R-96-08

# STATISTICAL OBJECTIVE ANALYSIS SCHEME (SOAS) FOR IMPROVING WSR-88D RAINFALL ESTIMATES

## VOLUME 2

## SOAS VERSION 1.0 USER'S GUIDE

December 1996

U.S. DEPARTMENT OF THE INTERIOR
Bureau of Reclamation
Reclamation Service Center
Dam Safety Office

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188





**1. AGENCY USE ONLY** *(Leave Blank)*

**2. REPORT DATE**
December 1996

**3. REPORT TYPE AND DATES COVERED**
Final

**4. TITLE AND SUBTITLE**
Statistical Objective Analysis Scheme (SOAS) for Improving WSR-88D Rainfall Estimates—Volume 2: SOAS Verson 1.0 User's Guide

**5. FUNDING NUMBERS**
PR

**6. AUTHOR(S)**
Augusto J. Pereira Fo., Kenneth C. Crawford, and Curtis L. Hartzell

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Oklahoma Climatological Survey
University of Oklahoma
Norman OK 73019

Bureau of Reclamation
Technical Service Center
Denver CO 80225

**8. PERFORMING ORGANIZATION REPORT NUMBER**
R-96-08

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Bureau of Reclamation
Denver Federal Center
PO Box 25007
Denver CO 80225-0007

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
DIBR

**11. SUPPLEMENTARY NOTES**
Hard copy available at the Technical Service Center, Denver, Colorado

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Available from the National Technical Information Service, Operations Division, 5285 Port Royal Road, Springfield, Virginia 22161

**12b. DISTRIBUTION CODE**

**13. ABSTRACT**

The Bureau of Reclamation and the Oklahoma Climatological Survey at the University of Oklahoma prepared volume 1 of this report to describe the SOAS (Statistical Objective Analysis Scheme) and its application to case studies in the Lake Altus area in southwestern Oklahoma. Volume 2 is a SOAS user's guide.

The WSR-88D (or NEXRAD) weather surveillance Doppler-radar network provides real-time precipitation estimates at unprecedented spatial and temporal scales. The accuracy of these estimates improves when radar data are adjusted with hourly precipitation gage measurements.

A SOAS was used to reanalyze hourly rainfall estimates from mosaiced digital rainfall arrays over the Lake Altus area using Oklahoma Mesonet rain gage data. Results indicate that current stage III processing underestimates total rainfall accumulations by as much as 40 percent compared to the SOAS reanalysis. Furthermore, the largest discrepancies between the stage III rainfall estimates and the SOAS reanalyses coincided with areas of overlapping coverage between individual WSR-88D umbrellas (230 km). This comparative study has shown that stage III processing can be significantly improved by the SOAS that uses Mesonet rain gage data.

**14. SUBJECT TERMS** - -hydrometeorology/ rainfall estimates/ WSR-88D (weather surveillance radar-Doppler) stage III/ Oklahoma Mesonet/ statistical objective analysis scheme/

**15. NUMBER OF PAGES**
40

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UL | UL | UL | UL |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

# STATISTICAL OBJECTIVE ANALYSIS SCHEME (SOAS) FOR IMPROVING WSR-88D RAINFALL ESTIMATES

## VOLUME 2

## SOAS VERSION 1.0 USER'S GUIDE

by

**Augusto J. Pereira Fo. and Kenneth C. Crawford**
**Oklahoma Climatological Survey,**
**University of Oklahoma**

**Curtis L. Hartzell**
**River Systems and Meteorology Group**
**Technical Service Center, Bureau of Reclamation**

Dam Safety Office
Reclamation Service Center
Denver, Colorado

December 1996

**UNITED STATES DEPARTMENT OF THE INTERIOR** ★ **BUREAU OF RECLAMATION**

## ACKNOWLEDGMENTS

The development of the SOAS (Statistical Objective Analysis Scheme) was sponsored by the Oklahoma Water Resources Board through an interagency contract (No. 960716) with the University of Oklahoma. Funding was provided by the following Reclamation (Bureau of Reclamation) offices: Commissioner's Office, Safety of Dams Office, Hydrologic Risk Reduction Program; and Great Plains Regional Office, Water Conservation Program. Partial support for the OCS (Oklahoma Climatological Survey) also was provided by UCAR/COMET under Grant No. NA37RJ0203.

*U.S. Department of the Interior*
*Mission Statement*

As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally-owned public lands and natural resources. This includes fostering sound use of our land and water resources; protecting our fish, wildlife, and biological diversity; preserving the environmental and cultural values of our national parks and historical places; and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to ensure that their development is in the best interests of all our people by encouraging stewardship and citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in island territories under U.S. administration.

## CONTENTS



## FIGURES

Figure

# 1. INTRODUCTION

This version of the SOAS (Pereira, 1996) performs rainfall analysis over a portion of the HRAP (hydrologic rainfall analysis project) grid of the Arkansas-Red River Basin. Its computer code (appendix) was written in FORTRAN 77 and was structured in a modular fashion to facilitate future modifications and upgrades (chap. 4). Internal code documentation is provided to complement the information provided in chapters 2 and 3.

The general sequence of processing in SOAS is indicated on figure 1. Stage III data from the WSR-88D (or NEXRAD [next generation radar]) network is archived using netCDF format and is indicated in the diagram as a NETCDF file. A variation of the SOAS that uses a Barnes first-pass scheme (Koch et al., 1983) was implemented to interpolate Mesonet observations to the HRAP grid whenever stage III data are not available. Thus, the output generated by the SOAS includes an analysis of rainfall for Mesonet-only and one for a combined Mesonet and stage III analysis.

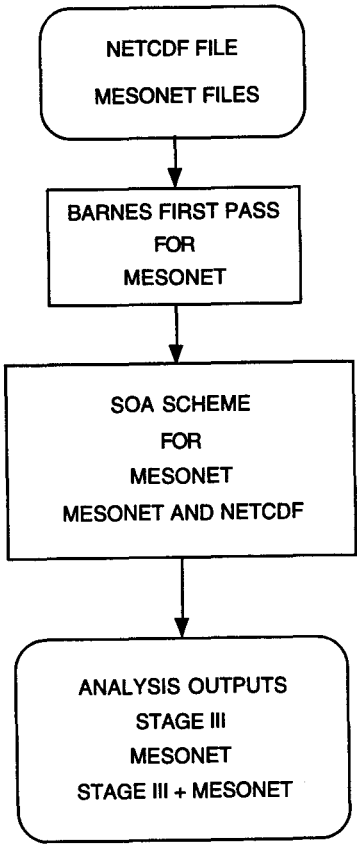


Figure 1. - General structure of SOAS.

This user's manual focuses on fundamentals that give the user a basic working knowledge of the SOAS. Because the SOAS will be applied to watershed projects of the Bureau of Reclamation across the 17 Western States, information on how to use the scheme elsewhere is provided as well as information on how to use WSR-88D data from stage I or II in the analysis.

A companion to this manuscript (vol. 1) was prepared to describe the mathematical development of the SOAS and its application to some case studies for the Lake Altus area in southwestern Oklahoma. Differences and similarities between stage III and the SOAS rainfall reanalysis are discussed. Based on the statistics available, a limited analysis of the number of rain gages in the Mesonet (Crawford, 1979) is provided to indicate error reduction versus rain gage network costs.

## 2. METHODOLOGY

### 2.1 SOA for WSR-88D Stage III and MESONET

The ABRFC (Arkansas-Red Basin River Forecast Center) currently provides an hourly stage III rainfall analysis over its HRAP grid (fig. 2), which is a composite of stage II data from all 17 WSR-88Ds (fig. 3) within their area of responsibility. Each polar stereographic grid cell has about 4- by 4-km resolution. Stage III data can be downloaded from ABRFC's WWW (world wide web) site *(http://info. abrfc. noaa. gov)*, where additional information is available to the general public.

Further improvement of their stage III analysis can be achieved by using the SOAS. For Oklahoma, more specifically over the Lake Altus basin project in southwestern Oklahoma, the stage III data are integrated with surface observations from the Oklahoma Mesonet (Brock et al., 1995; fig. 4), a network of 114 automatic stations which measure, among many other variables, the total rainfall accumulation at 5-minute time intervals. A description of Mesonet sites and products can be accessed through the Mesonet's WWW site (*http://geowww.gcn.uoknor.edu/WWW/Mesonet/Mesonet.html*).

These two data sources (stage III and Mesonet) are ingested into SOAS to produce the rainfall reanalysis. The stage III source is used as a background field and the Mesonet represents the required surface observations. Covariance statistics developed using the Twin Lakes WSR-88D are deemed to be representative of High-Plains weather systems. Previous studies (Pereira, 1996) indicated that a three-rain gage analysis ($K$=3) is needed to produce a minimum in the expected analysis error variance. Thus, the closest three rain gages to a grid point are identified to determine the background cross-correlations between the rain gages and the grid point and among the rain gages to determine weights, which are used to produce the final analysis. Observation errors are not used in this version of the SOAS. Its future implementation is discussed in chapter 4.

