 1 BYTE (IMAGE IN)
         A     11,FOUR        STEPPING R11 BY 1 FULLWORD (IMAGE OUT)
         BCT   5,REPEAT       BRANCHING TO REPEAT TILL ALL BYTES CONVERTED
***********************************************************************
*   PRINTING SAMPLED IMAGE OUTPUT FOR VERIFICATION                    *
***********************************************************************
         MVI   LINEOUT,C' '                  MOVES IN CARRIAGE CONTROL
         MVC   LINEOUT+1(132),IMAGE+400   SELECTS 400 BYTES INTO IMAGE
         L     15,=A(PRINT)                LOADS ADDRESS OF PRINT ROUTINE
         LA    1,PARMS                LOADS PARMS INTO REG 1 FOR TRANSFER
         BALR  14,15                  BRANCHING TO SUBROUTINE
***********************************************************************
*  WRITES OUT COMPLETE CONVERTED IMAGE LINE TO DISK AND BRANCHES      *
*             FOR MORE DATA TILL END OF FILE                          *
***********************************************************************
         PUT   DISKIO,IMAGE     PUTS TRANSFORMED DATA TO DISKIO
         B      MORE            BRANCHES FOR MORE DATA
***********************************************************************
*  CLOSING UP FILES AND FREEING ALL VIRTUAL STORAGE AREAS             *
***********************************************************************
ENDIO    CLOSE   (TAPEIN,,DISKIO)    CLOSES OPENED FILES
         CLOSE    (PRINTOUT)           CLOSES PRINTER FILES
         FREEMAIN R,LV=686,A=(12)      FREES VIRTUAL STORAGE FOR LINEIN
         FREEMAIN R,LV=136,A=(8)       FREES VIRTUAL STORAGE AREAS
         L     13,4(13)                LOADING BACK SYSTEM SAVE AREA
         LM    14,12,12(13)            LOADING SYSTEMS REGS
         LA    15,0                    SETS CONDITION CODE TO ZERO
         BR    14                      BRANCHES TO SYSTEM
       EJECT
***********************************************************************
***********************************************************************
*             PRINT SUBROUTINE STARTS HERE                            *
***********************************************************************
***********************************************************************
PRINT    CSECT
         DROP   13                 DROPS USE OF REG 13 AS BASE REG.
         USING  PRINT,15           USES REG 15 FOR TEMP. BASE REG.
         STM   14,12,12(13)        STORES REG 1-15 EXCEPT FOR 13
         LA     12,SUBSAVE         LOADS ADDRESS OF SAVEAREA INTO R12
         ST     13,4(12)           STORES REG 13 AT 4 OFFSET OF R12
         ST     12,8(13)           STORES ADDRESS AT 8 OFFSET OF R13
         LA     13,SUBSAVE         LOADS ADDRESS OF SUBSAVE INTO R13
         B     AROUND              BRANCH AROUND NON-EXECUTABLE STMTS
SUBSAVE  DS    18F               SUBROUTINE SAVE AREA
         DROP   15                 STOPS USE OF REG 15 AS BASE REG.
         USING  AROUND,12          STARTS USE OF REG 12 AS BASE REG.
AROUND   L      2,0(1)           LOADS ADDRESS OF PRINTOUT INTO REG 2
         PUT    0(2),LINEOUT       PRINTS LINEOUT
```

```
         MVI     LINEOUT,C' '      MOVES IN BLANK FOR CLEARING OUT AREA
         MVC     LINEOUT(132),LINEOUT+1    CARRIES BLANK THROUGH AREA
         L       13,4(13)          LOADS ADDRESS OF SUBROUTINE SAVE AREA
         LM      14,12,12(13)      RELOAD REGS
         BR      14                BRANCHES TO MAIN PROGRAMS LAST STMT+1
        EJECT
***********************************************************************
***********************************************************************
        DSECT          IMAGES FOR VIRTUAL STORAGE SPECIFIED HERE
***********************************************************************
*        MISCS AREA USED TO IMAGE VIRTUAL STORAGE AREA                *
***********************************************************************
LINEOUT  DS    CL133             OUTPUT AREA FOR PRINTOUT
LINEIN   DS    CL686             INPUT IMAGE AREA
         EJECT
***********************************************************************
***********************************************************************
         END     MAIN              ALL DONE
***********************************************************************
***********************************************************************
//GO.SYSUDUMP     DD SYSOUT=Z
//GO.OUTPUT DD SYSOUT=Z
//*    THIS DD CARD IS CHANGED FOR DIFFERENT IMAGE FILES THAT
//*      ARE TO BE READ OFF OF THE TAPE
//GO.INPUT DD DSN=RAW.TEMP.JL6,DISP=(OLD,KEEP),
// UNIT=TAPE,VOL=SER=IMAGE1,LABEL=(13,SL)
//GO.DISK DD DSN=&&IMAGE,VOL=SER=ZZRTCH,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(CYL,(10,2),RLSE,CONTIG,ROUND)
//TEMP EXEC FORTVCLG,GOREGN=7000K,PARM='XREF,OPT(3),GOSTMT',          X
// FVREGN=4500K,FVLNSPC='3200,(75,20)',OUT='*'
//FORT.SYSUDUMP DD SYSOUT=*
//FORT.SYSIN  DD *
C.................................................................
C      THIS PART OF THE PROCESS TAKES THE TEMPORARY DATA SET
C       &&IMAGE AND CONVERTS THE READING TO THE TEMPERATURE
C       AS CALCULATED BY THE EQUATION FROM NOAA USERS GUIDE,
C       USING THE CALIBRATION READINGS AS READ BY A PREVIOUS
C       PROGRAM.  THE OUTPUT WILL BE IN F8.4 FORMAT.
C      THIS PROGRAM IS SET UP FOR A 686 BY 454 ELEMENT IMAGE
C          IF DIFFERENT THESE VALUES MUST BE CHANGED
C.................................................................
C        SETTING UP PROGRAM STORAGE AREAS
C.................................................................
      REAL*8 TEMP(686,454)
      INTEGER*2 IMAGE(686,454)
      CHARACTER*1 INPUT(686)
      INTEGER VALUE,STNID
      REAL*8 TEMP1,TEMP2,TEMP81
      REAL*8 N,M1,M2,INLAT,INLONG,CONVRS,ALNGTH,ECCENT
      REAL*8 PRLAT1,PRLAT2,ORGLAT,ORGLNG,LINE,PIXEL
      REAL*8 QPRT1,QPRT2,QPRT3,QPRT4,QPRT5,QPRT23
```

```
      REAL*8 Q1,Q2,Q0,Q,C,ROWE,ROWE0,THETA
C..............................................................
C     READING IN INPUT FROM ASSEMBLER CONVERSION PROGRAM
C..............................................................
      DO 901 L=1,454
      READ(10,900) (IMAGE(I,L),I=1,686)
 900  FORMAT(200I4,200I4,200I4,186I4)
 901  CONTINUE
C..............................................................
C     CONVERTS TO ACTUAL TEMP, OR SET TO -999.99 TO INDICATE
C              MASK REGION SURROUNDING IMAGE
C..............................................................
      DO 922 L=1,454
      DO 605 I=1,686
      IF (IMAGE(I,L).EQ.0) THEN
           TEMP2=-726.99
      ELSE
      IF (IMAGE(I,L).LT.0) IMAGE(I,L)=-1*IMAGE(I,L)
      TEMP1=152.26-(0.15465*(IMAGE(I,L)*4))
      TEMP1=(1.1910659E-5*(927.22**3)/TEMP1)
      IF (TEMP1.LT.0) THEN
           TEMP2=146
      ELSE
           TEMP2=(1.438833*927.22)/DLOG((1+TEMP1))
      ENDIF
      ENDIF
      TEMP(I,L)=TEMP2-273
 605  CONTINUE
 922  CONTINUE
C..............................................................
C     READ USER INPUT OF DESIRED LOCATION OF A VALUE TO
C     BE PRINTED.  USES ALBERS EQUAL CONICAL AREA METHOD
C     TO CONVERT THE GEOGRAPHICAL LOCATION INPUT AND
C     CHANGE IT TO A LINE AND PIXEL VALUE.
C..............................................................
      CONVRS=3.141592654/180
C*************************************************************
C*     BELOW CONSTANTS ARE SATELLITE AND IMAGE LOCATION       *
C*      DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE        *
C*                AND OF SOUTH DAKOTA AREA                    *
C*************************************************************
      ALNGTH=6378206.4
      ECCENT=0.0822719
      PRLAT1=42.75*CONVRS
      PRLAT2=45.25*CONVRS
      ORGLAT=(44)*CONVRS
      ORGLNG=(-100)*CONVRS
C******************** END OF DEPENDENT CONSTANTS ********************
      WRITE (6,9)
  9   FORMAT('0','STATION ID',' LONGITUDE ',4X,' LATITUDE ',4X,
     -' LINE  ',4X,' PIXEL ',5X,'TEMP')
```

```
50    READ (5,10,END=3333) STNID,INLONG,INLAT
10    FORMAT(I4,F8.4,F8.4)
       INLONG=INLONG*CONVRS
       INLAT=INLAT*CONVRS
       M1=COS(PRLAT1)/((1-(ECCENT**2)*((DSIN(PRLAT1))**2))**0.5)
       M2=COS(PRLAT2)/((1-(ECCENT**2)*((DSIN(PRLAT2))**2))**0.5)
       QPRT1=(1-ECCENT**2)
       QPRT2=(1+ECCENT*DSIN(ORGLAT))
       QPRT3=(1-ECCENT*DSIN(ORGLAT))
       QPRT23=DLOG(QPRT3/QPRT2)
       QPRT4=1/(2*ECCENT)
       QPRT5=(1-((ECCENT**2)*((DSIN(ORGLAT))**2)))
       QPRT5=(DSIN(ORGLAT)/QPRT5)
       Q0=QPRT1*(QPRT5-QPRT4*QPRT23)
       QPRT1=(1-ECCENT**2)
       QPRT2=(1+ECCENT*DSIN(PRLAT1))
       QPRT3=(1-ECCENT*DSIN(PRLAT1))
       QPRT23=DLOG(QPRT3/QPRT2)
       QPRT4=1/(2*ECCENT)
       QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT1))**2)))
       QPRT5=(DSIN(PRLAT1)/QPRT5)
       Q1=QPRT1*(QPRT5-QPRT4*QPRT23)
       QPRT1=(1-ECCENT**2)
       QPRT2=(1+ECCENT*DSIN(PRLAT2))
       QPRT3=(1-ECCENT*DSIN(PRLAT2))
       QPRT23=DLOG(QPRT3/QPRT2)
       QPRT4=1/(2*ECCENT)
       QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT2))**2)))
       QPRT5=(DSIN(PRLAT2)/QPRT5)
       Q2=QPRT1*(QPRT5-QPRT4*QPRT23)
       QPRT1=(1-ECCENT**2)
       QPRT2=(1+ECCENT*DSIN(INLAT))
       QPRT3=(1-ECCENT*DSIN(INLAT))
       QPRT23=DLOG(QPRT3/QPRT2)
       QPRT4=1/(2*ECCENT)
       QPRT5=(1-((ECCENT**2)*((DSIN(INLAT))**2)))
       QPRT5=(DSIN(INLAT)/QPRT5)
       Q=QPRT1*(QPRT5-QPRT4*QPRT23)
       N=((M1**2)-(M2**2))/(Q2-Q1)
       C=(M1**2)+(N*Q1)
       ROWE=(ALNGTH*((C-N*Q)**0.5))/N
       ROWE0=(ALNGTH*((C-N*Q0)**0.5))/N
       THETA=N*(INLONG-ORGLNG)
       X=ROWE*(DSIN(THETA))
       Y=ROWE0-(ROWE*(DCOS(THETA)))
C*************************************************************************
C*      BELOW EQUATIONS ARE SATELLITE AND IMAGE LOCATION                 *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE                 *
C*                  AND OF SOUTH DAKOTA AREA                             *
C*************************************************************************
       LINE=((231000-Y)/1000)+1.5
```

```
      PIXEL=((X+352000)/1000)+1.5
C******************** END OF DEPENDENT EQUATIONS ********************
      TEMP81=0
      IPIXEL=INT(PIXEL)
      ILINE=INT(LINE)
      DO 555 J=ILINE-1,ILINE+1
      DO 666 K=IPIXEL-1,IPIXEL+1
      TEMP81=TEMP(K,J)+TEMP81
  666 CONTINUE
  555 CONTINUE
      TEMP81=TEMP81/9
      WRITE (6,7) STNID,INLONG/CONVRS,INLAT/CONVRS,LINE,PIXEL,TEMP81
  7   FORMAT(' ',3X,I4,3X,F10.5,4X,F10.5,5X,F5.0,6X,F5.0,5X,F8.4)
      GOTO 50
C...........................................................................
C         NOW WRITE THIS OUT TO A PERMANENT FILE
C          WILL BE IN REAL FORMAT OF F8.4THE
C          TEMPERATURES OR -999.99 WHICH
C          INDICATES THE IMAGE MASK AREA
C...........................................................................
 3333 DO 3000 J=1,454
      WRITE(11,3001) (TEMP(I,J),I=1,686)
 3001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
 3000 CONTINUE
      END
/*
//GO.FT05F001 DD *      (CHECK ONE POINT IN IMAGE)
0070-098.08544.377
/*
//GO.FT10F001 DD DSN=&&IMAGE,VOL=SER=ZZRTCH,DISP=(OLD,DELETE),
// UNIT=SYSDA
//GO.FT11F001 DD DSN=TEMP.IMAGE.JUL23,DISP=(NEW,KEEP),
// VOL=SER=RANDYC,UNIT=TAPE,LABEL=(1,SL),
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=27440,DEN=4)
```

Raster to Vector Plotting

```
//SASTEST JOB 1034091100,RLC,CLASS=M,TIME=999,REGION=6500K,MSGCLASS=D,
// TYPRUN=HOLD
//* Requires notification of system operator, due to
//* due to possibly large execution times.
//SAS EXEC SAS,OUT='*',TIME=999,REGION=6500K,PENS='LIQB'
//PLOT DD DSN=TEMP.IMAGE.JUL23,DISP=OLD,
// VOL=SER=RANDYC,UNIT=TAPE,LABEL=(1,SL),
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=27440,DEN=4)
//SYSIN DD *
************************************************************************;
*                                                                      *;
*            FOLLOWING JOB PRODUCES A PLOTTED DIGITIZED                *;
*            RASTER IMAGE ON A VECTOR PLOTTER.                         *;
*                                                                      *;
************************************************************************;
*;
*;
************************************************************************;
*                                                                      *;
*            READ IN INPUT FROM TAPE IN F8.X FORMAT.                   *;
*            LINESIZE IS FIXED AT 686: CHANGE ALL THE                  *;
*            686 VALUES TO THE APPROPRIATE LINESIZE                    *;
*                                                                      *;
************************************************************************;
DATA RAWDATA(KEEP=X VALUE);
      INFILE PLOT CLOSE=DISP;
             DO X=1 TO 686;
                OFFSET=1+((X-1)*8);
                INPUT @OFFSET VALUE 8. @;
                OUTPUT RAWDATA;
              END;
            RETURN;
************************************************************************;
*                                                                      *;
*            PRODUCE LINE AND PIXEL COORDINATES AS WELL AS             *;
*            CHANGE THE IMAGE TO APPEAR IN CORRECT                     *;
*            DIRECTIONS OF WEST ON LEFT AND NORTH ON TOP.              *;
*            CHANGE 686 VALUES TO THE APPROPRIATE LINESIZE             *;
*                                                                      *;
************************************************************************;
 DATA IMAGE(KEEP=LINE PIXEL VALUE);
  RETAIN YX 1;
      SET RAWDATA;
             YX=YX+1;
             PIXEL=X;
             LINE=INT(YX/686);
             LINE=(LINE+454)-((LINE*2)-1);
             IF VALUE=-999.99 THEN DELETE;
             ELSE OUTPUT IMAGE;
```

```
**********************************************************************************;
*                                                                               *;
*         PLOT IMAGE ON PEN PLOTTER WITH PATTERN AND JOIN                      *;
*         OPTIONS ON FOR CONTOUR ROUTINE, NO LEVELS ARE                        *;
*         DEFINED, BUT MAY BE IF DESIRED: SEE SAS/GRAPH                        *;
*         USERS MANUAL FOR DETAILS.                                            *;
*                                                                               *;
**********************************************************************************;
GOPTIONS DEVICE=CAL1051 HSIZE=10.5 VSIZE=8.5 NOTEXT82;
 TITLE1 C=BLACK F=SIMPLEX H=1 ' ';
 PROC GCONTOUR DATA=IMAGE;
  PLOT LINE*PIXEL=VALUE/PATTERN JOIN CAXIS=BLACK;
  PATTERN V=M1N90 R=4;
  PATTERN V=M1N0 R=4;
  PATTERN V=M1L R=4;
  PATTERN V=M1R R=4;
  PATTERN V=M1X R=4;
  PATTERN V=M1X45 R=4;
  PATTERN V=M2X R=4;
  PATTERN V=M2X45 R=4;
```

Generating Rainfall Images

