3-1997

Development and Implementation of an Isentropic Potential Vorticity Algorithm for use at Air Force Global Weather Center

Jay B. DesJardins Jr.

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

THESIS

Jay B DesJardins, Jr., Capt, USAF

AFIT/GM/ENP/97M-3

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GM/ENP/97M-3

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

THESIS

Jay B DesJardins, Jr., Capt, USAF

AFIT/GM/ENP/97M-3

AFIT/GM/ENP/97M-3

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GM/ENP/97M-3

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

THESIS

Presented to the Faculty of the Graduate School of Engineering

Air Force Institute of Technology

Air University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science (Meteorology)

Jay B DesJardins, Jr., B.S.

Captain, USAF

March 1997

AFIT/GM/ENP/97M-3

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

Jay B DesJardins, Jr., B.S.

Captain, USAF

Approved:

JASON P. TUELL, MAJ, USAF
Advisory Committee Chairman

14 Mar 97
Date

MICHAEL K. WALTERS, LT COL, USAF
Advisory Committee Member

14 Mar 97
Date

KIRK A. MATHEWS, Ph.D.
Advisory Committee Member

17 MAR 97
Date

ACKNOWLEDGEMENTS

I am indebted to my wife Debra, who eagerly managed to tend to the needs of our three children, including our new daughter Sidney Nicole, who (after brilliant planning) was born three weeks before the thesis research quarter. I also must thank my thesis advisor, Maj Jason Tuell, whose motivation propelled me through this project. I also wish to thank Capt Vincent Ries who provided me information on Navy products and coding practices at AFGWC.

Jay

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

Figure Description Page

Figure	Description	Page

LIST OF TABLES

Table Page

LIST OF SYMBOLS

Symbol   Meaning

a .... Radius of the Earth
A .... Surface area
b .... Intercept of a given linear relationship
c_p .... Specific heat of dry air at constant pressure
f .... Coriolis parameter ($= 2\,|\mathbf{\Omega}|\sin\phi$)
g .... Scalar force of gravity per unit mass
h .... Effective depth of vortex
i .... Column (longitudinal) indicator of an array
$\mathbf{i}$ .... Eastward (longitudinal) unit vector
j .... Row (latitudinal) indicator of an array
$\mathbf{j}$ .... Northward (latitudinal) unit vector
k .... Longitudinal wave number
$\mathbf{k}$ .... Vertical unit vector
l .... Latitudinal wave number
m .... Slope of a given linear relationship
p .... Pressure
p_0 .... Specified reference pressure (100 kPa)
p_{MSL} .... Mean sea-level pressure
p_n .... Pressure given by the nth successive Newton iteration
p_{sfc} .... Surface pressure
p_θ .... Isentropic pressure
P .... (Ertel's) isentropic potential vorticity
P_p .... (Ertel's) isentropic potential vorticity at constant pressure
P_R .... Rossby (barotropic) potential vorticity
P_θ .... (Ertel's) isentropic potential vorticity at constant potential temperature
RH_p .... Relative humidity at constant pressure
RH_θ .... Relative humidity at constant potential temperature
s_p .... Isobaric value for a scalar (for a given pressure)
s_p .... Isobaric value for a scalar (for a given pressure)
s_θ .... Isentropic value for a scalar (for a given potential temperature)
t .... Time
T .... Temperature
T_{MSL} .... Mean sea-level temperature