### 2.2 SOA for MESONET-only

When only Mesonet rain gages are available to produce the SOA, a background field is produced from a first pass Barnes analysis with the following weight function:

$$w_{ik} = \exp\left( -\frac{d_{ik}^2}{\kappa_o} \right) \tag{1}$$

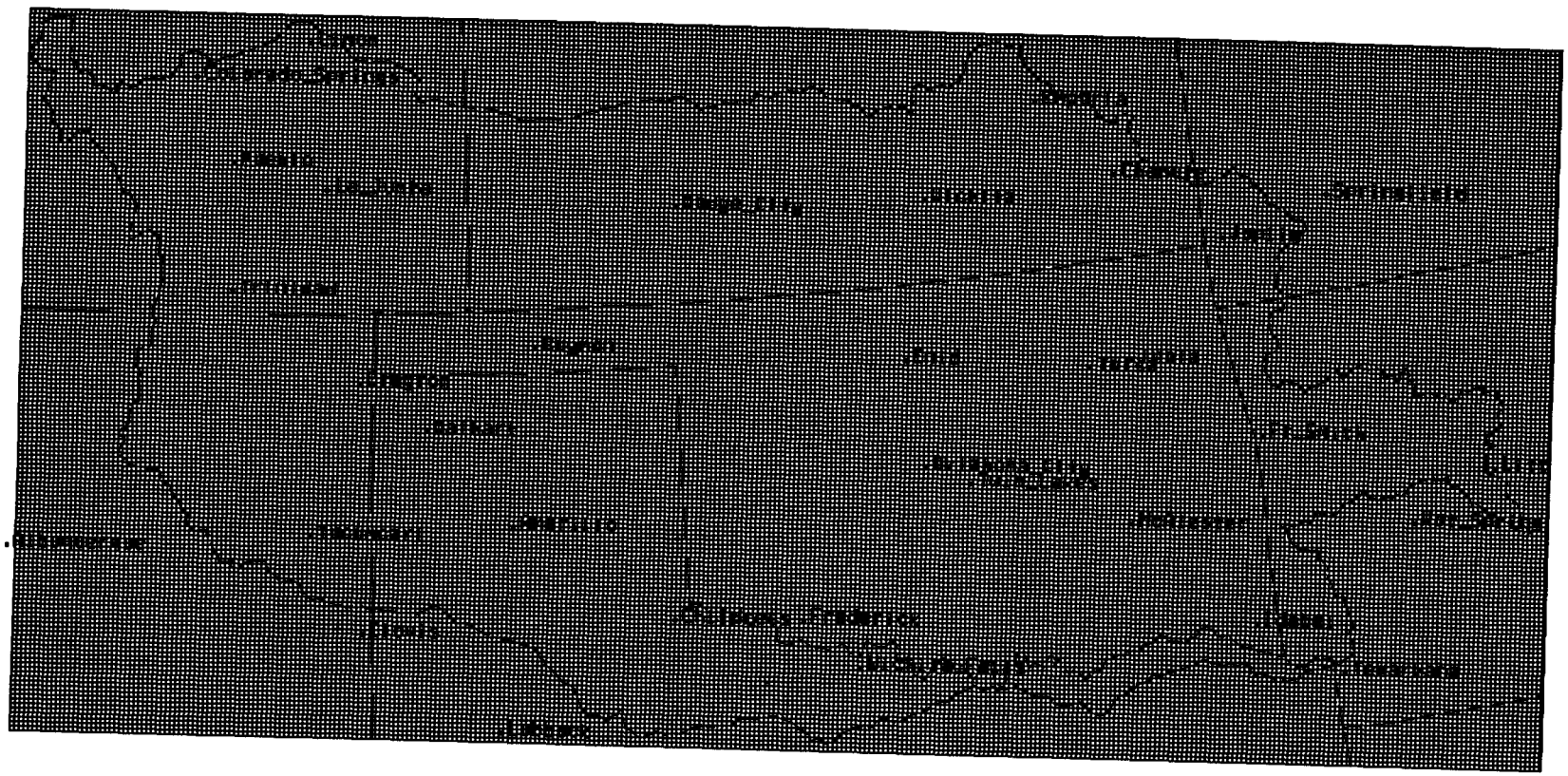

Figure 2. - Hydrologic rainfall analysis project (HRAP) grid of the Arkansas-Red Basin (adapted from ABRFC).

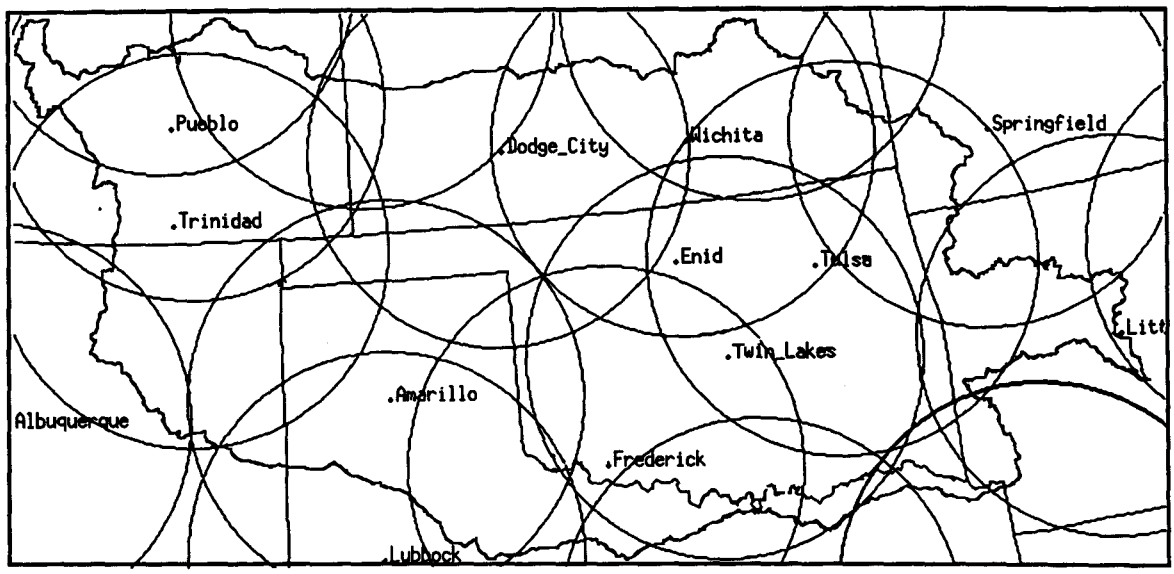


Figure 3. - WSR-88D network in the Arkansas-Red Basin area.

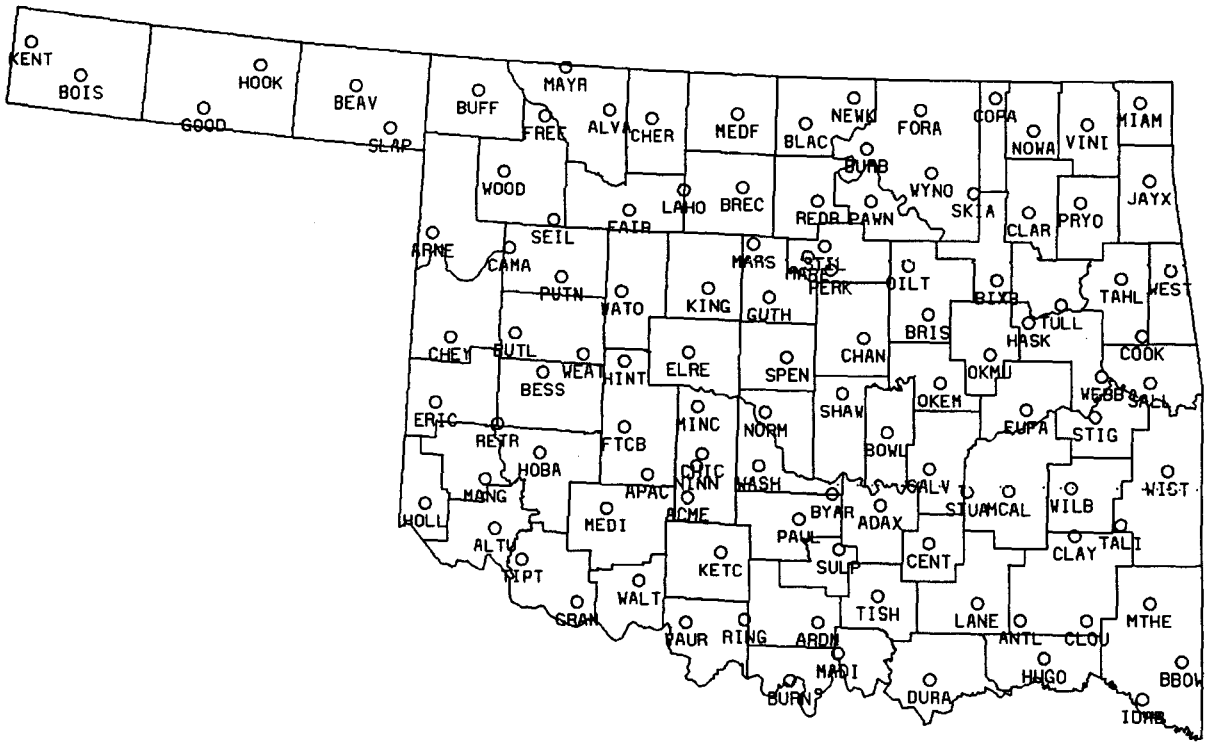


Figure 4. - The Oklahoma Mesonet sites (after Morris, 1996).

where: $d_{ik}$ = distance between grid point and rain gage (km)
$\kappa_o$ = distance constant (1924.72 km$^2$ for Mesonet)

All rain gages within a radius of 160 km of the analysis location are used to produce the background field. Once the field has been determined, the analysis field is interpolated back to rain gage locations before the SOAS (similar to section 2. 1) is used to produce the final (rain gage only) analysis.