```
//A4 JOB acctnumber,'MAKE RAINFALL MAP',CLASS=M,TIME=200,
// MSGCLASS=D,REGION=7000K
//* Requires notification of system operator, due to possibly
//* very large execution times (varies with number of available
//* points of ground gathered data)
//IMAGE EXEC FORTVCLG,GOREGN=7000K,PARM='XREF,OPT(3),GOSTMT',             X
// FVREGN=4500K,FVLNSPC='3200,(75,20)',OUT='*',TIME.GO=200
//FORT.SYSIN  DD *
C..............................................................
C          SETTING UP PROGRAM STORAGE AREAS
C..............................................................
      INTEGER VALUE,STNID
      REAL*8 N,M1,M2,INLAT,INLONG,CONVRS,ALNGTH,ECCENT
      REAL*8 PRLAT1,PRLAT2,ORGLAT,ORGLNG,LINE,PIXEL
      REAL*8 QPRT1,QPRT2,QPRT3,QPRT4,QPRT5,QPRT23
      REAL*8 Q1,Q2,Q0,Q,C,ROWE,ROWE0,THETA
      REAL*4 IMAGE(686,454),IMAGE2(686,454),VAL(9,2)
      REAL*4 DATA(100,100,3)
      REAL*4 LAT,LONG,FLAG,PIXL(8)
      INTEGER CNTY,STN,MONTH,DATE,PASS
C................................................................
C     FILL IN IMAGE WITH INITIAL VALUE FLAG FOR EMPTY CELLS
C................................................................
      DO 1 J=1,454
      DO 2 I=1,686
      IMAGE(I,J)=-1
    2 CONTINUE
    1 CONTINUE
C................................................................
C..        READ THE STATION LOCATION DATA INTO THE ARRAY              ..
C..     DATA (1,X,X)   =     STATION ID  COUNTY                     ..
C..     DATA (X,1,X)   =     STATION ID  INTERNAL COUNTY ID         ..
C..     DATA (X,X,1)   =     STATION LAT.                           ..
C..     DATA (X,X,2)   =     STATION LONG                           ..
C..     DATA (X,X,3)   =     STATION RAINFALL                       ..
C................................................................
  100 READ(5,102,END=101) CNTY,STN,LONG,LAT
  102 FORMAT(I2,1X,I2,15X,F10.5,F10.5)
      DATA(CNTY,STN,1)=LAT
      DATA(CNTY,STN,2)=LONG
      DATA(CNTY,STN,3)=-1
      GOTO 100
  101 CLOSE(UNIT=5,STATUS='KEEP')
C................................................................
C        NOW READ THE RAIN GAUGE REPORT DATA IN, AND DETERMINE
C        IF IT IS DURING THE DATES REQUESTED.  REQUESTED DATES
C          MODIFIED IN CODE, REMEMBER TO CHANGE WHEN NEEDED
C          WHEN CORRECT DATES ARE FOUND THE TOTAL DURING THE
C           PERIOD IS FILLED INTO THE ARRAY DESCRIPED ABOVE
```

```
C.....................................................................
  200 READ(10,201,END=202) CNTY,STN,MONTH,DATE,AMOUNT
  201 FORMAT(I2,I2,3X,I2,5X,I2,F4.2)
      Z=DATA(CNTY,STN,3)
      IF (Z.EQ.-1) DATA(CNTY,STN,3)=0
      IF (MONTH.NE.7) GOTO 200
      IF ((DATE.LT.$$).OR.(DATE.GT.%%)) GOTO 200
      DATA(CNTY,STN,3)=DATA(CNTY,STN,3)+AMOUNT
      GOTO 200
  202 CLOSE(UNIT=10,STATUS='KEEP')
C.....................................................................
C     WRITE THIS DATA OUT FOR A CHECK AND CONVERT TO LINE AND PIXEL
C
C.....................................................................
      DO 302 I=1,100
      DO 301 J=1,100
      IF ((DATA(I,J,1).EQ.0.0).OR.(DATA(I,J,3).EQ.-1)) GOTO 301
      WRITE(6,300) I,J,DATA(I,J,1),DATA(I,J,2),DATA(I,J,3)
  300 FORMAT(' ',I2,5X,I2,5X,F10.5,5X,F10.5,5X,F5.2)
C...................................................................
C    FINDING THE LINE AND PIXEL LOCATION FOR THESE STATIONS
C   REPORTING AND STORE LINE IN DATA(*,*,1) AND PIXEL IN DATA(*,*,2)
C       THEN REWRITE THIS DATA AGAIN AFTER THE CONVERSION
C          AND PLACE THE VALUE INTO THE IMAGE ARRAY
C....................................................................
      INLONG=(DATA(I,J,2)*(-1.0))
      INLAT=DATA(I,J,1)
      CONVRS=3.141592654/180
C*************************************************************
C*       BELOW CONSTANTS ARE SATELLITE AND IMAGE LOCATION       *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE        *
C*                 AND OF SOUTH DAKOTA AREA                     *
C*************************************************************
      ALNGTH=6378206.4
      ECCENT=0.0822719
      PRLAT1=42.75*CONVRS
      PRLAT2=45.25*CONVRS
      ORGLAT=(44)*CONVRS
      ORGLNG=(-100)*CONVRS
C******************* END OF DEPENDENT CONSTANTS *****************
      INLONG=INLONG*CONVRS
      INLAT=INLAT*CONVRS
      M1=COS(PRLAT1)/((1-(ECCENT**2)*((DSIN(PRLAT1))**2))**0.5)
      M2=COS(PRLAT2)/((1-(ECCENT**2)*((DSIN(PRLAT2))**2))**0.5)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(ORGLAT))
      QPRT3=(1-ECCENT*DSIN(ORGLAT))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(ORGLAT))**2)))
      QPRT5=(DSIN(ORGLAT)/QPRT5)
```

```
      Q0=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT1))
      QPRT3=(1-ECCENT*DSIN(PRLAT1))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT1))**2)))
      QPRT5=(DSIN(PRLAT1)/QPRT5)
      Q1=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT2))
      QPRT3=(1-ECCENT*DSIN(PRLAT2))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT2))**2)))
      QPRT5=(DSIN(PRLAT2)/QPRT5)
      Q2=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(INLAT))
      QPRT3=(1-ECCENT*DSIN(INLAT))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(INLAT))**2)))
      QPRT5=(DSIN(INLAT)/QPRT5)
      Q=QPRT1*(QPRT5-QPRT4*QPRT23)
      N=((M1**2)-(M2**2))/(Q2-Q1)
      C=(M1**2)+(N*Q1)
      ROWE=(ALNGTH*((C-N*Q)**0.5))/N
      ROWE0=(ALNGTH*((C-N*Q0)**0.5))/N
      THETA=N*(INLONG-ORGLNG)
      X=ROWE*(DSIN(THETA))
      Y=ROWE0-(ROWE*(DCOS(THETA)))
C*****************************************************************************
C*       BELOW EQUATIONS ARE SATELLITE AND IMAGE LOCATION                    *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE                     *
C*                   AND OF SOUTH DAKOTA AREA                                *
C*****************************************************************************
      LINE=((231000-Y)/1000)+1.5
      PIXEL=((X+352000)/1000)+1.5
C******************** END OF DEPENDENT EQUATIONS ******************
      DATA(I,J,2)=INT(PIXEL)
      DATA(I,J,1)=INT(LINE)
      X=DATA(I,J,2)
      Y=DATA(I,J,1)
      WRITE(6,304) I,J,DATA(I,J,1),DATA(I,J,2),DATA(I,J,3)
  304 FORMAT(5X,'**CONVERTED**  ',I2,5X,I2,5X,F10.5,5X,F10.5,5X,F5.2)
      IMAGE(X,Y)=DATA(I,J,3)
  301 CONTINUE
  302 CONTINUE
C.............................................................................
C  NOW TAKE THE RAINFALL AMOUNTS PLACED INTO THE IMAGE ARRAY
```

```
C     AND SPREAD THESE VALUES THROUGH OUT THE ENTIRE STATE
C     USING A 3X3 WINDOW FILLING THE CENTER WITH THE VALUE
C     WHICH IS THE MOST PREDOMINANT IN THE WINDOW.  KEEP ON
C      DOING THIS UNTIL NO -1 FLAGS APPEAR IN ANY CELL
C        EXCEPT VERY OUTER BORDER ROWS AND COLUMNS.
C.......................................................................
C.......................................................................
C
C.......................................................................
C   START THE ROUTINE OFF, BUT DON'T GO THROUGH CHECK FOR FINISH 1ST
C.......................................................................
      FLAG=-1
      GOTO 501
C.......................................................................
C        CHECK TO SET IF ITS DONE YET, ELSE UPDATE PASS COUNT
C                       AND CONTINUE ON
C.......................................................................
  500 IF (FLAG.EQ.0) GOTO 550
      FLAG=0
C.......................................................................
C      PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C.......................................................................
  501 DO 502,X=2,686-1
      DO 503,Y=2,454-1
C.......................................................................
C     SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C      SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                     FOUND AN EMPTY CELL
C.......................................................................
      IF (IMAGE(X,Y).NE.-1) GOTO 533
      FLAG=-1
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C.............................................................
C  NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C    FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C     BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
C         FOR HIGHEST STRIKE COUNT AT INDEX 2
C.............................................................
      DO 504,I=1,9
      VAL(I,1)=-9
      VAL(I,2)=0
  504 CONTINUE
      VAL(1,1)=-1
C.............................................................
```

```
C      BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C        ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C         STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C         IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                VALUE ARRAY FOR NEXT PASS
C...........................................................
      DO 520,J=1,8
      I=1
  505 IF (VAL(I,1).EQ.-9) GOTO 515
      IF (VAL(I,1).NE.PIXL(J)) GOTO 510
      VAL(I,2)=VAL(I,2)+1
      GOTO 520
  515 VAL(I,1)=PIXL(J)
      VAL(I,2)=1
      GOTO 520
  510 IF(I.EQ.9) GOTO 520
      I=I+1
      GOTO 505
  520 CONTINUE
C.................................................................
C    FIND MAJOR GRID VALUE AND STICK THIS INTO EMPTY CELL
C      ELSE DEFAULT TO 1,1 OF 3X3 GRID WHICH IS -1 FLAG
C                  AND CONTINUE ON
C.................................................................
      J=0
      K=2
      DO 530 I=2,9
      IF (VAL(I,2).LT.J) GOTO 530
      J=VAL(I,2)
      K=I
  530 CONTINUE
      IMAGE2(X,Y)=VAL(K,1)
      GOTO 503
  533 IMAGE2(X,Y)=IMAGE(X,Y)
  503 CONTINUE
  502 CONTINUE
      DO 531 I=1,686
      DO 532 J=1,454
      IMAGE(I,J)=IMAGE2(I,J)
  532 CONTINUE
  531 CONTINUE
      GOTO 500
  550 CONTINUE
C.....................................................................
C   NOW SMOOTH OUT THE SINGLE VALUES FROM THE IMAGE SO IT ISNT
C     SO SPARATIC, AND FLAG AGAIN WITH -1 FOR A SECOND PASS
C     WITH THE ABOVE, ONLY THIS TIME CHECKING THAT THE
C            STRIKE VALUE IS GREATER THAT 1
C            WILL PUT IMAGE TO BE REWORKED
C               INTO IMAGE ARRAY 2
C.....................................................................
```

```
C...........................................................
C
C...........................................................
C      PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C...........................................................
  601 DO 602,X=2,686-1
      DO 603,Y=2,454-1
C...........................................................
C     SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C      SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                     FOUND AN EMPTY CELL
C...........................................................
      FLAG=-1
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C...................................................
C   NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C     FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C      BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
C          FOR HIGHEST STRIKE COUNT AT INDEX 2
C...................................................
      DO 604,I=1,9
      VAL(I,1)=-9
      VAL(I,2)=0
  604 CONTINUE
      VAL(1,1)=-1
C..................................................
C      BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C        ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C         STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C         IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                VALUE ARRAY FOR NEXT PASS
C..................................................
      DO 620,J=1,8
      I=1
  605 IF (VAL(I,1).EQ.-9) GOTO 615
      IF (VAL(I,1).NE.PIXL(J)) GOTO 610
      VAL(I,2)=VAL(I,2)+1
      GOTO 620
  615 VAL(I,1)=PIXL(J)
      VAL(I,2)=1
      GOTO 620
  610 IF(I.EQ.9) GOTO 620
      I=I+1
      GOTO 605
```

```
  620 CONTINUE
C.................................................................
C    FIND ALL CELLS IN OUTLINING GRID THAT ARE LIKE THE CENTER
C        IF COUNT IS GREATER OR EQUAL TO 2 THEN LEAVE ALONE
C           ELSE FLAG FOR NEXT PASS THROUGH
C.................................................................
      I=2
      Z=IMAGE(X,Y)
  631 IF (VAL(I,1).NE.Z) GOTO 632
      IF (VAL(I,2).GE.2) GOTO 630
      Z=-1.0
      GOTO 630
  632 I=I+1
      IF (I.LT.10) GOTO 631
      Z=-1.0
  630 CONTINUE
      IMAGE2(X,Y)=Z
  603 CONTINUE
  602 CONTINUE
      DO 651 I=1,686
      DO 652 J=1,454
      IMAGE(I,J)=IMAGE2(I,J)
  652 CONTINUE
  651 CONTINUE
C.....................................................................
C   NOW FOR THE FINAL PASSES OVER TO REPLACE THE FLAGGED PIXELS
C.....................................................................
C.....................................................................
C
C.....................................................................
C   START THE ROUTINE OFF, BUT DON'T GO THROUGH CHECK FOR FINISH 1ST
C.....................................................................
      FLAG=-1
      GOTO 701
C.....................................................................
C       CHECK TO SET IF ITS DONE YET, ELSE UPDATE PASS COUNT
C                         AND CONTINUE ON
C.....................................................................
  700 IF (FLAG.EQ.0) GOTO 750
      FLAG=0
C.....................................................................
C      PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C.....................................................................
  701 DO 702,X=2,686-1
      DO 703,Y=2,454-1
C.....................................................................
C     SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C      SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                   FOUND AN EMPTY CELL
C.....................................................................
      IF (IMAGE(X,Y).NE.-1) GOTO 733
```

```
      FLAG=-1
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C.............................................................
C   NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C     FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C      BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
C          FOR HIGHEST STRIKE COUNT AT INDEX 2
C.............................................................
      DO 704,I=1,9
      VAL(I,1)=-9
      VAL(I,2)=0
  704 CONTINUE
      VAL(1,1)=-1
C............................................................
C      BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C        ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C         STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C         IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                VALUE ARRAY FOR NEXT PASS
C............................................................
      DO 720,J=1,8
      I=1
  705 IF (VAL(I,1).EQ.-9) GOTO 715
      IF (VAL(I,1).NE.PIXL(J)) GOTO 710
      VAL(I,2)=VAL(I,2)+1
      GOTO 720
  715 VAL(I,1)=PIXL(J)
      VAL(I,2)=1
      GOTO 720
  710 IF(I.EQ.9) GOTO 720
      I=I+1
      GOTO 705
  720 CONTINUE
C..................................................................
C    FIND MAJOR GRID VALUE AND STICK THIS INTO EMPTY CELL
C            ELSE DEFAULT TO 1,1 OF 3X3 GRID
C                  AND CONTINUE ON
C..................................................................
      J=0
      K=2
      DO 730 I=2,9
      IF (VAL(I,2).LT.J) GOTO 730
      J=VAL(I,2)
      K=I
```

