REPRESENTATION OF THE MESOSCALE WIND FIELD USING A LINE INTEGRAL TECHNIQUE

LP-175

TDWR CONTRACT NO. 14-00030

Prepared by:

DEPARTMENT OF METEOROLOGY
COLLEGE OF GEOSCIENCES
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS

Prepared for:

TEXAS DEPARTMENT OF WATER RESOURCES
AUSTIN, TEXAS

Funded by:

DEPARTMENT OF THE INTERIOR, WATER AND POWER RESOURCES SERVICE
TEXAS DEPARTMENT OF WATER RESOURCES

June 1982

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO.	2.	3. Recipient's Accession No.

4. Title and Subtitle
Representation of the mesoscale wind field using a line integral technique

5. Report Date
May 1982

6.

7. Author(s)
John S. Trares, Jr. and Phanindramohan Das

8. Performing Organization Rept. No.

9. Performing Organization Name and Address
Department of Meteorology
Texas A&M University
College Station, Texas 77843

10. Project/Task/Work Unit No.
5540

11. Contract(C) or Grant(G) No.
(C) TDWR No. 14-00030
(G) 14-06-D-7587

12. Sponsoring Organization Name and Address
Texas Department of Water Resources
P.O. Box 13087, Capitol Station
Austin, Texas 78711

13. Type of Report & Period Covered
Technical Report

14.

15. Supplementary Notes Prepared in cooperation with the Bureau of Reclamation, Division of Atmospheric Resources Research, Denver Federal Center, Denver Colorado 80225

16. Abstract (Limit: 200 words)

An interpolated representation of a measured wind field in a mesoscale network, with the constraint that divergence and vorticity fields remain essentially unaffected, has been developed. Point values of divergence and vorticity are approximated from line integrals around triangles. Wind observations define vertices of the triangles. A weighted averaging interpolation scheme, complete with isotropic exponential weighting, is utilized to obtain grid point estimates of divergence and vorticity. The gridded kinematic fields, obtained from line integrals, are relaxed for the purpose of constructing stream function and velocity potential distributions. The non-divergent ($\vec{V}_\psi$) and irrotational ($\vec{V}_\chi$) components of the horizontal wind vector, obtained from the gradients of stream functions and velocity potentials, respectively, are superposed to obtain the total analyzed wind ($\vec{V}_{\psi+\chi}$). A test was conducted on this rendition of a measured wind field using observed data.

17. Document Analysis a. Descriptors

Interpolation	Boundary Conditions	Initial Conditions
Vorticity	Divergence	Gridding of Data
Observed Wind	Line Integral Technique	Finite Difference Method

b. Identifiers/Open-Ended Terms

Mesoscale, Texas HIPLEX, High Plains Cooperative Program, HIPLEX

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
	20. Security Class (This Page)	22. Price

ACKNOWLEDGEMENTS

This research was partially supported by the Bureau of Reclamation and the State of Texas through the Texas Department of Water Resources under TDWR Contract No. 14-00030, Dr. James R. Scoggins, Principal Investigator. The support was very much appreciated. Partial support also was provided by the Department of Meteorology, Texas A&M University.

TABLE OF CONTENTS

Page

ABSTRACT i

ACKNOWLEDGEMENTS ii

LIST OF FIGURES v

1. INTRODUCTION 1

a. Nonuniqueness of Wind-Field Interpolation 3
b. Noise Generated in an Interpolation Scheme 4
c. "Roughing" Due to Finite Differences 5

2. THEORETICAL AND ANALYTICAL BACKGROUND 8

a. Determination of Vorticity and Divergence 9
b. Construction of ψ and χ Fields: Boundary Conditions 11

3. THE DATA 13

a. S-coordinates 13
b. Balloon Drift 13

4. ANALYTICAL AND COMPUTATIONAL PROCEDURE 16

a. Objective Analysis 16
b. Smoothing 17
c. Computation of Vorticity and Divergence 19
d. Construction of the Fields of ψ and χ 22
e. Test of Validity of the Procedure for Wind Field Reconstruction: Computation of Root Mean Square Vector Error 24

5. RESULTS AND DISCUSSION 25

a. Gridded Components of Observed Winds 25
b. Divergence and Vorticity by the Finite Difference Method and the Resulting Wind Field 26
c. Divergence and Vorticity by the Line Integral Technique and the Resulting Wind Field 32
d. Evaluation of Results 49

Page

6. SUMMARY 51

a. Comparison between the LIT and Finite Difference Method Computations of Divergence and Vorticity. 51
b. Comparison between the Reconstructed Wind Fields via the LIT and Finite Difference Method. 52
c. Future Research 52
d. Final Remarks 55

REFERENCES. 58

APPENDIX A. 60

APPENDIX B. 65

LIST OF FIGURES

Figure		Page
1	Gridded surface height analysis of the Texas HIPLEX region, with indicated radiosonde station locations	14
2	Isotach analysis of the original gridded wind field over the HIPLEX region	27
3	Wind direction analysis of the original gridded wind field over the HIPLEX region.	28
4	Differential evaluation of divergence, by centered differences, for the HIPLEX region.	29
5	Differential evaluation of vorticity, by centered differences, for the HIPLEX region.	31
6	Isotach analysis of the reconstructed wind field, via finite differences, for a subregion within HIPLEX	33
7	Wind direction analysis of the reconstructed wind field, via finite differences, for a subregion within HIPLEX . . .	34
8	Location of triangle centroids.	36
9	LIT evaluation of divergence for the HIPLEX region.	37
10	LIT evaluation of vorticity for the HIPLEX region	38
11	Isotach analysis of the reconstructed wind field, via LIT, for a subregion within HIPLEX	39
12	Wind direction analysis of the reconstructed wind field, via LIT, for a subregion within HIPLEX.	40
13	Isotach analysis for a subregion within HIPLEX, with twenty-two triangles retained for interpolation	41
14	Location of triangle centroids. Seventeen triangles have been retained for interpolation	43
15	Seventeen point values of divergence interpolated onto the grid.	45
16	Seventeen point values of vorticity interpolated onto the grid.	46

Figure Page

17 Isotach analysis of the reconstructed wind field, realized from one execution of the wind retrieval scheme. . 47

18 Wind direction analysis of the reconstructed wind field, realized from one execution of the wind retrieval scheme. . 48

1. INTRODUCTION

A mesoscale primitive equation model was developed by Djuric and Das (Scoggins et al., 1981) for the TAMU Texas HIPLEX studies for 1979. The model was initialized using the original gridded fields of temperature, dew point, u and v components of the wind, and heights. After a thirty-minute run of the model, it was found that noise, in the form of spurious waves, began to obscure the results. It was speculated that this noise was the result of defects in the initial data, presumably the winds. The present study was taken up with a view of developing a procedure for the initialization of the wind field, which, while retaining the mesoscale properties of the wind field, will be less prone to causing the development of noise in the numerical model.

V. Bjerknes (1904) is recognized as the first to suggest that, given observed initial fields of mass and velocity, it would be possible, at least in principle, to determine the mass and velocity distribution at any future time by solving the hydrodynamical equations as an initial value problem. A new direction in meteorology, namely numerical weather prediction, was founded upon this suggestion.

To carry out a numerical forecast, it is necessary to specify the wind field (and fields of other variables) at the beginning of the forecast period. These fields define the initial state from which the system of differential equations is integrated forward in time. A

common procedure for specifying the wind field is through the interpolation of observed winds onto a uniformly spaced mesh by means of some weighted averaging scheme (Bergthórsson and Döös, 1955). This type of analysis of the wind field is quite satisfactory for numerical prognosis with the vorticity models (Haltiner and Williams, 1980). However, integration of the momentum equations in their primitive form is much more sensitive to the initial state. If the wind field prepared by the weighted averaging scheme is inserted directly into a primitive equation model, considerable noise is generated while the mass and motion fields adjust to one another. This "shock" of data insertion is eventually reduced to an acceptable level as adjustment between the two fields occurs over time.

It would obviously be advantageous if this adjustment could be effected prior to the beginning of the forecast. Talagrand (1972) directly filters noise contained in the initial wind field through the addition of a divergence damping term in the momentum equation. Shuman and Hovermale (1968) initialize their primitive equation model by inserting analyzed geopotential fields into the balance equation obtaining the stream function ψ from which the rotational wind component may be realized. Sasaki (1958) developed a method founded on the calculus of variations in which differences between the objectively analyzed values and the newly adjusted values were minimized in a least squares approximation, subject to dynamical constraints. These constraints may include the balance equation, hydrostatic relation, steady-state momentum equations, integral relations conserving energy

and mass, and others.

The problem of initializing a mesoscale primitive equation model becomes especially acute since such a model aims at making short-range prediction of phenomena on the scale of organized convection. The organizing influence, on this scale, arises from mesoscale convergence. The convergence (divergence) field is usually obtained from the gradients of a "gridded" wind field. Grid point values of wind are determined from the interpolation of observed (measured) winds onto the mesh by means of a weighted averaging scheme. However, as discussed below, there are serious flaws in this approach.

a. Nonuniqueness of Wind-Field Interpolation

A fundamental problem in numerical analysis is the nonuniqueness of vector field interpolation (Schaefer and Doswell, 1979). This arises because a vector, such as wind, is a directed quantity. Interpolation of its magnitude and direction does not yield results identical to those from interpolated components. For example, consider two wind observations at the end points of an east-west line, which consist of a westerly at 10.0 m s^{-1} at one point and a southerly at 20.0 m s^{-1} at the other. Halfway between the two observation points, the linearly interpolated directions and speeds will give an interpolated southwesterly wind at 15.0 m s^{-1}, while a linear interpolation of components will give a south-southwesterly wind at 11.2 m s^{-1}. The differences may not be as drastic in day-to-day situations; in a comparison of data interpolated via computer to that

interpolated manually, Williams (1976) found that wind speed estimation differed by an average of 3.5 m s^{-1}, while wind directions were roughly equivalent. But a potentially serious problem remains in a simple-minded interpolation of winds for a mesoscale primitive equation model.

b. Noise Generated in an Interpolation Scheme

Mesoscale prediction suffers from the basic difficulty of being supplied with inaccurate, initial grid point values of some atmospheric variable. These grid point values are frequently generated by a weighted averaging interpolation scheme. The basic methodology of this scheme is as follows: an initial guess of the atmospheric variable (such as climatological mean, previously analyzed values of the variable, or a combination of these two approaches) is supplied to the grid points; the initial guess field is modified through the observations (weighted corrections to the initial guess field which are functions of the distance from the observation to the grid point); the corrected grid point values are "lightly" smoothed to remove obvious meteorological noise.

Barnes (1964) has proven that a weighted averaging interpolation scheme is convergent (in other words, the approximated spatial distribution of some atmospheric variable as determined from a weighted averaging scheme approaches the "real" spatial distribution of that variable). This is true provided the independent harmonic waves (which represent the spatial distribution of some atmospheric variable

(Sasaki, 1960)) have wavelengths greater than two mesh lengths. Unfortunately, all wavelengths are present in the approximated spatial distribution of the atmospheric variable. Judicious smoothing is not considered feasible since both short (meteorological noise) and long (meteorologically significant) waves would be affected.

Another difficulty inherent in a weighted averaging interpolation scheme is the insufficient number of observations to adequately "correct" the initial guess field. Because of this problem, inertia-gravity wave modes are poorly (erroneously) represented on the grid. In the case of a gridded wind field, these gravity waves may generate large and unrealistic divergence patterns (Warner et al., 1978).

c. "Roughing" Due to Finite Differences

Given a gridded data field, its further manipulation often involves the determination of derivatives by finite-difference methods. For example, the mesoscale initialization (described by Warner et al., 1978) involves the computation of vorticity from which the stream function and geopotential fields are derived. If the vorticities are determined by the method of finite differences from the gridded fields of wind components, they will be contaminated both by the nonuniqueness of vector field interpolation and by the uncertainties of the interpolation scheme. The inherent inaccuracies of the finite-difference method will exacerbate the above uncertainties (Conte and de Boor, 1980).

The foregoing discussion points to the need for a strategy for

the initialization of the wind field, in which the following should be achieved:

1) The interpolation of a vector field should be avoided.

2) The needed interpolation of scalar fields should maximize details while suppressing unresolvable short-wave noise.

3) A minimum number of finite-difference operations be performed.

Fortunately, recent research has devoted considerable attention to the elements needed for developing such a strategy. The two relevant studies, although aimed at synoptic-scale analysis, are those of Shukla and Saha (1974) and Schaefer and Doswell (1979). In principle, the procedure suggested is as follows: in a network of wind observations, the vorticity (ξ) and divergence (D) are determined from a line integral method such as that of Bellamy (1949) or Ceselski and Sapp (1975); the scalar fields of ξ and D are utilized to determine the fields of stream function and velocity potential which, in turn, will provide the grid point values of the rotational ($\vec{V}_\psi$) and irrotational ($\vec{V}_\chi$) parts of the wind; and, the final analyzed wind field is obtained by superposing the irrotational and nondivergent parts ($\vec{V}_{\psi+\chi}$).

The purpose of the present study is to ascertain (test) the "goodness" of the above strategy for determining the initial wind field by means of a comparison with other accepted strategies. The above strategy constitutes a wind field that is "reconstructed" from fields of divergence and vorticity calculated by line integrals. The

other accepted strategies for determining the initial wind field are: (i) a direct wind-component interpolation, and (ii) a wind field that is "reconstructed" from fields of divergence and vorticity calculated by centered differences. From this test, the most optimal strategy for determining the initial wind field will be ascertained for use in a mesoscale primitive equation model.

2. THEORETICAL AND ANALYTICAL BACKGROUND

The analytical strategy suggested by Shukla and Saha (1974) is based on a well-known theorem of Helmholtz according to which a vector $\vec{V}$, such as the horizontal wind, can be separated into a rotational ($\vec{V}_\psi$) and an irrotational ($\vec{V}_\chi$) part as follows:

$$\vec{V} = \vec{V}_\psi + \vec{V}_\chi = \hat{k} \times \nabla_h \psi + \nabla_h \chi \quad , \tag{1}$$

where $\hat{k}$ is a unit vertical vector, ψ is the horizontal stream function, ∇_h is the horizontal gradient operator, and χ is the horizontal velocity potential. The vertical component of the curl of the above equation gives

$$\hat{k} \cdot \nabla_h \times \vec{V} = \nabla_h^2 \psi = \xi \quad , \tag{2}$$

where ξ is the vertical component of the vorticity. The horizontal divergence of (1) gives

$$\nabla_h \cdot \vec{V} = \nabla_h^2 \chi = D \quad , \tag{3}$$

where D is divergence. Now, if one is given the ξ and D fields, corresponding ψ and χ fields can be constructed by solving (2) and (3) over a grid. The values of ψ and χ at the grid points can then be used to construct the wind field according to (1). The advantage of this reconstruction method will be to produce a wind field which conserves the kinematic properties, ξ and D, which are fundamental to the

original wind field. However, the advantage may or may not be meaningful depending on how we construct the fields of ξ and D.

a. Determination of Vorticity and Divergence

Given a set of wind observations, it would be quite straightforward to interpolate the individual wind components, u and v, in the customary x- and y-directions to obtain their gridded fields. These fields permit the computation of ξ and D from their differential definitions:

$$\xi = \partial v/\partial x - \partial u/\partial y \quad , \tag{4}$$

$$D = \partial u/\partial x + \partial v/\partial y \quad . \tag{5}$$

As already pointed out, however, this method involves two undesirable operations, namely, (i) interpolation of a vector field, and (ii) performance of a finite-difference operation.

In order to avoid the problems mentioned in the last paragraph, an integral method, originally due to Bellamy (1949), can be used to determine values of ξ and D directly from the wind observations. In Bellamy's method, a triangle is selected for the curve with vertices at the wind observation points. The winds are then allowed to displace the vertices for some time interval. Divergence is equal to the relative increase in area, enclosed within the triangle, per unit time. Vorticity is obtained by repeating the operation with the winds turned 90°.

Bellamy's method is a special case of what Ceselski and Sapp

(1975) call the "Line Integral Technique" to be referred to as LIT in the sequel. The principle of this technique can be explained as follows: a point value of divergence (Gauss' Theorem) is represented as

$$D \equiv \lim_{A \to o} (1/A) \oint \vec{V} \cdot \hat{n} \, ds \quad , \tag{6}$$

while, from Stokes' Theorem, a point value of vorticity becomes

$$\xi \equiv \lim_{A \to o} (1/A) \oint \vec{V} \cdot \hat{s} \, ds \quad , \tag{7}$$

where A is the area of some surface that lies in the horizontal plane, $\hat{s}$ and $\hat{n}$ are unit vectors tangent and normal to the path of integration, ds is the differential increment along the path, and $\oint$ constrains the integration to proceed in a counterclockwise sense along a path marking the periphery of A.

Schaefer and Doswell (1979) elucidated the dissimilarity between the methods of Bellamy and of Ceselski and Sapp. In the Bellamy method, the total change in area of the triangle (with the inclusion of the vertices) is realized in the determination of divergence. The method of Ceselski and Sapp also realizes the change in area, but excludes the expansion of the region around the vertices. Consequently, the dissimilarity between the two methods is directly proportional to the time increment chosen in implementing the Bellamy method.

It appears that the evaluation (determination) of kinematic variables through the above integral methods is superior to the

evaluation through differentiation. An initial interpolation of winds is not required for the implementation of integral definitions. Consequently, the divergence and vorticity fields obtained by integral methods appear to be most consistent with the original wind observations.

Finally, the "roughing" inherent in numerical differentiation is replaced by the "smoothing" of integration (Schaefer and Doswell, 1979).

b. Construction of ψ and χ Fields: Boundary Conditions

Once the values of vorticity and divergence are determined at every grid point, either through the combinations of derivatives of interpolated wind components or through the interpolation of point values of the above kinematic variables calculated from line integrals, Poisson-type equations (2) and (3) need to be solved obtaining values of ψ and χ, respectively. Unfortunately, in meteorological problems, no information about ψ or χ is available at the boundaries of the computational domain. Any attempt to compute ψ and χ corresponds to various degrees of approximation that must satisfy

$$\hat{n} \cdot \vec{V} = V_n = - \partial\psi/\partial s + \partial\chi/\partial n \tag{8}$$

$$\hat{s} \cdot \vec{V} = V_s = + \partial\psi/\partial n + \partial\chi/\partial s \tag{9}$$

at the boundary. Here n is distance on the earth normal to the

boundary increasing outward, and s is distance on the earth along the boundary positive in the counterclockwise sense.