Because the Barnes scheme is a low-pass filter analysis, small-scale features are smeared, which tends to broaden features in the analysis rainfall area. Clustering or sparse distribution of rain gages are not handled well by the Barnes scheme. Thus, one must be very cautious when using this scheme.

### 2.3 Code Structure and Subroutines

To facilate an understanding of the code and to make modifications more easily, the SOAS was structured such that no *goto* commands are used anywhere in the code. Additionally, each internal loop is offset from others to identify clearly the command lines. Equations, grid formats, and diagrams not shown in the main body of this manual are documented within the code (appendix). Supplementary references also are in the source code to guide the user to relevant reading material. All main components of the code are shown in the diagram of figure 5.

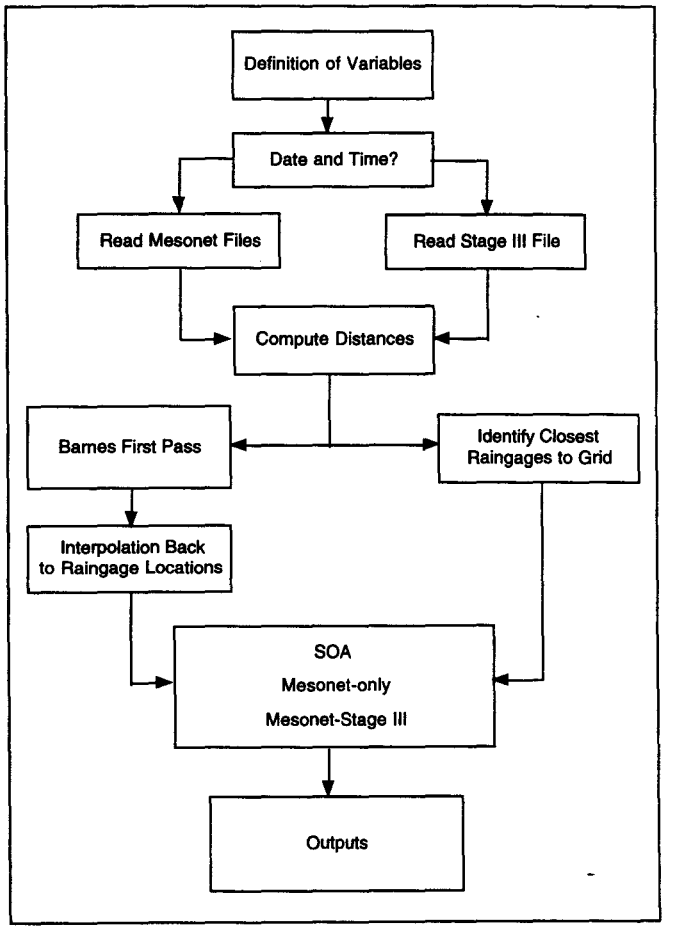


Figure 5. - Diagram of main code components for SOAS.

The source code has five subroutines:

(a) mesop: read Mesonet files
(b) netcdfread: read stage III files
(c) srdcmp: matrix factorization
(d) srsolv: matrix equation solver (*AX=B*)
(e) dtt : write analysis outputs

Subroutines (c) and (d) are used in the SOA to calculate the filtering weights. Subroutine (b) calls netcdf subroutines that are in the *netcdf.inc* include file. This file is available for various computer platforms for C and FORTRAN compilers. The user guide for this file can be found at *http://www.unidata.ucar.edulpackageslnetcdf/guide_toc.html.*

The subroutines used by the SOAS are described in the source code. Subroutine (a) was developed to read 5-minute Mesonet files, which is documented in the source code (appendix). Subroutine (e) generates four output files: SOA for Mesonet-only (*.sam), Mesonet-Stage III (*.soa), Stage III (*.stp), and the respective latitude and longitude of each grid in the analysis area. The default configuration generates output files for the Lake Altus area; grid coordinates are stored in the *latlonarea.altus* file.

## 3. USING THE SOFTWARE

### 3.1 Overview

Matrix and vector dimensions were declared as parameters to make modifications simple and easy. The three dimension parameters are: number of grid points in the east-west direction (*nx*), number of grid points in the south-north direction (*ny*), and maximum number of rain gages (*mxy*). These parameters have been preset for the HRAP grid and the number of Oklahoma Mesonet stations, respectively.

Four auxiliary parameters are used to set the initial and final indices of the analysis area: *ib* and *if*, and *jb* and *jf*, for east-west and south-north directions, respectively. The following auxiliary parameters also can be modified as needed: standard latitude *(stla)* and longitude *(stlo),* Barnes cutoff radius (*dic*) and distance constant (*k0*).

The standard latitude and longitude are polar stereographic parameters used to convert latitudes and longitudes into relative distance on the image plane and vice-versa. The Barnes cutoff radius and distance constants are associated with the Barnes first pass loop that depends on the density of the rain gage network.

To compile and link the program on a UNIX system, use:

(1) *f77 -c -l/[path]/netcdf.inc soas.f*
*(2) f77 soas.o -lnetcdf -osoas*

This code will generate the executable *soas.* The parameters required for the SOAS are the month (mm), day (dd), year (yy), and hour (hh). Any file name called by subroutine *mesop or netcdfread* that does not exist will return a message error and terminate the program.

If deemed necessary (e.g., to reduce computer time), the Mesonet-only analysis can be switched off or commented out from the main program code. This section of the code extends from the header "Precipitation analysis for MESONET..." until the line before the header "The SOA weights are calculated here."

Graphic interfaces, automatic data ingest and processing routines, "fool-proof" check points, and other features can be built around the SOAS as needed for different computer platforms. Except for these cosmetic details, development of the SOAS system is complete.

The formats for real-time and retrieved Mesonet data are different. In subroutine *mesop*, the format statement labeled **20** should be *64x* for retrieved and *69x* for real-time data (see appendix). Finally, in subroutine *netcdfread*, use *netcdf.inc* compatible with your computer platform (see section 2.3).

## 3.2 Step-by-Step Instructions on How to Use SOAS

### 3.2.1 Download Mesonet File from Your Mesonet ftp Account

Real-time Mesonet data are available at about 10 minutes after measurements are made. Rainfall measurements are accumulated from 0000 UTC to 0000 UTC. Thus, two Mesonet data files are needed to retrieve hourly rainfall accumulations (except for 0100 UTC): the present and the previous hour. Real time data files are labeled *mmddhhnn*.d05, where *mm* is the month, *dd* is the day, *hh* is the hour, and *nn* is the minute. Only files with *nn*=00 are needed.

Example: ftp> get 06041500.d05
ftp> get 06041600.d05

These two files are used to analyze the rainfall accumulation from 1500 UTC to 1600 UTC on June 4, 1995. Both the SOAS executable file and the data files should be in the same directory.

### 3.2.2 Download Stage III File from *http://info.abrfc.noaa. gov/pub/StageIII/1hr_netcdf*

This stage III binary file is available about 45 minutes after the hour. Hourly rainfall accumulations are estimated at the hour. Thus, just one data file is needed to run SOAS. Files are labeled: *mmddyyhh*z.nc, where *mm* is the month, *dd* is the day, *yy* is the year, *hh* the hour, and z represents UTC time.

Example: click on **06049516z.nc**

This step will download the hourly rainfall accumulation from 1500 UTC to 1600 UTC on June 4, 1995. Place the stage III file with Mesonet files and SOAS executable.

### 3.2.3 Run the SOAS Executable

Example: > type **soas** and press return.

The following will be displayed:

```
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
)))))                                                                        )))))
(((((                          Welcome to                                  (((((
)))))                                                                        )))))
(((((      Statistical Objective Analysis Scheme (SOAS)                      (((((
)))))                                                                        )))))
(((((                          Version 1.0                                 (((((
)))))                                                                        )))))
(((((                   Copyright (c) 1996                                 (((((
)))))          Oklahoma Climatological Survey                              )))))
(((((      University of Oklahoma.  All rights reserved.                    (((((
)))))                                                                        )))))
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
```

> enter data (mm,dd,yy,hh)
> type **06,04,95,16** and press return

The program (fig. 5) will read in Mesonet and stage III files. The driver program will retrieve the latitude and longitude of each Mesonet station and grid location, calculate distances between stations and grid points, and perform the analysis and generate four output files (section 2.3) before terminating.

The instructions above are valid for both retrieved and real-time Mesonet data. A directory should be made for each year to avoid overwriting files because Mesonet data files do not specify the year.

### 3.3 Using the SOAS with Other Data Sets

Individual WSR-88D hourly rainfall accumulations from stages I or II also can be used as background fields to produce the precipitation analysis anywhere in the WSR-88D network because other locations have a unique HRAP grid system. Moreover, stage III results from the mosaicing of stage II, which results from the processing of stage I hourly rainfall accumulations. More details on stage I (II) processing can be found at *http://www.abrfc.noaa.gov/StageI(II).html.*

To perform the rainfall analysis for a given National Weather Service RFC (river forecast center) area, check the RFC's HRAP array size and modify the parameters *nx* and *ny* to fit its dimensions. Also, check the number of rain gages available in the RFC area and adjust the parameter *mxy* accordingly.