```
  730 CONTINUE
       IF (VAL(K,2).GT.1) IMAGE2(X,Y)=VAL(K,1)
       GOTO 703
  733 IMAGE2(X,Y)=IMAGE(X,Y)
  703 CONTINUE
  702 CONTINUE
       DO 731 I=1,686
       DO 732 J=1,454
       IMAGE(I,J)=IMAGE2(I,J)
  732 CONTINUE
  731 CONTINUE
       GOTO 700
  750 CONTINUE
C.......................................................................
C     FINALLY ALL DONE.   NOW WRITE PASS COUNT OUT AND A SAMPLE
C          OF THE COMPLETED IMAGE FOR A CHECK
C            THEN WRITE IMAGE TO A DISK FILE
C             AND USE SAS GCONTOUR TO PLOT
C.......................................................................
       DO 806 J=1,454
       WRITE(6,807) (IMAGE2(I,J),I=300,315)
  807 FORMAT(16F8.2)
  806 CONTINUE
       DO 4000 J=1,454
       WRITE(12,4001) (IMAGE2(I,J),I=1,686)
 4001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
 4000 CONTINUE
       END
/*
//GO.FT05F001 DD DSN=RAINFALL.STNLOC,VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=OLD,LABEL=(20,SL,,IN)
//GO.FT10F001 DD DSN=RAIN.GAUGE.DATA81,VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=OLD,LABEL=(21,SL,,IN)
//GO.FT12F001 DD DSN=RAINFALL.JL$$JL%%,DISP=(NEW,KEEP),
// VOL=SER=RANDYC,UNIT=TAPE,LABEL=(2,SL),
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=27440,DEN=4)
```

Selection of Ground Gathered Moisture Data Readings

```
//A5 JOB acctnumber,'GATHER MOISTURE DATA',CLASS=G,TIME=4,
// MSGCLASS=D,REGION=3000K
// EXEC SAS,OUT='*'
//SAMPLES  DD DSN=SOIL.SAMP.DATA81,UNIT=TAPE,VOL=SER=IMAGE1,DISP=OLD,
//             LABEL=(23,SL)
//STNLOC   DD UNIT=TAPE,VOL=SER=IMAGE1,DISP=OLD,LABEL=(22,SL),
//             DSN=SOIL.SAMP.STNLOC
//PNCHDATA DD SYSOUT=*
//SAS.SYSIN DD *
*****************************************************************************;
*                                                                           ;
*                                                                           ;
*                                                                           ;
*                                                                           ;
*****************************************************************************;
OPTIONS MISSING=' ';
*---   D * A * T * A    C * O * L * L * E * C * T * I * O * N  ------;
*------------ READ DATA FROM SOIL MOISTURE DATA --------------------;
DATA MOIST(KEEP=SITE DATE LEVEL0 LEVEL1);
   INFILE SAMPLES;
                         INPUT @1    SITE       $3.
                               @5    DATE        MMDDYY6.
                               @12   LEVEL0      3.3
                               @16   LEVEL1      3.3   @;
                          IF LEVEL0=0.999 THEN LEVEL0=.;
                          IF LEVEL1=0.999 THEN LEVEL1=.;
                         OUTPUT MOIST;
                         RETURN;
*----------- READ IN RELATED SITE LOCATION DATA -------------------;
 DATA LOCATION(KEEP=SITEID LONG LAT);
    INFILE STNLOC;
          INPUT  @27 SITEID    $3.
                 @5  LONG       8.3
                 @13 LAT        8.3   @;
              IF SITEID NE '' THEN OUTPUT LOCATION;
*--------- SORT MOISTURE READINGS BY DATE AND PRINT ------------;
 PROC SORT DATA=MOIST;
   BY SITE DATE;
 PROC PRINT DATA=MOIST;
       FORMAT DATE WEEKDATE29.;
*-------------------------------------------------------------;
*---------     FIND ALL DATA PERTAINING TO THE DATE           --;
*---------     DESIRED OR EITHER:                             --;
*---------      +++ A SAMPLE WITHIN 3 DAYS PRIOR AND WITHIN 10 --;
*---------          DAYS AFTER DESIRED DATE FOR INTERPOLATION  --;
*---------      +++ A SAMPLE WITHIN 10 DAYS PRIOR AND WITHIN 3 --;
*---------          DAYS AFTER DESIRED DATE FOR INTERPOLATION  --;
*-------------------------------------------------------------;
DATA OUTDATA(KEEP=SITEID MOISTO MOIST1);
```

```
%INCLUDE DATEIN;
 CENTER=INPUT(DAY,DATE10.);
        RETAIN BEFORE BMOISTO AFTER AMOISTO SITEID CENTER CORRECTO EOF
               BMOIST1 AMOIST1 CORRECT1;
     SITEID='000';
     EOF=0;
   READ:;
       IF SITEID=SITE THEN DO;
          CURRENT=DATE-CENTER;
          IF CURRENT GT 0 THEN DO;
                   IF CURRENT LT AFTER THEN DO;
                                 AMOIST1=LEVEL1;
                                 AMOIST0=LEVEL0;
                                 AFTER=CURRENT;
                    END;
          END;
          ELSE;
          IF CURRENT LT 0 THEN DO;
                    IF CURRENT GT BEFORE THEN DO;
                                 BMOIST0=LEVEL0;
                                 BMOIST1=LEVEL1;
                                 BEFORE=CURRENT;
                    END;
          END;
          ELSE;
          IF CENTER=DATE THEN DO;
                             CORRECT0=LEVEL0;
                             CORRECT1=LEVEL1;
                             BEFORE=0;
                             AFTER=0;
                             END;
       END;
       ELSE DO;
       IF ((BEFORE NE 0) AND (AFTER NE 0))  THEN DO;
          IF (((ABS(BEFORE) LE 3) AND (AFTER LE 10))
             OR ((ABS(BEFORE) LE 10) AND (AFTER LE 3))) THEN DO;
           XMOIST0=BMOIST0-AMOIST0;
           XMOIST1=BMOIST1-AMOIST1;
           XMOIST8=BMOIST8-AMOIST8;
           INCR0=XMOIST0/(ABS(BEFORE)+AFTER);
           INCR1=XMOIST1/(ABS(BEFORE)+AFTER);
           MOIST0=BMOIST0+(INCR0*BEFORE);
           MOIST1=BMOIST1+(INCR1*BEFORE);
           IF SITEID NE '000' THEN OUTPUT OUTDATA;
       END;
       ELSE;
       END;
       ELSE IF BEFORE=0 AND AFTER=0 THEN DO;
                 MOIST0=CORRECT0;
                 MOIST1=CORRECT1;
                 IF SITEID NE '000' THEN OUTPUT OUTDATA;
```

```
                END;
                IF ((DATE-CENTER) LT 0) THEN DO;
                                  BEFORE=DATE-CENTER;
                                  BMOIST0=LEVEL0;
                                  BMOIST1=LEVEL1;
                                  END;
                    ELSE DO;
                                  BEFORE=0;
                                  BMOIST1=0;
                                  BMOIST0=0;
                                  END;
                IF ((DATE-CENTER) GT 0) THEN DO;
                                   AFTER=DATE-CENTER;
                                   AMOIST1=LEVEL1;
                                   AMOIST0=LEVEL2;
                                   END;
                  ELSE DO;
                                   AFTER=999;
                                   AMOIST1=0;
                                   AMOIST0=0;
                                   AGREEN=0;
                                   END;
            SITEID=SITE;
       END;
       IF EOF NE 0 THEN GOTO ALLDONE;
       ELSE DO;
             SET MOIST END=EOF;
             GOTO READ;
       END;
       ALLDONE:;
      OUTPUT OUTDATA;
*------------- SORT OUTPUT DATASETS BY MERGE VARIABLE ---------------;
*-------------  JOIN BOTH TOGETHER BY SITEID, THEN    ---------------;
*-------------  PRINT OUT DATA AND PUT A FORMATTED    ---------------;
*-------------  LIST OUT TO PNCHDATA DD FOR INPUT     ---------------;
*-------------  TO THE IMAGE GENERATION PROGRAM.      ---------------;
PROC SORT DATA=OUTDATA;
   BY SITEID;
PROC SORT DATA=LOCATION; BY SITEID;
DATA JOIN; MERGE OUTDATA LOCATION; BY SITEID;
PROC PRINT DATA=JOIN;
DATA;
  SET JOIN;
       FILE PNCHDATA NOTITLES;
       PUT @1 SITEID $3.  @5 LONG 8.3
           @15 LAT 8.3
           @25 MOIST0 4.3 @35 MOIST1 4.3;
       RETURN;
//DATEIN DD *   DATE OF DATE DESIRED USE SAS DATE FORMAT
 RETAIN DAY '6JUL81';
```

Generation of Soil Moisture Image

```
//A6 JOB acctnumber,'MAKE MOISTURE MAP',CLASS=M,TIME=200,
// MSGCLASS=D,MSGLEVEL=(1,1),REGION=6500K
/*JOBPARM LINES=100,TIME=999,ROOM=18
//* Requires notification of system operator, due to possibly
//* very large execution times (varies with number of available
//* points of ground gathered data)
//GENERATE EXEC FORTVCLG,GOREGN=6000K,PARM='XREF,OPT(3),GOSTMT',      X
// FVREGN=4500K,FVLNSPC='3200,(75,20)',OUT='*',TIME.GO=300
//FORT.SYSIN  DD *
C.........................................................................
C            SETTING UP PROGRAM STORAGE AREAS
C.........................................................................
      INTEGER VALUE,STNID
      REAL*8 N,M1,M2,INLAT,INLONG,CONVRS,ALNGTH,ECCENT
      REAL*8 PRLAT1,PRLAT2,ORGLAT,ORGLNG,LINE,PIXEL
      REAL*8 QPRT1,QPRT2,QPRT3,QPRT4,QPRT5,QPRT23
      REAL*8 Q1,Q2,Q0,Q,C,ROWE,ROWE0,THETA
      REAL*4 IMAGE(686,454),IMAGE2(686,454),VAL(9,2)
      REAL*4 DATA(200,3),MOIST0,MOIST1
      REAL*4 LAT,LONG,FLAG,PIXL(8)
      INTEGER CNTY,STN,MONTH,DATE,PASS
C.........................................................................
C     FILL IN IMAGE WITH INITIAL VALUE FLAG FOR EMPTY CELLS
C.........................................................................
      DO 1 J=1,454
      DO 2 I=1,686
      IMAGE(I,J)=-1
    2 CONTINUE
    1 CONTINUE
C.........................................................................
C..         READ THE STATION LOCATION DATA INTO THE ARRAY              ..
C..      DATA (1,X)   =     STATION ID                                  ..
C..      DATA (X,1)   =     STATION LAT.                                ..
C..      DATA (X,2)   =     STATION LONG                                ..
C..      DATA (X,3)   =     STATION TEMP                                ..
C.........................................................................
      I=0
  100 I=I+1
      READ(5,102,END=101) LONG,LAT,MOIST0,MOIST1
  102 FORMAT(4X,F8.3,2X,F8.3,2X,F4.3,6X,F4.3)
      DATA(I,1)=LAT
      DATA(I,2)=LONG
      DATA(I,3)=MOIST1
      GOTO 100
  101 CONTINUE
C.........................................................................
C     WRITE THIS DATA OUT FOR A CHECK AND CONVERT TO LINE AND PIXEL
C
C.........................................................................
```

```
      DO 301 J=1,I
      IF ((DATA(J,1).EQ.0.0).OR.(DATA(J,2).EQ.0.0)) GOTO 301
      WRITE(6,300) J,DATA(J,1),DATA(J,2),DATA(J,3)
  300 FORMAT(' ',I3,5X,F10.5,5X,F10.5,5X,F5.2)
C..............................................................
C    FINDING THE LINE AND PIXEL LOCATION FOR THESE STATIONS
C   REPORTING AND STORE LINE IN DATA(*,*,1) AND PIXEL IN DATA(*,*,2)
C       THEN REWRITE THIS DATA AGAIN AFTER THE CONVERSION
C          AND PLACE THE VALUE INTO THE IMAGE ARRAY
C..............................................................
      INLONG=DATA(J,2)
      INLAT=DATA(J,1)
      CONVRS=3.141592654/180
C**********************************************************************
C*       BELOW CONSTANTS ARE SATELLITE AND IMAGE LOCATION            *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE             *
C*                AND OF SOUTH DAKOTA AREA                           *
C**********************************************************************
      ALNGTH=6378206.4
      ECCENT=0.0822719
      PRLAT1=42.75*CONVRS
      PRLAT2=45.25*CONVRS
      ORGLAT=(44)*CONVRS
      ORGLNG=(-100)*CONVRS
C******************** END OF DEPENDENT CONSTANTS *******************
      INLONG=INLONG*CONVRS
      INLAT=INLAT*CONVRS
      M1=COS(PRLAT1)/((1-(ECCENT**2)*((DSIN(PRLAT1))**2))**0.5)
      M2=COS(PRLAT2)/((1-(ECCENT**2)*((DSIN(PRLAT2))**2))**0.5)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(ORGLAT))
      QPRT3=(1-ECCENT*DSIN(ORGLAT))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(ORGLAT))**2)))
      QPRT5=(DSIN(ORGLAT)/QPRT5)
      Q0=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT1))
      QPRT3=(1-ECCENT*DSIN(PRLAT1))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT1))**2)))
      QPRT5=(DSIN(PRLAT1)/QPRT5)
      Q1=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT2))
      QPRT3=(1-ECCENT*DSIN(PRLAT2))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT2))**2)))
```

```
      QPRT5=(DSIN(PRLAT2)/QPRT5)
      Q2=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(INLAT))
      QPRT3=(1-ECCENT*DSIN(INLAT))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(INLAT))**2)))
      QPRT5=(DSIN(INLAT)/QPRT5)
      Q=QPRT1*(QPRT5-QPRT4*QPRT23)
      N=((M1**2)-(M2**2))/(Q2-Q1)
      C=(M1**2)+(N*Q1)
      ROWE=(ALNGTH*((C-N*Q)**0.5))/N
      ROWE0=(ALNGTH*((C-N*Q0)**0.5))/N
      THETA=N*(INLONG-ORGLNG)
      X=ROWE*(DSIN(THETA))
      Y=ROWE0-(ROWE*(DCOS(THETA)))
C***************************************************************************
C*       BELOW EQUATIONS ARE SATELLITE AND IMAGE LOCATION                  *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE                   *
C*                  AND OF SOUTH DAKOTA AREA                               *
C***************************************************************************
      LINE=((231000-Y)/1000)+1.5
      PIXEL=((X+352000)/1000)+1.5
C******************* END OF DEPENDENT EQUATIONS *******************
      DATA(J,2)=INT(PIXEL)
      DATA(J,1)=INT(LINE)
      X=DATA(J,2)
      Y=DATA(J,1)
      WRITE(6,304) J,DATA(J,1),DATA(J,2),DATA(J,3)
  304 FORMAT(5X,'**CONVERTED**  ',I3,5X,F10.5,5X,F10.5,5X,F5.2)
      IMAGE(X,Y)=DATA(J,3)
  301 CONTINUE
  302 CONTINUE
C.......................................................................
C   NOW TAKE THE RAINFALL AMOUNTS PLACED INTO THE IMAGE ARRAY
C     AND SPREAD THESE VALUES THROUGH OUT THE ENTIRE STATE
C     USING A 3X3 WINDOW FILLING THE CENTER WITH THE VALUE
C     WHICH IS THE MOST PREDOMINANT IN THE WINDOW.  KEEP ON
C      DOING THIS UNTIL NO -1 FLAGS APPEAR IN ANY CELL
C        EXCEPT VERY OUTER BORDER ROWS AND COLUMNS.
C.......................................................................
C.......................................................................
C
C.......................................................................
C   START THE ROUTINE OFF, BUT DON'T GO THROUGH CHECK FOR FINISH 1ST
C.......................................................................
      PASS=0
      FLAG=-1
      GOTO 501
C.......................................................................
```

```
C         CHECK TO SET IF ITS DONE YET, ELSE UPDATE PASS COUNT
C                        AND CONTINUE ON
C....................................................................
  500 IF (FLAG.EQ.0) GOTO 550
      FLAG=0
      PASS=PASS+1
C....................................................................
C       PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C....................................................................
  501 DO 502,X=2,686-1
      DO 503,Y=2,454-1
C....................................................................
C      SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C       SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                      FOUND AN EMPTY CELL
C....................................................................
      IF (IMAGE(X,Y).NE.-1) GOTO 533
      FLAG=-1
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C.............................................................
C    NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C      FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C       BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
C           FOR HIGHEST STRIKE COUNT AT INDEX 2
C.............................................................
      DO 504,I=1,9
      VAL(I,1)=-9
      VAL(I,2)=0
  504 CONTINUE
      VAL(1,1)=-1
C............................................................
C       BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C         ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C          STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C          IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                 VALUE ARRAY FOR NEXT PASS
C............................................................
      DO 520,J=1,8
      I=1
  505 IF (VAL(I,1).EQ.-9) GOTO 515
      IF (VAL(I,1).NE.PIXL(J)) GOTO 510
      VAL(I,2)=VAL(I,2)+1
      GOTO 520
  515 VAL(I,1)=PIXL(J)
```