Sangster (1960) gave a detailed discussion of the means of specifying the boundary values of ψ, and brought out the necessity of considering both ψ and χ so as to satisfy (8) and (9) at the boundary. The method adopted in this work, which is an extension of Sangster's basic discussion, will be presented in the sequel.

3. THE DATA

The data utilized in this study originate from a mesoscale network located in western Texas (Texas HIPLEX region), where measurements were taken at seven radiosonde stations at 0300 GMT 28 May 1979. Variables available include pressure, height, u and v components of the wind, as well as range and azimuth of the balloon.

a. S-coordinates

The analysis is performed on a coordinate system in which the vertical coordinate represents a given fraction of the local depth of the model atmosphere from the earth's surface to the top of the model. This will be referred to as the S-coordinate where S is defined as

$$S = \left(z_S(j) - z_o(j)\right) / \left(z_t - z_o(j)\right) \quad . \tag{10}$$

Here $z_o(j)$ and $z_S(j)$, respectively, are the heights of the surface and of the level denoted by a constant value of S (both situated at a given radiosonde location j), and z_t is the height of the top of the model atmosphere. The gridded surface height (which is an approximation for a topographic map of this region) is given in Fig. 1.

b. Balloon Drift

For synoptic scale observations, an upper air wind measurement is assumed valid directly over the station. The net horizontal displacement of the balloon, realized from the prevailing flow, is small

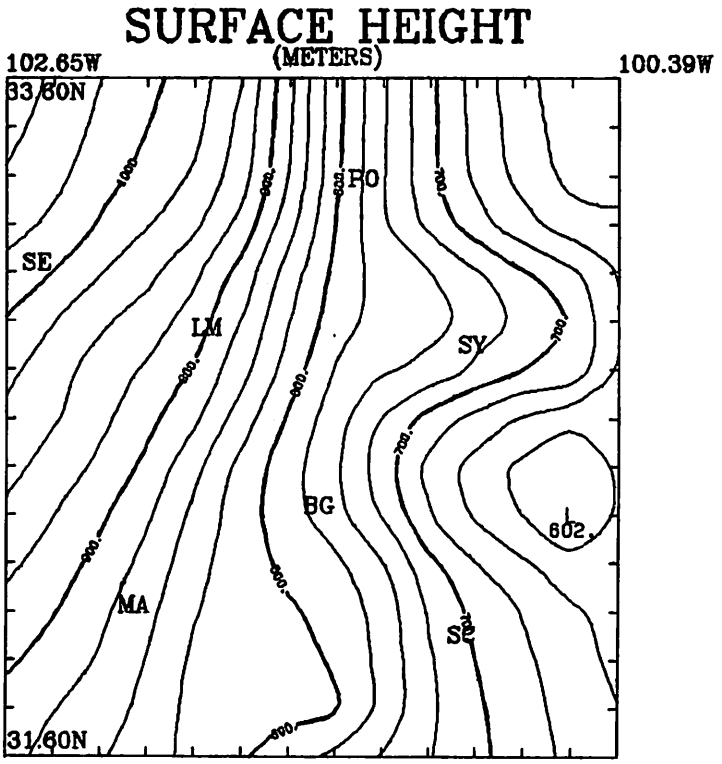

Fig. 1. Gridded surface height analysis of the Texas HIPLEX region, with indicated radiosonde station locations:

MA	Midland	PO	Post
SE	Seagraves	LM	Lamesa
BG	Big Spring	SY	Snyder
SC	Sterling City.		

The positions of these stations correspond to their geographical location, rather than their adjusted position considering balloon drift (next section). Isopleth interval is 25.0 m.

compared to station separation. For a mesoscale network, however, net horizontal displacement of the balloon may be of the same order of magnitude as station separation. A balloon drift correction, for a mesoscale network, insures that the datum is "gridded up" from its true location (realized from the prevailing flow) rather than from the launch site. Therefore, a more representative field is provided for the analysis of meteorological variables at all levels in the predictive model. It should be emphasized that displacement data must be interpolated from the pressure heights to the appropriate S level heights. In other words, the height of the S level pertaining to a given "moveable" observation must be recalculated when the balloon moves into a region where the terrain height is different from that at the launch site.

The computational procedure is based on the assumption that all balloons arrive simultaneously at a fixed altitude, while in fact, balloons released from neighboring stations, generally, reach a given altitude at different times. It is believed that this time difference, the order of several minutes, is negligible when compared to the time scale of changes in the mesoscale environment. Fankhauser (1969) has made analyses of mesonetwork data in cases where departures from scheduled release times were important; accordingly, his scheme provided for adjustment.

The computer implementation of the balloon drift calculation procedure is detailed in Appendix A.

4. ANALYTICAL AND COMPUTATIONAL PROCEDURE

a. Objective Analysis

A recursive weighted averaging interpolation scheme (Cressman, 1959) was utilized to obtain grid point estimates of meteorological variables. This particular effort at objective analysis involves the determination of these variables at grid points as the sum of the weighted values of the individual data f(j). In the Cressman approach, variables are interpolated onto a uniformly spaced mesh by the application of weight factors, using successively smaller influence radii with each iteration (or "pass" through the field). These weight factors are expressed in terms of grid length; therefore, the detail of the analysis is limited by one's choice of grid length. This method is suitable for objective analyses of mesoscale regions where mesh lengths are comparatively small.

The generalized smoothed function, employed in this effort, is given by

$$g(x,y) = \sum_{j=1}^{m} \eta \, f(j) \; / \; \sum_{j=1}^{m} \eta \quad , \tag{11}$$

where η is the weight factor, m is the number of observation points, and f(j) is the atmospheric variable measured at each observation j. The influence of each datum is weighted according to its distance from the grid point. The specific weight factor utilized for any given "pass" through the field is

$$n = \exp\left((-4r^2) / R^2\right) \quad , \tag{12}$$

where r is the distance from the individual observation j to the given grid point, R is the influence radius which is expressed in terms of grid distances, and 4 is a reliability coefficient.

In this work, four "passes" were made through the field, decreasing the radius of influence for each successive pass. The radii varied from ten grid distances to one, with mesh length taken as 15.8 km. It should be noted that applications of a recursive weighted averaging scheme require reasonably uniform data distributions to insure stability in the results.

b. Smoothing

With the implementation of a weighted averaging scheme, discontinuities will develop on the grid at a distance from an isolated observation j where the weighting factor n is rapidly approaching zero. This problem is especially conspicuous near the boundary. These discontinuities in the final output give rise to a further depreciation in the stability of the results. This problem is alleviated through the introduction of a smoothing scheme.

In this study, a systematic explicit smoothing is included for the final products. The simple smoother

$$\bar{f}_j = (1-\nu)f_j + (\nu/2)(f_{j+1} + f_{j-1}) \tag{13}$$

can be applied sequentially to the two directions (Shuman, 1957); that

is, smooth in each dimension independently of the other dimension, to give

$$\bar{f}_{ij} = (1-\nu)^2 f_{ij}$$

$$+ \nu(1-\nu)(f_{i-1\ j} + f_{i+1\ j} + f_{i\ j-1} + f_{i\ j+1}) / 2$$

$$+ \nu^2(f_{i-1\ j-1} + f_{i-1\ j+1} + f_{i+1\ j-1} + f_{i+1\ j+1}) / 4$$

$$= R_{ij} f_{ij} \quad , \tag{14}$$

where ν is the smoothing element index and R_{ij} is the response function. If the distribution of an atmospheric variable is represented by a Fourier integral (Sasaki, 1960), then the corresponding response function is obtained by inserting the simple harmonic function (harmonic form of the dependent variable)

$$f(x,y) = A \exp\left(i\ (kx + \ell y)\right) \quad , \tag{15}$$

with k and ℓ as the wavenumbers in the x and y directions, respectively, and A representing wave amplitude, into (14), leaving the response function

$$R(k,\ell) = \left(1 - 2\nu \sin^2(k\Delta x/2)\right)$$

$$*\left(1 - 2\nu \sin^2(\ell\Delta y/2)\right) \quad . \tag{16}$$

It should be emphasized that the above smoothing routine does not affect the wavenumber nor the phase of the original wave; only its

amplitude.

On my grid, the interior points are smoothed using the nine point smoother (14). Only one "pass" is made through the field employing a smoothing element index of $\nu = 0.10$. The corresponding response function is

$$R(k,\ell) = \left(1 - .2 \sin^2(\alpha \Delta h/2)\right) * \left(1 - .2 \sin^2(\alpha \Delta h/2)\right) \quad , \tag{17}$$

provided $\Delta x = \Delta y = \Delta h$ and $k = \ell = \alpha$. The response function illustrates that the smoothing is extremely light.

The border points are smoothed using the smoother illustrated in (13). One pass is made through the boundary values employing a smoothing element index of $\nu = 0.45$. The corresponding response function is

$$R(k,\ell) = 1 - .9 \sin^2(\alpha \Delta h/2) \quad , \tag{18}$$

provided $\Delta x = \Delta y = \Delta h$ and $k = \ell = \alpha$. Considerable smoothing is indicated by this response function.

c. Computation of Vorticity and Divergence

The computation of ξ and D, as already indicated, is done in two different ways for the sake of comparison. One of the methods employs finite differencing ("Finite Difference Method") in the field of gridded u and v components and the other uses the "Line Integral Technique." The details of these methods are as follows:

1) Finite Difference Method

From the interpolated wind components, the grid point values of divergence and vorticity are computed via centered differences. The difference expressions are

$$D(x,y) = \Big(u\,(x+\Delta x) - u\,(x-\Delta x)\Big) \,/\, (2\Delta x) + \Big(v\,(y-\Delta y) - v\,(y+\Delta y)\Big) \,/\, (2\Delta y) \quad , \tag{19}$$

and

$$\xi(x,y) = \Big(v\,(x+\Delta x) - v\,(x-\Delta x)\Big) \,/\, (2\Delta x) - \Big(u\,(y-\Delta y) - u\,(y+\Delta y)\Big) \,/\, (2\Delta y) \quad , \tag{20}$$

where $\Delta x = \Delta y$. Note; the y axis points approximately toward the south over this computational domain.

2) Line Integral Technique

The LIT does not require an initial interpolation of the winds. Instead, the kinematic variables are computed from the randomly spaced wind observations according to (6) and (7). The paths of integration implied in these relations are triangles, each of which is defined by three neighboring wind observations (corrected for balloon drift). Since a minimum of three points is required to define an enclosed area, the triangle produces the smallest scale and least areally-smoothed information.

This technique has the ability of producing large quantities of computed data points. The number of triangles that can be produced,

given n wind observations, is

$$n(n-1)(n-2) / 3! \quad . \tag{21}$$

The purpose of generating data points is to assign the available information contained in the raw data to triangle centroids, which are more densely distributed than the original observations.

The wind observations, at the end of each of the three line segments, are utilized to compute a simple mean wind for that segment. Eqs. (6) and (7) are evaluated from these mean winds situated along each segment of the triangle. The divergence (or vorticity) value is assigned to each triangle centroid. These values are then considered as observations which can be interpolated to a uniform grid.

Both theoretical and practical considerations limit the triangles for which divergence computations can be reliably made. The process of choosing a particular subset of triangles, from the set of all possible triangles, has an influence on the divergence calculation since the computed divergence values are not independent. Each represents an average divergence over its associated triangle, which shares at least one side with adjacent triangles.

The triangles must be assigned some weighting factor that filters areal mean divergence and vorticity values which cannot be assumed to represent point values over the computational grid. The following criteria should serve as a guideline:

1) Uniform triangle sizes should be used.

2) Triangles should be approximately equilateral.

3) Area of triangles should be commensurate with that of the grid used in the final objective analysis.

Schaefer and Doswell (1979) determined an empirical weight, equal to the square of the tangent of the minimum angle divided by the area. This weighting factor follows the above guidelines adequately. It should be noted that this weighting is not used in the interpolation (Ceselski and Sapp, 1975). Its only purpose is to choose those candidate triangles which are closest to being ideal.

The line integrals are evaluated from all possible triangles, with computed values of divergence and vorticity assigned to the centroids. The weight is then applied, filtering the faulty triangles. Kinematic values at the centroids of the successful triangles are considered as observations, and interpolated to a uniform grid by the weighted averaging scheme previously described.

The computer implementation of the Line Integral Technique calculation procedure is detailed in Appendix A.

d. Construction of the Fields of ψ and χ

The procedure for the construction of ψ and χ fields follows from the relevant discussion in Section 2. Given the fields of D and ξ, obtained either through finite differencing or from the Line Integral Technique, (2) and (3) are solved by sequential relaxation subject to the boundary conditions (8) and (9). The values of V_n and V_s appearing in (8) and (9) are read from the original gridded wind field, irrespective of the method used to determine D and ξ.

The details of the procedure to determine ψ and χ are as follows:

1) Obtain χ by solving

$$\nabla^2\chi = D \qquad (\chi = 0.0 \text{ on the boundary}).$$

2) Knowing the χ values, evaluate $\partial\chi/\partial n$ at the boundary points, and trapezoidally integrate the following equation along the boundary

$$\partial\psi/\partial s = - V_n + \partial\chi/\partial n \quad .$$

3) Knowing the values of ψ at the boundary from 2), obtain the field by solving

$$\nabla^2\psi = \xi \quad .$$

4) Evaluate $\partial\psi/\partial n$ at the boundary points, and trapezoidally integrate the following equation along the boundary

$$\partial\chi/\partial s = + V_s - \partial\psi/\partial n \quad .$$

5) Knowing the values of χ at the boundary from 4), obtain the χ field by solving

$$\nabla^2\chi = D \quad .$$

6) Repeat entire procedure three times.

Once the ψ and χ fields are known, reconstruction of the wind field is realized through the following equations:

$$u = - \partial\psi/\partial y + \partial\chi/\partial x \tag{22}$$

$$v = + \partial\psi/\partial x + \partial\chi/\partial y \quad . \tag{23}$$

e. Test of Validity of the Procedure for Wind Field Reconstruction: Computation of Root Mean Square Vector Error

When real data are analyzed, it is not possible to declare one analysis technique to be definitely better than another as true divergence and vorticity fields are unknown. There also is no unique bench mark against which the wind fields, reconstructed by the various methods, can be compared. The bench mark arbitrarily chosen in this study is due to Shukla and Saha (1974) and is the root mean square vector error defined by

$$E = \left\{ \sum_{j=1}^{m} \left[(u_o - u_a)^2 + (v_o - v_a)^2 \right]_j \; / \; m \right\}^{\frac{1}{2}} , \tag{24}$$

where the subscript o stands for observed and a for analyzed; m indicates the number of observation points. Bilinear interpolation is utilized to obtain wind estimates at the observation points from the gridded fields.

5. RESULTS AND DISCUSSION

The results of this study can be discussed in two parts, one of which is just a statement of the innovations implemented. The other is a qualitative analysis of the merit of the innovative procedure.

The items of innovation that should prove useful to future efforts in mesoscale modeling are:

1) Placement of the radiosonde observations on surfaces of constant S as defined in (10).

2) Implementation of a method to account for balloon drift while developing the initial fields of dependent variables in the mesoscale prediction scheme.

3) Systematic construction of wind fields consistent with the observations and the fundamental dynamical constraints.

Of these items, 1) and 2) are just technical advances and are of unquestionable value. The final evaluation of 3) can only emerge from the application and behavior of the wind field in actual mesoscale prediction.

The major effort in this study is the construction of the wind field. This construction has been done in three different ways and the results presented hereafter bring out a comparison between them.

a. Gridded Components of Observed Winds

The development of the fields of u- and v-components proceeded as follows. At first, the radiosonde observations were processed for

balloon drift while being vertically interpolated onto the appropriate S-surface. The resulting observations of u and v were then individually gridded by the weighted averaging scheme described in Section 4. Subsequently the fields of u and v were smoothed through the application of the smoothing operation (14) in the interior, and (13) on the boundary.

Figs. 2 and 3 illustrate the isotach and direction analyses derived from the gridded wind field developed in the above procedure. Of interest in the given plots is the velocity convergence situated along a line from Big Spring to Sterling City extending northeastward. The corresponding velocity divergence is situated southeast of Lamesa. Velocity convergence, located northeast of Seagraves, may be deduced from Fig. 3, with corresponding velocity divergence located west of Post. Pronounced cyclonic curvature is evident within the Lamesa/Snyder/Sterling City triangle. Corresponding anticyclonic curvature is evident with the Midland/Big Spring/Lamesa triangle.

A root mean square vector error, as defined in (24), was performed on the interpolated results. The error was found to be 0.3 m s^{-1}.

b. Divergence and Vorticity by the Finite Difference Method and the Resulting Wind Field

Given the gridded components, D and ζ were computed by finite differences. The divergence field (Fig. 4) shows an area of strong convergence located along and either side of a line from Big Spring to Sterling City. Three distinct closed-off centers are illustrated.

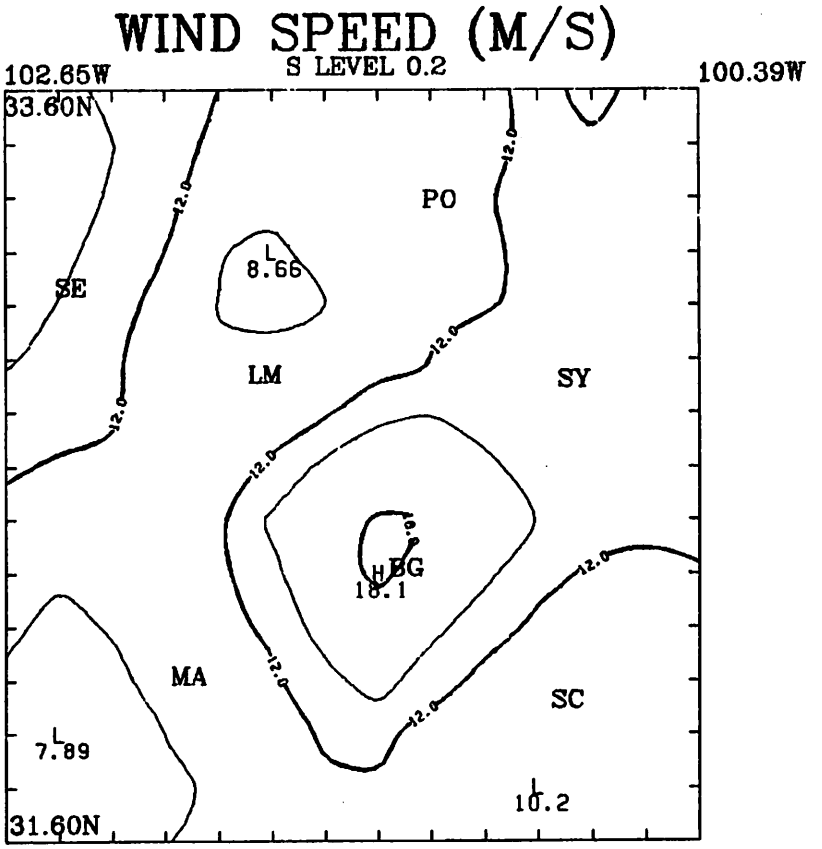

Fig. 2. Isotach analysis of the original gridded wind field over the HIPLEX region. Isotach interval is 3.0 m s^{-1}.