Symbol	Meaning
T_n	Temperature given by the *n*th successive Newton iteration
T_p	Temperature at constant pressure
T_θ	Temperature at constant potential temperature
u	Eastward (longitudinal) wind component
u_g	Eastward (longitudinal) geostrophic wind component
u_p	Eastward (longitudinal) wind component at constant pressure
u_θ	Eastward (longitudinal) wind component at constant potential temperature
u_ϕ	Eastward (longitudinal) wind component at constant latitude
$\overline{u}_\phi$	Mean eastward (longitudinal) wind component for a given latitude
v	Northward (latitudinal) wind component
v_g	Northward (longitudinal) geostrophic wind component
v_p	Northward (latitudinal) wind component at constant pressure
v_θ	Northward (latitudinal) wind component at constant potential temperature
$\mathbf{V}$	Horizontal wind vector
x	Distance component in **i**-direction
y	Distance component in **j**-direction
Z	Geopotential height
Z_p	Geopotential height at constant pressure
Z_θ	Geopotential height at constant potential temperature
ϕ	Latitudinal distance (radians)
Φ	Geopotential
Φ_0	Mean-state geopotential
Γ	Vertical lapse rate
λ	Longitudinal distance (radians)
θ	Potential temperature
$\mathbf{\Omega}$	Angular rotational velocity of the Earth
ψ	Montgomery streamfunction
ζ	Relative vorticity (at constant pressure)
ζ_a	Absolute vorticity
ζ_θ	Relative vorticity at constant potential temperature

AFIT/GM/ENP/97M-3

ABSTRACT

This thesis presents and validates methods for calculating isentropic potential vorticity (IPV) and applies these methods in software programs planned for implementation at the Air Force Global Weather Center (AFGWC). The IPV programs will benefit Air Force Weather forecasters by providing them additional tools to diagnose atmospheric kinematics and understand atmospheric dynamics. A formula translation (FORTRAN) program is recommended using coarse-grain mandatory-level isobaric data projected to be available on AFGWC computer systems. Specifically, atmospheric models such as the Navy Operational Global Atmosphere Prediction System model and the National Centers for Environmental Prediction's Medium Range Forecast model are used. Program development and analysis consists of three main steps: (1) data retrieval; (2) IPV calculations; and, (3) interpolation to an isentropic vertical coordinate system. This thesis recommends performing IPV calculations at constant pressure for comparison with other mandatory-level isobaric parameters, or in routine cross-sectional analysis. Additionally, a recommendation is made to calculate IPV at constant potential temperature from interpolated isentropic state variables instead of interpolating isobaric IPV fields. Applications of the developed programs and subroutines include visualization of synoptic-scale vertical motions critical to cloud and precipitation forecasts, and an alternative method of locating the tropopause in cross-sectional analysis. This thesis is a significant effort to move toward operational use of isentropic analysis and the incorporation of IPV analysis into forecasting techniques at AFGWC.

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

1. Introduction

This thesis presents and validates methods for calculating isentropic potential vorticity (IPV) and applies these methods in software programs planned for implementation at the Air Force Global Weather Center (AFGWC). The algorithm and methods are implemented in formula translation (FORTRAN) routines designed for implementation at AFGWC using *coarse-grain* (Hoskins *et al*., 1985) mandatory-level meteorological model output available at AFGWC. These routines will benefit Air Force Weather (AFW) forecasters by providing additional tools to diagnose atmospheric kinematics and understand atmospheric dynamics and by employing *IPV thinking* (Hoskins *et al*., 1985) techniques. Zapotocny and Runk (1995) documented operational plans to incorporate and apply isentropic and IPV analysis at the AFGWC that are the foundation of this thesis. This thesis will be a significant effort to move toward operational use of isentropic analysis and the incorporation of IPV analysis into forecasting techniques at the AFGWC.

a. Research objective

With the advent of faster computer systems and new research, a move has already been made by national weather services to view weather products on isentropic surfaces in real time (Zapotocny and Runk, 1995; Carlson, 1991). But, since World War II, the aviation and meteorological community, including AFGWC, has focused almost exclusively on isobaric products (Bluestein, 1993; Moore, 1993). However, many synoptic-scale dynamic and kinematic features are more easily visualized and simplified in the quasi-Lagrangian reference frame offered by an isentropic analysis.