Each HRAP grid point and rain gage has a latitude and longitude coordinate so that distances can be calculated. If a limited portion of the entire array is to be analyzed, change the indices *ib, if, jb, ff* corresponding to the limits of the area of interest.

Similarly, stage I and II rainfall arrays are loaded to their respective HRAP grid location within the RFC area. All parameters and indices above are declared in the main program. The input format for stage III (I and II) and rain gages are given by *mesop* and *netcdfread* subroutines, respectively. Output data are formatted by the *dtt* subroutine.

The background error covariances were developed for the Great Plains (specifically Oklahoma) rainfall systems. New statistics should be checked and/or developed for other areas of the United States. Refer to volume 1 for details on how to estimate background error covariances.

## 4. UPGRADING SOAS

As new data files become available, the SOAS can be further refined with the inclusion of a 2-D background error cross-correlation function that can be subdivided by season (winter and summer), by geographic location, by system type (stratiform and convective), and by any other features that may be relevant to reduce the expected analysis error variance. These refinements can be added to function *corr(d)* in the computer code by adding the relative coordinates of two points and by modifying the cross-correlation function from 1-D to 2-D.

Additionally, observational errors can be incorporated into the analysis equations (vol. 1). Methodologies can be (and have been) developed to measure rain gage errors that are caused by exposure of the equipment and the physical characteristics of the gage. Incorporating observation errors will render a more realistic analysis with a higher degree of reliability upon which economic decisions can be made. For the Mesonet, sensor errors are minimized by following strict guidelines that call for constant maintenance and calibration of the rain gages.

These additions will allow determination of the degree of improvement achieved by including 2-D statistics and observation errors. Nevertheless, a better understanding of the errors involved not only will improve the analysis but also will produce a more realistic analysis that can be of economic benefit to a user.

## 5. REFERENCES


Brock, F. V., K. C. Crawford, R. L. Elliott, G. W. Cuperus, S. J. Stadler, H. L. Johnson, and M. D. Eilts, 1995: The Oklahoma Mesonet: A technical overview. J. *Atmos. and Oc. Tech.,* **12**, 5-19.

Crawford, K. C., 1979: Considerations for the design of a hydrologic data network using multivariate sensors. *Water Resour. Res.,* **15**, 1752-1762.

Koch S., M. Desjardins, and P. Kocin, 1983: An interactive Barnes objective map analysis scheme for use with satellite and conventional data. *J. Climate Appl. Meteor.,* **22**, 1487-1503.

Moffis, D., 1996: The utility of mesoscale observations in refining diagnostics for the short-term prediction of convection. M.S. Thesis, Department of Meteorology, University of Oklahoma. In preparation.

Pereira, A. J. Fo., 1996: Mesoscale Rainfall fields: analysis and hydrometeorologic modeling. Ph.D. Dissertation, Department of Meteorology, University of Oklahoma. 150 pp.