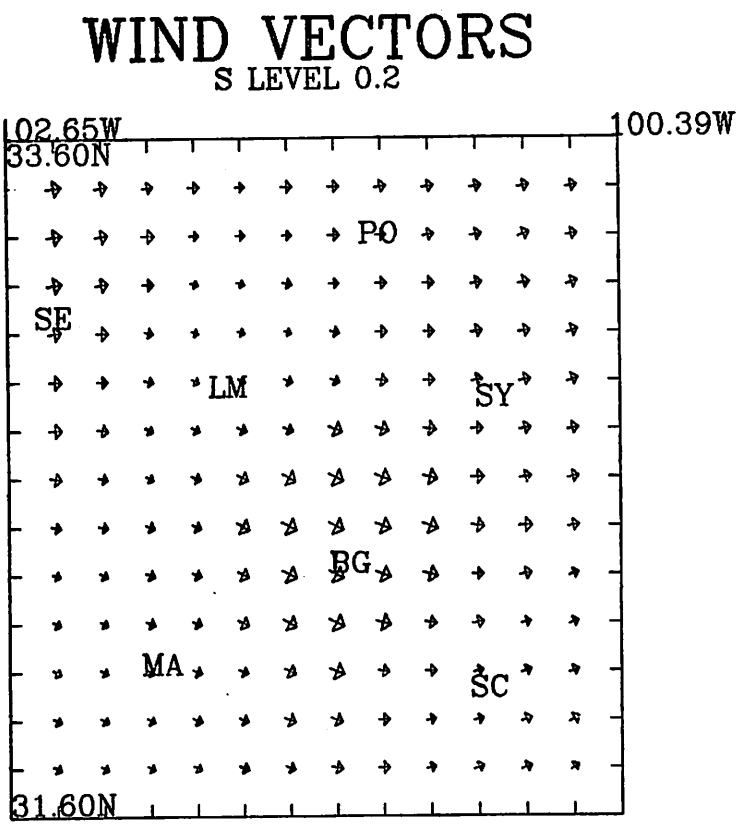

Fig. 3. Wind direction analysis of the original gridded wind field over the HIPLEX region. Size of arrowhead indicates relative magnitude of wind speed.

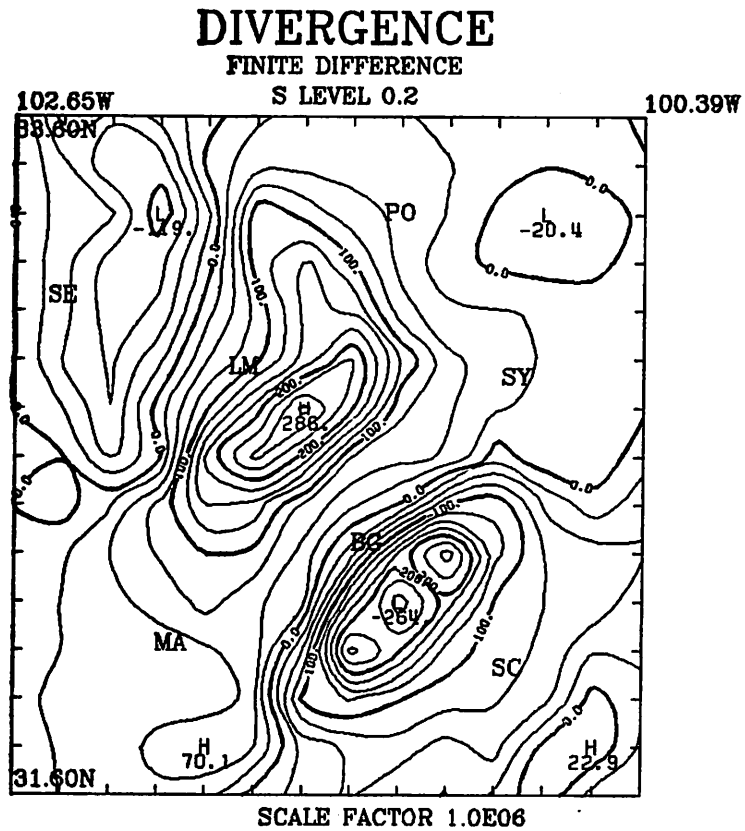

Fig. 4. Differential evaluation of divergence, by centered differences, for the HIPLEX region. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

The corresponding distinct area of strong divergence is indicated southeast of Lamesa. The above findings appear to be heuristically consistent with the original observations. Distinct areas of convergence are shown northeast of Seagraves and east of Post. It is difficult to support this finding in view of the original observations. It is quite possible that this finding is an artifact of the gridding scheme.

The vorticity field (Fig. 5) shows a strong distinct area of positive vorticity in the Lamesa/Snyder/Big Spring triangle. A strong area of negative vorticity (of approximately the same strength) lies northeastward of Midland. An extensive area of negative vorticity is shown, situated southeastward of Seagraves. The above findings appear to be heuristically consistent with the original observations. A distinct area of negative vorticity is shown northeast of Sterling City, while a distinct area of positive vorticity is shown on the southern boundary of the computational domain. The original observations cannot support this finding. Again, this finding may be an artifact of the gridding scheme.

Horizontal divergence and the vertical component of vorticity are fundamental aspects of the horizontal vector wind field. It is clearly desirable to preserve these properties during an analytical representation of the winds. As explained in Section 4, the calculated values of divergence and vorticity are used in solving the Poisson type equations which yield values of χ and ψ, respectively. The wind components are computed through the gradients of velocity potential

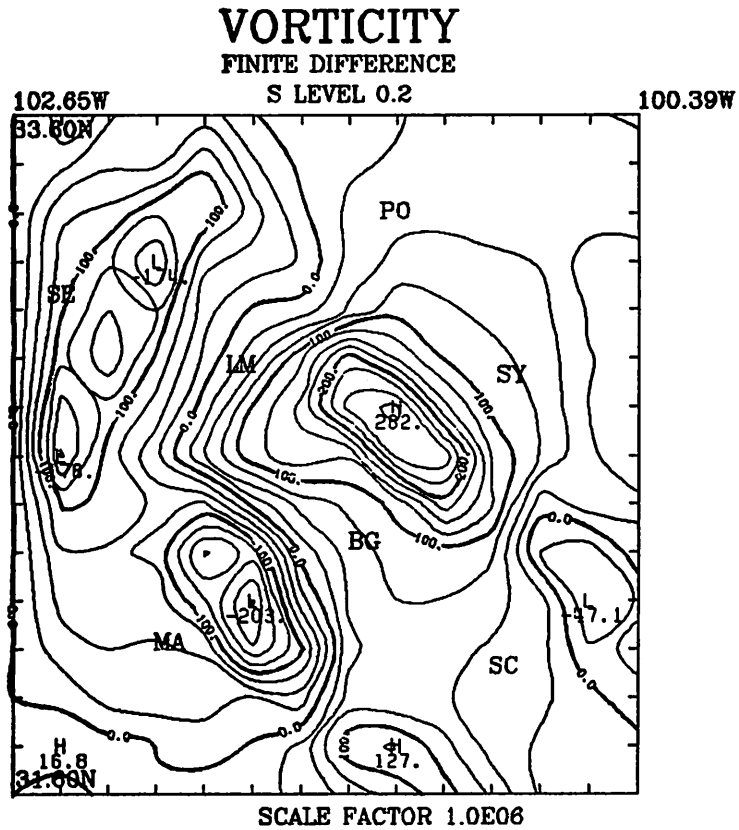

Fig. 5. Differential evaluation of vorticity, by centered differences, for the HIPLEX region. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

and stream function.

A particular subregion of the computational domain is examined in the following presentation. This subregion includes all nine triangle centroids (clustered within the Lamesa/Big Spring/Snyder triangle) and six radiosonde stations (Seagraves' location deleted).

The reconstructed wind field (with D and ξ determined by the Finite Difference Method) is shown in Figs. 6 and 7, and is similar to the original gridded wind field. The root mean square vector error for this retrieved wind field is about 1.3 m s^{-1}. It is obvious that the original gridded wind field (Figs. 2 and 3) and the reconstructed wind field (which preserves the divergence and vorticity aspects of the original gridded wind field) are not identical. The reconstructed wind field, via finite differences, recognizes a wind speed maximum (19.3 m s^{-1}) west of Big Spring. No wind speed as strong as 19.3 m s^{-1} was measured at the given radiosonde stations. However, conservation of the kinematic fields (through finite-difference reconstruction) "forces" the existence of that wind value over this computational domain. Other dissimilarities between the original and reconstructed wind fields may be noted.

c. Divergence and Vorticity by the Line Integral Technique and the Resulting Wind Field

In implementing the LIT, it may be recalled, triangles are formed with the observation points as vertices. Given the seven radiosonde stations, thirty-five triangles are created through the combination of

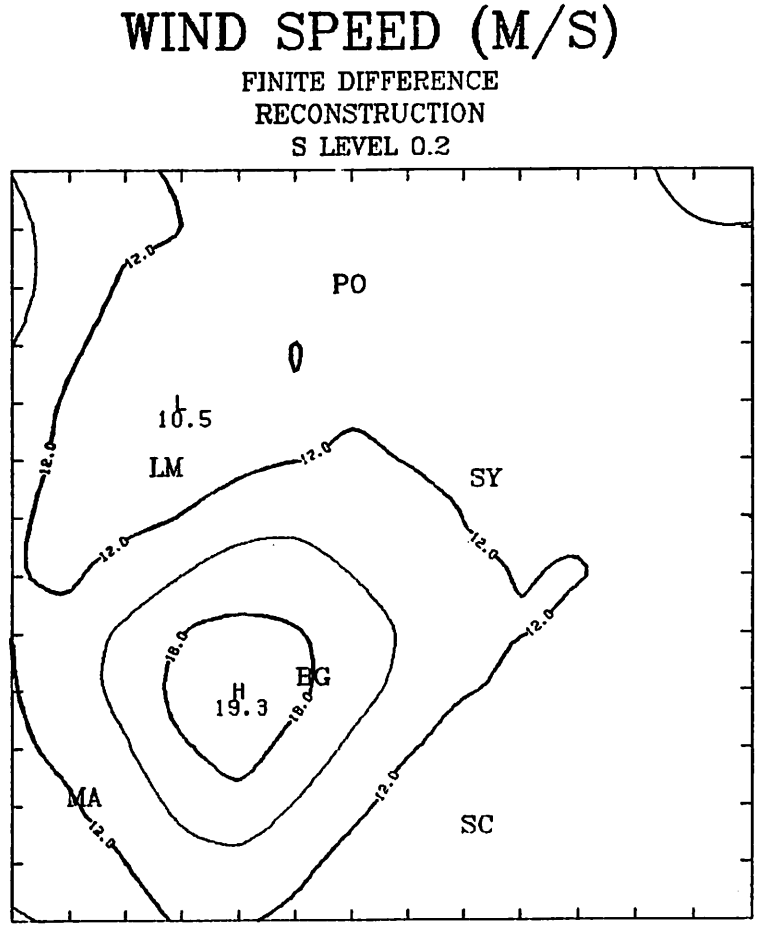

Fig. 6. Isotach analysis of the reconstructed wind field, via finite differences, for a subregion within HIPLEX. Isotach interval is 3.0 m s^{-1}.

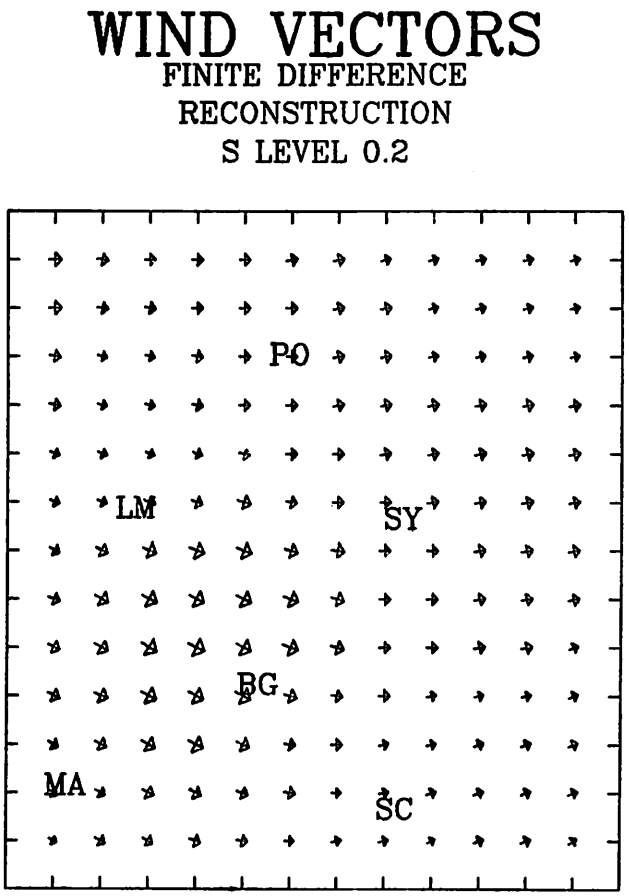

Fig. 7. Wind direction analysis of the reconstructed wind field, via finite differences, for a subregion within HIPLEX.

each observation with its neighbor. Areal mean values of divergence and vorticity are determined through their integral definitions, and assigned to all triangle centroids. Upon implementation of the triangle weighting factor, nine triangles survive the filtering (Fig. 8). These nine triangle centroids, considered as point values of the kinematic variables, are interpolated onto a uniformly spaced mesh by the weighted averaging scheme previously described. The fields of divergence and vorticity are shown in Figs. 9 and 10.

The wind field, reconstructed from the fields of divergence and vorticity determined by the LIT, is given in Figs. 11 and 12. A root mean square vector error computation also was performed, and was found to be 6.9 m s^{-1}. This is much too high. Obviously this application of the Line Integral Technique has produced an unacceptable reconstruction of the wind field.

The large vector errors in the reconstruction of the wind through the LIT-determined vorticity and divergence values naturally cast suspicion on the implementation of the LIT. In developing the results presented above, a very stringent criterion was applied for the selection of suitable triangles from the possible population. Of the thirty-five triangles generated, only nine were retained for subsequent interpolation. Dr. Joseph T. Schaefer (personal communication) suggested a "relaxation of the criterion." According to his advice, the selection criterion was relaxed, and this resulted in a retention of thirteen additional triangles for interpolation. The results, however, proved poor (Fig. 13) when compared to the original gridded

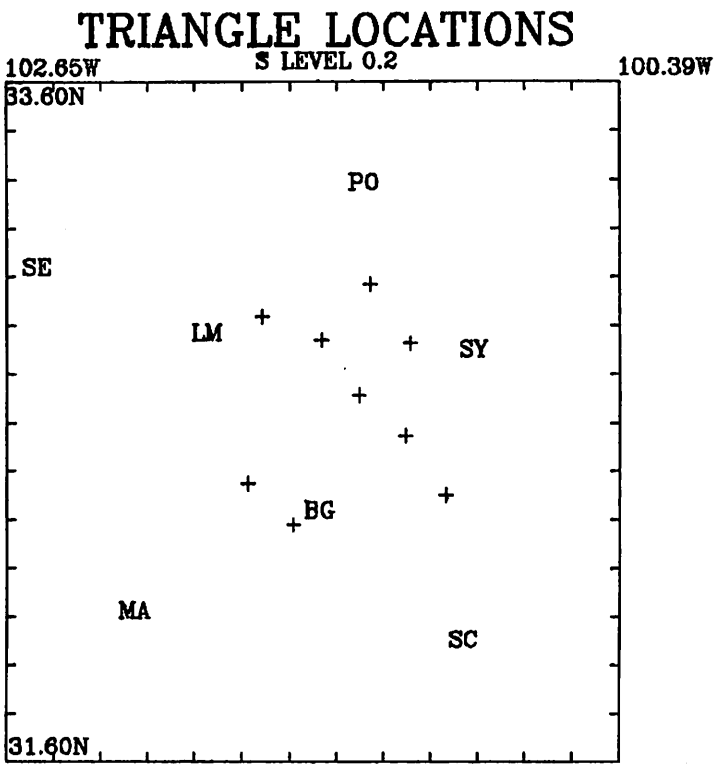

Fig. 8. Location of triangle centroids. These centroids (indicated by +) are treated as observations. Nine triangles have been retained for interpolation.