The algorithms developed, along with their proposed applications, will help keep AFGWC products and analysis techniques consistent with current theory being taught at major learning institutions (e.g., universities and training centers), and applied operationally by forecasters at other meteorological organizations. The IPV products produced by the developed routines will aid forecasting and defining the structure of frontal zones, vertical motion fields, moisture fields, depth of an atmospheric disturbance, and the dynamic tropopause (Zapotocny and Runk, 1995). Upper-level IPV anomalies used in conjunction with surface potential temperature anomalies are useful tools in describing the quasi-geostrophic (QG) forcing terms owing to vertical motion, cyclogenesis and frontogenesis. Depiction and visualization of moisture advection have direct application to forecasting regions of likely cloud, contrail, and precipitation development significant to military aviation operations. Locating the dynamic tropopause through IPV analysis will allow AFGWC personnel to accurately forecast severe weather, turbulence, and better define upper boundary conditions for any nested (mesoscale) models. IPV also allows visualization of the combined effects of vorticity advection and stability when deducing vertical motion strengths and the relative vertical extent.

b. Overview

This thesis will focus on validating IPV calculation methods, implementing them in software programs, and briefly discussing their proposed applications. The next chapter will briefly outline the history of IPV and isentropic analysis and provide an overview of the equations that will form a foundation for further algorithm development. Chapter 3 involves the methodology and development of the algorithm and its implementation in a viable FORTRAN program. The methodology includes a look at existing IPV calculation methods, software, standards, and available data. From this analysis, an implementation plan is formed. Fig. 1 is a flow chart of the major issues and decisions that will be addressed during algorithm development and implementation. Development addresses

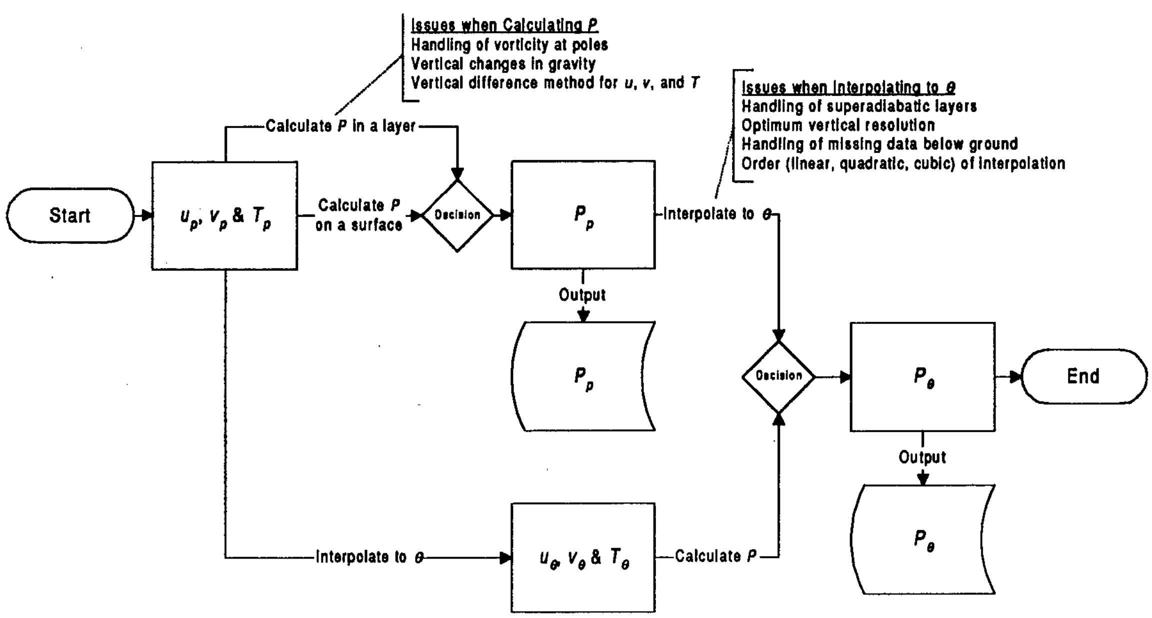


FIG. 1. Flow diagram of major issues and decisions addressed during IPV (P) algorithm development and implementation. u, v, and T are the horizontal wind components and temperature, respectively. Subscripts p and θ represent isobaric and isentropic data, respectively.

three general areas: (1) data retrieval, (2) IPV calculation methods, and (3) isentropic interpolation techniques. The proposed algorithm implementation by FORTRAN routines as a result of the development effort is shown in Fig. 2. Following this, chapter 4 provides a brief demonstration of proposed applications and a demonstration of the utility of the output produced by the FORTRAN routines.