# APPENDIX—SOAS SOURCE CODE

```
      program SOAS
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c     ((((((                                                                        ((((((
c     ))))))    Statistical Objective Analysis Scheme (SOAS)                       ))))))
c     ((((((                    Version 1.0                                         ((((((
c     ))))))                                                                        ))))))
c     ((((((                 Copyright (c) 1996                                     ((((((
c     ))))))            Oklahoma Climatological Survey                              ))))))
c     ((((((    University of Oklahoma.  All rights reserved.                       ((((((
c     ))))))                                                                        ))))))
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c
c     Oklahoma Climatological Survey
c     University of Oklahoma
c     Sarkeys Energy Center,
c     100 East Boyd, Suite 1210
c     Norman, OK  73019
c     Phone:  (405) 325-2542
c     FAX  :  (405) 325-2550
c     e-mail: apereira@stratus.gcn.uoknor.edu
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     program SOAS
c
c     created by Augusto J. Pereira 6-20-96
c
c     purpose:
c
c     This is the main program for the Statistical Objective Analysis
c     Scheme (SOAS).
c
c     SOAS integrates hourly precipitation estimates from Stage III data
c     (or any other WSR-88D hourly product on the HRAP grid format)
c     with surface observations. Both hourly precipitation estimates
c     and measurements are integrated through the SOAS to reduce errors,
c     thus refining the estimates of the spatial distribution of rainfall.
c
c     Please consult the SOAS user's manual for further details.
c
c
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     chronogram of modifications:
c
c     7-9-96  changed mesop to get generic preceding date
c             checked code for errors
c
c     7-10-96 added control to netcdfread since, for some reason,
c             some archives do not have reference lat & lon (e.g,
c             0300-0600 UTC on 7-9-96)
c
c     7-15-96 included the normalized analysis error
c             variance (naev) matrix
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     grid arrangement
c
c
```

```
c                 x-direction, West-East:
c
c            +-----+-----+-----+-----+-----+-----+
c        i=  1     2      . . .     NX-2  NX-1   NX
c
c
c                 y-direction, South-North:
c
c            +-----+-----+-----+-----+-----+-----+
c        j=  1     2      . . .     NY-2  NY-1   NY
c
c
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     Declaration of variables
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      implicit none            ! explicit declarations only

      integer nx               ! number of grid points on x-direction
      integer ny               ! number of grid points on y-direction
      integer mxy              ! number of raingages (maximum)
      integer nr               ! number of raingages (actual)
      integer na               ! number of raingages to be used for each
                               ! analysis grid point
      integer ib               ! index of starting grid point analysis on
                               ! x-direction
      integer if               ! index of ending grid point analysis on
                               ! x-direction
      integer jb               ! index of starting grid point analysis on
                               ! y-direction
      integer jf               ! index of ending grid point analysis on
                               ! y-direction
      integer mm               ! month
      integer dd               ! day
      integer yy               ! year
      integer hh               ! hour

      parameter (nx=335)
      parameter (ny=159)
      parameter (mxy=200)
      parameter (na=3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     Lake Altus area coordinates
c
c                                          HRAP coordinates
c     lat                                (146,60)   (168,62)
c     35.6  +--------+                              +
c           |        |                 +
c           |        |
c           |        |
c           |        |                                   +
c     34.4  +------- +                    +
c                                        (149,27)     (172,29)
c         100.0    99.0 lon
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      parameter (ib=146)
      parameter (if=172)
      parameter (jb=27)
```

```
      parameter (jf=62)

c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     variables:
c
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      real stp(nx,ny)            ! Stage III precipitation (mm)
      real mep(mxy)              ! raingage precipitation (mm)
      real sop(nx,ny)            ! SOAS for stp + mep (mm)
      real sbp(nx,ny)            ! Barnes (1st pass) + SOAS (2nd pass)
                                 ! for mep only (mm)
      real fbp(nx,ny)            ! Barnes 1st pass analysis for mep only
      real mbp(mxy)              ! Barnes 1st pass interpolation back to
                                 ! mep location (mm)
      real naev(nx,ny)           ! normalized analysis error variance

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     arrays of grid coordinates:
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real sla(nx,ny)            !latitude of stp grid (deg)
      real slo(nx,ny)            !longitude of stp grid (deg)
      real mla(mxy)              !latitude of mep grid (deg)
      real mlo(mxy)              !longitude of mep grid (deg)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     arrays of grid polar stereographic distances
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real sxd(nx,ny)            !x-direction relative distance
                                 !of stp grid on image plane (km)

      real syd(nx,ny)            !y-direction relative distance
                                 !of stp grid on image plane (km)

      real mxd(mxy)              !x-direction relative distance
                                 !of mep grid on image plane (km)

      real myd(mxy)              !y-direction relative distance
                                 !of mep grid on image plane (km)

      real dis(nx,ny,na)         !relative distances between stp grid
                                 !point and the 3 closest mep points

      real dim(mxy)              !relative distance between a mep
                                 !to the closest stp grid

      integer idex(nx,ny,na)     !corresponding mxy value of 3 closest
                                 !mep points to stp grid point

      integer  ic(mxy)           !corresponding nx value of closest
                                 !stp grid point to mep

      integer  jc(mxy)           !corresponding ny value of closest
                                 !stp grid point to mep
```

```
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     polar stereographic parameters
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real stla                  !standard latitude (deg)
      real stlo                  !standard longitude (deg)
      real erad                  !radius of the earth (km)
      real pi                    !trigonometric constant

      parameter (stla=60)
      parameter (stlo=105)
      parameter (erad=6371.2)
      parameter (pi=3.14159)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c      Barnes parameters (valid for MESONET area only)
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real dic                  !cutoff radius (km)
      real k0                   !distance constant for MESONET (km2)

      parameter (dic=160)
      parameter (k0=1924.72)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     auxiliary parameters
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real co(na,na)            !matrix of cross-correlations used by SOA
      real coin(na)             !vector of cross-correlations used by SOA
      real smat(na,na)          !matrix of upper triangular cross-corr
      real we(na)               !vector of weights for SOA
      real corr                 !real function which calculates
                                !cross-correlations
      real sigma                !real function wich estimates average map
                                !factor for polar stereographic distances
      real  aux1
      real  aux2
      real  aux3
      real  aux4
      real  duax
      real  dmin
      real  dmax
      real  det
      real  d
      real  dx
      real  dy
      real  d1
      real  d2
      real  a1
      real  a2
      real  a3
      real  a4
      real  wt
      real  w
      integer i
      integer j
      integer k
```

```
      integer l
      integer i1
      integer i2
      integer j1
      integer j2
      integer ia
      integer ja
      integer l1
      integer k1
      integer ist

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                  four letter name of mesonet stations
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      character*4  mname(mxy)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     beginning of executable
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c
      write(6,'(/16(/5x,a)//)')

     # ')))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))',
     # '((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((',
     # ')))))                                                          )))))',
     # '(((((                      Welcome to                          (((((',
     # ')))))                                                          )))))',
     # '(((((   Statistical Objective Analysis Scheme (SOAS)           (((((',
     # ')))))                                                          )))))',
     # '(((((                     Version 1.0                          (((((',
     # ')))))                                                          )))))',
     # '(((((                  Copyright (c) 1996                      (((((',
     # ')))))            Oklahoma Climatological Survey                )))))',
     # '(((((   University of Oklahoma.  All rights reserved.           (((((',
     # ')))))                                                          )))))',
     # '((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((',
     # ')))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))'

      print *, 'enter data (mm,dd,yy,hh)'
      read(6,*) mm,dd,yy,hh

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     read in mesonet data file
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call mesop(mm,dd,yy,hh,mep,mla,mlo,mname,mxy,nr)


c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     read in Stage III data file
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call netcdfread(mm,dd,yy,hh,sla,slo,stp,nx,ny)
```

```
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     calculation of distances on the image plane:
c
c     x=erad*([1+sin(stla)]/[1+sin(sla)]}*cos(sla)*sin(stlo-slo)
c     y=erad*([1+sin(stla)]/[1+sin(sla)]}*cos(sla)*cos(stlo-slo)
c
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      aux1=pi/180.                        !conversion from deg to rad

      aux2=erad*(1.+sin(stla*aux1))       !take this portion out of the
                                          !internal loop to reduce number
                                          !of calculations

c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     stp matrix
c
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      do i=1,nx
           do j=1,ny
              duax=(aux2/(1+(sin(sla(i,j)*aux1))))*cos(sla(i,j)*aux1)
              aux3=(stlo-slo(i,j))*aux1
              sxd(i,j)=duax*sin(aux3)
              syd(i,j)=duax*cos(aux3)
         enddo
      enddo


c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     mep vector
c
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      do i=1,nr
         duax=(aux2/(1+(sin(mla(i)*aux1))))*cos(mla(i)*aux1)
         aux3=(stlo-mlo(i))*aux1
         mxd(i)=duax*sin(aux3)
         myd(i)=duax*cos(aux3)
      enddo


c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     identify na closest raingages (mep) to grid point (stp)
c          and the closest grid point to raingage
c
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      do k=1,nr
         dim(k)=999999999.        !load minimum distance with big number
      enddo

      do l=1,na
```

```
      do i=1,nx
         do j=1,ny
            dmin= 999999999.
            do k=1,nr
               d=sqrt((mxd(k)-sxd(i,j))**2.+(myd(k)-syd(i,j))**2.)
               if(d.lt.dim(k)) then
                                    dim(k)=d
                                    ic(k)=i
                                    jc(k)=j
                 endif
               if(l.eq.1) then
                            if(d.lt.dmin) then
                                          dmin=d
                                          idex(i,j,l)=k
                            endif
                         else
                            if(d.lt.dmin.and.d.gt.dis(i,j,l-1)) then
                                         dmin=d
                                         idex(i,j,l)=k
                            endif
               endif
            enddo
            dis(i,j,l)=dmin
         enddo
      enddo
   enddo


c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     Precipitation analysis for MESONET only uses a first pass Barnes