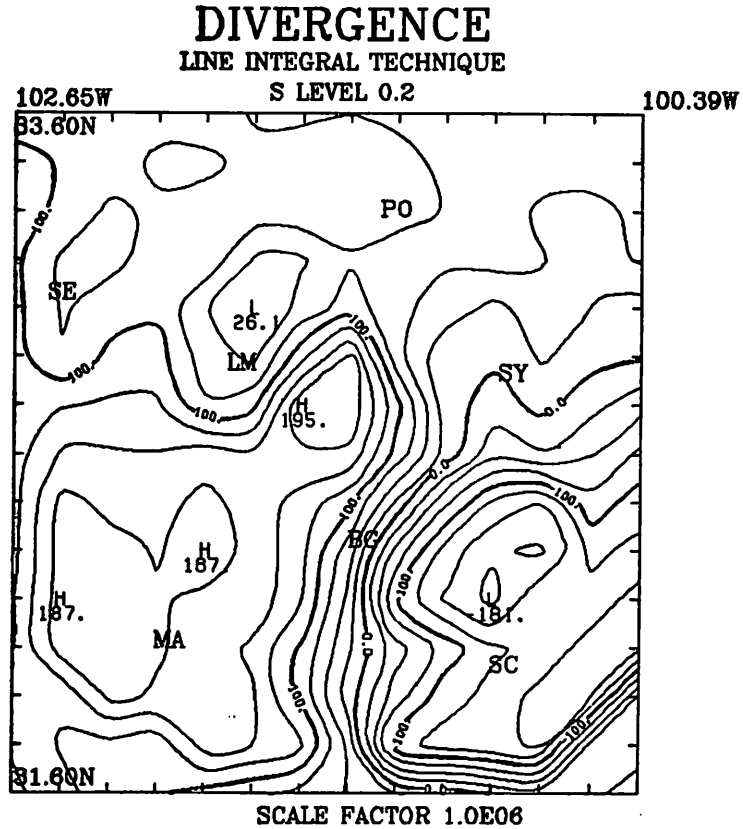

Fig. 9. LIT evaluation of divergence for the HIPLEX region. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

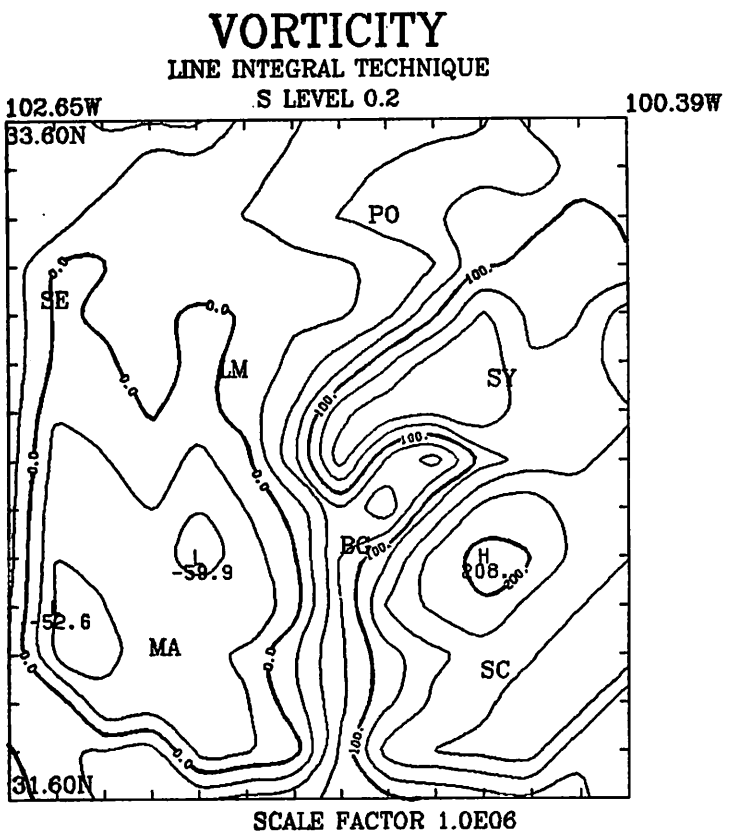

Fig. 10. LIT evaluation of vorticity for the HIPLEX region. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

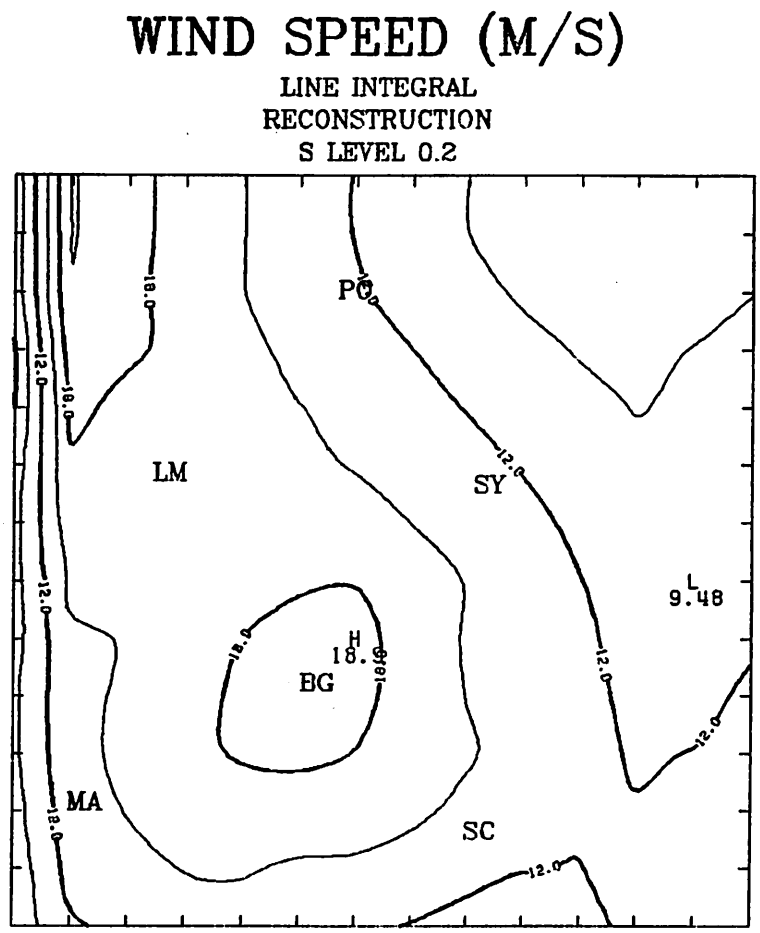

Fig. 11. Isotach analysis of the reconstructed wind field, via LIT, for a subregion within HIPLEX. Nine triangles retained for interpolation. Isotach interval is 3.0 m s^{-1}.

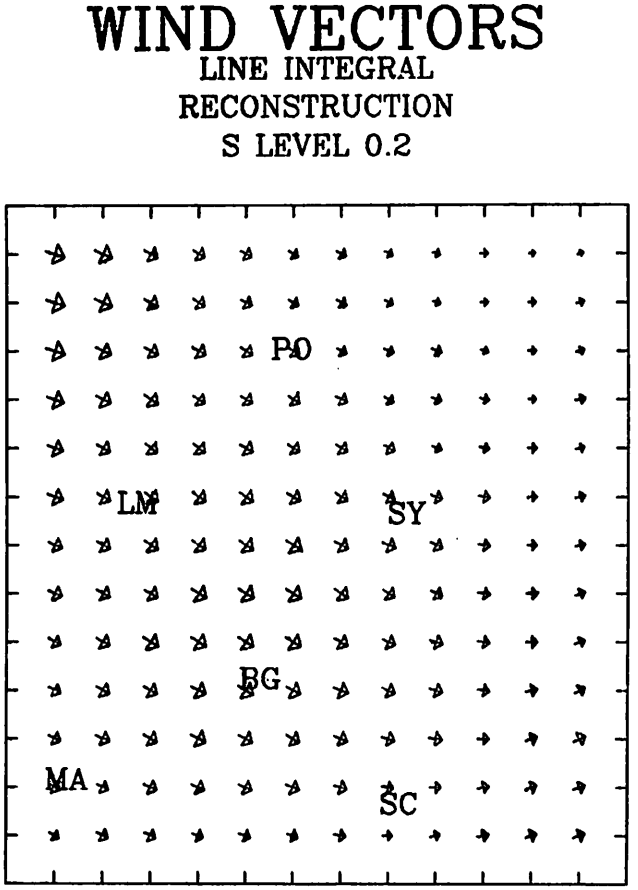

Fig. 12. Wind direction analysis of the reconstructed wind field, via LIT, for a subregion within HIPLEX. Nine triangles retained for interpolation.

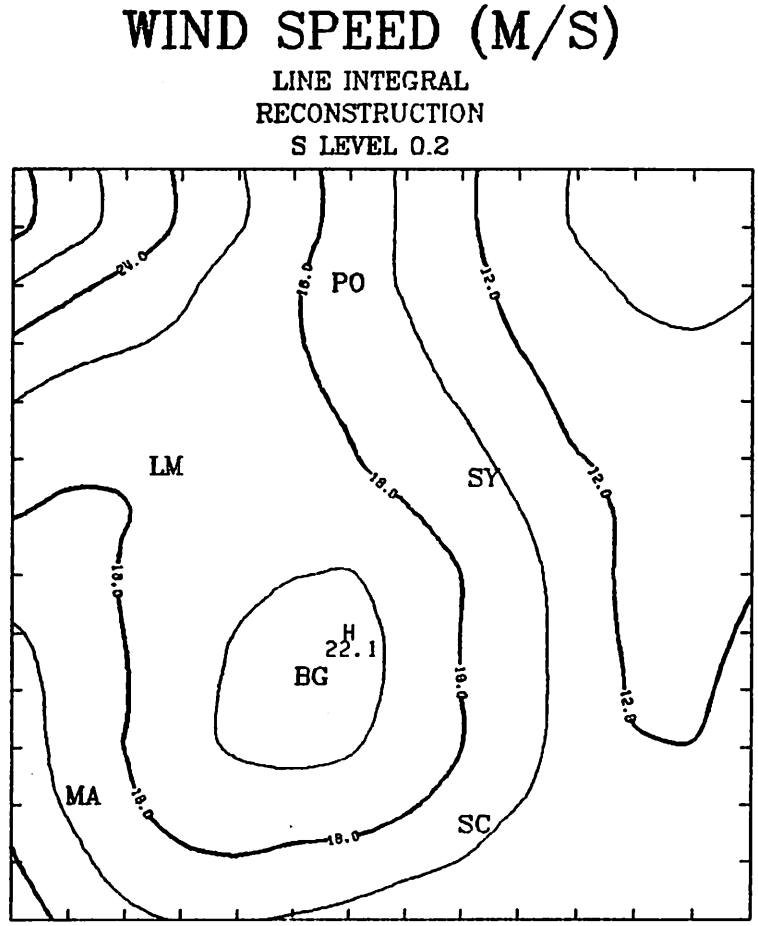

Fig. 13. Isotach analysis for a subregion within HIPLEX, with twenty-two triangles retained for interpolation.

wind field (Figs. 2 and 3, pp. 27-28). The root mean square vector error of the reconstructed wind field, developed from this version of the LIT, was 10.7 m s^{-1}.

Obviously, the selection criterion was "overrelaxed." A careful scrutiny of Fig. 8 revealed that no centroids (observations) were present in certain critical areas on the grid. In particular, centroids were absent for the Seagraves/Midland/Lamesa triangle, the Seagraves/Lamesa/Post triangle, and the Big Spring/Midland/Sterling City triangle. The criterion was altered, so that the above critical triangles (observations) survived the filtering. Seventeen triangles survived and, intuitively speaking, were well distributed (Fig. 14). The root mean square vector error, for this version of the LIT, was 5.6 m s^{-1}.

A final attempt was made to further reduce the vector errors for the LIT. We may recall that the methodology for reconstructing the wind field from the fields of divergence and vorticity entailed four iterations (see p. 23). This particular number of iterations proved optimal when reconstructing the wind field via the Finite Difference Method. However, this finding may not be true for the LIT. In fact, I found, through a trial-and-error approach, that the most optimal results were secured with only one execution of the wind retrieval scheme. It is obvious that the proper number of iterations (for the wind retrieval scheme) will be realized when the root mean square vector error between the observed and reconstructed wind is a minimum. With one execution of the scheme, the root mean square vector error

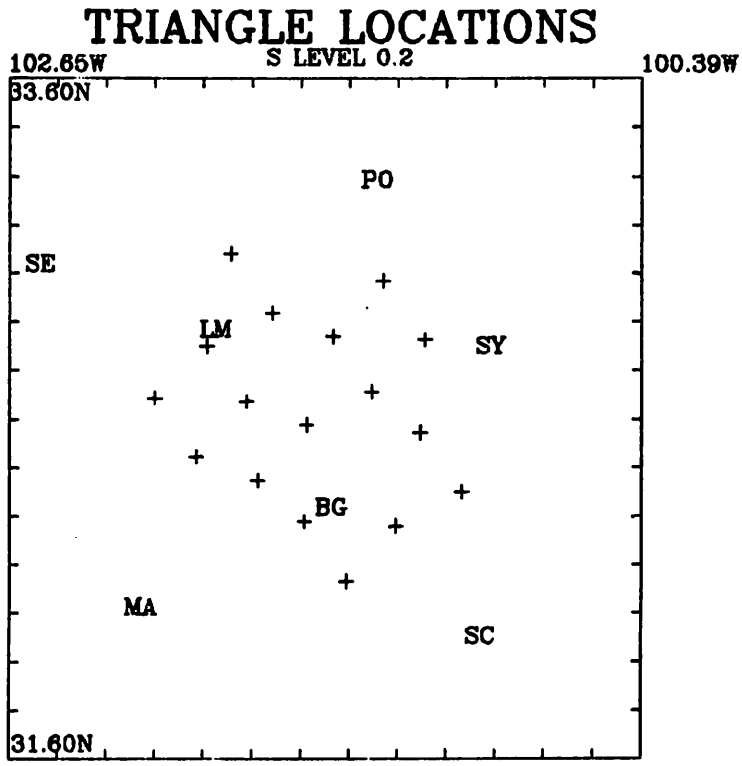

Fig. 14. Location of triangle centroids. Seventeen triangles have been retained for interpolation.

was reduced to 5.0 m s^{-1}.

Fig. 15 is the divergence field computed from the most recent version of the LIT. The divergence field shows distinct areas of strong convergence located either side of a line from Big Spring to Sterling City. The corresponding distinct areas of divergence are indicated either side of a line from Lamesa to Big Spring. These findings are roughly similar to those of the Finite Difference Method (Fig. 4, p. 29). Note that the distinct areas of convergence northeast of Seagraves and east of Post (suspicious findings in the Finite Difference Method, see p. 30) are absent in the LIT evaluation. Except for the area of divergence southwest of Midland, all distinct areas of divergence (convergence) can be supported from the original observations.

The vorticity field (Fig. 16) shows distinct areas of positive vorticity situated south and west of Snyder. Distinct areas of negative vorticity are shown south of Big Spring and northwest of Lamesa. These findings appear to be heuristically consistent with the original observations, and are roughly similar to the findings in the differential evaluation (Fig. 5, p. 31). Note that the distinct area of negative vorticity northeast of Sterling City and the distinct area of positive vorticity on the southern boundary of the grid (suspicious findings in the Finite Difference Method, see p. 30) are absent in the LIT evaluation. Again, all distinct areas of negative (positive) vorticity can be supported from the original observations. The resulting wind field is illustrated in Figs. 17 and 18.

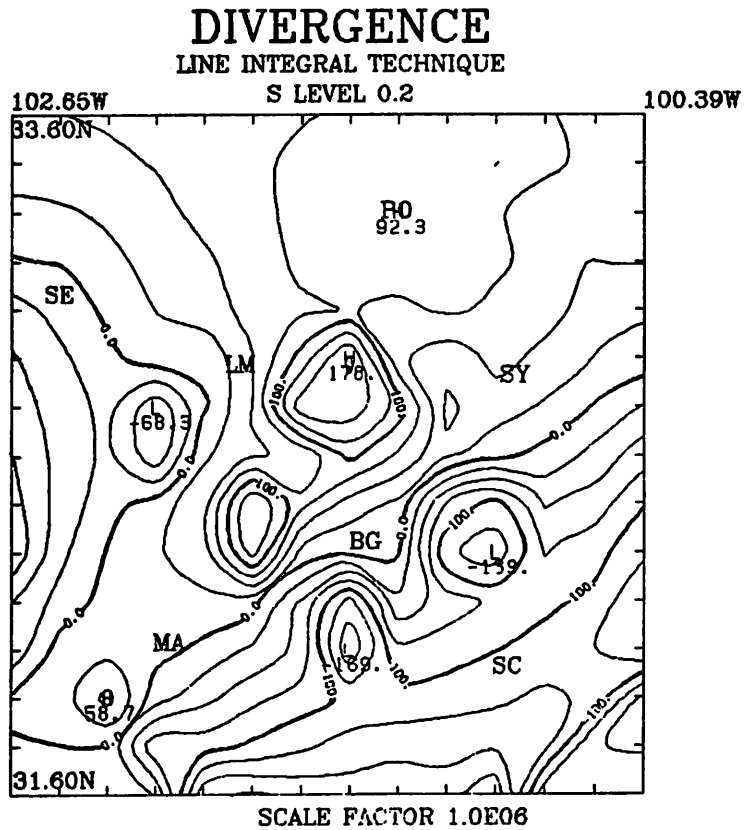

Fig. 15. Seventeen point values of divergence interpolated onto the grid. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

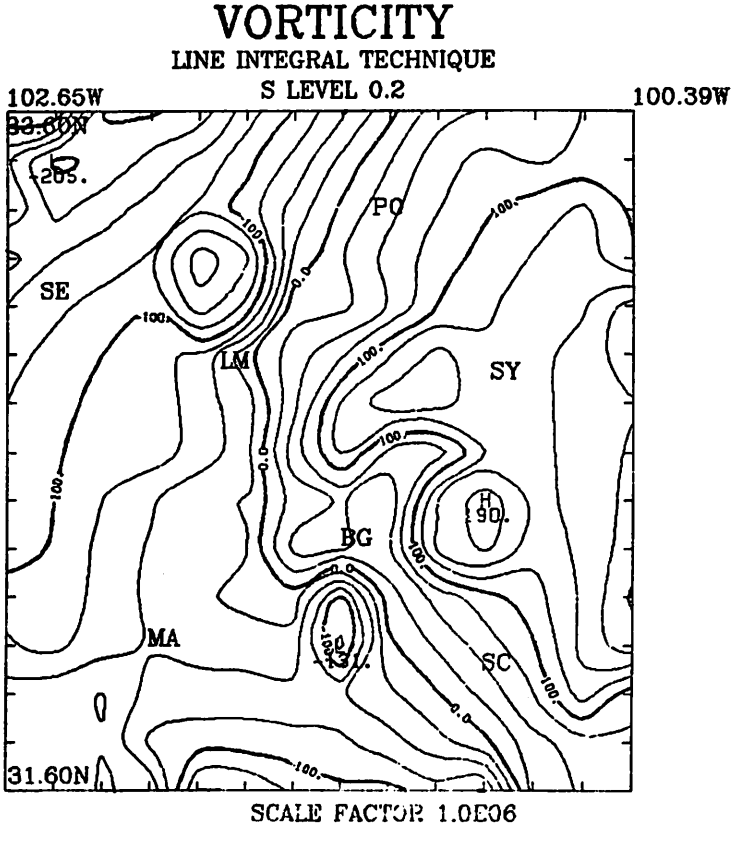

Fig. 16. Seventeen point values of vorticity interpolated onto the grid. Isopleth interval is 25.0 * 10^{-6} s^{-1}.