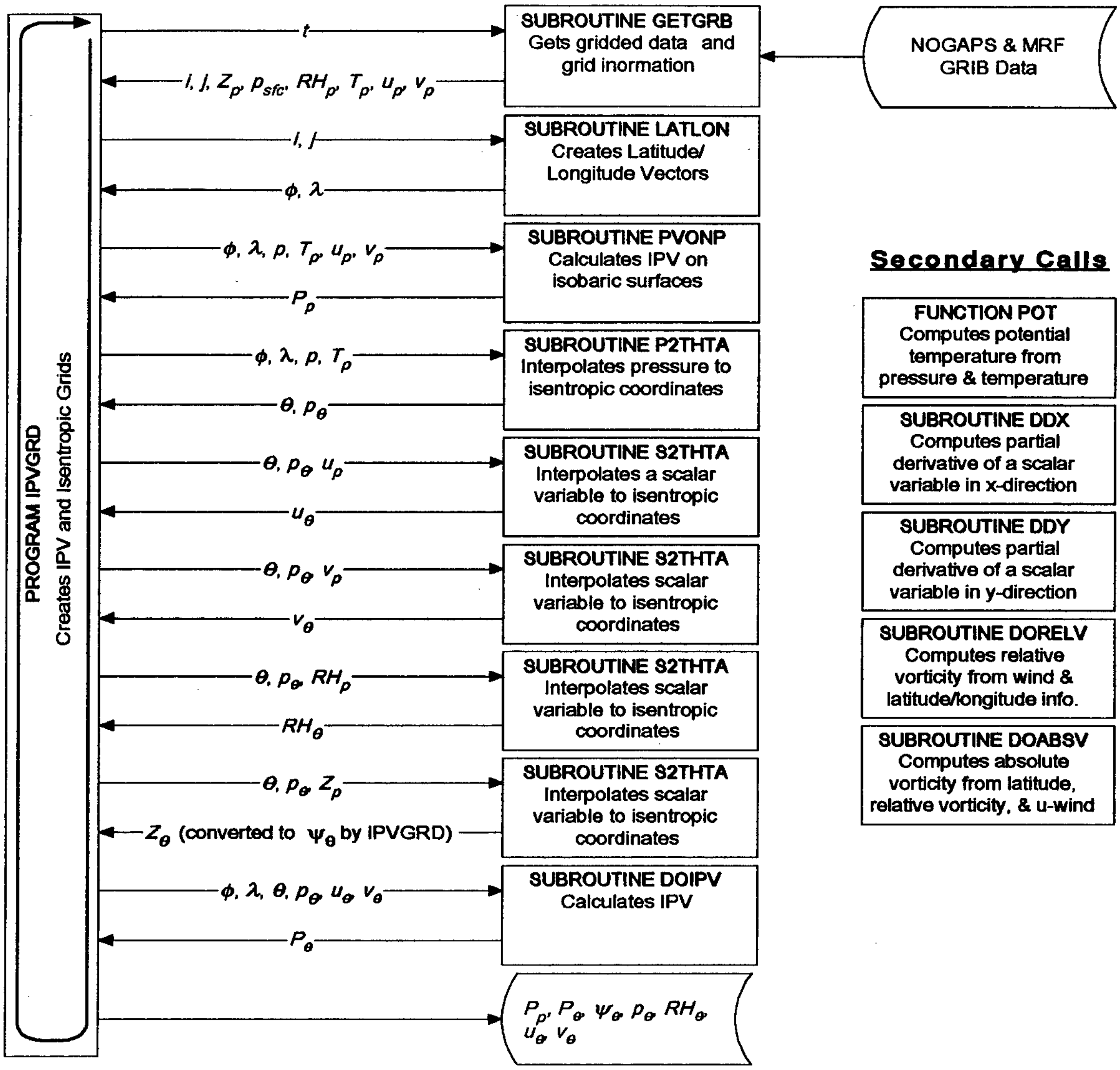


FIG. 2. Schematic indicating flow logic used by FORTRAN programs developed and documented at Appendix A-M used to create isentropic and IPV data fields. Primary calls indicate key variables passed between program and subroutines.

2. Background

This chapter briefly describes the evolution of IPV and its practicality as a forecasting and analysis tool. IPV products can simplify the visualization of dynamic and kinematic processes key to understanding past, existing, and future states of the atmosphere. AFGWC products are typically depicted on isobaric surfaces. A translation of these products to isentropic surfaces allows a more realistic Lagrangian view of the atmospheric motions. When atmospheric processes are adiabatic and frictionless, isentropic (constant entropy) surfaces are equivalent to surfaces of constant potential temperature, θ. Such isentropic surfaces are defined using Poisson's equation:

$$\theta = T\left(\frac{p_0}{p}\right)^{\frac{R_d}{c_p}} \tag{1}$$

where T is temperature, p represents pressure, R_d is the gas constant for dry air, c_p is the specific heat of dry air at constant pressure, and p_0 is a reference pressure typically taken to be 100 kPa. Since equation (1) shows that potential temperature is inversely proportional to pressure, it seems reasonable that isentropic surfaces may be used as a vertical coordinate instead of the more conventional height or pressure coordinate systems.

Since synoptic motions are inherently dominated by adiabatic, frictionless forcing, a projection of IPV, or other scalar parameters, onto isentropic surfaces gives an almost pure Lagrangian view of advective processes. The only motions contributing to cross-isentropic flow are those owing to diabatic (non-adiabatic) processes or friction.