c                analysis to generate a background field.
c
c     Reference: Koch et al.,1983,J. Climate Appl. Meteor,22,1487-1503
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      do i=1,nx
         do j=1,ny
            wt=0
            do k=1,nr

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                        image plane distance
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

               d=sqrt((sxd(i,j)-mxd(k))**2.+(syd(i,j)-myd(k))**2.)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                       distance on the earth
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

               d=(d*2)/(sigma(sla(i,j)) + sigma(mla(k)))

               if(d.lt.dic) then
                                w=exp(-d*d/(1.0*k0))
                                fbp(i,j)=fbp(i,j)+w*mep(k)
                                wt=w+wt
               endif
```

```
            enddo
            if(wt.gt.0) fbp(i,j)=fbp(i,j)/wt
         enddo
      enddo


c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                     Interpolation of fbp grid
c                 back to mep locations for second pass
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      do k=1,nr

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     closest fbp grid point (calculated through stp grid) to mbp is
c     used to identify the other three grid points that contain mbp
c
c
c                       +         +          +  j+1
c
c                           II         I
c
c                       +         x          +   j
c
c                           III        IV
c
c                       +         +          +  j-1
c
c                     i-1         i         i+1
c
c               x = closest point to raingage location
c
c             dx > 0 & dy > 0 => raingage in quadrant II
c             dx > 0 & dy < 0 => raingage in quadrant III
c             dx < 0 & dy > 0 => raingage in quadrant I
c             dx < 0 & dy < 0 => raingage in quadrant IV
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

         ia=ic(k)
         ja=jc(k)
         dx=sxd(ia,ja)-mxd(k)
         dy=syd(ia,ja)-myd(k)
         i1=ia
         j1=ja
         if(dx.gt.0) then
                        i2=ia-1
                        if(dy.gt.0) then
                                        j2=ja+1
                                    else
                                        j2=ja-1
                        endif
                     else
                        i2=ia+1
                        if(dy.gt.0) then
                                        j2=ja+1
                                    else
                                        j2=ja-1
                        endif
         endif
```

```
         if(i2.lt.i1) then
                        ia=i2
                        i2=i1
                        i1=ia
         endif
         if(j2.lt.j1) then
                        ja=j2
                        j2=j1
                        j1=ja
         endif

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c            estimates bilinear interpolation coefficients
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

         d1=(myd(k)-syd(i1,j2))/(syd(i1,j1)-syd(i1,j2))
         d2=(mxd(k)-sxd(i1,j2))/(sxd(i2,j1)-sxd(i1,j1))
         a1=d1-d1*d2
         a2=d1*d2
         a3=d2-d1*d2
         a4=d1*d2-d1-d2+1.

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c            interpolation of mbp grid back to raingage location
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

         mbp(k)=a1*fbp(i1,j1)+a2*fbp(i2,j1)+a3*fbp(i2,j2)+a4*fbp(i1,j2)
      enddo



c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     the SOA weights are calculate here
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      do i=ib,if
      do j=jb,jf
            do k=1,na
            do l=1,na
                  k1=idex(i,j,k)
                  l1=idex(i,j,l)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                 distance between raingages on image plane
c
c
c                    3rd raingage            2nd raingage
c                         x                       x
c
c
c                                      grid
c                                        +
c
c
c                                        x
c                                   1st raingage
```

```
c
c      distances: d12, d13 and d23
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                     d=sqrt((mxd(k1)-mxd(l1))**2.+(myd(k1)-myd(l1))**2.)

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                    distance between raingages on the earth
c                           (get average image factor)
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                     d=(2.*d)/(sigma(mla(k1))+sigma(mla(l1)))

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                    fill in cross-correlation matrix
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                     co(k,l)=corr(d)
              enddo
              enddo

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c      prepare cross-correlation matrix inversion
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

              call srdcmp(na,na,co,smat,det,ist)

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c      determines cross-correlation vector between raingages and grid
c
c      distances dg1, dg2 and dg3
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

              do k=1,na
                     k1=idex(i,j,k)

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                    distance between raingage and grid on image plane
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                     d=sqrt((mxd(k1)-sxd(i,j))**2.+(myd(k1)-syd(i,j))**2.)

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                    distance between raingage and grid on the earth
c
c      ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                        d=(2.*d)/(sigma(mla(k1))+sigma(sla(i,j)))

c      ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                    fill in vector of cross-correlations
```

```
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                  coin(k)=corr(d)
            enddo

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     finally, solve for the weights
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

            call srsolv(na,na,smat,coin,we)



c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                                    SOA equation
c
c                 Reference: Daley,1991, atmospheric data analusis
c                        Cambridge University Press, chpt 4.
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

            do k=1,na
               kl=idex(i,j,k)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c          for mep and stp
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

               sop(i,j)=sop(i,j)+we(k)*(mep(kl) - stp(ic(kl),jc(kl)))

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c          for mep only (used Barnes on first pass)
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

               sbp(i,j)=sbp(i,j)+we(k)*(mep(kl) - mbp(kl))

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c      optional : normalized analysis error variance
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

c              naev(i,j)=naev(i,j)-we(k)*coin(k)

            enddo
            sop(i,j)=sop(i,j)+stp(i,j)
            sbp(i,j)=sbp(i,j)+fbp(i,j)
c           naev(i,j)=naev(i,j)+1.

      enddo
      enddo



c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
```

```
c
c     print results to files:
c                              mmddyyhh.stp  for stage III only
c
c                              mmddyyhh.soa  for stage III + mesonet
c
c                              mmddyyhh.sam  for mesonet only
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call dtt(mm,dd,yy,hh,stp,sop,sbp,sla,slo,nx,ny,ib,if,jb,jf)

      stop

      end




c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     cross-correlation model for 1 hour precipitation accumulation
c                       (polynomial fit)
c
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      real function corr(d)
      real d                           !distance between two points (km)
      real caux                        !auxiliary number

      caux = 0.99782             -0.037183*d
     #     +0.0015733*(d**2)    -5.9063e-5*(d**3)
     #     +1.4311e-6*(d**4)    -2.1448e-8*(d**5)
     #     +1.9885e-10*(d**6)   -1.1110e-12*(d**7)
     #     +3.4260e-15*(d**8)   -4.4737e-18*(d**9)

      corr=caux
      return
      end




c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
c
c     calculation of image scale factor of a given latitude
c               (polar stereographic projection)
c
c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

      real function sigma(lat)
      real lat
      real stla
      real saux
      real pi
      parameter (stla=60)
      parameter (pi=3.14159)
      saux = 1 + sin(stla*pi/180.)
      saux = saux/(1 + sin(lat*pi/180.))
      sigma=saux
```

```
      return
      end


c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c         The following three subroutines were taken from an objective
c                 analysis package to solve for the SOA weights
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

        SUBROUTINE SRDCMP(NMAX,N,A,S,DET,STATUS)
C
C PURPOSE        THE SQUARE ROOT FACTORIZATION OF SYMMETRIC, POSITIVE
C                DEFINITE MATRIX A.
C
C ARGUMENTS      NMAX - I  MAXIMUM NUMBER OF ROWS IN MATRIX A
C                N ---- I  ACTUAL NUMBER OF ROWS IN MATRIX A
C                A ---- I  SYMMETRIC, POSITIVE DEFINITE MATRIX OF
C                          ORDER N
C                S ----  O UPPER TRIANGULAR FACTOR OF MATRIX A
C                DET --  O DETERMINANT OF MATRIX A
C
C DIMENSIONS     A(NMAX,N), S(NMAX,N)
C
C                [RGR.LIB.TEST]TEST.COM        TEST PROCEDURE
C                [RGR.LIB.TEST]LINALG.IM       TEST CODE
C
C REFERENCE      CARNAHAN, B., H.A. LUTHER, AND J.O. WILKES, 1969.
C                APPLIED NUMERICAL METHODS. JOHN WILEY AND SONS,
C                NEW YORK. PAGE 334. (604 PP.)
C
C ALGORITHM      MATRIX A IS FACTORED INTO THE PRODUCT OF A LOWER
C                TRIANGULAR MATRIX, S', AND AN UPPER TRIANGULAR
C                MATRIX, S. A = S'S. NO PIVOTING IS PERFORMED.
C
C REMARKS        SUBROUTINE SRSOLV IS USED TO SOLVE THE LINEAR
C                EQUATION AX = B.
C
C                MATRIX S MAY OVERWRITE MATRIX A IF DESIRED.
C
C HISTORY        MAY 83 - CONVERTED ACCUMULATORS FROM SINGLE TO
C                         DOUBLE PRECISION.
C                NOV 82 - CONVERTED FROM IFTRAN TO IFMAC.
C                         MODIFIED AT PROFS BY G. RASMUSSEN.
C                JAN 81 - WRITTEN AT NCAR BY G. RASMUSSEN.
C                APR 85 - FOR'S CHANGED TO DO'S FOR SPEED BY S. BENJAMIN.
C                MAY 86 - CONVERTED FROM IFMAC TO FORTRAN BY S. BENJAMIN.
C                         ALSO ADDED STATUS PARAMETER.


      IMPLICIT NONE

      INTEGER I,J,K,IM1,N, NMAX, STATUS

      DOUBLE PRECISION SII, SIJ, DT
      REAL A(NMAX,Nmax), S(NMAX,Nmax), DET

      STATUS = 1
```

```
      DT = 1.D0
      DO I=1,N
         IM1 = I - 1
         SII = A(I,I)
         DO K = 1,IM1
            SII = SII - S(K,I)*S(K,I)
         END DO
         IF (SII.LE.0.D0) THEN
                        DET = 0.
                        WRITE (6,5) I
5                       FORMAT (/,' ERROR AT ROW AND COLUMN ',I3,
     1        ' MATRIX IS NOT POSITIVE DEFINITE')
                        STATUS = 0
                        RETURN
        END IF
        SII = dSQRT(SII)
        S(I,I) = SII
        DT = DT*SII
        DO J = I+1,N
           SIJ = A(I,J)
           DO K = 1 , IM1
              SIJ = SIJ - S(K,I)*S(K,J)
           END DO
           S(I,J) = SIJ/SII
        END DO
      END DO
      DET = DT*DT
      RETURN
      END
```

```
      SUBROUTINE SRINV(NMAX,N,S,AI)
C
C PURPOSE        COMPUTE THE INVERSE OF MATRIX A = S'S.
C
C ARGUMENTS      NMAX ---- I  MAXIMUM NUMBER OF ROWS IN MATRIX A
C                N ------- I  ACTUAL NUMBER OF ROWS IN MATRIX A
C                S ------- I  UPPER TRIANGULAR FACTOR OF MATRIX A
C                AI ------  O INVERSE OF MATRIX A
C
C DIMENSIONS     S(NMAX,NMAX), AI(NMAX,NMAX)
C
C FILES          [RGR.LIB.TEST]TEST.COM        TEST PROCEDURE
C                [RGR.LIB.TEST]LINALG.IM       TEST CODE
C
C REFERENCE      CARNAHAN, B., H.A. LUTHER, AND J.O. WILKES, 1969.
C                APPLIED NUMERICAL METHODS. JOHN WILEY AND SONS,
C                NEW YORK. PAGE 334. (604 PP.)
C
C
C ALGORITHM      MATRIX AI IS GIVEN BY AI = SI X SI(T) WHERE SI IS THE
C                INVERSE OF MATRIX S.
C
C REMARKS        SUBROUTINE SRDCMP FACTORS MATRIX A.
C
C                MATRIX AI MAY OVERWRITE MATRIX S.
C
C HISTORY        MAY 83 - CONVERTED ACCUMULATORS FROM SINGLE TO
C                         DOUBLE PRECISION.
C                NOV 82 - CONVERTED FROM IFTRAN TO IFMAC.
```

```
C                       MODIFIED AT PROFS BY G. RASMUSSEN.
C               JAN 81 - WRITTEN AT NCAR BY G. RASMUSSEN.
C               MAY 86 - CONVERTED FROM IFMAC TO FORTRAN BY S. BENJAMIN

      IMPLICIT NONE

      INTEGER I,J,K,N,NMAX
      DOUBLE PRECISION SUM
      REAL S(NMAX,NMAX), AI(NMAX,NMAX)

      DO I = N , 1 , -1                    ! COMPUTE SI'
         AI(I,I) = 1./S(I,I)
         DO J = I+1 , N
            SUM = 0.D0
            DO K = I+1 , J
               SUM = SUM - S(I,K)*AI(J,K)
            END DO
            AI(J,I) = SUM*AI(I,I)
         END DO