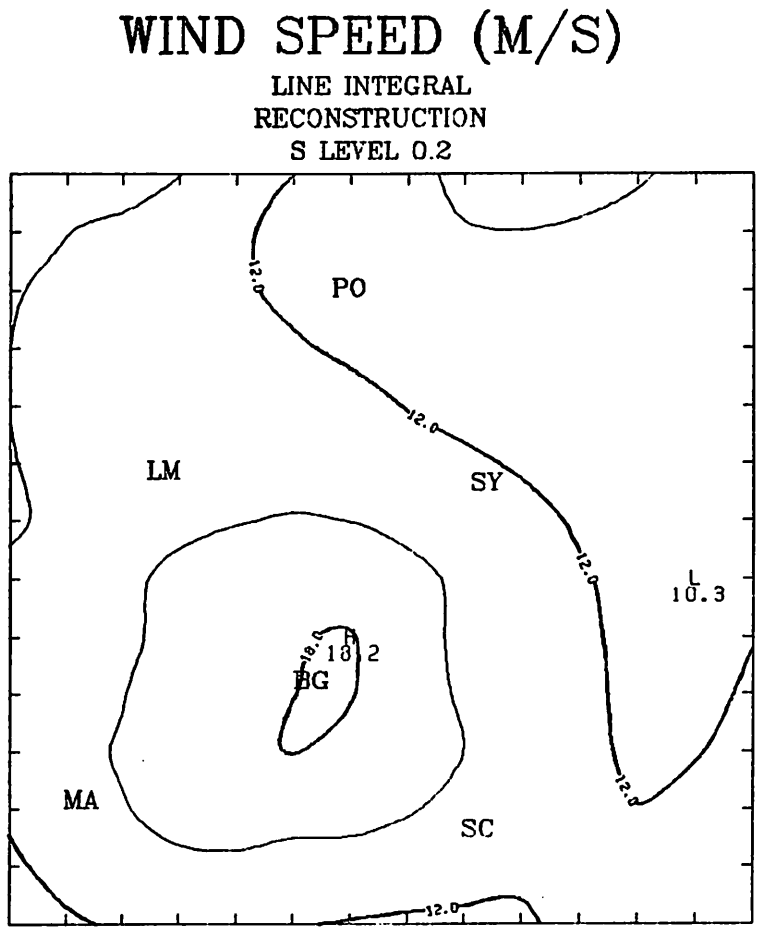

Fig. 17. Isotach analysis of the reconstructed wind field, realized from one execution of the wind retrieval scheme. Isotach interval is 3.0 m s^{-1}.

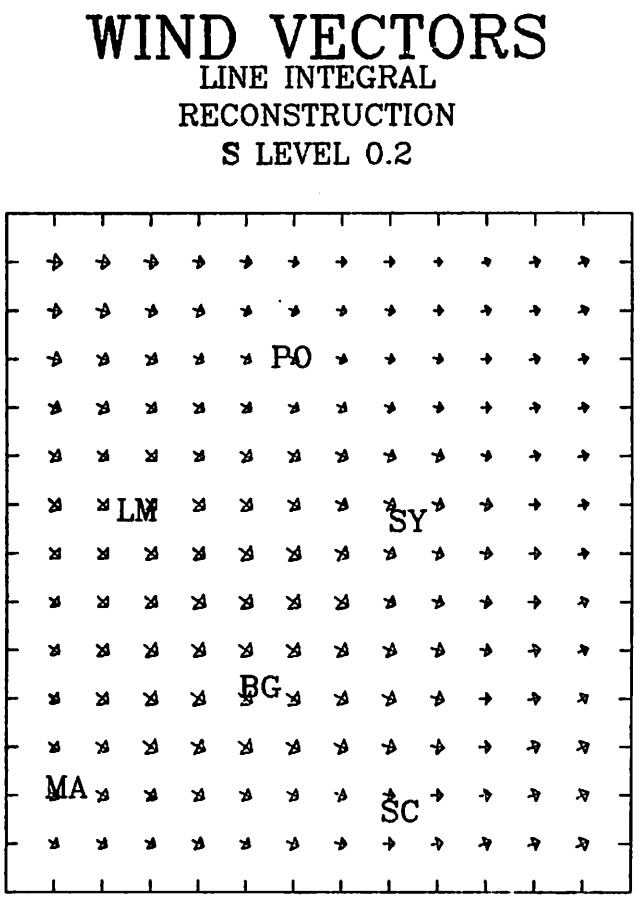

Fig. 18. Wind direction analysis of the reconstructed wind field, realized from one execution of the wind retrieval scheme.

d. Evaluation of Results

An appraisal of the given methods, founded solely on the magnitudes of their vector errors, is rather arbitrary. However, the vector error does compare the analyzed fields (realized by different methods) with the actual observations and, as is obvious, one cannot do better than the original observations.

Comparison of the reconstructed wind field via finite differences (Figs. 6 and 7, pp. 33-34) with the original gridded wind field (Figs. 2 and 3, pp. 27-28) exhibits good agreement at the Post, Snyder, and Sterling City stations. The Finite Difference Method computes slightly stronger wind speeds at the Big Spring and Lamesa stations. The greatest difference between the two wind fields was indicated at Midland, with its vector error of 2.5 m s^{-1}. The distribution of the vector error population was negatively skewed to the left, with a median of .6 m s^{-1}. The root mean square vector error was 1.3 m s^{-1}.

Comparison of the reconstructed wind field via the LIT (Figs. 17 and 18, pp. 47-48) with the original wind field (Figs. 2 and 3, pp. 27-28) exhibits good agreement for the Big Spring, Snyder, and Post stations. The LIT generates stronger wind speeds at the Midland and Sterling City stations. The greatest difference between the two wind fields was indicated at Lamesa, with its vector error of 6.3 m s^{-1}. The distribution of the vector error population was approximately normal, with a median of 6.0 m s^{-1}. The root mean square vector error was 5.0 m s^{-1}.

From the preceding arguments, it appears that the Finite Difference Method produces a more representative wind field. The root mean square vector error was considerably lower, when compared to the Line Integral Technique. Individual station vector errors were all lower. However, the distribution of the vector error population was more erratic in the finite-difference approach.

6. SUMMARY

A computer program (Appendix B) has been developed for computing divergence and vorticity directly from wind observations, and, thus having available what are effectively "observed" kinematic fields. The wind field is reconstructed so as to be consistent with these kinematic fields.

a. Comparison between the LIT and Finite Difference Method Computations of Divergence and Vorticity

The evaluations of the kinematic variables, via line integrals and centered differences, were applied to an actual meteorological data set. With one exception (see p. 44), all distinct areas of divergence (vorticity), realized through the LIT evaluation, could be supported in view of the original observations. This support was lacking for "certain" distinct areas of divergence (vorticity) realized through the Finite Difference Method evaluation. These "certain" distinct areas are suspicious, probably an artifact of the gridding scheme.

It is not difficult to see that "certain" distinct areas of the kinematic variables can be falsely created by the Finite Difference Method, especially near the boundaries of the computational domain. In the differential evaluation, it is the "change" in the grid point values of the wind which determine divergence and vorticity. The orientation of stations (with respect to each other) and the determination of grid point values of wind (realized through a distance-

dependent weighting from grid point to observation) may interact in such a fashion so as to create these "certain" distinct areas (through falsely constructed wind gradients). The LIT does not require an interpolation of the winds (or the gradient of the winds); therefore, it is immune to this problem.

It is also apparent (from the plots) that integration produces much "smoother" kinematic fields when compared to differentiation. The term "smoother" suggests the amount (degree) of variability in the computed values over the computational domain. This term does not pertain to the "wiggles" inherent in the isopleths; idiosyncrasies in the versatec plotter are not considered here.

b. Comparison between the Reconstructed Wind Fields via the LIT and Finite Difference Method

The reconstructed wind field, via the Line Integral Technique, was compared to a reconstructed wind field, via the Finite Difference Method. Although the methodologies are different, both techniques are expected to retain the original measured winds. The observations are the only definitive information on the grid. From this perspective, the reconstruction of a wind field, by using the kinematic fields obtained through the Finite Difference Method, appears superior to its competitor.

c. Future Research

A pertinent problem in the implementation of the wind retrieval

scheme (see p. 23) was the determination of proper boundary values (normal and tangential wind components, V_n and V_s) for the subsequent calculations. In this study, the needed winds (for the boundary) were obtained from the original gridded wind field. Obviously, this choice of boundary values will favor the finite-difference reconstruction of the wind field since the kinematic variables are computed from the derivatives of the original gridded wind field. A systematic procedure is required to secure a "better" choice of boundary values for the LIT.

The following is a suggestion:

> It is a well-known fact (Shukla and Saha, 1974) that the boundary values (V_s and V_n) must satisfy the integral constraints of
>
> $$\int_A\int \xi_{LIT}\, dA = \oint V_s ds$$
>
> $$\int_A\int D_{LIT}\, dA = \oint V_n ds$$
>
> where A denotes the area of computation. Usually, a residual exists from the computations such that
>
> $$E_s = \oint V_s ds - \int_A\int \xi_{LIT}\, dA$$
>
> $$E_n = \oint V_n ds - \int_A\int D_{LIT}\, dA \quad .$$
>
> If the divergence and vorticity values are considered absolute, then the above residuals can be reduced

by suitably altering the boundary values of V_s and V_n. If we let

$$E_s = \bar{V}_s \oint ds$$

$$E_n = \bar{V}_n \oint ds \quad ,$$

where the overhead bar indicates some arbitrary value, then

$$\bar{V}_s = \left[\oint V_s ds - \iint_A \xi_{LIT} dA \right] / \oint ds$$

$$\bar{V}_n = \left[\oint V_n ds - \iint_A D_{LIT} dA \right] / \oint ds \quad .$$

Knowing the values of $\bar{V}_s$ and $\bar{V}_n$, we can systematically alter each given boundary value of V_s and V_n (obtained from the original gridded wind field) by the following equations

$$(V_s')_i = (V_s - \bar{V}_s)_i$$

$$(V_n')_i = (V_n - \bar{V}_n)_i$$

where V_s' and V_n' are the altered boundary values, and i designates a given boundary point on the grid.

The author feels that the above suggestion may reduce the large root mean square vector error inherent in the LIT reconstruction of the winds. Furthermore, this suggestion should also be applied to the finite-difference reconstruction of the wind field for additional refinement.

Finally, the criterion used for the selection of suitable triangles (from a given population) for the purpose of obtaining point values of divergence and vorticity appears to be inadequate. A subjective trial-and-error approach was the only means available (in this study) for selecting a suitable "group" of point values for subsequent interpolation (to obtain grid point values of divergence and vorticity). In using this approach, the most ideal triangle was assigned a weight identical to less ideal, but suitable, triangles for implementing the interpolation phase of the LIT. Obviously, this is a source of error. One must determine a criterion which assigns an appropriate weight to each member of this select group of triangles before implementing the interpolation phase. The reader is directed to a paper by Ceselski and Sapp (1975) for additional insight.

d. Final Remarks

Dr. James R. Scoggins (personal communication) reports that the "suspicious" distinct areas of divergence and vorticity near the grid boundary, generated by finite-difference calculations, are usually ignored. If this rule-of-thumb is implemented, then the wind field reconstructed from fields of divergence and vorticity, calculated by centered differences, is clearly superior to the other two methods in determining the initial grid point values of wind for use in a mesoscale primitive equation model. The method of initializing the wind field by using grid point values of wind derived directly from the observations (through interpolation) is inferior despite its small

root mean square vector error. This given wind field appears likely to be kinematically inconsistent with the derived boundary values. The other method which consists of reconstructing the wind field from LIT-computed fields of divergence and vorticity possesses a large root mean square vector error. This given wind field appears to be less consistent with the original observations when compared to the other two methods.

It must be pointed out that the complete initialization of a mesoscale primitive equation model was not realized in this study. The mesoscale primitive equation model developed by Djuric and Das (Scoggins et al., 1981) was initialized from the gridded fields of geopotential, wind, and temperature, as derived directly from the observations (through interpolation). It is speculated that the vitiation in the skill of their model was the result of these initial fields (which were not in a state of balance).

If a balanced state between mass, wind, and temperature is to be achieved, the wind field must be reconstructed so as to (at least) preserve its vorticity. Stream functions, computed by a relaxation of the vorticity field (see p. 23), are utilized in a balance equation (Warner et al., 1978) for the purpose of constructing geopotentials consistent with the analyzed (or reconstructed) winds. The temperature field is constructed from the hydrostatic relationship utilizing the derived geopotentials. It is believed that these balanced fields of geopotential, wind, and temperature (realized from the above procedure) will reduce the problem of noise generation in

the initial stage of a numerical model run.

Finally, it should be emphasized that Shukla and Saha (1974) have proven that an analyzed wind field which preserved both its vorticity and divergence always gives a better representation of the original observed wind, than a wind field that preserves only its vorticity. In this study, the gradients of the stream function were an order of magnitude larger than the gradients of the velocity potentials (computed by a relaxation of the divergence field). However, in other cases, both gradients may have comparable magnitudes so that the method of Shukla and Saha will be more relevant.

REFERENCES

Barnes, S. L., 1964: A technique for maximizing details in numerical weather map analysis. J. Appl. Meteor., 3, 396-409.

Bellamy, J. C., 1949: Objective calculations of divergence, vertical velocity, and vorticity. Bull. Amer. Meteor. Soc., 30, 45-49.

Bergthórsson, P., and B. Döös, 1955: Numerical weather map analysis. Tellus, 7, 329-340.

Bjerknes, V., 1904: Das Problem von der Wettervorhersage, betrachtet vom Standpunkt der Mechanik und der Physik. Meteor. Z., 21, 1-7.

Ceselski, B. F., and L. L. Sapp, 1975: Objective wind field analysis using line integrals. Mon. Wea. Rev., 105, 885-892.

Conte, S. D., and C. de Boor, 1980: Elementary Numerical Analysis. McGraw-Hill Book Company, 432 pp.

Cressman, G. P., 1959: An operational objective analysis system. Mon. Wea. Rev., 87, 367-374.

Fankhauser, J. C., 1969: Convective processes resolved by a meso-scale rawinsonde network. J. Appl. Meteor., 5, 778-798.

Fuller, G., 1972: Plane Trigonometry with Tables. McGraw-Hill Book Company, 271 pp.

Halliday, D., and R. Resnick, 1974: Fundamentals of Physics. John Wiley & Sons, Inc., 827 pp.

Haltiner, G. J., and R. T. Williams, 1980: Numerical Prediction and Dynamic Meteorology. John Wiley & Sons, Inc., 477 pp.

Sangster, W. E., 1960: A method of representing the horizontal pressure force without reduction of station pressures to sea level. J. Meteor., 17, 166-176.

Sasaki, Y., 1958: An objective analysis based on the variational method. Soc. Japan, 36, 77-78.

_____, 1960: An objective analysis for determining initial conditions for the primitive equations. Tech. Rep. (Ref. 60-16T), Dept. of Oceanography and Meteorology, Texas A&M University, 22 pp.

Schaefer, J. T., and C. A. Doswell III, 1979: On the interpolation of a vector field. Mon. Wea. Rev., 107, 458-476.

Scoggins, J. R., D. Djuric, P. Das, G. L. Huebner, and A. B. Long, 1981: TAMU Texas HIPLEX studies for 1979. Tech. Rep. (TDWR Contract No. 14-00030), Dept. of Oceanography and Meteorology, Texas A&M University, 301 pp.

Shukla, J., and K. R. Saha, 1974: Computation of nondivergent streamfunction and irrotational velocity potential from the observed winds. Mon. Wea. Rev., 102, 419-425.

Shuman, F. G., 1957: Numerical methods in weather prediction: II. Smoothing and filtering. Mon. Wea. Rev., 85, 357-361.

_____, and J. B. Hovermale, 1968: An operational six-layer primitive equation model. J. Appl. Meteor., 7, 525-547.

Talagrand, O., 1972: On the damping of high frequency motions in four-dimensional assimilation of meteorological data. J. Atmos. Sci., 29, 1571-1574.

Warner, T. T., R. A. Anthes, and A. L. McNab, 1978: Numerical simulations with a three-dimensional mesoscale model. Mon. Wea. Rev., 106, 1079-1099.

Williams, R. J., 1976: Surface parameters associated with tornadoes. Mon. Wea. Rev., 104, 540-545.

APPENDIX A

A1) Balloon Drift Routine

The procedure is applied to each radiosonde station j:

1) Find the required S level height (S = 0.2) from the equation

$$z_S(j) = z_o(j) + S\left(z_t - z_o(j)\right)$$

(obtained from (10), p. 13).

2) Interpolate range and azimuth values of the balloon onto the S levels by the equation

$$V_S(j) = V_B(j) + \left(\left(z_S(j) - z_B(j)\right) / \left(z_A(j) - z_B(j)\right)\right) * \left(V_A(j) - V_B(j)\right) \quad ,$$

where V_B and V_A refer to either range or azimuth values measured at given heights (from radiosonde calibration charts) below and above the required S level height respectively, z_B and z_A refer to these given heights (which sandwich the S level height), and V_S is the interpolated value (range or azimuth). Azimuth (AZM) is the angle between true north and balloon location. It is a horizontal direction expressed in degrees. Range (RNG) is the line-of-sight distance between balloon and radiosonde station.

3) Determine the x and y displacements of the balloon (XDSP and YDSP, respectively) by the equations

$$XDSP = (RNG)\left(\sin(AZM)\right) / GD$$

$$YDSP = (RNG)\left(\cos(AZM)\right) / GD \quad ,$$

where GD is grid distance (15.8 km).

4) Determine the new grid locations (x',y') of the balloon from its launch site (x,y) by the equations

$$x' = x + XDSP$$

$$y' = y - YDSP \quad .$$

The y axis points approximately toward the south.

5) Find the surface height at the new location (x',y'). This is realized through a bilinear interpolation of gridded surface heights onto the new location $z_o'(j)$.

6) Find the adjusted S level height $(z_S'(j))$ from the equation

$$z_S'(j) = z_o'(j) + S\left(z_t - z_o'(j)\right) \quad .$$

7) Interpolate desired meteorological values onto the adjusted S level height by the equation listed in step 2 (V_B and V_A infer desired meteorological values).

A2) Routine for Line Integral Technique

1) Construct all thirty-five triangles by the combination of each radiosonde observation with its neighbor. Information at the first vertex consists of i and j grid locations (FI1, FJ1) and u and v components of the wind (U1,V1). Information at the vertices are

ordered as follows:

vertex1	FI1, FJ1, U1, V1
vertex2	FI2, FJ2, U2, V2
vertex3	FI3, FJ3, U3, V3 .

This ordering is realized in the subsequent computer program (Appendix B) through the subroutines METH1, METH2, METH3, METH4, METH5.