```
      VAL(I,2)=1
      GOTO 520
  510 IF(I.EQ.9) GOTO 520
      I=I+1
      GOTO 505
  520 CONTINUE
C.................................................................
C    FIND MAJOR GRID VALUE AND STICK THIS INTO EMPTY CELL
C         ELSE DEFAULT TO 1,1 OF VAL ARRAY WHICH IS THE
C          EMPTY VALUE, -1, AND CONTINUE ON
C.................................................................
      J=0
      K=2
      DO 530 I=2,9
      IF (VAL(I,2).LT.J) GOTO 530
      J=VAL(I,2)
      K=I
  530 CONTINUE
      IMAGE2(X,Y)=VAL(K,1)
      GOTO 503
  533 IMAGE2(X,Y)=IMAGE(X,Y)
  503 CONTINUE
  502 CONTINUE
      DO 531 I=1,686
      DO 532 J=1,454
      IMAGE(I,J)=IMAGE2(I,J)
  532 CONTINUE
  531 CONTINUE
      GOTO 500
  550 CONTINUE
C.....................................................................
C   NOW SMOOTH OUT THE SINGLE VALUES FROM THE IMAGE SO IT ISNT
C     SO SPARATIC, AND FLAG AGAIN WITH -1 FOR A SECOND PASS
C     WITH THE ABOVE, ONLY THIS TIME CHECKING THAT THE
C        STRIKE VALUE IS GREATER THAN OR EQUAL TO 2
C             WILL PUT IMAGE TO BE REWORKED
C                INTO IMAGE ARRAY 2
C.....................................................................
C.....................................................................
C
C.....................................................................
C      PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C.....................................................................
  601 DO 602,X=2,686-1
      DO 603,Y=2,454-1
C.....................................................................
C     SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C      SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                    FOUND AN EMPTY CELL
C.....................................................................
      FLAG=-1
```

```
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C.............................................................
C   NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C     FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C      BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
C          FOR HIGHEST STRIKE COUNT AT INDEX 2
C.............................................................
      DO 604,I=1,9
      VAL(I,1)=-9
      VAL(I,2)=0
  604 CONTINUE
      VAL(1,1)=-1
C............................................................
C      BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C        ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C         STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C         IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                VALUE ARRAY FOR NEXT PASS
C............................................................
      DO 620,J=1,8
      I=1
  605 IF (VAL(I,1).EQ.-9) GOTO 615
      IF (VAL(I,1).NE.PIXL(J)) GOTO 610
      VAL(I,2)=VAL(I,2)+1
      GOTO 620
  615 VAL(I,1)=PIXL(J)
      VAL(I,2)=1
      GOTO 620
  610 IF(I.EQ.9) GOTO 620
      I=I+1
      GOTO 605
  620 CONTINUE
C.................................................................
C    FIND THE CENTER GRIDS STRIKE COUNT FOR OUT LINING
C    IF COUNT IS GREATER THAN OR EQUAL TO 1 THEN LEAVE
C         ELSE FLAG FOR NEXT PASS THROUGH
C.................................................................
      I=2
      Z=IMAGE(X,Y)
  631 IF (VAL(I,1).NE.Z) GOTO 632
      IF (VAL(I,2).GE.2) GOTO 630
      Z=-1.0
      GOTO 630
  632 I=I+1
```

```
      IF (I.LT.10) GOTO 631
      Z=-1.0
  630 CONTINUE
      IMAGE2(X,Y)=Z
  603 CONTINUE
  602 CONTINUE
      DO 651 I=1,686
      DO 652 J=1,454
      IMAGE(I,J)=IMAGE2(I,J)
  652 CONTINUE
  651 CONTINUE
C.......................................................................
C   NOW FOR THE FINAL PASSES OVER TO REPLACE THE FLAGGED PIXELS
C.......................................................................
C.......................................................................
C
C.......................................................................
C   START THE ROUTINE OFF, BUT DON'T GO THROUGH CHECK FOR FINISH 1ST
C.......................................................................
      FLAG=-1
      GOTO 701
C.......................................................................
C        CHECK TO SET IF ITS DONE YET, ELSE UPDATE PASS COUNT
C                     AND CONTINUE ON
C.......................................................................
  700 IF (FLAG.EQ.0) GOTO 750
      FLAG=0
C.......................................................................
C     PASS THROUGHT ENTIRE IMAGE ONCE WITH 3X3 WINDOW
C.......................................................................
  701 DO 702,X=2,686-1
      DO 703,Y=2,454-1
C.......................................................................
C     SEE IF THIS CELL NEEDS REPLACING, IF SO FILL OUT 3X3 GRID
C      SETUP WITH CENTER OUT AND TURN ON FLAG THAT THIS PASS
C                    FOUND AN EMPTY CELL
C.......................................................................
      IF (IMAGE(X,Y).NE.-1) GOTO 733
      FLAG=-1
      PIXL(1)=IMAGE(X-1,Y-1)
      PIXL(2)=IMAGE(X,Y-1)
      PIXL(3)=IMAGE(X+1,Y-1)
      PIXL(4)=IMAGE(X-1,Y)
      PIXL(5)=IMAGE(X+1,Y)
      PIXL(6)=IMAGE(X-1,Y+1)
      PIXL(7)=IMAGE(X,Y+1)
      PIXL(8)=IMAGE(X+1,Y+1)
C.............................................................
C   NOW CLEAR OUT VALUE ARRAY WHICH COUNTS NUMBER OF STRIKES
C     FOR A SPECIFIC PIXEL VALUE, FILL INDEX 1 WITH -1 FLAG
C      BECAUSE THIS ISN'T A VALID VALUE, WILL START CHECK
```

```
C         FOR HIGHEST STRIKE COUNT AT INDEX 2
C.............................................................
       DO 704,I=1,9
       VAL(I,1)=-9
       VAL(I,2)=0
  704 CONTINUE
       VAL(1,1)=-1
C............................................................
C      BELOW WILL CHECK EACH GRID VALUE AGAINST THE VALUE
C        ARRAY, IF VALUE IS FOUND WILL INCREMENT THE
C         STRIKE COUNT ONE, IF VALUE DOESN'T APPEAR
C          IN THE VALUE ARRAY, THEN PLACE IT IN THE
C                VALUE ARRAY FOR NEXT PASS
C............................................................
       DO 720,J=1,8
       I=1
  705 IF (VAL(I,1).EQ.-9) GOTO 715
       IF (VAL(I,1).NE.PIXL(J)) GOTO 710
       VAL(I,2)=VAL(I,2)+1
       GOTO 720
  715 VAL(I,1)=PIXL(J)
       VAL(I,2)=1
       GOTO 720
  710 IF(I.EQ.9) GOTO 720
       I=I+1
       GOTO 705
  720 CONTINUE
C.................................................................
C    FIND MAJOR GRID VALUE AND STICK THIS INTO EMPTY CELL
C      ELSE DEFAULT TO 1,1 OF VAL ARRAY WHICH IS THE
C       EMPTY VALUE OF -1, AND CONTINUE ON
C.................................................................
       J=0
       K=2
       DO 730 I=2,9
       IF (VAL(I,2).LT.J) GOTO 730
       J=VAL(I,2)
       K=I
  730 CONTINUE
       IF (VAL(K,2).GT.1) IMAGE2(X,Y)=VAL(K,1)
       GOTO 703
  733 IMAGE2(X,Y)=IMAGE(X,Y)
  703 CONTINUE
  702 CONTINUE
       DO 731 I=1,686
       DO 732 J=1,454
       IF (IMAGE2(I,J).GT.0.0) THEN
               IMAGE(I,J)=IMAGE2(I,J)
           ELSE
               IMAGE(I,J)=-1
           ENDIF
```

```
  732 CONTINUE
  731 CONTINUE
      GOTO 700
  750 CONTINUE
C.....................................................................
C     FINALLY ALL DONE.   NOW WRITE PASS COUNT OUT AND A SAMPLE
C          OF THE COMPLETED IMAGE FOR A CHECK
C            THEN WRITE IMAGE TO A DISK FILE
C             AND USE SAS GCONTOUR TO PLOT
C.....................................................................
      DO 806 J=1,454
      WRITE(6,807) (IMAGE2(I,J),I=300,315)
  807 FORMAT(16F8.2)
  806 CONTINUE
      DO 4000 J=1,454
      WRITE(12,4001) (IMAGE2(I,J),I=1,686)
 4001 FORMAT(200F8.3,200F8.3,200F8.3,86F8.3)
 4000 CONTINUE
      END
/*
//GO.FT12F001 DD DSN=TEMP.MOISTO,DISP=(NEW,PASS),
// VOL=SER=XCRTCH,UNIT=SYSDA,SPACE=(5488,(460,25)),
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//* Following DD precedes data generated from program in Appendix A5
//* All incomplete data cards must be deleted before execution.
//* Soil level of generated image is determined by the FORMAT
//* statement reading in this data, and must be changed to desired.
//GO.SYSIN DD *
E01  -97.198    44.377  .120      .121
E02  -97.520    44.361  .034      .068
E03  -97.620    44.376  .079      .087
E04  -98.085    44.377  .032      .096
E05  -98.222    44.189  .029      .074
E06  -98.303    44.001  .035      .068
E07  -97.900    44.044  .049      .093
E08  -97.780    44.000  .053      .187
E09  -97.521    43.986  .093      .157
E10  -97.120    43.987  .066      .174
N01  -96.733    44.563  .101      .107
N02  -96.712    44.665  .119      .109
N03  -96.733    44.737  .093      .157
N04  -96.672    44.810  .093      .114
N05  -96.692    44.883  .127      .093
N06  -96.650    45.190  .056      .117
N07  -96.771    45.219  .064      .106
N08  -96.854    45.291  .021      .066
S01  -96.757    44.146  .083      .121
S02  -96.636    43.914  .074      .074
S03  -96.722    43.783  .273      .216
S04  -96.702    43.754  .089      .185
S05  -96.897    43.410  .124      .196
```

S06	-96.877	43.271	.047	.144
S07	-96.778	43.257	.074	.178
S08	-96.778	43.077	.159	.192
S09	-96.660	42.924	.143	.160
W01	-98.031	45.465	.048	.138
W02	-98.155	45.494	.053	.129
W03	-98.647	45.479	.090	.107
W04	-98.831	45.450	.081	.130
W05	-99.017	45.497	.004	.113
W06	-99.121	45.454	.068	.107
W07	-99.348	45.469	.115	.066
W08	-99.659	45.469	.110	.137
W09	-99.721	45.455	.074	.132
W12	-102.135	45.450	.011	.073
W14	-102.589	45.493	.061	.092
W15	-102.753	45.536	.039	.041
W17	-103.554	45.523	.005	.053
W18	-103.598	45.306	.012	.032
W19	-103.556	44.973	.021	.141
W21	-103.761	44.887	.027	.093
W22	-103.822	44.771	.009	.071
W23	-103.513	44.684	.049	.107
W24	-103.182	43.945	.043	.113
W25	-103.222	43.788	.038	.084
W26	-103.348	43.340	.010	.031
W27	-103.117	43.194	.045	.099
W28	-101.365	43.208	.010	.029
W29	-101.286	43.193	.020	.058
W30	-101.524	43.266	.020	.081
W32	-101.194	43.872	.035	.123
W33	-102.256	44.001	.048	.155
W34	-102.216	44.030	.026	.074
W35	-102.188	43.988	.091	.147
W37	-101.910	44.481	.020	.024
W38	-102.030	44.625	.036	.076
W39	-102.055	44.740	.040	.090
W40	-102.035	44.886	.029	.081
W43	-101.548	44.408	.150	.112
W44	-101.106	44.366	.197	.175
W45	-101.134	43.946	.175	.153
W46	-101.711	43.814	.166	.143
W47	-99.915	44.525	.254	.277
W48	-99.605	44.556	.357	.313
W49	-99.471	44.525	.423	.213
W50	-99.091	44.540	.233	.281
W51	-98.751	44.422	.100	.142
W52	-98.468	44.480	.101	.090
W53	-98.468	44.378	.101	.090

/*

Data Base Generation

```
//A8 JOB acctnumber,'JOIN IMAGES',CLASS=M,TIME=999,MSGCLASS=D,
// REGION=6000K,MSGLEVEL=(1,1)
/*JOBPARM ROOM=18
//* Requires notification of system operator, due to possibly
//* very large execution times.  This program copies all images
//* to disk files for joint sequential reads from all images,
//* and will require about 20 megabytes of disk storage in
//* addition to other required temporary files.
//* NOTE:  date and time must be changed in Fortran program
//*        to that read from the image label program (A1)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&MASK,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=MASK.IMAGE.SD,LABEL=(19,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&STEMP,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=TEMP.IMAGE.JL6,LABEL=(1,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&RAIN,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=RN.IMAGE.JL2JL6,LABEL=(7,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&AIRTMP,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=AIRTEMP.JL6.IMAGE,LABEL=(25,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&MOISTO,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=SOILMTO.JL6.IMG,LABEL=(29,SL),VOL=SER=IMAGE1,
```

```
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&MOIST1,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=SOILMT1.JL6.IMG,LABEL=(30,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&CROPUS,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=CROP.LAND.USE.IMG,LABEL=(26,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD DSN=&&ND,
// DISP=(NEW,PASS),SPACE=(5488,(460)),UNIT=3340,
// DCB=(RECFM=FB,LRECL=5488,BLKSIZE=5488)
//SYSUT1 DD DSN=ND.IMAGE.JUL6,LABEL=(27,SL),VOL=SER=IMAGE1,
// UNIT=TAPE,DISP=(OLD,PASS)
//IMAGE EXEC FORTVCLG,PARM='XREF,OPT(3),GOSTMT',TIME.GO=300,OUT='*'
//FORT.SYSUDUMP DD SYSOUT=*
//FORT.SYSIN  DD *
       REAL*4 IMAGE1(686),IMAGE2(686),IMAGE3(686),IMAGE4(686)
       REAL*4 IMAGE5(686),IMAGE6(686),IMAGE7(686),IMAGE8(686)
       REAL*8 DATE,NOON
       INTEGER LINE,PIXEL
       REAL*8 TEMP1,TEMP2,TEMP3,VALUE,LONG
       REAL*8 N,M1,M2,INLAT,INLONG,CONVRS,ALNGTH,ECCENT
       REAL*8 PRLAT1,PRLAT2,ORGLAT,ORGLNG,LAT0,LAT1
       REAL*8 QPRT1,QPRT2,QPRT3,QPRT4,QPRT5,QPRT23
       REAL*8 Q1,Q2,Q0,Q,C,ROWE,ROWE0,THETA
C************* HARD CODED DATE AND TIME (GMT) ***********************
       YEAR=1981
       DATE=187
       OVERHD=14.633
C************    CALCUTING CONSTANTS NEEDED BY SOLAR ROUTINE *******
       CONVRS=3.141592654/180
       XLEAP=((YEAR-1976)/4)-(INT((YEAR-1976)/4))
       IF (XLEAP.EQ.0) XLEAP=1
       IF (XLEAP.NE.1) XLEAP=0
       VALUE=(((YEAR-1972)/7)-(INT((YEAR-1972)/7)))*7
       INTVAL=VALUE
       IF ((VALUE-INTVAL).GT.0.90) VALUE=INTVAL+1
       VALUE=VALUE+INT((((INT(YEAR/4))*4)-1972)/4)
       VALUE=VALUE-(INT((YEAR-((INT(YEAR/4))*4))/4))
```