      END DO
      DO I=1,N                             ! COMPUTE AI = SI SI'
         DO J=1,I
            SUM = 0.D0
            DO K=I,N
               SUM = SUM + AI(K,I)*AI(K,J)
            END DO
            AI(I,J) = SUM
            AI(J,I) = SUM
         END DO
      END DO
      RETURN
      END



      SUBROUTINE SRSOLV(NMAX,N,S,B,X)
C
C PURPOSE        SOLVE MATRIX EQUATION AX = B.
C
C ARGUMENTS      NMAX ---- I  MAXIMUM NUMBER OF ROWS IN MATRIX A
C                N ------- I  ACTUAL NUMBER OF ROWS IN MATRIX A
C                S ------- I  UPPER TRIANGULAR FACTOR OF MATRIX A
C                B ------- I  RIGHT-HAND SIDE VECTOR OF LENGTH N.
C                X -------  O SOLUTION VECTOR OF LENGTH N
C
C DIMENSIONS     S(NMAX,N), B(N), X(N)
C
C FILES          [RGR.LIB.TEST]TEST.COM        TEST PROCEDURE
C                [RGR.LIB.TEST]LINALG.IM       TEST CODE
C
C REFERENCE      CARNAHAN, B., H.A. LUTHER, AND J.O. WILKES, 1969.
C                APPLIED NUMERICAL METHODS. JOHN WILEY AND SONS,
C                NEW YORK. PAGE 334. (604 PP.)
C
C ALGORITHM      THE SOLUTION IS OBTAINED IN TWO STEPS.
C                FIRST S'Y = B IS SOLVED FOR VECTOR Y BY FORWARD
C                SUBSTITUTION. NEXT SX = Y IS SOLVED FOR VECTOR X
C                BY BACKWARD SUBSTITUTION.
C
C REMARKS        MATRIX S IS OBTAINED BY A CALL TO SUBROUTINE SRDCMP.
C
```

```
C               VECTOR X MAY OVERWRITE VECTOR B.
C
C HISTORY       MAY 83 - CONVERTED ACCUMULATORS FROM SINGLE TO
C                        DOUBLE PRECISION.
C               NOV 82 - CONVERTED FROM IFTRAN TO IFMAC.
C                        MODIFIED AT PROFS BY G. RASMUSSEN.
C               JAN 81 - WRITTEN AT NCAR BY G. RASMUSSEN.
C               MAY 86 - CONVERTED FROM IFMAC TO FORTRAN BY S. BENJAMIN
C

      IMPLICIT NONE

      INTEGER I,J,K,N,NMAX
      DOUBLE PRECISION SUM
      REAL S(NMAX,Nmax), X(Nmax), B(Nmax)

      DO K=1,N                      ! FORWARD SUBSTITUTION, S'Y = B
         SUM = B(K)
         DO J=1 ,K-1
            SUM = SUM-S(J,K)*B(J)
         END DO
         B(K) = SUM/S(K,K)
      END DO
      DO K = N , 1 , -1            ! BACKWARD SUBSTITUTION,  SX = Y
         SUM = B(K)
         DO J = K+1 , N
            SUM = SUM-S(K,J)*X(J)
         END DO
         X(K) = SUM/S(K,K)
      END DO
      RETURN
      END

      end



c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     subroutine mesop extracts hourly mesonet precipitation from 5 min.
c     ascii files. The filename format is mmddhhnn.d05, where:
c
c                              mm  = month
c                              dd  = day
c                        yy  = year
c                              hh  = hour
c                              nn  = minute
c                              d05 = identifier for 5 min data
c
c          Precipitation is accumulated from 0000 UTC to 0000 UTC.
c
c                      written by Augusto Pereira 6-26-96
c
c          Modified on 7-9-96 to generalize the calculus of preceding date
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

        subroutine mesop(mm,dd,yy,hh,mep,mla,mlo,mname,mxy,nr)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
```

```
c     Declaration of variables
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      implicit none                  !explicit declarations only

      integer mxy                    !maximum number of raingages
      integer nr                     !number of raingages on filename
      integer mm                     !month
      integer dd                     !day
      integer yy                     !year
      integer hh                     !hour
      integer i
      integer j
      integer k
      integer l
      integer m
      integer n
      integer coun
      integer coun1
      integer coun2

      real mep(mxy)                  !hourly raingage precipitation (mm)
      real amp(mxy,2)                !auxiliary array to store precip
      real mla(mxy)                  !latitude of raingage  (deg)
      real ala(mxy)                  !auxiliary latitude (deg)
      real mlo(mxy)                  !longitude of raingage (deg)
      real alo(mxy)                  !auxiliary longitude (deg)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     more auxiliary variables
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      integer lat
      integer lam
      integer las
      integer log
      integer lom
      integer los
      real paux
      integer*4 ier
      integer*4 iostat
      integer nd(12)                 !number of days in each month of
                                     !a nonleap year
      character*4 mname(mxy)         !four letter station name
      character*4 aname(mxy)         !auxiliary station name vector
      character*4 xname(mxy,2)       !auxiliary station name array
      character*15 filename          !input filename
      character*2 num(32)            !encode numbers from 0 to 31
      character*4 auxs               !auxiliary station

      data num/'01','02','03','04','05','06','07','08','09','10',
     #         '11','12','13','14','15','16','17','18','19','20',
     #         '21','22','23','24','25','26','27','28','29','30',
     #         '31','00'/

      data nd/31,28,31,30,31,30,31,31,30,31,30,31/

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c           open archive with mesonet stations and coordinates
c
```

```
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

        open(unit=3,file='mesoinfo.tbl',status='old',form='formatted')
        ier=0
        j=0
        do while(ier.eq.0)
        read(3,10,iostat=ier)auxs,lat,lam,las,log,lom,los
10      format(5x,a4,56x,3i2,2x,i3,2i2)
            if(ier.eq.0) then
                             j=j+1
                             ala(j)=lat+lam/60.+las/3600.
                             alo(j)=log+lom/60.+los/3600.
                             aname(j)=auxs
            endif
        enddo
        coun=j
        close(3)
        ier=0

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                                encode filename
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

        filename(1:2)=num(mm)
      filename(3:4)=num(dd)
      if(hh.eq.0) then
                      filename(5:6)=num(32)
                        else
                      filename(5:6)=num(hh)
      endif
      filename(7:12)='00.d05'

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     read in just precipitation accumulation
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      open(3,file=filename,status='old',form='formatted')

      i=0
      ier=0
      read(3,*)                        !read header
      do while(ier.eq.0)
              i=i+1
              read(3,20,iostat=ier) xname(i,1),amp(i,1)
20            format(1x,a4,64x,f6.2) !69x for real time data
                                     !64x is used for retrieved data
                                     !check mesonet archive format
      enddo
      coun1=i-1
      close(3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c      except for hh=01, all other time intervals need the previous hour
c        precipitation accumulation to calculate 1 hour accumulation
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      if(hh.ne.1) then
                     if(hh.eq.0) then
```

```
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     to be generic, all possibilities need to be considered to calculate
c                    the date of preceding day when hh=0
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

              filename(5:6)=num(23)
              if(dd-1.eq.0) then
                        if(mm-1.eq.0) then
                                     filename(1:2)=num(12)
                                     filename(3:4)=num(31)
                                  else
                                  if(mm-1.eq.2) then
                                           filename(1:2)=num(2)
                                     if(mod(yy,4).eq.0) then
                                           filename(3:4)=num(29)
                                                        else
                                           filename(3:4)=num(28)
                                     endif
                                                 else
                                     filename(1:2)=num(mm-1)
                                     filename(3:4)=num(nd(mm-1))
                                   endif
                        endif
                            else
                        filename(3:4)=num(dd-1)
               endif

                             else
                                  filename(5:6)=num(hh-1)
                     endif
                       open(3,file=filename,status='old',form='formatted')
                       i=0
                       ier=0
                     read(3,*)                          !read header
                       do while(ier.eq.0)
                             i=i+1
                               read(3,20,iostat=ier) xname(i,2),amp(i,2)

                       enddo
       endif
         close(3)
       coun2=i-1

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                fill in mep vector with 1 hour accumulations
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

        if(hh.eq.1) then

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c          first hour of precipitation accumulation (hh=01) is ready
c            just need to get corresponding latitude and longitude
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                      do i=1,coun1
                         mep(i)=amp(i,1)
                         mname(i)=xname(i,1)
```

```
                    j=0
                    k=0
                    do while(k.eq.0)
                       j=j+1
                       if(xname(i,1).eq.aname(j)) then
                                                  mla(i)=ala(j)
                                                  mlo(i)=alo(j)
                                                  k=1
                       endif
                       if(j.eq.coun) k=1
                    enddo
                 enddo
                 k=coun1

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                 other time intervals need extra work
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                else
                  k=0
                  do i=1,coun1
                     m=0
                     j=0
                     do while(m.eq.0)
                     j=j+1

c      )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c          first verify if station reported data for both hh & hh-1
c
c      (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                        if(xname(i,1).eq.xname(j,2)) then
                                     paux=amp(i,1)-amp(j,2)

c      )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c           if so, than verify if the accumulation is greater
c                        or equal to zero
c
c      (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                                     if(paux.ge.0) then
                                          k=k+1
                                          mep(k)=paux
                                          mname(k)=xname(i,1)

c      )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                 get latitude and longitude of the station
c
c      (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

                        n=0
                        l=0
                        do while(n.eq.0)
                           l=l+1
                           if(xname(i,1).eq.aname(l)) then
                                              mla(k)=ala(l)
                                              mlo(k)=alo(l)
                                              n=1
                                              m=1
```

```
                        endif
                        if(l.eq.coun) n=1
                     enddo
                                 endif
                     endif
                     if(j.eq.coun2) m=1
                  enddo
               enddo
      endif
      nr=k
      return
        end


c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     subroutine netcdfread extracts hourly Stage III precipitation
c            from netCDF files generated by Tulsa RFC.
c           the filename format is mmddyyhhz.nc, where:
c
c                              mm    =  month
c                              dd    =  day
c                              yy    =  year
c                              hh    =  hour
c                              z.nc  =  identifier for 1 hour data
c
c              written by Augusto Pereira 6-28-96
c
c              modified on 7-10-96 to use internal values of
c              reference latitude and longitude if they  are
c              not written to netcdf file.
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      subroutine netcdfread(mm,dd,yy,hh,sla,slo,stp,nx,ny)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     declaration of variables
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      implicit none                 !explicit declarations only

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c        data extraction is done by netCDF functions and subroutines
c      thus, need to include path of netcdf include file on your system
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
      include '/usr/local/include/netcdf.inc' !path on caps cluster

      integer nx                    !x-direction dimension of stp matrix
      integer ny                    !y-direction dimension of stp matrix