2) Reorder the second and third vertices (if needed) to ensure that all subsequent summations proceed in a counterclockwise direction (integrations commence at vertex1). This reordering is found in Appendix B under subroutine SORT. In addition, the position of a given vertex (situated either above or below its corresponding triangle segment, defined by the remaining vertices) must be established. This entails a test for each triangle segment (DO LOOP 7813, Appendix B).

3) For each triangle, evaluate:

Triangle centroid (Halliday and Resnick, 1974, p. 138).
Area of triangle (Fuller, 1972, p. 167).
Slope of triangle segment (Fuller, 1972, p. 164).
Weight of triangle.

(DO LOOP 2899, Appendix B).

4) Calculate mean u and v wind components from the vertices, and assign these values to the midpoints of the triangle segments. From these averaged components, a wind direction (DIR) and speed (SPD) may be realized from the equations

$$DIR = (57.29)\ \tan^{-1}(v/u)$$

$$SPD = \left(u^2 + v^2\right)^{\frac{1}{2}} \quad .$$

The x coordinate axis serves as the reference direction for the angles (DIR).

5) Determine the tangent (VT1) and normal (VN1) components of the mean wind which is situated at the midpoint of each triangle segment (subroutine SIGN, Appendix B). Point values of divergence (DIV) and vorticity (VOR) may be realized from these winds, and assigned to the triangle centroids by the equations

$$DIV = \Big[(VN1)(SIDE1) + (VN2)(SIDE2) + (VN3)(SIDE3)\Big] / (TRIANGLE\ AREA)$$

$$VOR = \Big[(VT1)(SIDE1) + (VT2)(SIDE2) + (VT3)(SIDE3)\Big] / (TRIANGLE\ AREA)$$

where SIDE1, SIDE2, SIDE3 refer to the lengths of the triangle segments.

6) Filter "bad" triangles.

7) Interpolate "good" triangles onto a uniform mesh (by the weighted averaging scheme mentioned earlier) to obtain grid point estimates of divergence and vorticity (subroutines GUESS, ANAL, SMOOTH, Appendix B).