```
      IF (VALUE.GE.7) VALUE=VALUE-7
      SAVDYS=182
      IF ((VALUE.EQ.5.).OR.(VALUE.EQ.6)) SAVDYS=189
      BEGSAV=(120-VALUE)+XLEAP
      ENDSAV=BEGSAV+SAVDYS
      BX=0.0061912879*(4713+((INT(YEAR/4))*4))
      BX=BX-(INT((YEAR-1976)/4))*0.0268171723
      IF (XLEAP.EQ.1) GOTO 20
      BX=BX-0.002052
      BX=BX-((((((INT(YEAR/4))+1)*4)-YEAR)/3)*0.047743)
   20 CONTINUE
C.....................................................................
C    DETERMINE CONSTANTS NEEDED BY THE CELL TO GEO LOCATION ROUTINE
C.....................................................................
      CONVRS=3.141592654/180
C*****************************************************************
C*       BELOW CONSTANTS ARE SATELLITE AND IMAGE LOCATION        *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE          *
C*                  AND OF SOUTH DAKOTA AREA                      *
C*****************************************************************
      ALNGTH=6378206.4
      ECCENT=0.0822719
      PRLAT1=42.75*CONVRS
      PRLAT2=45.25*CONVRS
      ORGLAT=(44)*CONVRS
      ORGLNG=(-100)*CONVRS
C******************* END OF DEPENDENT CONSTANTS *******************
      M1=COS(PRLAT1)/((1-(ECCENT**2)*((DSIN(PRLAT1))**2))**0.5)
      M2=COS(PRLAT2)/((1-(ECCENT**2)*((DSIN(PRLAT2))**2))**0.5)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(ORGLAT))
      QPRT3=(1-ECCENT*DSIN(ORGLAT))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(ORGLAT))**2)))
      QPRT5=(DSIN(ORGLAT)/QPRT5)
      Q0=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT1))
      QPRT3=(1-ECCENT*DSIN(PRLAT1))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT1))**2)))
      QPRT5=(DSIN(PRLAT1)/QPRT5)
      Q1=QPRT1*(QPRT5-QPRT4*QPRT23)
      QPRT1=(1-ECCENT**2)
      QPRT2=(1+ECCENT*DSIN(PRLAT2))
      QPRT3=(1-ECCENT*DSIN(PRLAT2))
      QPRT23=DLOG(QPRT3/QPRT2)
      QPRT4=1/(2*ECCENT)
      QPRT5=(1-((ECCENT**2)*((DSIN(PRLAT2))**2)))
```

```
      QPRT5=(DSIN(PRLAT2)/QPRT5)
      Q2=QPRT1*(QPRT5-QPRT4*QPRT23)
      N=((M1**2)-(M2**2))/(Q2-Q1)
      C=(M1**2)+(N*Q1)
      ROWE0=(ALNGTH*((C-N*Q0)**0.5))/N
      DO 9000 LINE=1,454
C.......................................................................
      READ(11,1001) (IMAGE1(I),I=1,686)
 1001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      READ(12,2001) (IMAGE2(I),I=1,686)
 2001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      READ(13,3001) (IMAGE3(I),I=1,686)
 3001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      READ(14,4001) (IMAGE4(I),I=1,686)
 4001 FORMAT(200F8.3,200F8.3,200F8.3,86F8.3)
C.......................................................................
      READ(15,5001) (IMAGE5(I),I=1,686)
 5001 FORMAT(200F8.3,200F8.3,200F8.3,86F8.3)
C.......................................................................
      READ(16,6001) (IMAGE6(I),I=1,686)
 6001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      READ(17,7001) (IMAGE7(I),I=1,686)
 7001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      READ(18,8001) (IMAGE8(I),I=1,686)
 8001 FORMAT(200F8.2,200F8.2,200F8.2,86F8.2)
C.......................................................................
      DO 9001 PIXEL=1,686
      IF (IMAGE7(PIXEL).NE.0.0) GOTO 9001
C******** TAKES LINE AND PIXEL AND CONVERTS TO ***********************
C********   A LATITUDE AND LONGITUDE          ***********************
C***************************************************************
C*       BELOW EQUATIONS ARE SATELLITE AND IMAGE LOCATION       *
C*        DEPENDENT. CHANGE IF NOT FROM A NOAA SATELLITE        *
C*                   AND OF SOUTH DAKOTA AREA                   *
C***************************************************************
      Y=231000-((LINE-1.5)*1000)
      X=((PIXEL-1.5)*1000)-352000
C****************** END OF DEPENDENT EQUATIONS ******************
      ROWE=(X**2+((ROWE0-Y)**2))**0.5
      THETA=DATAN(X/(ROWE0-Y))
      Q=(C-((ROWE**2)*(N**2)/(ALNGTH**2)))/N
      LAT0=DASIN(Q/2)
 555  TEMP=((1-((ECCENT**2)*((DSIN(LAT0))**2)))**2)/(2*(DCOS(LAT0)))
      TEMP2=(Q/(1-ECCENT**2))
      TEMP2=TEMP2-(DSIN(LAT0))/(1-(ECCENT**2)*((DSIN(LAT0))**2))
      TEMP3=(DLOG((1-ECCENT*(DSIN(LAT0)))/(1+ECCENT*(DSIN(LAT0)))))
```

```
      TEMP2=TEMP2+(1/(2*ECCENT))*TEMP3
      LAT1=LAT0+TEMP*TEMP2
      IF ((INT(LAT0*10000000)).EQ.(INT(LAT1*10000000))) GOTO 666
      LAT0=LAT1
      GOTO 555
  666 CONTINUE
      XLONTD=ORGLNG/CONVRS+((THETA/CONVRS)/N)
      XLATD=LAT1/CONVRS
C******  NOW DETERMINE SPECIFIC LOCATIONS SOLAR VALUES******************
      SAVTIM=0
 121  IF ((DATE.GE.BEGSAV).AND.(DATE.LE.ENDSAV)) SAVTIM=1
      DAYS=(YEAR-1975)*365+(INT((YEAR-(INT(YEAR/4)-1)*4)/4))+DATE
      XN=DAYS*(360/365.25)
 135  IF (XN.LE.360.) GOTO 125
      XN=XN-360
      GOTO 135
 125  CONTINUE
      XM=XN+279.041470-282.510396
      IF (XM.LT.0.) XM=XM+360.
      EC=(360/3.141592654)*0.016720*(SIN(((XM*3.141592654)/180)))
      XLAMDA=XN+EC+279.041470
      IF (XLAMDA.GT.360.)' XLAMDA=XLAMDA-360.
      CONVRS=3.141592654/180
      DECLIN=ARSIN((SIN(23.43*CONVRS)*SIN(XLAMDA*CONVRS)))
      RTASCN=ATAN((TAN(XLAMDA*CONVRS)*COS(23.43*CONVRS)))
      RTASCN=(RTASCN/CONVRS)+180.
      DECLIN=DECLIN/CONVRS
      QUARDT=1
      IF (XLAMDA.GT.90.) QUARDT=2
      IF (XLAMDA.GT.180.) QUARDT=3
      IF (XLAMDA.GT.270.) QUARDT=4
      VALUE=(90.*QUARDT)-RTASCN
      IF ((VALUE.LE.90).AND.(VALUE.GE.0.)) GOTO 15
      IF ((180-VALUE).LT.0.) RTASCN=RTASCN-180
      IF ((180-VALUE).GT.0.) RTASCN=RTASCN+180
      IF ((RTASCN-360.).GT.0.) RTASCN=RTASCN-360
      IF (RTASCN.LT.0.) RTASCN=RTASCN+360
 15   CONTINUE
C********************** FINDING SOLAR MAX. ALT. *************************
      TIMLST=(RTASCN/15)
      NOON=TIMLST+(ABS((XLONTD/15)))
      DAYS=(DATE*0.065709)-BX
      IF (DAYS.LT.0) DAYS=DAYS+24.065832
      NOON=NOON-DAYS
      IF (NOON.LT.0) NOON=NOON+24.065832
      NOON=NOON*0.9972645
C     NOON=NOON-6+SAVTIM    TO CONVERT TO LOCAL TIME
      IF (NOON.GT.24) NOON=NOON-24
      SINEA1=SIN((DECLIN*CONVRS))*SIN((XLATD*CONVRS))
      SINEA2=COS((DECLIN*CONVRS))*COS((XLATD*CONVRS))
      AX=ARSIN((SINEA1+SINEA2))
```

```
      ALT=AX/CONVRS
C******************* FIND SUNRISE/SUNSET  **********************************
      SUNSET=ARCOS(((TAN((XLATD*CONVRS)))*(TAN((DECLIN*CONVRS)))*(-1)))
      SUNSET=SUNSET/CONVRS
      SUNRIS=(24.065832-(SUNSET/15)+(RTASCN/15))-(XLONTD/15)
      SUNSET=((SUNSET/15)+(RTASCN/15))-(XLONTD/15)
      DAYS=(DATE*0.065709)-BX
      IF (DAYS.LT.0.) DAYS=DAYS+24.065832
      SUNRIS=SUNRIS-DAYS
      SUNSET=SUNSET-DAYS
      IF (SUNRIS.LT.0.) SUNRIS=SUNRIS+24.065832
      IF (SUNSET.LT.0.) SUNSET=SUNSET+24.065832
      SUNRIS=SUNRIS*0.9972645
      SUNSET=SUNSET*0.9972645
      IF (SUNSET.GT.24) SUNSET=SUNSET-24
      IF (SUNRIS.GT.24) SUNRIS=SUNRIS-24
      IF (SUNRIS.GT.24) SUNRIS=SUNRIS-24
      IF (SUNRIS.GT.SUNSET) THEN
            DLNGTH=(SUNSET+24)-SUNRIS
         ELSE
            DLNGTH=SUNSET-SUNRIS
      ENDIF
      SUNSET=NOON+(DLNGTH/2)
      IF (SUNSET.LT.0.) SUNSET=SUNSET+24.0
      IF (SUNSET.GT.24.) SUNSET=SUNSET-24.0
      SUNRIS=NOON-(DLNGTH/2)
      IF (SUNRIS.LT.0.) SUNRIS=SUNRIS+24.0
      IF (SUNRIS.GT.24.) SUNRIS=SUNRIS-24.0
C********** FIND AZMTH AND ALT AT THE SPECIFIED TIME *******************
      GMTTIM=OVERHD
      IF (GMTTIM.LT.0.) GMTTIM=GMTTIM+24.065832
      GSTTIM=GMTTIM*1.002743
      GSTTIM=GSTTIM+((DATE*0.065709)-BX)
      IF (GSTTIM.GT.24.0) GSTTIM=GSTTIM-24.065832
      IF (GSTTIM.LT.0.0)  GSTTIM=GSTTIM+24.065832
      TIMLST=GSTTIM+(XLONTD/15)
      IF (TIMLST.GT.24.0) TIMLST=TIMLST-24.065832
      IF (TIMLST.LT.0.0)  TIMLST=TIMLST+24.065832
      HX=TIMLST-(RTASCN/15)
      HX=HX*15
      SINEA1=SIN((DECLIN*CONVRS))*SIN((XLATD*CONVRS))
      SINEA2=COS((DECLIN*CONVRS))*COS((XLATD*CONVRS))*COS((HX*CONVRS))
      AX=ARSIN((SINEA1+SINEA2))
      OALT=AX/CONVRS
      AX=AX/CONVRS
      AZCOS1=SIN((DECLIN*CONVRS))-SIN((XLATD*CONVRS))*SIN((AX*CONVRS))
      AZCOS2=COS((XLATD*CONVRS))*COS((AX*CONVRS))
      AZMTH=ARCOS((AZCOS1/AZCOS2))
      AZMTH=AZMTH/CONVRS
      VALUE=SIN((HX*CONVRS))
      IF (VALUE.GE.0) AZMTH=360-AZMTH
```

```
      OAZMTH=AZMTH-180.
C************************ CALCULATING SOLAR EXPOSURE TIME ****************
      IF ((OVERHD.LT.SUNRIS).AND.(SUNSET.GT.OVERHD)) THEN
              SUNEXP=(OVERHD+24.0)-SUNRIS
        ELSE
      IF ((SUNSET.LT.OVERHD).AND.(SUNRIS.GT.OVERHD)) THEN
            SUNEXP=0
        ELSE
          SUNEXP=OVERHD-SUNRIS
      ENDIF
      ENDIF
C********** NOW WRITE A DATA CARD FOR THE SAMPLE POINT  **************
C********** EXCLUDING ALL VALUES THAT ARE CONSIDERED    ***************
C********** INVALID POINTS REPLACING WITH A -9.9X       ***************
      IF ((IMAGE1(PIXEL).GT.999.9).OR.(IMAGE1(PIXEL).LT.-99.9))
     _IMAGE1(PIXEL)=-9.9
      IF ((IMAGE2(PIXEL).GT.99.99).OR.(IMAGE2(PIXEL).LT.-9.99))
     _IMAGE2(PIXEL)=-9.99
      IF ((IMAGE3(PIXEL).GT.99.9).OR.(IMAGE3(PIXEL).LT.-9.9))
     _IMAGE3(PIXEL)=-9.9
      IF ((IMAGE4(PIXEL).GT.99.999).OR.(IMAGE4(PIXEL).LT.-9.999))
     _IMAGE4(PIXEL)=-9.999
      IF ((IMAGE5(PIXEL).GT.99.999).OR.(IMAGE5(PIXEL).LT.-9.999))
     _IMAGE5(PIXEL)=-9.999
      IF ((IMAGE6(PIXEL).GT.99.9).OR.(IMAGE6(PIXEL).LT.-9.9))
     _IMAGE6(PIXEL)=-9.9
      IF ((IMAGE8(PIXEL).GT.999.9).OR.(IMAGE8(PIXEL).LT.-99.9))
     _IMAGE8(PIXEL)=-9.9
      WRITE(19,9010) PIXEL,LINE,IMAGE1(PIXEL),IMAGE2(PIXEL),
     _IMAGE3(PIXEL),IMAGE4(PIXEL),IMAGE5(PIXEL),IMAGE6(PIXEL),
     _IMAGE8(PIXEL),OAZMTH,OALT,SUNEXP
 9010 FORMAT(I4,1X,I4,1X,F5.1,1X,F5.2,1X,F4.1,1X,F6.3,1X,
     _F6.3,1X,F4.1,1X,F5.1,1X,F7.2,1X,F7.2,1X,F5.2)
 9001 CONTINUE
 9000 CONTINUE
      END
//GO.FT11F001 DD DSN=&&STEMP,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT12F001 DD DSN=&&RAIN,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT13F001 DD DSN=&&AIRTMP,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT14F001 DD DSN=&&MOISTO,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT15F001 DD DSN=&&MOIST1,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT16F001 DD DSN=&&CROPUS,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT17F001 DD DSN=&&MASK,DISP=(OLD,DELETE),
// UNIT=3340
//GO.FT18F001 DD DSN=&&ND,DISP=(OLD,DELETE),
```

```
// UNIT=3340
//GO.FT19F001 DD LABEL=(x,SL),VOL=SER=RANDYC,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=32000),UNIT=TAPE,DISP=(NEW,PASS)
// EXEC SAS,OUT='*',TIME=200
//WORK      DD UNIT=3340,SPACE=(6160,(2000,750),,,ROUND),DSN=&TEMP,
// VOL=SER=(XCRTCH,ZZRTCH,VS1D01,MVSPG1,MVSPG2,MVSPG3),
// DISP=(NEW,DELETE,DELETE)
//TAPEDATA DD LABEL=(x,SL),VOL=SER=RANDYC,UNIT=TAPE,DISP=OLD
//SAS.SYSIN DD *
****************************************************************************;
*                                                                          ;
*     PRODUCE A HISTOGRAM OF THE INDIVIDUAL IMAGES FOR USE                 ;
*                LATER IN THE STUDY.                                       ;
*                                                                          ;
****************************************************************************;
*---   D * A * T * A    C * O * L * L * E * C * T * I * O * N  ------;
*---------------- SET UP DATASET AREAS -------------------------------;
DATA IMAGE1(KEEP=STEMP) IMAGE2(KEEP=RAINFALL) IMAGE3(KEEP=AIRTEMP)
     IMAGE4(KEEP=MOISTO) IMAGE5(KEEP=MOIST1) IMAGE6(KEEP=LANDUSE)
     IMAGE7(KEEP=ND) IMAGE8(KEEP=AZMUTH) IMAGE9(KEEP=ALT)
     IMAGE10(KEEP=SUNEXP);
*------------ READ DATA PRODUCED BY THE FORTRAN STEP1 --------------;
   INFILE TAPEDATA;
                              INPUT @1    PIXEL      4.
                                    @6     LINE       4.
                                    @11    STEMP      5.1
                                    @17    RAINFALL   5.2
                                    @23    AIRTEMP    4.1
                                    @28    MOISTO     6.3
                                    @35    MOIST1     6.3
                                    @42    LANDUSE    4.1
                                    @47    ND         5.1
                                    @53    AZMUTH     7.2
                                    @61    ALT        7.2
                                    @69    SUNEXP     5.2 ;
            OUTPUT IMAGE1;
            OUTPUT IMAGE2;
            OUTPUT IMAGE3;
            OUTPUT IMAGE4;
            OUTPUT IMAGE5;
            OUTPUT IMAGE6;
            OUTPUT IMAGE7;
            OUTPUT IMAGE8;
            OUTPUT IMAGE9;
            OUTPUT IMAGE10;
PROC FREQ DATA=IMAGE1;
PROC FREQ DATA=IMAGE2;
PROC FREQ DATA=IMAGE3;
PROC FREQ DATA=IMAGE4;
PROC FREQ DATA=IMAGE5;
PROC FREQ DATA=IMAGE6;
```