Therefore, using potential temperature as a vertical coordinate minimizes flow perpendicular to a given isentropic surface. The result is a coordinate transformation where 2D atmospheric motions are maximized. In addition, isentropic surfaces act as a material surfaces in the absence of diabatic processes and friction.

IPV products have been a valuable tool in identifying air of stratospheric origin and providing a more useful definition of the dynamic tropopause than the lapse-rate definition (Danielsen, 1968). The lapse rate definition produces ambiguities in the vicinity of upper-level jets and fronts. However, better spatial and temporal continuity is possible using a constant IPV surface to define the tropopause (Spaete *et al.*, 1994). Generally, values of IPV less than 1.5 potential vorticity units (PVU, 1 PVU $\equiv 10^{-6}\ m^2\ K\ kg^{-1}\ s^{-1}$) represent tropospheric conditions (Davis, 1991). However, these values fluctuate seasonally. Spaete *et al.* (1994) reference standard tropopause values ranging from the World Meteorological Organization (WMO)-accepted value of 1.6 PVU up to the 3.0 through 4.0 PVU range derived from January 1979 model data from the European Centre for Medium Range Weather Forecasts global analyses.

Like vorticity advection, motions of IPV anomalies in the upper troposphere can also be used to explain cyclogenesis events. One of the largest advantages of IPV over absolute vorticity, is that the horizontal scale of the anomaly implies a specific vertical depth to which the effects are felt (Bluestein, 1993). IPV anomalies in the upper tropospheric can be used to identify potential areas for future cyclogenesis. A paper by Hoskins *et al.* (1985) presents a historical overview on the use and significance of IPV charts that is generally referenced in most texts today and summarized in the remainder of this section. Hoskins *et al.* (1985) also introduced the phrase *IPV thinking* to denote application of IPV products to reinforce QG theory and atmospheric forcing.