APPENDIX B

Computer Program

```
//SOPTIONS
C HIPLEX RESEARCH
C <16> CORRESPONDS TO THE NUMBER OF TOTAL STATIONS
C < 7> CORRESPONDS TO THE NUMBER OF UPPER LEVEL STATIONS
C <10> CORRESPONDS TO THE NUMBER OF VARIABLES TO BE GRIDDED
C <21,21> CORRESPONDS TO THE GRID SIZE
C < 8> CORRESPONDS TO THE NUMBER OF S LEVELS INITIALIZED IN THE
C       DATA STATEMENT
C VALUE(NUMBER OF STATIONS, LEVELS, VARIABLES)
      CHARACTER*3  YES1 (35)
      REAL         DLAT (35),FJ2  (35),B     (18,18),VT1  (35)
     1,            DLON (35),FI3  (35),C     (18,18),VN1  (35)
     2,            FFI  (35),FJ3  (35),              VT2  (35)
     3,            FFJ  (35),DTU1 (35),              VN2  (35)
     4,            GPM  (35),DTV1 (35),              VT3  (35)
     5,            DATA (35),DTU2 (35),              VN3  (35)
      REAL         DATA1(35),DTV2 (35),              AREA1(35)
     1,            SFCHT(35),DTU3 (35),              SPD1 (35)
     2,            FI   (35),DTV3 (35),              DIR1 (35)
     3,            FJ   (35),SIDE1(35),              SPD2 (35)
     4,            ANGLE(35),SIDE2(35),              DIR2 (35)
     5,            WHT  (35),SIDE3(35),              SPD3 (35)
      REAL                   CRX  (35),A     (18,18),DIR3 (35)
     1,      V ( 7, 8),      CRY  (35),A1    (18,18),ANG1 (35)
     2,                      SCANR(35),TC1   (18,18),ANG2 (35)
     3,      U ( 7, 8),      FI1  (35),TC    (18,18),ANG3 (35)
     4,      ZS( 7, 8),      FJ1  (35),TW1   (18,18),DIV  (35)
     5,                      FI2  (35),TW    (18,18),VORT (35)
      REAL         ANGL1(35),ANGL2(35),              ANGL3(35)
      REAL  E(50,16,16),VALUE(16, 8,10),SLVL( 8)
      INTEGER      NO   (35),ITIME( 6),KDAY (23)
     1,DAY,T,HIPL,HIPLX,IFLAG
      DATA SLVL/0.0 ,.10,.20,.30,.40,.50,.60,.70/
      DATA KDAY/21,25,26,27,28,1,4,5,8,9,19,21,24,2,3,4,5,6,7,14
     $,15,17,18/
      DATA ITIME/15,18,21,00,03,12/
      COMMON DATA,DATA1,FFI,FFJ,A,A1,TC,TC1,TW,TW1
      COMMON /GRIDIJ/ISTART,JSTART,IEND,JEND
      COMMON /GRIDJK/ SLAT,SLON,M1,N1,GD,M2,N2
      DEFINE FILE 62(805,900,U,NB)
C M1 AND N1 ARE THE IJ LOCATION OF THE NORTHWEST CORNER OF A
C (18,18) GRID WHICH IS CONTAINED IN THE LARGER (18,18) GRID.
C 35.6N AND 105.5W IS THE LATITUDE AND LONGITUDE OF THE NORTHWEST
C CORNER.  GRID SPACING IS 40 KILOMETERS.
      CUE    =0.0
      GEO    =0.0
      GEO1   =0.0
      M1                              = 1
      M2                              =18
      N1                              = 1
      N2                              =18
      ISTA                            =16
      IPAR                            =10
      SCALE                           = 1.0
      SMO                             = 0.1
      NTS                             = 7
      SCAN                            = 3.0
      NOSCAN                          = 4
      ISTART                          = 1
```

```
      IEND                                =18
      JSTART                              = 1
      JEND                                =18
      ZT                                  =18000.0
      LEV                                 = 8
      NRPTS                               =35
      ILM                                 = 1
      MNY                                 =50
      GD                                  =15.867
      SLAT                                =33.80
      SLON                                =102.65
      DO 5000   I=1,NTS
      DO 5000   J=1,LEV
      ZS(I,J)             =0.0
      U (I,J)             =0.0
      V (I,J)             =0.0
5000  CONTINUE
      DO 5001   I=ISTART,IEND
      DO 5001   J=JSTART,JEND
      A  (I,J)            =0.0
      A1 (I,J)            =0.0
      TC (I,J)            =0.0
      TW (I,J)            =0.0
      TW1(I,J)            =0.0
      B  (I,J)            =0.0
      TC1(I,J)            =0.0
5001  CONTINUE
      DO 5002   I=1,ISTA
      DATA (I)            =0.0
      DATA1(I)            =0.0
      FFI  (I)            =0.0
      DLAT (I)            =0.0
      FFJ  (I)            =0.0
      DLON (I)            =0.0
      NO   (I)            =0.0
      GPM  (I)            =0.0
      SECHT(I)            =0.0
5002  CONTINUE
      DO 7002   I=1,NRPTS
      SCANR(I)            =SCAN
      DTU1 (I)            =0.0
      DTV1 (I)            =0.0
      FI1  (I)            =0.0
      FJ1  (I)            =0.0
      FI2  (I)            =0.0
      FJ2  (I)            =0.0
      FI3  (I)            =0.0
      FJ3  (I)            =0.0
      CRX  (I)            =0.0
      CRY  (I)            =0.0
      DTU2 (I)            =0.0
      DTV2 (I)            =0.0
      DTU3 (I)            =0.0
      DTV3 (I)            =0.0
      SIDE1(I)            =0.0
      SIDE2(I)            =0.0
      SIDE3(I)            =0.0
      AREA1(I)            =0.0
7002  CONTINUE
      DO 5003   I=1,ISTA
```

```
      DO 5003  J=1,LEV
      DO 5003  K=1,IPAR
      VALUE(I,J,K)     =0.0
 5003 CONTINUE
C
      DO 5004  I=1,ISTA
      READ (5,500) NO(I),DLAT(I),DLON(I),SFCHT(I),YES1(I)
  500 FORMAT (I3,1X,F4.2,1X,F5.2,2X,F4.0,4X,A3)
 5004 CONTINUE
      DO 5005  LST=1,ISTA
        CALL DEGRID (DLAT(LST),DLON(LST),FI(LST),FJ(LST))
      FFI(LST)=FI(LST)
      FFJ(LST)=FJ(LST)
 5005 CONTINUE
C
      IDAY     = 5
      KTIME    = 5
      IMON                        = 5
      IF (IDAY .GE.  6) IMON      = 6
      IF (IDAY .GE. 14) IMON      = 7
      IIMON      =IMON
      IIDAY      =KDAY (IDAY)
      IITIME     =ITIME(KTIME)
      WRITE (6,503)
  503 FORMAT ('1')
      WRITE (6,504)
  504 FORMAT (' ',T10,'HEIGHTS',T25,'PRESSURE',T40,'TEMPERATURE',
     $T56,'DEW POINT',T72,'DIRECTION',T86,'SPEED',T95,'U WIND',
     $T105,'V WIND')
      WRITE (6,610)
  610 FORMAT (' ',T10,'(METERS)',T27,'(MB)',T41,'(CELSIUS)',T73,
     $'(DEG)',T86,'M/SEC',T96,'M/SEC',T106,'M/SEC')
C
C
  505 FORMAT('0',T45,A3,T50,I2)
  506 FORMAT ('0')
  508 FORMAT(' ',T 9,F7.1,T26,F6.1,T43,F5.1,T58,F5.1,
     #T74,F5.1,T87,F4.1,T95,F5.1,T105,F5.1,T118,F4.2)
      DO 6006  LST=1,NTS
      WRITE (6,505) YES1(LST),NO(LST)
      WRITE (6,506)
      N3 = (LST-1)*11S+(IDAY-1)*5+KTIME
      READ (02'N3) ((E(I,J,LST),I=1,50),J=1,16)
      DO 6005  I=1,50
      WRITE (6,508)  (E(I,J,LST),J=3,10)
 6005 CONTINUE
      GPM(LST)=E(1,3,LST)
 6006 CONTINUE
      DO 5070  K=1,ISTA
      DATA (K)=SFCHT(K)
 5070 CONTINUE
      DO 5008  LT=1,ISTA
      CUE      =CUE+1
      GEO      =DATA (LT)+GEO
 5008 CONTINUE
      GEO1     =GEO/CUE
        CALL GUESS     (A,GEO1)
        CALL ANAL      (ISTA,SCANR,NOSCAN,1)
        CALL SMOOTH    (A,B,SMO)
C
```

```
C
C
C
      DO 5010   IPAR=3,10
      DO 5011   LST=1,NTS
      DO 5012   I=1,3
      ZS(LST,I)=E(1,3,LST)+SLVL(I)*(ZT-E(1,3,LST))
      IF (I .EQ. 1) VALUE(LST,I,IPAR)=E(I,IPAR,LST)
      IF (I .EQ. 1) GO TO 5012
      DO 5013   J=1,MNY
      IF (ZS(LST,I) .LE. E(J,3,LST)) GO TO 510
 5013 CONTINUE
      ZS(LST,I)=0.0
      GO TO 5011
  510   CALL RATIO  (ZS,E,LST,I,J,15,RNGE)
        CALL RATIO  (ZS,E,LST,I,J,16,AZM)
      ANG                    =AZM*0.017453
      XDSP                   =(RNGE*SIN(ANG))/GD
      YDSP                   =(RNGE*COS(ANG))/GD
      FFI(LST)               =XDSP + FI(LST)
      FFJ(LST)               =FJ(LST) - YDSP
        CALL INTRP  (A,FFI(LST),FFJ(LST),DINT)
      GPM(LST)               =DINT
      ZS(LST,I)              =GPM(LST)+SLVL(I)*(ZT-GPM(LST))
      DO 5014   J=1,MNY
      IF (ZS(LST,I) .LE. E(J,3,LST)) GO TO 511
 5014 CONTINUE
      ZS(LST,I)              =0.0
      GO TO 5011
  511   CALL RATIO  (ZS,E,LST,I,J,IPAR,VLE)
      VALUE(LST,I,IPAR)      =VLE
 5012 CONTINUE
 5011 CONTINUE
 5010 CONTINUE
      WRITE (6,501)
  501 FORMAT('1','I AND J GRID LOCATIONS')
      WRITE (6,601)
  601 FORMAT ('0',T15,'OBSERVATION',T45,'BALLOON DRIFT')
      WRITE (6,502) (FI(LST),FJ(LST),YES(LST),FFI(LST),FFJ(LST),
     *LST=1,ISTA)
  502 FORMAT (' ',T14,F6.2,F6.2,T30,A7,T44,F6.2,F6.2)
C
      WRITE (6,512)
  512 FORMAT('1','DATA ON S LEVELS',T40,'(0) IMPLIES MISSING DATA')
      WRITE (6,513)
  513 FORMAT ('0',T10,'HEIGHTS',T25,'PRESSURE',T40,'TEMPERATURE',
     $T56,'DEW POINT',T72,'DIRECTION',T86,'SPEED',T95,'U WIND',
     $T105,'V WIND',T116,'S LEVEL')
      DO 5015   LST=1,NTS
      WRITE (6,505)   YES(LST),NO(LST)
      WRITE (6,506)
      DO 5015   I=1,LEV
      WRITE (6,508) ZS(LST,I),(VALUE(LST,I,IPAR),IPAR=4,10),SLVL(I)
 5015 CONTINUE
      WRITE (6,509)   IIMON,IIDAY,IITIME
  509 FORMAT ('1',10X,'SURFACE HEIGHT ANALYSIS ON ',I2,
     $'/',I4.3X,'AT',3X,I4,' GMT')
        CALL OUTPUT   (A,M1,M2,N1,N2,1.0,1)
C
      IL=3
```

```
C
  520 IPAR=9
      DO 5020  LST=1,NTS
      IF (IL .EQ. 9) GO TO 6000
      DATA (LST)=VALUE(LST,IL,IPAR)
      DATA1(LST)=VALUE(LST,IL,IPAR+1)
 5020 CONTINUE
      L1=1
      K1=1
      CALL METH1 (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,DTU1,DTV1,
     $FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      CALL METH2 (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,DTU1,DTV1,
     $FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      CALL METH3 (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,DTU1,DTV1,
     $FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      CALL METH4 (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,DTU1,DTV1,
     $FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      CALL METH5 (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,DTU1,DTV1,
     $FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      CALL SORT (FI1,FJ1,DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,
     &DTV3)
      WRITE (6,2897)
 2897 FORMAT ('1',T10,'VERTEX1',T30,'VERTEX2',T50,'VERTEX3',
     &T70,'CENTROID',T90,'TRIANGLE')
      DO 2899  I= 1,35
      CRX  (I)=(FI1(I)+FI2(I)+FI3(I))/3
      CRY  (I)=(FJ1(I)+FJ2(I)+FJ3(I))/3
C
      SIDE1(I)=((FI1(I)-FI2(I))**2+(FJ1(I)-FJ2(I))**2)**0.5
      SIDE2(I)=((FI2(I)-FI3(I))**2+(FJ2(I)-FJ3(I))**2)**0.5
      SIDE3(I)=((FI3(I)-FI1(I))**2+(FJ3(I)-FJ1(I))**2)**0.5
C
      S       =0.5*(SIDE1(I)+SIDE2(I)+SIDE3(I))
      AREA1(I)=(ABS(S*(S-SIDE1(I))*(S-SIDE2(I))*(S-SIDE3(I))))**0.5
      ANGL1(I)=ARCOS((SIDE3(I)**2)+(SIDE1(I)**2)-SIDE2(I)**2)/
     &(2*SIDE3(I)*SIDE1(I)))
      ANGL2(I)=ARCOS((SIDE1(I)**2)+(SIDE2(I)**2)-SIDE3(I)**2)/
     &(2*SIDE1(I)*SIDE2(I)))
      ANGL3(I)=ARCOS((SIDE2(I)**2)+(SIDE3(I)**2)-SIDE1(I)**2)/
     &(2*SIDE2(I)*SIDE3(I)))
      ANGLE(I)=AMIN1(ANGL1(I),ANGL2(I),ANGL3(I))
      WHT  (I)=(TAN(ANGLE(I)))**2/AREA1(I)
 2899 CONTINUE
      DO 2900  I= 1,35
      WRITE (6,121) FI1(I),FJ1(I),FI2(I),FJ2(I),FI3(I),FJ3(I),
     $CRX(I),CRY(I),I
  121 FORMAT (' ',T6,F6.2,F6.2,T26,F6.2,F6.2,T46,F6.2,F6.2,T68,
     $F6.2,F6.2,T93,I2)
 2900 CONTINUE
      WRITE (6,122)
  122 FORMAT ('1',T4,'TRIANGLE',T21,'SIDE1',T31,'SIDE2',T41,'SIDE3',
     $T60,'AREA',T80,'WEIGHT')
      DO 2901  I= 1,35
      WRITE (6,133) I,SIDE1(I),SIDE2(I),SIDE3(I),AREA1(I),WHT(I)
  133 FORMAT (' ',T7,I2,T20,F6.2,T30,F6.2,T40,F6.2,T59,F6.2,T79,F6.2)
 2901 CONTINUE
      DO 777  I=1,35
      UWND1          =(DTU1(I)+DTU2(I))/2
      VWND1          =(DTV1(I)+DTV2(I))/2
      CALL DIRSPD (UWND1,VWND1,SPD,DIR)
```

```
      SPD1 (I)        =SPD
      DIR1 (I)        =DIR
      UWND2           =(DTU2(I)+DTU3(I))/2
      VWND2           =(DTV2(I)+DTV3(I))/2
      CALL DIRSPD (UWND2,VWND2,SPD,DIR)
      SPD2 (I)        =SPD
      DIR2 (I)        =DIR
      UWND3           =(DTU3(I)+DTU1(I))/2
      VWND3           =(DTV3(I)+DTV1(I))/2
      CALL DIRSPD (UWND3,VWND3,SPD,DIR)
      SPD3 (I)        =SPD
      DIR3 (I)        =DIR
  777 CONTINUE
      WRITE (6,778)
  778 FORMAT ('1',T5,'TRIANGLE',T20,'WIND AT SIDE1',T40,
     $'WIND AT SIDE2',T60,'WIND AT SIDE3')
      WRITE (6,779)
  779 FORMAT (' ',T21,'DIR',T29,'SPD',T41,'DIR',T49,'SPD',
     &T61,'DIR',T69,'SPD')
      WRITE (6,780)
  780 FORMAT ('0')
      DO 7810  I=1,35
      WRITE (6,781) I,DIR1(I),SPD1(I),DIR2(I),SPD2(I),DIR3(I),SPD3(I)
  781 FORMAT (' ',T9,I2,T20,F5.1,T28,F5.1,T40,F5.1,T48,F5.1,T60,F5.1,
     &T68,F5.1)
 7810 CONTINUE
      WRITE (6,881)
  881 FORMAT ('1',T5,'TRIANGLE',T20,'ANGLE OF SIDE1',T40,
     %'ANGLE OF SIDE2',T60,'ANGLE OF SIDE3')
      DO 7811  I=1,35
      ANG1 (I)=ATAN2(FJ1(I)-FJ2(I),FI2(I)-FI1(I))*57.2958
      IF (ANG1(I) .LT. -90.0  .OR.  ANG1(I) .GT. 90.0) GO TO 893
      ANG1 (I)=90.0-ANG1(I)
  793 IF (ANG1(I) .GT. 360.0) ANG1(I)=ANG1(I)-360.0
      ANG2 (I)=ATAN2(FJ2(I)-FJ3(I),FI3(I)-FI2(I))*57.2958
      IF (ANG2(I) .LT. -90.0  .OR.  ANG2(I) .GT. 90.0) GO TO 894
      ANG2 (I)=90.0-ANG2(I)
  794 IF (ANG2(I) .GT. 360.0) ANG2(I)=ANG2(I)-360.0
      ANG3 (I)=ATAN2(FJ3(I)-FJ1(I),FI1(I)-FI3(I))*57.2958
      IF (ANG3(I) .LT. -90.0  .OR.  ANG3(I) .GT. 90.0) GO TO 895
      ANG3 (I)=90.0-ANG3(I)
  795 IF (ANG3(I) .GT. 360.0) ANG3(I)=ANG3(I)-360.0
      GO TO 7811
  893 ANG1 (I)=270.0-ANG1(I)
      GO TO 793
  894 ANG2 (I)=270.0-ANG2(I)
      GO TO 794
  895 ANG3 (I)=270.0-ANG3(I)
      GO TO 795
 7811 CONTINUE
      DO 7812  I=1,35
      WRITE (6,882) I,ANG1(I),ANG2(I),ANG3(I)
  882 FORMAT (' ',T9,I2,T25,F5.1,T45,F5.1,T65,F5.1)
 7812 CONTINUE
      DO 7813  I=1,35
C IFLAG = 2 SUGGESTS THAT THE TRIANGLE IS BELOW ITS LEG
      IFLAG = 2
      IF (FI1(I) .LT. FI2(I)) IFLAG = 1
      IF (FI1(I) .EQ. FI2(I)  .AND.  FJ1(I) .GT. FJ2(I)) IFLAG=1
      IF (FI1(I) .EQ. FI2(I)  .AND.  FJ1(I) .LT. FJ2(I)) IFLAG=2
```

```
      CALL SIGN (ANG1(I),DIR1(I),IFLAG,SPD1(I),VT,VN)
      VT1  (I)=VT
      VN1  (I)=VN
      IFLAG = 1
      IF (FI2(I) .GT. FI3(I)) IFLAG=2
      CALL SIGN (ANG2(I),DIR2(I),IFLAG,SPD2(I),VT,VN)
      VT2  (I)=VT
      VN2  (I)=VN
      IFLAG = 2
      IF (FI3(I) .LT. FI1(I)) IFLAG=1
      CALL SIGN (ANG3(I),DIR3(I),IFLAG,SPD3(I),VT,VN)
      VT3  (I)=VT
      VN3  (I)=VN
 7813 CONTINUE
      WRITE (6,409)
  409 FORMAT ('1',T5,'TRIANGLE',T20,'WIND AT SIDE1',T40,
     &'WIND AT SIDE2',T60,'WIND AT SIDE3')
      WRITE (6,408)
  408 FORMAT (' ',T21,'TAN',T29,'NOR',T41,'TAN',T49,'NOR',
     &T61,'TAN',T69,'NOR')
      WRITE (6,407)
  407 FORMAT ('0')
      DO 7713  I=1,35
      WRITE (6,406) I,VT1(I),VN1(I),VT2(I),VN2(I),VT3(I),VN3(I)
  406 FORMAT (' ',T9,I2,T20,F5.1,T28,F5.1,T40,F5.1,T48,F5.1,
     &T60,F5.1,T68,F5.1)
 7713 CONTINUE
      WRITE (6,123)
  123 FORMAT ('1',T5,'TRIANGLE',T40,'DIVERGENCE',T80,'VORTICITY')
      DO 8022  I=1,35
      SIDE1(I)=SIDE1(I)*15800
      SIDE2(I)=SIDE2(I)*15800
      SIDE3(I)=SIDE3(I)*15800
      DIV  (I)=((VN1(I)*SIDE1(I))+(VN2(I)*SIDE2(I))+(VN3(I)*
     $SIDE3(I)))/(AREA1(I)*15800**2)
      VORT (I)=((VT1(I)*SIDE1(I))+(VT2(I)*SIDE2(I))+(VT3(I)*
     $SIDE3(I)))/(AREA1(I)*15800**2)
      WRITE (6,102) I,DIV(I),VORT(I)
  102 FORMAT (' ',T9,I2,T36,E14.7,T75,E14.7)
 8022 CONTINUE
      IPTS=0
      IR=1
      DO 7292  I=1,35
      IF (WHT(I) .LT. 0.05) GO TO 7292
      SCANR(IR)       =1.0
      DATA (IR)       =DIV  (I)
      DATA1(IR)       =VORT (I)
      FFI  (IR)       =CRX  (I)
      FFJ  (IR)       =CRY  (I)
      IPTS=IPTS+1
      IR=IR+1
 7292 CONTINUE
        CALL GUESS  (A ,0.0)
        CALL GUESS  (A1,0.0)
        CALL ANAL   (IPTS ,SCANR,NOSCAN,2)
        CALL SMOOTH (A ,B,SMO)
      WRITE(6,515) SLVL(IL),IIMON,IIDAY,IITIME
  515 FORMAT('1',10X,'DIVERGENCE AT S LEVEL ',F4.2,' ON '
     %,I2,'/',I4,3X,'AT',3X,I4,' GMT')
        CALL SMOOTH (A1,B,SMO)
```

```
      CALL OUTPUT (A ,M1,M2,N1,N2,1.0E06,1)
      WRITE (6,516) SLVL(IL),IIMON,IIDAY,IITIME
  516 FORMAT('1',10X,'VORTICITY AT S LEVEL ',F4.2,' ON '
     #,I2,'/',I4,3X,'AT',3X,I4,' GMT')
      CALL OUTPUT (A1,M1,M2,N1,N2,1.0E06,1)
 6000 CONTINUE
      STOP
      END

      SUBROUTINE RATIO (ZS,E,LST,I,J,T,VLE)
      INTEGER T,IFLAG
      DIMENSION ZS( 7,8),E(50,16,16)
      VLE=E(J-1,T,LST)+(ZS(LST,I)-E(J-1,3,LST))/(E(J,3,LST)-
     $E(J-1,3,LST))*(E(J,T,LST)-E(J-1,T,LST))
      RETURN
      END

      SUBROUTINE ANAL (NRPTS,SCANR,NOSCAN,IGO)
C     THIS ROUTINE ANALYZES THE REPORTED DATA AND PUTS THE ADJUSTED
C     VALUES AT GRID POINTS.
      DIMENSION TC1(18,18),          SCANR(35)
     1,         TW1(18,18),FFI  (35)
     2,         TC (18,18),FFJ  (35)
     3,         TW (18,18),DATA1(35)
     4,         A1 (18,18),DATA (35)
     5,         A  (18,18)
      COMMON DATA,DATA1,FFI,FFJ,A,A1,TC,TC1,TW,TW1
      COMMON /GRIDIJ/ ISTART,JSTART,IEND,JEND
      DO 400  NSCAN=1,NOSCAN
      DO 401  J=JSTART,JEND
      DO 401  I=ISTART,IEND
      TC (I,J)                =0.0
      TC1(I,J)                =0.0
      TW1(I,J)                =0.0
  401 TW (I,J)                =0.0
      NCNT                    =0
      DO 402  K=1,NRPTS
      M = 1
      N = 1
      IF (DATA (K) .EQ. 0.0)   M=2
      IF (DATA1(K) .EQ. 0.0)   N=2
      IF (M .EQ. 2 .AND. N .EQ. 2)  GO TO 402
      II=FFI(K)
      JJ=FFJ(K)
      RMAX                                    =SCANR(K)
      IF (NRPTS .NE. 16) GO TO 31
      IF (NSCAN .EQ. 1)                    RMAX=18.0
      IF (NSCAN .EQ. 2)                    RMAX=10.0
      IF (NSCAN .EQ. 3)                    RMAX= 5.0
      GO TO 32
   31 IF (NSCAN .EQ. 1)  RMAX=10.0
      IF (NSCAN .EQ. 2)  RMAX= 5.0
      IF (NSCAN .EQ. 3)  RMAX= 2.0
   32 RMSQ = RMAX**2
      NCNT = NCNT + 1
C
      GO TO (40,41),M
   40   CALL INTRP  (A ,FFI(K),FFJ(K),DINT)
      ERROR =DATA (K)-DINT
   41 CONTINUE
```

```
      GO TO (42,43),N
   42   CALL INTRP  (A1,FFI(K),FFJ(K),DINT1)
      ERROR1=DATA1(K)-DINT1
   43 CONTINUE
C
      I1=FFI(K) - RMAX+0.5
      I2=FFI(K) + RMAX+0.5
      J1=FFJ(K) - RMAX+0.5
      J2=FFJ(K) + RMAX+0.5
      IMIN = MAX0(ISTART,I1)
      IMAX = MIN0(IEND,I2)
      JMIN = MAX0(JSTART,J1)
      JMAX = MIN0(JEND,J2)
C
      DO 403  J=JMIN,JMAX
      DO 404  I=IMIN,IMAX
      RSQ = (FLOAT(I)-FFI(K))**2 + (FLOAT(J) - FFJ(K))**2
      IF (RSQ .GE. RMSQ) GO TO 404
      WGT = EXP(-4.*RSQ/RMSQ)
      GO TO (44,45),M
   44 TC (I,J)=TC (I,J)+WGT*ERROR
      TW (I,J)=TW (I,J)+WGT
   45 GO TO (46,404),N
   46 CONTINUE
      IF (!GO .EQ. 1)  GO TO 404
      TC1(I,J)=TC1(I,J)+WGT*ERROR1
      TW1(I,J)=TW1(I,J)+WGT
  404 CONTINUE
  403 CONTINUE
C     STATION  OUTSIDE OF THE GRID WON'T BE CONSIDERED.
  402 CONTINUE
      DO 405  J=JSTART,JEND
      DO 405  I=ISTART,IEND
      IF (TW(I,J) .LE. 0.0) GO TO 47
      A (I,J) = A (I,J) + TC (I,J)/TW (I,J)
   47 IF ((GO .EQ. 1)                GO TO 405
      IF (TW1(I,J) .LE. 0.0)         GO TO 405
      A1(I,J) = A1(I,J) + TC1(I,J)/TW1(I,J)
  405 CONTINUE
  400 CONTINUE
      RETURN
      END

      SUBROUTINE DECGRID (DLAT,DLON,FI,FJ)