      integer mm                    !month
      integer dd                    !day
      integer yy                    !year
      integer hh                    !hour
      integer ll                    !number of reference latitudes and
                                    !longitudes (4 corners of hrap grid)
```

```
      real stp(nx,ny)              !precipitation matrix
      real sla(nx,ny)              !matrix of stp latitudes
      real slo(nx,ny)              !matrix of stp longitudes

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     netCDF auxiliaries
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      parameter (ll=4)

      integer ncid                 !ID number return from ncopn
      integer rcode                !returned error code, 0 = ok
      integer prid                 !precipitation ID from ncvid
      integer laid                 !latitude ID from ncvid
      integer loid                 !longitude ID from ncvid
      integer prst(2)              !index of first value to be
                                   !accessed on each dimension of pri
      integer prco(2)              !index of last value to be
                                   !accessed on each dimension of pri
      integer last                 !index of first value to be
                                   !accessed on vector rla
      integer laco                 !index of last value to be
                                   !accessed on vector rla
      integer lost                 !index of first value to be
                                   !accessed on vector rlo
      integer loco                 !index of last value to be
                                   !accessed on vector rlo
      integer*2 pri(nx,ny)         !precipitation array (1/100 mm)
      real rla(ll)                 !reference latitudes of 4 corners:
                                   !SW,SE,NE,NW
      real rlo(ll)                 !reference longitudes of 4 corners:
                                   !SW,SE<NE,NW
      data prst /1,1/
      data last /1/
      data lost /1/
      data laco /ll/
      data loco /ll/

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     auxiliary variables
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      integer i
      integer j
      real aux1
      real aux2
      real aux3
      real aux4

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     polar stereographic parameters
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      real stla                    !standard latitude (deg)
      real stlo                    !standard longitude (deg)
      real erad                    !radius of the earth (km)
```

```
      real pi                     !trigonometric constant
      real rx                     !reference distance on x-direction
                                  !on image plane (km)
      real ry                     !reference distance on y-direction
                                  !on image plane
      real dxy                    !resolution of grid on image plane (km)

      parameter (stla=60.)
      parameter (stlo=105.)
      parameter (erad=6371.2)
      parameter (pi=3.14159)
      parameter (dxy=4.7625)

      character*15 filename       !name of netCDF file
      character*2 cod(100)

      data cod/'01','02','03','04','05','06','07','08','09','10',
     #         '11','12','13','14','15','16','17','18','19','20',
     #         '21','22','23','24','25','26','27','28','29','30',
     #         '31','32','33','34','35','36','37','38','39','40',
     #         '41','42','43','44','45','46','47','48','49','50',
     #         '51','52','53','54','55','56','57','58','59','60',
     #         '61','62','63','64','65','66','67','68','69','70',
     #         '71','72','73','74','75','76','77','78','79','80',
     #         '81','82','83','84','85','86','87','88','89','90',
     #         '91','92','93','94','95','96','97','98','99','00'/
       prco(1)=nx
      prco(2)=ny

      filename(1:2) = cod(mm)
      filename(3:4) = cod(dd)
      filename(5:6) = cod(yy)
      if(hh.eq.0) then
                      filename(7:8) = cod(100)
                  else
                      filename(7:8) = cod(hh)
      endif
      filename(9:12)='z.nc'

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     open netCDF file with function ncopn
c     ncnowrit = ready-only mode
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      ncid = ncopn(filename,ncnowrit,rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract id of precipitation array with netCDF function ncvid
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      prid = ncvid(ncid,'amountofprecip',rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract id of reference latitude vector with netCDF function ncvid
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      laid = ncvid(ncid,'lat',rcode)
```

```
c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract id of reference longitude vector with netCDF function ncvid
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      loid = ncvid(ncid,'lon',rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract reference latitude vector with netCDF subroutine ncvgt
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call ncvgt(ncid,laid,last,laco,rla,rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract reference longitude vector with netCDF subroutine ncvgt
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call ncvgt(ncid,loid,lost,loco,rlo,rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c          sometimes, for some reason, the archives come without the
c                    reference latitudes and longitudes
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      if(int(rla(1)).ne.int(33.603)) then
                  rla(1) = 33.60300064
                  rla(2) = 32.43299866
                  rlo(1) = 106.4560013
                  rlo(2) = 92.32199880
      print *,'warning : archive ',filename,' does not have references'
      print *,'              using internal values'
      endif

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     extract array of precipitation with netCDF subroutine ncvgt
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call ncvgt(ncid,prid,prst,prco,pri,rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     close netCDF file
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      call ncclos(ncid,rcode)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     convert precipitation data into mm
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      do i=1,nx
         do j=1,ny
```

```
            stp(i,j)=float(pri(i,j))/100.
         enddo
      enddo

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     generate matrix of latitudes and longitudes based on HRAP SW corner
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      aux1 = pi/180.                          !conversion from deg to rad
      aux2 = erad*((1.+sin(stla*aux1))/(1.+ sin(rla(1)*aux1)))
      aux2 = aux2*cos(rla(1)*aux1)
      aux3 = (stlo-rlo(1))*aux1
      rx = aux2*sin(aux3)
      ry = aux2*cos(aux3)

      do i=1,nx
         aux2 = rx + dxy*i
         do j=1,ny

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     calculate the latitude of the grid point
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

            aux3 = ry -dxy*j
            aux4 = sqrt(aux2*aux2 + aux3*aux3)
            aux4 = aux4/(erad*(1+sin(stla*aux1)))
            aux4 = 2.*atan (aux4)
            aux4 = pi/2. -  aux4
            sla(i,j) = aux4/aux1

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     calculate the longitude of the grid point
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

            aux4 = atan(aux2/aux3)
            slo(i,j) = stlo - aux4/aux1
         enddo
      enddo

      return
      end



c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c       subroutine dtt writes the results of the analysis for Lake Altus
c     area into three separate files. The filename is mmddyyhh.ext, where:
c
c                                   mm  = month
c                                   dd  = day
c                                   yy  = year
c                                   hh  = hour
c                                   ext = stp (stage III only)
```

```
c                                    soa (stage III + mesonet)
c                                    sam (mesonet only)
c
c                      written by Augusto Pereira 7-2-96
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      subroutine dtt(mm,dd,yy,hh,stp,sop,sbp,sla,slo,nx,ny,ib,if,jb,jf)

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     declaration of variables
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      implicit none                 !explicit declarations only
      integer nx                    !number of grid points on x-direction
      integer ny                    !number of grid points on y-direction
      integer ib                    !index of starting grid point on x-dir
      integer if                    !index of ending grid point on x-dir
      integer jb                    !index of starting grid point on y-dir
      integer jf                    !index of ending grid point on y-dir
      integer mm                    !month
      integer dd                    !day
      integer yy                    !year
      integer hh                    !hour
      real stp(nx,ny)               !stage III precipitation matrix
      real sop(nx,ny)               !SOA precipitation analysis matrix
      real sbp(nx,ny)               !SOA (+ Barnes) precipitation analysis
                                    !for mesonet only
      real sla(nx,ny)               !latitude matrix of HRAP grid
      real slo(nx,ny)               !longitude matrix of HRAP grid
      character*15 filename         !name of file
      character*2  cod(100)         !encoding vector

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c     auxiliaries
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      integer i
      integer j

      data cod/'01','02','03','04','05','06','07','08','09','10',
     #         '11','12','13','14','15','16','17','18','19','20',
     #         '21','22','23','24','25','26','27','28','29','30',
     #         '31','32','33','34','35','36','37','38','39','40',
     #         '41','42','43','44','45','46','47','48','49','50',
     #         '51','52','53','54','55','56','57','58','59','60',
     #         '61','62','63','64','65','66','67','68','69','70',
     #         '71','72','73','74','75','76','77','78','79','80',
     #         '81','82','83','84','85','86','87','88','89','90',
     #         '91','92','93','94','95','96','97','98','99','00'/

c     ))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                   write latitudes and longitudes to file
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      open(3,file='latlonarea.altus',status='unknown',form='formatted')
      write(3,100) jb,jf,ib,if       !hrap coordinates
      do j=jf,jb,-1
```

```
         write(3,200) (sla(i,j),i=ib,if) !latitudes first
      enddo
      do j=jf,jb,-1
         write(3,200) (slo(i,j),i=ib,if) !longitudes second
      enddo
      close(3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c           write stage III precipitation data (Lake Altus area) (in)
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      filename(1:2) = cod(mm)
      filename(3:4) = cod(dd)
      filename(5:6) = cod(yy)
      if(hh.eq.0) then
                     filename(7:8) = cod(100)
                  else
                     filename(7:8) = cod(hh)
      endif
      filename(9:12)= '.stp'

      open(3,file=filename,status='unknown',form='formatted')
      write(3,100) jb,jf,ib,if        !hrap coordinates
      do j=jf,jb,-1
         write(3,300) ((stp(i,j)/25.4),i=ib,if)
      enddo
      close(3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c           write SOA precipitation data (Lake Altus area) (in)
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      filename(9:12)= '.soa'

      open(3,file=filename,status='unknown',form='formatted')
      write(3,100) jb,jf,ib,if        !hrap coordinates
      do j=jf,jb,-1
         do i=ib,if
            if(sop(i,j).ge.0) then
                                 sop(i,j)=sop(i,j)/25.4  !convert to inches
                              else
                                 sop(i,j)=0              !noise
            endif
         enddo
         write(3,300) (sop(i,j),i=ib,if)
      enddo
      close(3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c           write SOA for mesonet precipitation data (Lake Altus area) (in)
c
c     (((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((

      filename(9:12)= '.sam'

      open(3,file=filename,status='unknown',form='formatted')
      write(3,100) jb,jf,ib,if        !hrap coordinates
      do j=jf,jb,-1
         do i=ib,if
```

```
            if(sbp(i,j).ge.0) then
                                sbp(i,j)=sbp(i,j)/25.4  !convert to inches
                            else
                                sbp(i,j)=0              !noise
            endif
         enddo
         write(3,300) (sbp(i,j),i=ib,if)
      enddo
      close(3)

c     )))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))
c
c                                   formats
c
c     ((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((
100   format(4(1x,i3))
200   format(1x,100f8.4)
300   format(1x,100f4.2)

      return
      end
```

## Mission

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American Public.