```
PROC FREQ DATA=IMAGE7;
PROC FREQ DATA=IMAGE8;
PROC FREQ DATA=IMAGE9;
PROC FREQ DATA=IMAGE10;
```

Appendix B

STATISTICAL DATA

Variable Abbreviations Used

Standard Terms

STMPM = Satellite indicated temperature regional mean
STEMP = Satellite indicated pixel temperature
AIRM = Ground based air temperature regional mean reading
AIRTEMP = Pixel position ground read air temperature reading
VEGM = Cropland Use regional mean
VEG = Pixel position cropland use
RAINM = Regional mean rainfall reading
RAINFALL= Pixel position rainfall reading
NDM = Regional mean normalized difference from satellite image
ND = Pixel position normalized difference satellite reading
MOM = Regional mean surface soil moisture mean
MOISTO = Pixel position surface soil moisture reading
M1M = Regional mean near surface soil moisture mean
MOIST1 = Pixel position near surface soil moisture reading

Interaction Terms

	Reduced data set		Individual pixel data
T1	= STMPM/(AIRM*NDM)	or	STEMP/(AIRTEMP*ND)
T2	= STMPM/VEGM	or	STEMP/VEG
T3	= STMPM/(NDM*VEGM)	or	STEMP/(ND*VEG)
T4	= STMPM*AIRM	or	STEMP/AIRTEMP

Reduced Data Set Readings

SAS 21:07 SUNDAY, MARCH 2, 1986 1

OBS	SECTION	NUMOBS	ATRM	RATRM	NDM	SUNM	STMPM	VLCM	MUM	MIM
1	0.00	1247	42.4789	0.06736	145.261	3.50720	27.6662	0.066728	0.005000	0.053000
2	0.01	2658	41.6774	0.07073	148.647	3.55071	27.5963	0.182743	0.020759	0.047438
3	0.02	2468	39.2813	0.13727	147.636	3.61125	26.8382	0.285247	0.031863	0.076522
4	0.03	2399	38.0626	0.15198	151.370	3.67173	26.7468	0.180521	0.011000	0.073000
5	0.04	2222	35.6175	0.67575	157.141	3.73152	14.1536	0.349977	0.073036	0.131097
6	0.05	2339	36.0469	0.58839	163.476	3.79194	11.9015	0.308756	0.093559	0.109926
7	0.06	2375	37.4514	0.46291	160.902	3.85135	13.4896	0.537196	0.060816	0.116044
8	0.07	2389	36.0523	0.34216	164.029	3.91072	7.7841	0.546764	0.048984	0.136229
9	0.08	1806	33.9725	0.02405	163.655	3.96305	7.2643	0.571041	0.023252	0.070530
10	1.00	1643	41.1676	0.26408	141.911	3.48250	26.8623	0.019848	0.010144	0.041825
11	1.01	3500	40.3187	0.01915	144.099	3.52752	31.3423	0.104937	0.021643	0.041654
12	1.02	3500	39.4777	0.01393	148.512	3.58821	25.0137	0.229177	0.029949	0.077861
13	1.03	3500	37.1912	0.06386	149.138	3.64851	28.2676	0.212506	0.020528	0.078174
14	1.04	3317	34.7301	0.72371	158.339	3.70796	11.3696	0.235472	0.074890	0.131854
15	1.05	3498	34.9218	0.67164	162.549	3.76791	14.5584	0.500546	0.093180	0.110453
16	1.06	3500	36.4916	0.22565	157.384	3.82694	16.2871	0.693091	0.068187	0.116345
17	1.07	3469	35.3950	0.02134	161.739	3.88528	10.7732	0.689072	0.047814	0.131099
18	1.08	2857	33.5589	0.07407	164.361	3.93858	5.9197	0.551040	0.034085	0.081605
19	1.09	25	32.8000	0.00000	165.000	3.96240	4.6080	0.784000	0.056000	0.117000
20	2.00	1794	39.7022	0.11071	140.964	3.45513	24.0291	0.093846	0.021542	0.091980
21	2.01	3500	39.0764	0.05875	143.222	3.50125	23.3008	0.081246	0.030957	0.119927
22	2.02	3500	39.1195	0.02488	149.220	3.56146	22.2166	0.063669	0.033674	0.082870
23	2.03	3492	38.3047	0.10776	149.054	3.62123	23.6609	0.208960	0.094441	0.115431
24	2.04	3209	34.7223	0.98508	152.122	3.68141	12.0004	0.277040	0.193193	0.210243
25	2.05	3466	34.2079	2.22799	158.297	3.73992	7.2448	0.527383	0.227982	0.204271
26	2.06	3500	36.7675	0.49753	150.178	3.79832	24.1275	0.684257	0.113599	0.144428
27	2.07	3496	34.9855	0.15367	155.765	3.85646	17.9924	0.683741	0.057063	0.107176
28	2.08	3418	32.8051	0.39978	163.073	3.91438	4.3584	0.609947	0.098229	0.111599
29	2.09	120	32.8000	0.30000	161.833	3.94183	8.0517	0.443333	0.100775	0.112450
30	3.00	1640	38.3446	0.19299	159.502	3.43166	15.9839	0.129567	0.023463	0.084017
31	3.01	3500	40.3039	0.31254	146.328	3.47547	25.3707	0.196794	0.045177	0.102933
32	3.02	3500	39.5266	0.16075	144.803	3.53519	26.2749	0.091934	0.028482	0.055664
33	3.03	3493	38.6165	0.32654	148.650	3.59449	21.0639	0.217698	0.156454	0.132422
34	3.04	3233	35.7582	1.58041	150.781	3.65362	8.8999	0.301144	0.229158	0.232547
35	3.05	3492	34.9827	3.05688	153.093	3.71211	3.4646	0.511140	0.326425	0.267338
36	3.06	3500	35.4853	0.65522	148.388	3.77024	26.4737	0.676580	0.157751	0.160384
37	3.07	3478	34.3597	0.33559	151.539	3.82789	19.8303	0.714626	0.051495	0.087558
38	3.08	3408	32.5263	0.30641	163.033	3.88553	6.6073	0.694484	0.105685	0.112495
39	3.09	224	32.4009	0.40643	163.540	3.91420	5.8420	0.450000	0.098268	0.111625
40	4.00	944	32.8099	0.10536	158.254	3.40601	0.0238	0.026674	0.041809	0.104962
41	4.01	3495	37.6728	0.58635	150.276	3.45024	18.0531	0.146664	0.042620	0.111401
42	4.02	3500	38.7384	0.48466	142.947	3.50943	21.6725	0.208983	0.055118	0.118534
43	4.03	3500	37.5034	0.61726	148.697	3.56832	17.6924	0.275500	0.155945	0.142907
44	4.04	3498	35.8575	1.67487	148.212	3.62682	13.9394	0.266366	0.184545	0.186570
45	4.05	3265	34.9639	1.50411	148.828	3.68486	11.1461	0.287335	0.325243	0.263995
46	4.06	3500	33.4138	1.37421	146.526	3.74272	19.3324	0.567014	0.083965	0.119014
47	4.07	3500	34.1286	0.76694	150.613	3.80002	14.4690	0.678249	0.054653	0.116738
48	4.08	3286	31.3026	1.10217	161.133	3.85725	-1.0135	0.717185	0.091578	0.141470
49	4.09	365	31.1000	0.90532	160.093	3.88704	-1.5973	0.687671	0.091036	0.099052
50	5.00	2149	36.3636	1.08044	147.208	3.37712	10.8198	0.045672	0.020775	0.051396
51	5.01	3496	37.5875	0.62773	143.814	3.42537	17.4174	0.083395	0.033287	0.075458
52	5.02	3500	38.4806	0.74110	141.243	3.48414	18.5723	0.064817	0.081408	0.136779
53	5.03	3500	36.3743	0.94654	149.173	3.54259	15.8249	0.151157	0.063039	0.108436
54	5.04	3499	35.5138	1.23913	147.590	3.60064	16.8119	0.237985	0.044298	0.116715

OBS	SECTION	NUMOBS	AIRM	RAINM	NDM	SUNM	STMPM	VLGM	MOM	MIM
55	5.05	3421	34.7701	1.40488	145.405	3.65785	15.3103	0.461049	0.325451	0.262916
56	5.06	3484	33.1745	1.99904	152.026	3.71580	6.8843	0.745304	0.059722	0.094594
57	5.07	3496	32.5433	1.74301	154.477	3.77256	2.6091	0.718520	0.060835	0.129981
58	5.08	3230	31.4384	2.81547	156.872	3.82809	-4.1424	0.740015	0.109613	0.188141
59	5.09	218	30.9739	1.77486	160.899	3.86523	-5.8757	0.608257	0.105706	0.187899
60	6.00	1635	40.1391	1.28423	141.008	3.35628	22.3942	0.047535	0.010535	0.032040
61	6.01	2706	39.1494	0.35497	139.740	3.40399	22.0989	0.077694	0.039089	0.087516
62	6.02	2911	36.8770	0.93866	146.644	3.46167	12.0243	0.225012	0.028949	0.087443
63	6.03	3021	34.3683	1.40161	155.021	3.51919	7.2888	0.154300	0.018527	0.061829
64	6.04	3077	33.1012	2.28383	155.294	3.57642	4.6657	0.120751	0.020000	0.058000
65	6.05	2987	34.6355	2.50762	154.106	3.63389	4.3859	0.337020	0.144615	0.132057
66	6.06	2765	34.0787	1.62556	157.027	3.69181	0.7136	0.365714	0.060994	0.095963
67	6.07	3488	32.3237	1.56625	156.392	3.74548	0.3895	0.709278	0.092361	0.140691
68	6.08	2242	30.7076	2.62703	161.852	3.79841	-5.7620	0.716012	0.108352	0.169357
69	6.09	19	30.0000	3.29316	166.737	3.83263	-7.4579	0.547368	0.118158	0.176000
70	7.07	319	32.4245	1.43727	157.251	3.72596	-0.5888	0.352727	0.138260	0.155962
71	7.08	311	32.1389	1.89363	166.286	3.79431	-5.2013	0.636013	0.143000	0.160000

Full Image SAS PROC LEAPS Output

```
REGRESSION BY LEAPS AND BOUNDS   BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
             SAS INTERFACE       BY WARREN S. SARLE

NUMBER OF VARIABLES............          8
MATRIX TYPE....................          0
NUMBER OF PREDICTORS...........          7
NUMBER FORCED..................          0
NUMBER OF OBSERVATIONS.........     190098
SELECTION CRITERION CODE.......          1
NUMBER OF BEST REGRESSIONS.....         10
COLLINEARITY CODE..............          2
MAXIMUM PIVOTS.................          0
OUTPUT CODE....................          2
VARIABLE PENALTY...............     2.0000
TOLERANCE......................0.0
RESIDUAL VARIANCE..............         .0

INDICES:  1  2  3  4  5  6  7  8

VARIABLE NAMES
   AIRTEMP      ND          VLG          STEMP        T1           T2          T3           RAINFALL

MEANS
   36.06        152.1       .3749        14.78        .2672D-02    604.8       .2891D-01    .8483

SSCP
   .2488D+09   .1040D+10   .2483D+07   .1055D+09   .1902D+05   .4419D+10   .2002D+06   .5570D+07
   .1040D+10   .4412D+10   .1102D+08   .4159D+09   .7511D+05   .1669D+11   .8241D+06   .2469D+08
   .2483D+07   .1102D+08   .4845D+05   .8241D+06   151.4       .2809D+07   3563.       .7191D+05
   .1055D+09   .4159D+09   .8241D+06   .6479D+08   .1161D+05   .2791D+10   .1208D+06   .1289D+07
   .1902D+05   .7511D+05   151.4       .1161D+05   2.106       .5010D+06   22.29       240.6
   .4419D+10   .1669D+11   .2809D+07   .2791D+10   .5010D+06   .2633D+12   .4382D+06   .4623D+08
   .2002D+06   .8241D+06   3563.       .1208D+06   22.29       .4382D+06   523.7       2673.
   .5570D+07   .2469D+08   .7191D+05   .1289D+07   240.6       .4623D+08   2673.       .3194D+06
```

MODEL: RAINFALL ON AIRTEMP ND VEG STEMP T1 T2
T3

R**2	1	2	3	4	5	6	7
0.43275		+					
0.39039	+						
0.33416			+				
0.08605					+		
0.08090				+			
0.04269							+
0.02541						+	
0.57446	+			-			
0.56754		+		-			
0.55516		+			-		
0.55493	+				-		
0.48757	-	+					
0.46754		+				-	
0.46462		+					-
0.44441		+	+				
0.41742	+					-	
0.42564	+		+				
0.60108	+			-	+		
0.58644		+		-	+		
0.57661	+		+	-			
0.57472	+			-			+
0.57460	+	+		-			
0.57458	+			-		-	
0.56784		+	+	-			
0.56780		+		-			-
0.56778		+		-		+	
0.55924	+		+		-		
0.60212	+	-		-	+		
0.60201	+			-	+		-
0.60123	+		+	-	+		
0.60108	+			-	+	+	
0.58963		+		-	+		-
0.58715		+		-	+	+	
0.58651		+	-	-	+		
0.57776	+		+	-			-
0.57692	+		+	-		+	
0.57668	+	-	+	-			
0.60544	+		+	-	+		-
0.60297	+			-	+	-	-
0.60287	+	-	+	-	+		
0.60261	+	-		-	+		-
0.60217	+	-		-	+	-	
0.60136	+		+	-	+	+	
0.59289		+	+	-	+		-
0.59018		+		-	+	-	-
0.58721		+	+	-	+	+	
0.57783	+	-	+	-			-

```
0.60726    + - + - +   -
0.60671    +   + - + - -
0.60374    + -   - + - -
0.60293    + - + - + +
0.59373      + + - + - -

0.57784    + - + -   - -
0.56376    + + +     - - -

0.60894    + - + - + - -
```

12. PIVOTS 0.178D-13 DISCREPANCY 0.393D-02 TOLERANCE

REGRESSION BY LEAPS AND BOUNDS BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
SAS INTERFACE BY WARREN S. SARLE