      COMMON /GRIDJK/ SLAT,SLON,M1,N1,GO,M2,N2
C     LATITUDE AND LONGITUDE OF THE NORTHWEST CORNER OF THE
C     GRID IS 35.60N AND 105.50W.  GRID SPACING IS 40 KILOMETERS
      CNST = .2679492
      COST = .7155668
      CST  = 57.29578
      FAC1 =4450.4231/GO
      PHI  =(90-SLAT)/CST
      FAC2 =FAC1*(TAN(PHI/2.0)/CNST)**COST
      ALAT =(90.-DLAT)/CST
      R    =FAC1*(TAN(ALAT/2.0)/CNST)**COST
      ALON =(SLON-DLON)*COST/CST
      FI   = M1+R*SIN(ALON)
      FJ   = N1+R*COS(ALON)-FAC2
      RETURN
      END
```

```
      SUBROUTINE INTRP (A,FFI,FFJ,DINT)
      DIMENSION A(18,18)
      II    = FFI
      JJ    = FFJ
      DI    = FFI-FLOAT(II)
      DJ    = FFJ-FLOAT(JJ)
      Z1    = A(II,JJ)
      Z2    = A(II+1,JJ)
      Z3    = A(II,JJ+1)
      Z4    = A(II+1,JJ+1)
      Z5    =Z1+(Z2-Z1)*DI
      Z6    =Z3+(Z4-Z3)*DI
      DINT =Z5+(Z6-Z5)*DJ
      RETURN
      END

      SUBROUTINE SMOOTH (A,B,C)
C     A GOES IN UNSMOOTHED, A COMES OUT SMOOTHED, B IS ORIGINAL A
      DIMENSION A(18,18),B(18,18)
      COMMON /GRIDIJ/ ISTART,JSTART,IEND,JEND
C       SMOOTH INTERIOR POINTS
      J1 = JSTART + 1
      J2 = JEND - 1
      I1 = ISTART + 1
      I2 = IEND - 1
      C2 = C*(1.-C)/2.
      C3 = C*C/4.
      C4 = 1. - 4.*C2 - 4.*C3
      DO 100  J=J1,J2
      DO 100  I=I1,I2
      B(I,J) = C4*A(I,J)+C2*(A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1))
     *          +C3*(A(I-1,J-1)+A(I-1,J+1)+A(I+1,J+1)+A(I+1,J-1))
  100 CONTINUE
C     SMOOTH BORDER POINTS
      C1 = (1.0 - C)/2.0
      K  = JSTART
      L  = JEND
      M  = ISTART
      N  = IEND
      DO 101  I=I1,I2
      B(I,K) = A(I,K)*C + C1*(A(I-1,K) + A(I+1,K))
  101 B(I,L) = A(I,L)*C + C1*(A(I-1,L) + A(I+1,L))
      DO 102  J=J1,J2
      B(M,J) = A(M,J)*C + C1*(A(M,J-1) + A(M,J+1))
  102 B(N,J) = A(N,J)*C + C1*(A(N,J-1) + A(N,J+1))
C       SMOOTH CORNERS
      B(M,K) = A(M,K)*C + C1*(A(M,K+1) + A(M+1,K))
      B(N,K) = A(N,K)*C + C1*(A(N-1,K) + A(N,K+1))
      B(M,L) = A(M,L)*C + C1*(A(M,L-1) + A(M+1,L))
      B(N,L) = A(N,L)*C + C1*(A(N-1,L) + A(N,L-1))
        CALL MOVE    (B,A)
      RETURN
      END

      SUBROUTINE GUESS (A,VAL)
      COMMON /GRIDIJ/ ISTART,JSTART,IEND,JEND
C     INITIALIZE THE BARNES FIRST GUESS BY READING
C     IN VAL AND SETTING A=VAL
      REAL A(18,18)
      DO 1 I=ISTART,IEND
```

```
      DO 1 J=JSTART,JEND
      A(I,J)=VAL
    1 CONTINUE
      RETURN
      END

      SUBROUTINE OUTPUT (A,I1,IMAX,J1,JMAX,SCALE,IJK)
      DIMENSION A(18,18),PLINE(40)
      IIL   =I1-1
      IIMAX=IMAX-I1+1
      WRITE (6,100) I1,IMAX,J1,JMAX,SCALE
  100 FORMAT(' ','I1 = ',I2,' IMAX = ',I2,' J1 = ',I2,' JMAX = ',I2,' SC
     1ALE FACTOR = ',E10.2)
      DO 101  J= 3,16
      DO 102  I= 1,18
      PLINE(I)=0.0
      IF (J .LT. J1 .OR. J .GT. JMAX)  GO TO 102
      IF (I .LT. I1 .OR. I .GT. IMAX)  GO TO 102
      PLINE(I-IIL)=A(I,J)*SCALE
  102 CONTINUE
      IF (IJK .EQ. 0)  GO TO 12
      WRITE (6,10) (PLINE(I),I=1,IIMAX)
   10 FORMAT(1H0,10X,18F6.0)
      WRITE (6,199)
  199 FORMAT (1H0)
      GO TO 101
   12 WRITE (6,11) (PLINE(I),I=1,IIMAX)
   11 FORMAT('  ',21F6.1)
  101 CONTINUE
      RETURN
      END

      SUBROUTINE MOVE(A,B)
      COMMON /GRIDIJ/ ISTART,JSTART,IEND,JEND
      DIMENSION A(18,18),B(18,18)
      DO 1 J=JSTART,JEND
      DO 1 I=ISTART,IEND
    1 B(I,J)=A(I,J)
      RETURN
      END

      SUBROUTINE METH1   (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,
      DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      DIMENSION  FFI   (35),FI1   (35),FJ1   (35),DTU1 (35),DTV1 (35)
     1,          FFJ   (35),FI2   (35),FJ2   (35),DTU2 (35),DTV2 (35)
     2,          DATA (35),FI3   (35),FJ3   (35),DTU3 (35),DTV3 (35)
     3,          DATA1(35)
          DO 1    L= 1, 5
      FI1   (L)=FFI   (1)
      FJ1   (L)=FFJ   (1)
      DTU1 (L)=DATA (1)
      DTV1 (L)=DATA1(1)
    1     CONTINUE
          DO 2    L= 6, 9
      FI1   (L)=FFI   (2)
      FJ1   (L)=FFJ   (2)
      DTU1 (L)=DATA (2)
      DTV1 (L)=DATA1(2)
    2     CONTINUE
          DO 3    L=10,12
```

```
      FI1  (L)=FFI  (3)
      FJ1  (L)=FFJ  (3)
      DTU1 (L)=DATA (3)
      DTV1 (L)=DATA1(3)
3     CONTINUE
      DO 4    L=13,14
      FI1  (L)=FFI  (4)
      FJ1  (L)=FFJ  (4)
      DTU1 (L)=DATA (4)
      DTV1 (L)=DATA1(4)
4     CONTINUE
      FI1 (15)=FFI  (5)
      FJ1 (15)=FFJ  (5)
      DTU1(15)=DATA (5)
      DTV1(15)=DATA1(5)
      DO 5    L= 2, 6
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
5     CONTINUE
      DO 6    L= 3, 6
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
6     CONTINUE
      DO 7    L= 4, 6
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
7     CONTINUE
      DO 8    L= 5, 6
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
8     CONTINUE
      FI2 (L1)=FFI  (6)
      FJ2 (L1)=FFJ  (6)
      DTU2(L1)=DATA (6)
      DTV2(L1)=DATA1(6)
      L1      =L1+1
      DO 9    L= 3, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
9     CONTINUE
      DO 10   L= 4, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
```

```
      K1       =K1+1
10    CONTINUE
      DO 11  L= 5, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1       =K1+1
11    CONTINUE
      DO 12  L= 6, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1       =K1+1
12    CONTINUE
      FI3 (K1)=FFI  (7)
      FJ3 (K1)=FFJ  (7)
      DTU3(K1)=DATA (7)
      DTV3(K1)=DATA1(7)
      K1       =K1+1
      RETURN
      END

      SUBROUTINE METH2   (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,
      DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      DIMENSION  FFI  (35),FI1  (35),FJ1  (35),DTU1 (35),DTV1 (35)
     1,          FFJ  (35),FI2  (35),FJ2  (35),DTU2 (35),DTV2 (35)
     2,          DATA (35),FI3  (35),FJ3  (35),DTU3 (35),DTV3 (35)
     3,          DATA1(35)
      DO 1   L=16,18
      FI1  (L)=FFI  (1)
      FJ1  (L)=FFJ  (1)
      DTU1 (L)=DATA (1)
      DTV1 (L)=DATA1(1)
1     CONTINUE
      DO 2   L=19,20
      FI1  (L)=FFI  (2)
      FJ1  (L)=FFJ  (2)
      DTU1 (L)=DATA (2)
      DTV1 (L)=DATA1(2)
2     CONTINUE
      FI1 (21)=FFI  (3)
      FJ1 (21)=FFJ  (3)
      DTU1(21)=DATA (3)
      DTV1(21)=DATA1(3)
      DO 3   L= 3, 5
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1       =L1+1
3     CONTINUE
      DO 4   L= 4, 5
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1       =L1+1
4     CONTINUE
```

```
      FI2 (L1)=FFI   (5)
      FJ2 (L1)=FFJ   (5)
      DTU2(L1)=DATA (5)
      DTV2(L1)=DATA1(5)
      L1      =L1+1
          DO 5    L= 5, 7
      FI3 (K1)=FFI   (L)
      FJ3 (K1)=FFJ   (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
    5 CONTINUE
          DO 6    L= 6, 7
      FI3 (K1)=FFI   (L)
      FJ3 (K1)=FFJ   (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
    6 CONTINUE
      FI3 (K1)=FFI   (7)
      FJ3 (K1)=FFJ   (7)
      DTU3(K1)=DATA (7)
      DTV3(K1)=DATA1(7)
      K1      =K1+1
      RETURN
      END

      SUBROUTINE METH3   (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,
      DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      DIMENSION  FFI  (35),FI1  (35),FJ1  (35),DTU1 (35),DTV1 (35)
     1,          FFJ  (35),FI2  (35),FJ2  (35),DTU2 (35),DTV2 (35)
     2,          DATA (35),FI3  (35),FJ3  (35),DTU3 (35),DTV3 (35)
     3,          DATA1(35)
          DO 1    L=22,24
      FI1  (L)=FFI   (1)
      FJ1  (L)=FFJ   (1)
      DTU1 (L)=DATA (1)
      DTV1 (L)=DATA1(1)
    1 CONTINUE
          DO 2    L=25,26
      FI1  (L)=FFI   (2)
      FJ1  (L)=FFJ   (2)
      DTU1 (L)=DATA (2)
      DTV1 (L)=DATA1(2)
    2 CONTINUE
      FI1 (27)=FFI   (3)
      FJ1 (27)=FFJ   (3)
      DTU1(27)=DATA (3)
      DTV1(27)=DATA1(3)
          DO 3    L= 2, 4
      FI2 (L1)=FFI   (L)
      FJ2 (L1)=FFJ   (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
    3 CONTINUE
          DO 4    L= 3, 4
      FI2 (L1)=FFI   (L)
      FJ2 (L1)=FFJ   (L)
      DTU2(L1)=DATA (L)
```

```
      DTV2(L1)=DATA1(L)
      L1      =L1+1
    4 CONTINUE
      FI2 (L1)=FFI  (4)
      FJ2 (L1)=FFJ  (4)
      DTU2(L1)=DATA (4)
      DTV2(L1)=DATA1(4)
      L1      =L1+1
      DO 5   L= 5, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
    5 CONTINUE
      DO 6   L= 6, 7
      FI3 (K1)=FFI  (L)
      FJ3 (K1)=FFJ  (L)
      DTU3(K1)=DATA (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
    6 CONTINUE
      FI3 (K1)=FFI  (7)
      FJ3 (K1)=FFJ  (7)
      DTU3(K1)=DATA (7)
      DTV3(K1)=DATA1(7)
      K1      =K1+1
      RETURN
      END

      SUBROUTINE METHA   (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,
      DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      DIMENSION   FFI   (35),FI1   (35),FJ1   (35),DTU1 (35),DTV1 (35)
     1,           FFJ   (35),FI2   (35),FJ2   (35),DTU2 (35),DTV2 (35)
     2,           DATA  (35),FI3   (35),FJ3   (35),DTU3 (35),DTV3 (35)
     3,           DATA1(35)
      DO 1   L=28,29
      FI1  (L)=FFI  (1)
      FJ1  (L)=FFJ  (1)
      DTU1 (L)=DATA (1)
      DTV1 (L)=DATA1(1)
    1 CONTINUE
      FI1 (30)=FFI  (2)
      FJ1 (30)=FFJ  (2)
      DTU1(30)=DATA (2)
      DTV1(30)=DATA1(2)
      DO 2   L= 2, 3
      FI2 (L1)=FFI  (L)
      FJ2 (L1)=FFJ  (L)
      DTU2(L1)=DATA (L)
      DTV2(L1)=DATA1(L)
      L1      =L1+1
    2 CONTINUE
      FI2 (L1)=FFI  (3)
      FJ2 (L1)=FFJ  (3)
      DTU2(L1)=DATA (3)
      DTV2(L1)=DATA1(3)
      L1      =L1+1
      DO 3   L= 6, 7
      FI3 (K1)=FFI  (L)
```

```
      FJ3 (K1)=FFJ   (L)
      DTU3(K1)=DATA  (L)
      DTV3(K1)=DATA1(L)
      K1      =K1+1
3     CONTINUE
      FI3 (K1)=FFI   (7)
      FJ3 (K1)=FFJ   (7)
      DTU3(K1)=DATA  (7)
      DTV3(K1)=DATA1(7)
      K1      =K1+1
      RETURN
      END

      SUBROUTINE METHS   (L1,K1,FFI,FFJ,DATA,DATA1,FI1,FJ1,
     DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,FJ3,DTU3,DTV3)
      DIMENSION  FFI   (35),FI1   (35),FJ1   (35),DTU1 (35),DTV1 (35)
     1,          FFJ   (35),FI2   (35),FJ2   (35),DTU2 (35),DTV2 (35)
     2,          DATA (35),FI3   (35),FJ3   (35),DTU3 (35),DTV3 (35)
     3,          DATA1(35)
          DO 1    L= 1, 4
      FI1 (L1)=FFI   (L)
      FJ1 (L1)=FFJ   (L)
      DTU1(L1)=DATA  (L)
      DTV1(L1)=DATA1(L)
      L1      =L1+1
1     CONTINUE
          DO 2    L= 2, 5
      FI2 (K1)=FFI   (L)
      FJ2 (K1)=FFJ   (L)
      DTU2(K1)=DATA  (L)
      DTV2(K1)=DATA1(L)
      K1      =K1+1
2     CONTINUE
      M1      =31
          DO 3    L= 4, 7
      FI3 (M1)=FFI   (L)
      FJ3 (M1)=FFJ   (L)
      DTU3(M1)=DATA  (L)
      DTV3(M1)=DATA1(L)
      M1      =M1+1
3     CONTINUE
      FI1 (35)=FFI   (1)
      FJ1 (35)=FFJ   (1)
      DTU1(35)=DATA  (1)
      DTV1(35)=DATA1(1)
      FI2 (35)=FFI   (2)
      FJ2 (35)=FFJ   (2)
      DTU2(35)=DATA  (2)
      DTV2(35)=DATA1(2)
      FI3 (35)=FFI   (7)
      FJ3 (35)=FFJ   (7)
      DTU3(35)=DATA  (7)
      DTV3(35)=DATA1(7)
      RETURN
      END

      SUBROUTINE SORT (FI1,FJ1,DTU1,DTV1,FI2,FJ2,DTU2,DTV2,FI3,
     &FJ3,DTU3,DTV3)
      DIMENSION  FI1   (35),FJ1   (35),DTU1 (35),DTV1 (35)
     1,          FI2   (35),FJ2   (35),DTU2 (35),DTV2 (35)
```

```
     2.           FI3  (35),FJ3  (35),DTU3 (35),DTV3 (35)
          DO 6    I= 1,35
          IF (I .EQ. 14  .CR.  I .EQ. 31) GO TO 6
          IF (FI3(I) .GT. FI2(I)  .AND.  FI1(I) .GT. FI2(I)) GO TO 1
          IF (FI3(I) .GT. FI2(I)  .AND.  FI1(I) .LT. FI2(I)) GO TO 2
          IF (FI3(I) .LT. FI2(I)  .AND.  FI1(I) .LT. FI3(I)) GO TO 3
          IF (FI3(I) .LT. FI2(I)  .AND.  FI1(I) .GT. FI3(I)) GO TO 4
          GO TO 6
        1 IF (FJ3(I) .LT. FJ2(I)  .AND.  FJ1(I) .GT. FJ2(I)) GO TO 5
          IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .GT. FJ3(I)) GO TO 5
          IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .GT. FJ2(I)) GO TO 5
          GO TO 6
        2 IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .GT. FJ3(I)) GO TO 5
          IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .LT. FJ3(I)) GO TO 5
          IF (FJ3(I) .LT. FJ2(I)  .AND.  FJ1(I) .GT. FJ2(I)) GO TO 5
          GO TO 6
        3 IF (FJ3(I) .LT. FJ2(I)  .AND.  FJ1(I) .GT. FJ2(I)) GO TO 5
          IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .LT. FJ2(I)) GO TO 5
          IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .GT. FJ2(I)) GO TO 5
          GO TO 6
        4 IF (FJ3(I) .GT. FJ2(I)  .AND.  FJ1(I) .LT. FJ2(I)) GO TO 5
          GO TO 6
        5 A          =FI2  (I)
          B          =FJ2  (I)
          C          =DTU2 (I)
          D          =DTV2 (I)
          A1         =FI3  (I)
          B1         =FJ3  (I)
          C1         =DTU3 (I)
          D1         =DTV3 (I)
          FI3  (I) =A
          FJ3  (I) =B
          DTU3 (I) =C
          DTV3 (I) =D
          FI2  (I) =A1
          FJ2  (I) =B1
          DTU2 (I) =C1
          DTV2 (I) =D1
        6     CONTINUE
          RETURN
          END

          SUBROUTINE DIRSPD (U,V,SPD,DIR)
     C THE ANGLES ARE DETERMINED FROM CARTESIAN NORTH
          IF (U .EQ. 0.0  .AND.  V .EQ. 0.0) GO TO 2
          SPD  =SQRT(U**2+V**2)
          IF (U .EQ. 0.0)                    GO TO 3
          IF (V .EQ. 0.0)                    GO TO 4
          GO TO 5
        2 DIR  =0.0
          GO TO 6
        3 DIR  =360.0
          IF (V .LT. 0.0)                    DIR=180.0
          GO TO 6
        4 DIR  =90.0
          IF (U .LT. 0.0)                    DIR=270.0
          GO TO 6
        5 DIR  =ATAN2(V,U)*57.2958
          DIR  =90.0-DIR
          IF (DIR .GT. 180.0)                DIR=DIR-180.0
```

```
    6 RETURN
      END

      SUBROUTINE SIGN (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (CHI .LT. 90.0  .AND.  PHI .EQ.  0.0) GO TO 3
      IF (CHI .LT. 90.0  .AND.  PHI .LT. 90.0) GO TO 4
      IF (CHI .LT. 90.0  .AND.  PHI .EQ. 90.0) GO TO 1
      IF (CHI .LT. 90.0  .AND.  PHI .GT. 90.0) GO TO 2
C
      IF (CHI .GT. 90.0  .AND.  PHI .EQ.  0.0) GO TO 7
      IF (CHI .GT. 90.0  .AND.  PHI .LT. 90.0) GO TO 8
      IF (CHI .GT. 90.0  .AND.  PHI .EQ. 90.0) GO TO 5
      IF (CHI .GT. 90.0  .AND.  PHI .GT. 90.0) GO TO 6
C
    1 CALL SIGN1 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    2 CALL SIGN2 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    3 CALL SIGN3 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    4 CALL SIGN4 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    5 CALL SIGN5 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    6 CALL SIGN6 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    7 CALL SIGN7 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
    8 CALL SIGN8 (PHI,CHI,IFLAG,SPD,VT,VN)
      RETURN
      END

      SUBROUTINE SIGN1 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (IFLAG .EQ. 2) GO TO 1
      VT    = SPD*SIN(CHI/57.296)
      VN    =-SPD*COS(CHI/57.296)
      RETURN
    1 VT    =-SPD*SIN(CHI/57.296)
      VN    = SPD*COS(CHI/57.296)
      RETURN
      END

      SUBROUTINE SIGN2 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF ((PHI-CHI) .LT. 90.0) GO TO 1
      IF ((PHI-CHI) .GT. 90.0) GO TO 2
      WRITE (6,100)
  100 FORMAT ('0','SOMETHING IS WRONG2')
      RETURN
    1 IF (IFLAG .EQ. 2) GO TO 10
      VT    = SPD*COS((PHI-CHI)/57.296)
      VN    =-SPD*SIN((PHI-CHI)/57.296)
      RETURN
   10 VT    =-SPD*COS((PHI-CHI)/57.296)
      VN    = SPD*SIN((PHI-CHI)/57.296)
      RETURN
    2 IF (IFLAG .EQ. 2) GO TO 20
      VT    =-SPD*SIN((PHI-CHI-90.0)/57.296)
      VN    =-SPD*COS((PHI-CHI-90.0)/57.296)
      RETURN
   20 VT    = SPD*SIN((PHI-CHI-90.0)/57.296)
```

```
      VN    = SPD*COS((PHI-CHI-90.0)/57.296)
      RETURN
      END

      SUBROUTINE SIGN3 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (IFLAG .EQ. 2) GO TO 1
      VT    = SPD*COS(CHI/57.296)
      VN    = SPD*SIN(CHI/57.296)
      RETURN
    1 VT    =-SPD*COS(CHI/57.296)
      VN    =-SPD*SIN(CHI/57.296)
      RETURN
      END

      SUBROUTINE SIGN4 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (CHI .GT. PHI) GO TO 1
      IF (CHI .LT. PHI) GO TO 2
      WRITE (6,100)
  100 FORMAT ('0','SOMETHING IS WRONG4')
      RETURN
    1 IF (IFLAG .EQ. 2) GO TO 10
      VT    = SPD*COS((CHI-PHI)/57.296)
      VN    = SPD*SIN((CHI-PHI)/57.296)
      RETURN
   10 VT    =-SPD*COS((CHI-PHI)/57.296)
      VN    =-SPD*SIN((CHI-PHI)/57.296)
      RETURN
    2 IF (IFLAG .EQ. 2) GO TO 20
      VT    = SPD*COS((PHI-CHI)/57.296)
      VN    =-SPD*SIN((PHI-CHI)/57.296)
      RETURN
   20 VT    =-SPD*COS((PHI-CHI)/57.296)
      VN    = SPD*SIN((PHI-CHI)/57.296)
      RETURN
      END

      SUBROUTINE SIGN5 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (IFLAG .EQ. 2) GO TO 1
      VT    = SPD*COS((CHI-PHI)/57.296)
      VN    = SPD*SIN((CHI-PHI)/57.296)
      RETURN
    1 VT    =-SPD*COS((CHI-PHI)/57.296)
      VN    =-SPD*SIN((CHI-PHI)/57.296)
      RETURN
      END

      SUBROUTINE SIGN6 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (CHI .GT. PHI) GO TO 1
      IF (CHI .LT. PHI) GO TO 2
      WRITE (6,100)
  100 FORMAT ('0','SOMETHING IS WRONG6')
      RETURN
    1 IF (IFLAG .EQ. 2) GO TO 10
      VT    = SPD*COS((CHI-PHI)/57.296)
      VN    = SPD*SIN((CHI-PHI)/57.296)
      RETURN
   10 VT    =-SPD*COS((CHI-PHI)/57.296)
      VN    =-SPD*SIN((CHI-PHI)/57.296)
      RETURN
    2 IF (IFLAG .EQ. 2) GO TO 20
```

```
      VT    = SPD*COS((PHI-CHI)/57.296)
      VN    =-SPD*SIN((PHI-CHI)/57.296)
      RETURN
   20 VT    =-SPD*COS((PHI-CHI)/57.296)
      VN    = SPD*SIN((PHI-CHI)/57.296)
      RETURN
      END

      SUBROUTINE SIGN7 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF (IFLAG .EQ. 2) GO TO 1
      VT    =-SPD*SIN((CHI-90.0)/57.296)
      VN    = SPD*COS((CHI-90.0)/57.296)
      RETURN
    1 VT    = SPD*SIN((CHI-90.0)/57.296)
      VN    =-SPD*COS((CHI-90.0)/57.296)
      RETURN
      END

      SUBROUTINE SIGN8 (PHI,CHI,IFLAG,SPD,VT,VN)
      IF ((CHI-PHI) .LT. 90.0) GO TO 1
      IF ((CHI-PHI) .GT. 90.0) GO TO 2
      WRITE (6,100)
  100 FORMAT ('0','SOMETHING IS WRONG')
      RETURN
    1 IF (IFLAG .EQ. 2) GO TO 10
      VT    = SPD*COS((CHI-PHI)/57.296)
      VN    = SPD*SIN((CHI-PHI)/57.296)
      RETURN
   10 VT    =-SPD*COS((CHI-PHI)/57.296)
      VN    =-SPD*SIN((CHI-PHI)/57.296)
      RETURN
    2 IF (IFLAG .EQ. 2) GO TO 20
      VT    =-SPD*SIN((CHI-PHI-90.0)/57.296)
      VN    = SPD*COS((CHI-PHI-90.0)/57.296)
      RETURN
   20 VT    = SPD*SIN((CHI-PHI-90.0)/57.296)
      VN    =-SPD*COS((CHI-PHI-90.0)/57.296)
      RETURN
      END

//GO.SYSIN DD *
```