```
NUMBER OF VARIABLES............        10
MATRIX TYPE....................         0
NUMBER OF PREDICTORS...........         9
NUMBER FORCED..................         0
NUMBER OF OBSERVATIONS.........    190098
SELECTION CRITERION CODE.......         1
NUMBER OF BEST REGRESSIONS.....        10
COLLINEARITY CODE..............         2
MAXIMUM PIVOTS.................         0
OUTPUT CODE....................         2
VARIABLE PENALTY...............    2.0000
TOLERANCE.......................0.0
RESIDUAL VARIANCE..............      .0
```

INDICES: 1 2 3 4 5 6 7 8 9 10

VARIABLE NAMES

AIRTEMP	ND	VEG	STEMP	RAINFALL	T1	T2	T3	T4	MOISTO

MEANS

AIRTEMP	ND	VEG	STEMP	RAINFALL	T1	T2	T3	T4	MOISTO
36.06	152.1	.3749	14.78	.8483	.2672D-02	604.8	.2891D-01	554.9	.8598D-01

SSCP

.2488D+09	.1040D+10	.2483D+07	.1055D+09	.5570D+07	.1902D+05	.4419D+10	.2002D+06	.3979D+10	.5767D+06
.1040D+10	.4412D+10	.1102D+08	.4159D+09	.2469D+08	.7511D+05	.1669D+11	.8241D+06	.1560D+11	.2494D+07
.2483D+07	.1102D+08	.4845D+05	.8241D+06	.7191D+05	151.4	.2809D+07	3563.	.3001D+08	7084.
.1055D+09	.4159D+09	.8241D+06	.6429D+08	.1289D+07	.1161D+05	.2791D+10	.1208D+06	.2435D+10	.1972D+06
.5570D+07	.2469D+08	.7191D+05	.1289D+07	.3194D+06	240.6	.4623D+08	2673.	.4701D+08	.2087D+05
.1902D+05	.7511D+05	151.4	.1161D+05	240.6	2.106	.5010D+06	22.29	.4382D+06	36.43
.4419D+10	.1669D+11	.2809D+07	.2791D+10	.4623D+08	.5010D+06	.2633D+12	.4382D+06	.1082D+12	.6389D+07
.2002D+06	.8241	3563.	.1208D+06	2673.	22.29	.4382D+06	523.7	.4419D+07	464.9
.3979D+10	.1560D+11	.3001D+08	.2435D+10	.4701D+08	.4382D+06	.1082D+12	.4419D+07	.9263D+11	.7213D+07
.5767D+06	.2494D+07	7084.	.1972D+06	.2087D+05	36.43	.6389D+07	464.9	.7213D+07	2941.

MODEL: MOISTO ON AIRTEMP ND VEG STEMP RAINFALL T1
T2 T3 T4

R**2	1	2	3	4	5	6	7	8	9
0.47943		+							
0.46377					+				
0.45448	+								
0.35218			+						
0.21424						+			
0.20560				+					
0.19097									+
0.14031								+	
0.05271							+		
0.56909		+			+				
0.56520	+				+				
0.53951					+	+			
0.53718				+	+				
0.53179					+				+
0.52372			+		+				
0.52090					+			+	
0.50288		+							-
0.50180	+								-
0.49897		+		-					
0.57428		+			+			+	
0.57269		+	+		+				
0.57189		+			+		-		
0.57186	+		+		+				
0.57094	+				+		-		
0.57070	+				+			+	
0.56945		+			+	+			
0.56939	-	+			+				
0.56915		+			+				-
0.56914		+		+	+				
0.58684	+			+	+				-
0.58448		+		+	+				-
0.58406	+				+	+			-
0.58308		+			+	+			-
0.57876		+		-	+	+			
0.57699	+			-	+	+			
0.57545		+			+			+	-
0.57500		+			+		-	+	
0.57488	-	+			+			+	
0.57485	+				+			+	-
0.58805	+		+	+	+				-
0.58787	+	-		+	+				-
0.58773	+			+	+		-		-
0.58745	+			+	+			+	-
0.58684	+			+	+	-			-
0.58615	+				+	+	-		-
0.58590	+		+		+	+			-
0.58558		+	+	+	+				-

0.58549	+				+	+		+	-
0.58522		+		+	+		-		-
0.59002	+	-	+	+	+				-
0.58898	+	-		+	+		-		-
0.58860	+	-		+	+			+	-
0.58824	+		+	+	+		-		-
0.58813	+	-		+	+	-			-
0.58806	+		+	+	+			-	-
0.58805	+		+	+	+	+			-
0.58776	+			+	+	+	-		-
0.58775	+			+	+		-	+	-
0.58745	+			+	+	+		+	-
0.59039	+	-	+	+	+	-			-
0.59024	+	-	+	+	+			-	-
0.59015	+	-	+	+	+		-		-
0.58906	+	-		+	+	-	-		-
0.58901	+	-		+	+		-	+	-
0.58881	+	-		+	+	-		+	-
0.58843	+		+	+	+		-	-	-
0.58826	+		+	+	+	+	-		-
0.58806	+		+	+	+	+		-	-
0.58778	+			+	+	+	-	+	-
0.59085	+	-	+	+	+		-	-	-
0.59073	+	-	+	+	+	-		-	-
0.59042	+	-	+	+	+	-	-		-
0.58910	+	-		+	+	-	-	+	-
0.58846	+		+	+	+	+	-	-	-
0.58706	+	-	+		+	+	-	-	-
0.58608		+	+	+	+	+	-	-	-
0.58253	+	+	+	-	+	+	-	-	
0.53710	+	-	+	+		+	-	-	-
0.59112	+	-	+	+	+	-	-	-	-

20. PIVOTS 0.114D-12 DISCREPANCY 0.321D-03 TOLERANCE

REGRESSION BY LEAPS AND BOUNDS BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
SAS INTERFACE BY WARREN S. SARLE

```
NUMBER OF VARIABLES............         10
MATRIX TYPE....................          0
NUMBER OF PREDICTORS...........          9
NUMBER FORCED..................          0
NUMBER OF OBSERVATIONS.........     190098
SELECTION CRITERION CODE.......          1
NUMBER OF BEST REGRESSIONS.....         10
COLLINEARITY CODE..............          2
MAXIMUM PIVOTS.................          0
OUTPUT CODE....................          2
VARIABLE PENALTY...............     2.0000
TOLERANCE......................0.0
RESIDUAL VARIANCE..............         .0

INDICES:  1  2  3  4  5  6  7  8  9  10

VARIABLE NAMES
   AIRTEMP   ND   STEMP   VEG   RAINFALL   T1   T2   T3   T4   MOIST1
```

MEANS

36.06	152.1	14.78	.3749	.8483	.2672D-02	604.8	.2891D-01	554.9	.1209

SSCP

.2488D+09	.1040D+10	.1055D+09	.2483D+07	.5570D+07	.1902D+05	.4419D+10	.2002D+06	.3979D+10	.8181D+06
.1040D+10	.4412D+10	.4159D+09	.1102D+08	.2469D+08	.7511D+05	.1669D+11	.8241D+06	.1560D+11	.3505D+07
.1055D+09	.4159D+09	.6429D+08	.8241D+06	.1289D+07	.1161D+05	.2791D+10	.1208D+06	.2435D+10	.2986D+06
.2483D+07	.1102D+08	.8241D+06	.4845D+05	.7191D+05	151.4	.2809D+07	3563.	.3001D+08	9549.
.5570D+07	.2469D+08	.1289D+07	.7191D+05	.3194D+06	240.6	.4623D+08	2673.	.4701D+08	.2450D+05
.1902D+05	.7511D+05	.1161D+05	151.4	240.6	2.106	.5010D+06	22.29	.4382D+06	54.61
.4419D+10	.1669D+11	.2791D+10	.2809D+07	.4623D+08	.5010D+06	.2613D+12	.4382D+06	.1082D+12	.1123D+08
.2002D+06	.8241D+06	.1208D+06	3563.	2673.	22.29	.4382D+06	523.7	.4419D+07	648.3
.3979D+10	.1560D+11	.2435D+10	.3001D+08	.4701D+08	.4382D+06	.1082D+12	.4419D+07	.9263D+11	.1107D+08
.8181D+06	.3505D+07	.2986D+06	9549.	.2450D+05	54.61	.1123D+08	648.3	.1107D+08	3591.

MODEL: MOIST1 ON AIRTEMP ND STEMP VEG RAINFALL T1
T2 T3 T4

R**2	1	2	3	4	5	6	7	8	9
0.77554		+							
0.74904	+								
0.52417				+					
0.52338					+				
0.39429						+			
0.38632			+						
0.36858									+
0.22351								+	
0.13338							+		
0.81216		+			+				
0.80379	+				+				
0.78855		+							-
0.78781	+								-
0.78677		+	-						
0.78420		+				-			
0.78395		+		+					
0.78209	+		-						
0.78125		+					-		
0.78081	-	+							
0.81585		+		+	+				
0.81384		+			+			+	
0.81318	+			+	+				
0.81301		+			+		-		
0.81271		+			+				-
0.81237		+	-		+				
0.81235	-	+			+				
0.81218		+			+	-			
0.81012	+				+				-
0.80810	+				+		-		
0.82147	+		+		+				-
0.82130	+				+	+			-
0.81937		+			+	+			-
0.81888		+	-		+	+			
0.81818		+	+		+				-
0.81753	+		-		+	+			
0.81605	+			+	+				-
0.81596	+	+		+	+				
0.81594		+		+	+	+			
0.81591		+		+	+		+		
0.82408	+			+	+	+			-
0.82383	+		+	+	+				-
0.82214	+				+	+		+	-
0.82197	+		+		+			+	-
0.82180	+		-	+	+	+			
0.82178	+		+		+	+			-
0.82160	+				+	+	-		-
0.82158	+	-	+		+				-

0.82151	+		+		+		-		-
0.82139	+	+			+	+			-
0.82451	+	-	+	+	+				-
0.82444	+			+	+	+		-	-
0.82439	+		+	+	+			-	-
0.82431	+		+	+	+		+		-
0.82427	+		+	+	+	+			-
0.82425	+			+	+	+	+		-
0.82416	+	-		+	+	+			-
0.82235	+		+		+	+		+	-
0.82217	+				+	+	+	+	-
0.82216	+		+		+		+	+	-
0.82547	+	-	+	+	+			-	-
0.82505	+	-	+	+	+		+		-
0.82477	+		+	+	+	+		-	-
0.82463	+	-	+	+	+	+			-
0.82456	+		+	+	+	+	+		-
0.82456	+	-		+	+	+		-	-
0.82450	+		+	+	+		+	-	-
0.82444	+			+	+	+	+	-	-
0.82432	+	-		+	+	+	+		-
0.82243	+		+		+	+	+	+	-
0.82552	+	-	+	+	+		+	-	-
0.82551	+	-	+	+	+	+		-	-
0.82507	+	-	+	+	+	+	+		-
0.82480	+		+	+	+	+	+	-	-
0.82456	+	-		+	+	+	+	-	-
0.82245	+	-	+		+	+	+	+	-
0.82220	+	+	-	+	+	+	-	-	
0.82145		+	-	+	+	+	+	-	-
0.80882	+	-	+	+		+	-	-	-
0.82554	+	-	+	+	+	+	+	-	-

20. PIVOTS 0.227D-12 DISCREPANCY 0.321D-03 TOLERANCE

```
1         SAS(R) LOG   OS SAS 5.08          VS2/MVS JOB T3409520 STEP SAS                       20:13 MONDAY, FEBRUARY 17, 1986

NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C.  27511, U.S.A.
NOTE: THE JOB T3409520 HAS BEEN RUN UNDER RELEASE 5.08 OF SAS AT SOUTH DAKOTA STATE UNIVERSITY COMPUTER CENTER (03163001).

NOTE: CPUID   VERSION = 04  SERIAL = 040078  MODEL = 3031 .
      CPUID   VERSION = 00  SERIAL = 010205  MODEL = 3031 .

NOTE: SAS OPTIONS SPECIFIED ARE:
       SORT=10

          *-----------------------------------------------------------;
          *--    BELOW IS FOR THE REDUCED DATASET OF MEAN VALUES ----;
          *-----------------------------------------------------------;
           DATA COMPSITE; SET DISK.COMPSITE;
            TATRM 1/ATRM;
            TNDM 1/NDM;
            TVEGM 1/VEGM;
             T1 SIMPM*TATRM*TNDM;
             T2 SIMPM*TVEGM;
             T3-SIMPM*TNDM*VEGM;
             T4-SIMPM*ATRM;
          *---  SAS STATISTICAL ANALYSIS BELOW  -------------;

NOTE: DATA SET WORK.COMPSITE HAS 71 OBSERVATIONS AND 17 VARIABLES. 334 OBS/TRK.
NOTE: THE DATA STATEMENT USED 0.96 SECONDS.

                PROC LEAPS DATA COMPSITE LONG PRINT-10 NOINT;
           VAR  ATRM NDM VEGM SIMPM
                      T1
                      T2
                      T3
                      RATNM;
                WEIGHT NUMOBS;
          *;
          *---  SAS STATISTICAL ANALYSIS BELOW  -------------;

NOTE: THE PROCEDURE LEAPS USED 2.98 SECONDS.

                PROC LEAPS DATA COMPSITE LONG PRINT-10 NOINT;
              VAR ATRM NDM VEGM SIMPM RATNM
                      T1
                      T2
                      T3
                      T4
                      MOM;
                WEIGHT NUMOBS;
          *;
          *---  SAS STATISTICAL ANALYSIS BELOW  -------------;

NOTE: THE PROCEDURE LEAPS USED 3.84 SECONDS.

                PROC LEAPS DATA-COMPSITE LONG PRINT-10 NOINT;
            VAR ATRM NDM SIMPM VEGM RATNM
                      T1
                      T2
                      T3
```

```
                 T4
                 M1M;
               WEIGHT NUMOBS;

NOTE: THE PROCEDURE LEAPS USED 3.80 SECONDS.

NOTE: SAS INSTITUTE INC.
      SAS CIRCLE
      PO BOX 8000
      CARY, N.C.  27511-8000
```

REGRESSION BY LEAPS AND BOUNDS   BY GEORGE M. FURNIVAL AND ROBERT W. WILSON
SAS INTERFACE   BY WARREN S. SARLE

```
NUMBER OF VARIABLES...........           8
MATRIX TYPE...................           0
NUMBER OF PREDICTORS..........           7
NUMBER FORCED.................           0
NUMBER OF OBSERVATIONS........          71
SELECTION CRITERION CODE......           1
NUMBER OF BEST REGRESSIONS....          10
COLLINEARITY CODE.............           2
MAXIMUM PIVOTS................           0
OUTPUT CODE...................           2
VARIABLE PENALTY..............      2.0000
TOLERANCE.....................0.0
RESIDUAL VARIANCE.............    .0
```

INDICES: 1 2 3 4 5 6 7 8

VARIABLE NAMES

AIRM	NDM	VEGM	STMPM	T1	T2	T3	RAINM

MEANS

AIRM	NDM	VEGM	STMPM	T1	T2	T3	RAINM
36.06	152.0	.3712	14.71	.2666D-02	97.55	.2889D-01	.8502

SSCP

.9279D+05	.3883D+06	918.2	.3908D+05	7.007	.2680D+06	74.44	2090.
.3883D+06	.1643D+07	4074.	.1559D+06	28.02	.1013D+07	310.7	9228.
918.2	4074.	13.85	310.7	.5669D-01	1045.	.9380	26.03
.3908D+05	.1559D+06	310.7	.2105D+05	3.760	.1522D+06	38.64	547.0
7.007	28.02	.5669D-01	3.760	.6732D-03	26.78	.7042D-02	.1014
.2680D+06	.1013D+07	1045.	.1522D+06	26.78	.2385D+07	142.2	3068.
74.44	310.7	.9380	38.64	.7042D-02	142.2	.1135	1.195
2090.	9228.	26.03	547.0	.1014	3068.	1.195	95.12

MODEL: RAINM ON AIRM NDM VIGM SIMPM T1 T2
T3

R**2	1	2	3	4	5	6	7
0.54481		+					
0.51437			+				
0.49484	+						
0.16053					+		
0.14941				+			
0.13235							+
0.04149						+	
0.74889	+			-			
0.72738	+				-		
0.72620		+		-			
0.71315		+			-		
0.62721	-	+					
0.60279		+					-
0.58583		+				-	
0.58233			+				-
0.57263		+	+				
0.55799	+		+				
0.77961	+			-	+		
0.75724	+	-		-			
0.74944	+			-			+
0.74900	+			-		-	
0.74897	+		-	-			
0.74877		+		-	+		
0.72889	+				-		+
0.72881		+	-	-			
0.72871	+				-	-	
0.72863	+	-			-		
0.[illegible]588	+	-		-	+		
0.78590	+			-	+		-
0.78395	+		-	-	+		
0.78100	+			-	+	+	
0.77300	+	-		-			+
0.76550		+		-	+		-
0.76462	+	-	+	-			
0.76145	+	-		-		-	
0.75794		+	-	-	+		
0.75370		+		-	+	+	
0.81058	+	-	+	-	+		
0.80748	+	-		-	+	-	
0.80726	+	-		-	+		+
0.78595	+			-	+	-	-
0.78590	+		+	-	+		-
0.78408	+		-	-	+	+	
0.77337	+	-		-		-	+
0.77302	+	-	+	-			+
0.76742	+	-	+	-		-	
0.76581		+	+	-	+		-

0.81153	+ - + - + -
0.81078	+ - + - + -
0.80807	+ - - + - +
0.78595	+ + - + - -
0.77341	+ - + - - +
0.76582	+ + - + - -
0.73902	+ - + - - +
0.81231	+ - + - + - -

12. PIVOTS 0.114D-12 DISCREPANCY 0.1[illegible]D-0[illegible] TOLERANCE

REGRESSION BY LEAPS AND BOUNDS BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
SAS INTERFACE BY WARREN S. SARLE

NUMBER OF VARIABLES............	10
MATRIX TYPE....................	0
NUMBER OF PREDICTORS...........	9
NUMBER FORCED..................	0
NUMBER OF OBSERVATIONS.........	71
SELECTION CRITERION CODE.......	1
NUMBER OF BEST REGRESSIONS.....	10
COLLINEARITY CODE..............	2
MAXIMUM PIVOTS.................	0
OUTPUT CODE....................	2
VARIABLE PENALTY...............	2.0000
TOLERANCE......................	0.0
RESIDUAL VARIANCE..............	.0

INDICES: 1 2 3 4 5 6 7 8 9 10

VARIABLE NAMES

AIRM	NDM	VEGM	STMPM	RAINM	I1	I2	I3	I4	MOM
MEANS									
36.06	152.0	.3712	14.71	.8502	.2644D-02	97.55	.2889D-01	550.4	.8689D-01
SSCP									
.9279D+05	.3883D+06	918.2	.3908D+05	2090.	7.007	.2680D+06	74.44	.1467D+07	218.0
.3883D+06	.1643D+07	4074.	.1559D+06	9228.	28.02	.1013D+07	310.7	.5826D+07	939.8
918.2	4074.	13.85	310.7	26.03	.5669D-01	1045.	.9380	.1126D+05	2.601
.3908D+05	.1559D+06	310.7	.2105D+05	547.0	3.760	.1522D+06	38.64	.7980D+06	75.99
2090.	9228.	26.03	547.0	95.12	.1014	3068.	1.195	.1982D+05	7.498
7.007	28.02	.5669D-01	3.760	.1014	.6732D-03	26.78	.7042D-02	142.2	.1395D-01
.2680D+06	.1013D+07	1045.	.1522D+06	3068.	26.78	.2385D+07	142.2	.5956D+07	327.0
74.44	310.7	.9380	38.64	1.195	.7042D-02	142.2	.1135	1415.	.1840
.1467D+07	.5826D+07	.1126D+05	.7980D+06	.1982D+05	142.2	.5956D+07	1415.	.3035D+08	2769.
218.0	939.8	2.601	75.99	7.498	.1395D-01	327.0	.1840	2769.	.9515

MODEL: MOM ON AIRM NDM VICM SIMPM RAINM T1
T2 T3 T4

R**2	1	2	3	4	5	6	7	8	9
0.61925					+				
0.56314		+							
0.53666	+								
0.51190			+						
0.31266								+	
0.30285						+			
0.28733				+					
0.26473									+
0.04696							+		
0.70507					+			+	
0.68506					+	+			
0.68269	+				+				
0.68246				+	+				
0.68243		+			+				
0.67744					+				+
0.66626			+		+				
0.62257					+		+		
0.60413	+								-
0.60106		+					-		
0.70898	+				+			+	
0.70856		+			+			+	
0.70843				+	+				-
0.70731					+	+		+	
0.70715				+	+			+	
0.70680					+			+	+
0.70627					+	+	-		
0.70511					+		-	+	
0.70510			-		+			+	
0.70401				+	+		-		
0.71414				+	+		-		-
0.71315	+				+		-	+	
0.71224					+	+	-	+	
0.71175				+	+		-	+	
0.71158				+	+			+	-
0.71136					+	+	-		-
0.71101		+			+		-	+	
0.71084					+		-	+	+
0.71071			+	+	+				-
0.71043				+	+	-			-
0.71641	+	-			+		-	+	
0.71574	+	-			+			+	-
0.71547	+			+	+		-		-
0.71528	+		-		+		-	+	
0.71518			+	+	+		-		-
0.71511				+	+		-	+	-
0.71483		+		+	+		-		-
0.71469				+	+	-	-		-

```
0.71437   +       + + -   -
0.71409     + -   +   - +

0.72095   + -     +   - + -
0.72086   + -   + +   -   -
0.72064   + -   + +     + -
0.72005   + - + + +       -
0.71871   + -   - +   - +

0.71835   + -     + +   + -
0.71763   + -     + + -   -
0.71743   + -     + - - +
0.71660   +     + +   - + -
0.71642   + - -   +   - +

0.72642   + - + + +   -   -
0.72542   + -   + +   - + -
0.72487   + - + + + -     -
0.72386   + -     + + - + -
0.72291   + -   + + -   + -

0.72229   + - +   + + -   -
0.72196   + -   + + - -   -
0.72134   + - + + +     + -
0.72131   + -   - + + - +
0.72113   + - -   +   - + -

0.72871   + - + + + - -   -
0.72665   + - + + +   - + -
0.72640   + -   + + - - + -
0.72503   + - + + + -   + -
0.72411   + - +   + + - + -

0.72132   + - - - + + - +
0.71723   +   - + + + - + -
0.71595     + - + + - - + -
0.67794   + - + +   + - + -

0.72871   + - + + + - - + -

          30. PIVOTS   0.178D-13 DISCREPANCY   0.846D-04 TOLERANCE

REGRESSION BY LEAPS AND BOUNDS   BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
           SAS INTERFACE         BY WARREN S. SARLE

NUMBER OF VARIABLES............      10
MATRIX TYPE....................       0
NUMBER OF PREDICTORS...........       9
NUMBER FORCED..................       0
NUMBER OF OBSERVATIONS.........      71
SELECTION CRITERION CODE.......       1
NUMBER OF BEST REGRESSIONS.....      10
COLLINEARITY CODE..............       2
MAXIMUM PIVOTS.................       0
OUTPUT CODE....................       2
VARIABLE PENALTY...............  2.0000
TOLERANCE......................0.0
RESIDUAL VARIANCE..............      .0

INDICES:  1  2  3  4  5  6  7  8  9 10

VARIABLE NAMES
  AIRM     NDM     SIMPM     VEGM     RAINM     T1     T2     T3     T4     MIM
```

MEANS

36.06	152.0	14.71	.3712	.8502	.2644D-02	97.55	.2889D-01	550.4	.1215

SSCP

.9279D+05	.3883D+06	.3908D+05	918.2	2090.	7.007	.2680D+06	74.44	.1467D+07	307.1
.3883D+06	.1643D+07	.1559D+06	4074.	9228.	28.02	.1013D+07	310.7	.5826D+07	1314.
.3908D+05	.1559D+06	.2105D+05	310.7	547.0	3.760	.1522D+06	38.64	.7980D+06	113.8
918.2	4074.	310.7	13.85	26.03	.5669D-01	1045.	.9380	.1126D+05	3.503
2090.	9228.	547.0	26.03	95.12	.1014	3068.	1.195	.1982D+05	8.932
7.007	28.02	3.760	.5669D-01	.1014	.6732D-03	26.78	.7042D-02	142.2	.2066D-01
.2680D+06	.1013D+07	.1522D+06	1045.	3068.	26.78	.2385D+07	142.2	.5956D+07	606.3
74.44	310.7	38.64	.9380	1.195	.7042D-02	142.2	.1135	1415.	.2508
.1467D+07	.5826D+07	.7980D+06	.1126D+05	.1982D+05	142.2	.5956D+07	1415.	.3035D+08	4198.
307.1	1314.	113.8	3.503	8.932	.2066D-01	606.3	.2508	4198.	1.245

MODEL: M1M ON AIRM NDM SIMPM VIGM RAINM T1
T2 T3 T4

R**2	1	2	3	4	5	6	7	8	9
0.84385		+							
0.81626	+								
0.71181				+					
0.67363					+				
0.50949						+			
0.49365			+						
0.46642									+
0.44514								+	
0.12380							+		
0.88859		+			+				
0.88416	+				+				
0.86493	+								-
0.86278		+					-		
0.86141		+							-
0.85915		+	-						
0.85871	+		-						
0.85674		+		+					
0.85625		+				-			
0.85558	+						-		
0.89790	+				+		-		
0.89678		+			+			+	
0.89527	+			+	+				
0.89463		+			+		-		
0.89410	+				+			+	
0.89259		+		+	+				
0.88906		+			+	+			
0.88875		+	+		+				
0.88861	-	+			+				
0.88859		+			+				+
0.90136	+				+		-	+	
0.90057	+			+	+		-		
0.90009	+				+			+	-
0.89960		+			+		-	+	
0.89940	+		+		+				-
0.89938	+		-		+			+	
0.89854	+				+	-		+	
0.89844		+			+	+	-		
0.89838	+				+	+			-
0.89825	+				+	+	-		
0.90352	+				+	+	-		-
0.90318	+		+		+		-		-
0.90306	+		-		+	+	-		
0.90263	+				+		-	+	-
0.90235	+	-		+	+		-		
0.90229	+		-		+		-	+	
0.90212	+			+	+		-	+	
0.90198	+	-			+			+	-

0.90193	+		+	+	+				-
0.90190	+				+	-	-	+	
0.90734	+	-	+	+	+				-
0.90527	+	-		+	+	+			-
0.90511	+	-			+		-	+	-
0.90481	+			+	+	+	-		-
0.90478	+	-	+		+		-		-
0.90469	+		-	+	+	+	-		
0.90453	+	-			+	+	-		-
0.90438	+		+	+	+		-		-
0.90433	+		-		+	+	-	+	
0.90424	+				+	+	-	+	-
0.91063	+	-	+	+	+		-		-
0.90978	+	-		+	+	+	-		-
0.90803	+	-	+	+	+	-			-
0.90779	+	-	-	+	+	+	-		
0.90740	+	-	+	+	+			-	-
0.90676	+	-	+		+		-	+	-
0.90676	+	-			+	+	-	+	-
0.90616	+	-	-		+	+	-	+	
0.90587	+	-		+	+		-	+	-
0.90551	+	-		+	+	+		+	-
0.91122	+	-	+	+	+		-	-	-
0.91072	+	-	+	+	+	-	-		-
0.90984	+	-		+	+	+	-	-	-
0.90835	+	-	+	+	+	-		-	-
0.90787	+	-	-	+	+	+	-	+	
0.90680	+	-	+		+	+	-	+	-
0.90483	+		-	+	+	+	-	-	-
0.90203		+	-	+	+	+	-	-	-
0.89996	+	-	+	+		+	-	-	-
0.91152	+	-	+	+	+	-	-	-	-

26. PIVOTS 0.568D-13 DISCREPANCY 0.846D-04 TOLERANCE

Reduced Data Set PROC LEAPS with Forced Term

```
1        SAS(R) LOG   OS SAS 5.08         VS2/MVS JOB T3409582 STEP SAS                 13:01 SATURDAY, FEBRUARY 22, 1986

NOTE: COPYRIGHT (C) 1984 SAS INSTITUTE INC., CARY, N.C.  27511, U.S.A.
NOTE: THE JOB T3409582 HAS BEEN RUN UNDER RELEASE 5.08 OF SAS AT SOUTH DAKOTA STATE UNIVERSITY COMPUTER CENTER (03163001).

NOTE: CPUID   VERSION = 04  SERIAL = 040078  MODEL   3031 .
      CPUID   VERSION = 00  SERIAL = 030205  MODEL   3031 .

NOTE: SAS OPTIONS SPECIFIED ARE:
       SORT=10

           *-----------------------------------------------------------;
           *--    BELOW IS FOR THE REDUCED DATASET OF MEAN VALUES ----;
           *-----------------------------------------------------------;
            DATA COMPSITE; SET DISK.COMPSITE;
              IAIRM=1/AIRM;
              INDM=1/NDM;
              IVEGM=1/VEGM;
               T1=STMPM*IAIRM*INDM;
               T2=STMPM*IVEGM;
               T3=STMPM*INDM*VEGM;
               T4=STMPM*AIRM;
            *;
            *---  SAS STATISTICAL ANALYSIS BELOW  -------------;

NOTE: DATA SET WORK.COMPSITE HAS 71 OBSERVATIONS AND 17 VARIABLES. 334 OBS/TRK.
NOTE: THE DATA STATEMENT USED 0.78 SECONDS.

               PROC LEAPS DATA=COMPSITE LONG PRINT=10 NOINT INCLUDE=1;
              VAR STMPM AIRM NDM VEGM RAINM
                   T1
                   T2
                   T3
                   T4
                   MDM;
               WEIGHT NUMOBS;

NOTE: THE PROCEDURE LEAPS USED 3.16 SECONDS.

NOTE: SAS INSTITUTE INC.
      SAS CIRCLE
      PO BOX 8000
      CARY, N.C.  27511-8000
```

```
REGRESSION BY LEAPS AND BOUNDS   BY GEORGE. M. FURNIVAL AND ROBERT. W. WILSON
                   SAS INTERFACE   BY WARREN S. SARLE

NUMBER OF VARIABLES............          10
MATRIX TYPE....................           0
NUMBER OF PREDICTORS...........           9
NUMBER FORCED..................           1
NUMBER OF OBSERVATIONS.........          71
SELECTION CRITERION CODE.......           1
NUMBER OF BEST REGRESSIONS.....          10
COLLINEARITY CODE..............           2
MAXIMUM PIVOTS.................           0
OUTPUT CODE....................           2
VARIABLE PENALTY...............      2.0000
TOLERANCE......................0.0
RESIDUAL VARIANCE..............      .0

INDICES:  1  2  3  4  5  6  7  8  9 10
```

VARIABLE NAMES

MEANS

SIMPM	AIRM	NDM	VLCM	RAINM	T1	T2	T3	T4	MOM
14.71	36.06	152.0	.3712	.8502	.2644D-02	97.55	.2889D-01	550.4	.8689D-01

SSCP

SIMPM	AIRM	NDM	VLCM	RAINM	T1	T2	T3	T4	MOM
.2105D+05	.3908D+05	.1559D+06	310.7	547.0	3.760	.1522D+06	38.64	.7980D+06	75.99
.3908D+05	.9279D+05	.3883D+06	918.2	2090.	7.007	.2680D+06	74.44	.1467D+07	218.0
.1559D+06	.3883D+06	.1643D+07	4074.	9228.	28.02	.1013D+07	310.7	.5826D+07	939.8
310.7	918.2	4074.	13.85	26.03	.5669D-01	1045.	.9380	.1126D+05	2.601
547.0	2090.	9228.	26.03	95.12	.1014	3068.	1.195	.1982D+05	7.498
3.760	7.007	28.02	.5669D-01	.1014	.6732D-03	26.78	.7042D-02	142.2	.1395D-01
.1522D+06	.2680D+06	.1013D+07	1045.	3068.	26.78	.2385D+07	142.2	.5956D+07	327.0
38.64	74.44	310.7	.9380	1.195	.7042D-02	142.2	.1135	1415.	.1840
.7980D+06	.1467D+07	.5826D+07	.1126D+05	.1982D+05	142.2	.5956D+07	1415.	.3035D+08	2769.
75.99	218.0	939.8	2.601	7.498	.1395D-01	327.0	.1840	2769.	.9545

MODEL: MOM ON STMPM AIRM NDM VEGM RAINM T1
T2 T3 T4

FIRST 1 PREDICTORS MUST APPEAR IN EVERY REGRESSION

R**2	1	2	3	4	5	6	7	8	9
0.68246	+				+				
0.59384	-	+							
0.59225	-		+						
0.53501	+			+					
0.40892	+								-
0.37558	-					+			
0.33622	+							+	
0.32772	+						-		
0.70843	+				+				-
0.70715	+				+			+	
0.70401	+				+		-		
0.69524	+			+	+				
0.69341	-				+	+			
0.68617	+		+		+				
0.68473	+	+			+				
0.64050	-	+				+			
0.63103	-		+			+			
0.63057	+	+							-
0.71414	+				+		-		-
0.71175	+				+		-	+	
0.71158	+				+			+	-
0.71071	+			+	+				-
0.71043	+				+	-			-
0.70971	+	+			+				-
0.70926	-	+			+			+	
0.70923	+		+		+				-
0.70872	-				+	+	-		
0.70856	+		+		+			+	
0.71547	+	+			+		-		-
0.71518	+			+	+		-		-
0.71511	+				+		-	+	-
0.71483	+		+		+		-		-
0.71469	+				+	-	-		-
0.71368	+	+			+		-	+	
0.71315	+	+			+			+	-
0.71293	+		+		+		-	+	
0.71284	-	+	-		+			+	
0.71276	+				+	-		+	-
0.72086	+	+	-		+		-		-
0.72064	+	+	-		+			+	-
0.72005	+	+	-	+	+				-
0.71871	-	+	-		+		-	+	
0.71660	+	+			+		-	+	-
0.71582	+		+		+		-	+	-
0.71561	+	+		+	+		-		-

0.71561	+	+	-		+	-			-
0.71560	-	+			+	+	-	+	
0.71560	-	+		-	+		-	+	
0.72642	+	+	-	+	+		-		-
0.72542	+	+	-		+		-	+	-
0.72487	+	+	-	+	+	-			-
0.72291	+	+	-		+	-		+	-
0.72196	+	+	-		+	-	-		-
0.72134	+	+	-	+	+			+	-
0.72131	-	+	-		+	+	-	+	
0.71906	-	+	-	-	+		-	+	
0.71722	+	+		-	+		-	+	-
0.71714	-	+	-	+	+	+	-		
0.72871	+	+	-	+	+	-	-		-
0.72665	+	+	-	+	+		-	+	-
0.72640	+	+	-		+	-	-	+	-
0.72503	+	+	-	+	+	-		+	-
0.72132	-	+	-	-	+	+	-	+	
0.71723	+	+		-	+	+	-	+	-
0.71595	+		+	-	+	-	-	+	-
0.67794	+	+	-	+		+	-	+	-
0.72871	+	+	-	+	+	-	-	+	-

18. PIVOTS   0.236D-15 DISCREPANCY   0.846D-04 TOLERANCE