The largest advantage of isentropic analysis in itself is an opportunity to revise antiquated thinking from the static Norwegian *air mass* concept in favor of a Lagrangian *air stream* concept more consistent with quasi-geostrophic forcing. Systematic use of isentropic charts began as early as the 1930s with the work of Namias. A later standardization, heavily influenced by the aviation community, led to the wide traditional use of isobaric analysis and the decrease in popularity of isentropic charts. A revitalization in isentropic analysis after development of quasi-geostrophic theory has been building since the late 1950s (Carlson, 1991).

In 1939, Rossby realized the vertical component of absolute vorticity, ζ_a, is dominant in large-scale atmospheric flow in comparison to the horizontal components (due to relatively small vertical velocities). Thus, synoptic-scale vorticity analysis focused can be approximated by its vertical component, where:

$$\zeta_a = f + \zeta = f + \mathbf{k} \cdot (\nabla \times \mathbf{V}) \tag{2}$$

where, f represents the latitude-dependent Coriolis parameter, **k** is the vertical unit vector, ζ is the vertical component of relative vorticity, and **V** is the horizontal wind vector. In 1940, using a barotropic model, Rossby expressed the simplest form of potential vorticity as the measure of the ratio of the absolute vorticity to the effective depth of the vortex tube, h, defined by the vorticity:

$$\left(\frac{\zeta + f}{h} \right) = \text{Constant} . \tag{3}$$

This form conveniently accounts for the two dominant processes in the vorticity budget: the creation of vorticity by vortex tube stretching and by the horizontal advection of absolute vorticity (either by increasing relative vorticity or increasing latitude). Expressing *h* as the material surface thickness between isentropic layers, and using the hydrostatic approximation that incorporates gravity, *g*, equation (3) becomes the form generally referred to as the Rossby, or barotropic, potential vorticity (Holton, 1992), P_R:

$$P_R = -g \frac{(f + \zeta_\theta)}{\delta p} \tag{4}$$

where ζ_θ is the relative vorticity on an isentropic surface, and *p* represents pressure. Hence, the terminology *isentropic* potential vorticity. The need to calculate the vorticity in equation (4) on an isentropic surface first highlights the advantage and simplification of calculating potential vorticity directly from an isentropic analysis, vice an isobaric analysis.

In 1942, the independent work of Ertel further confirmed Rossby's work and extended the results to a continuous atmosphere. Ertel's potential vorticity[1], *P*, when applied to isentropic surfaces, is expressed as:

$$P = -g (f + \zeta_\theta) \frac{\partial \theta}{\partial p} \tag{5}$$

or, expressed in isobaric coordinates:

$$P_p = -g \left[\zeta_a + \left(\mathbf{k} \times \frac{\partial \mathbf{V}}{\partial \theta} \right) \bullet \nabla_p \theta \right] \frac{\partial \theta}{\partial p}. \tag{6}$$

---

[1] In some texts, Rossby's and Ertel's potential vorticity are used interchangably.

where, the subscript p on the gradient operator represents changes at constant pressure. Ertel's work, represented in equations (5) and (6), will be the basis of subsequent IPV calculations.

3. Methodology and development

Methodology will discuss different approaches to calculating IPV using existing software, standards, and data. The algorithm and program development process will follow the proposed methodology previously outlined in Fig. 1 and provide rationale for decisions made during the algorithm design and implementation. Development will first include an investigation of possible data sources for both operational and developmental use. Next, calculation methods for IPV algorithms will be investigated. Development will conclude with research into an isentropic interpolation scheme for mandatory-level isobaric data.

a. Methodology

Existing software e.g., GEMPAK (desJardins *et al.*, 1996) and National Centers for Environmental Prediction (NCEP) data unpacking routines, will be exploited to the extent possible. In order to meet AFGWC coding standards (AFGWC/SY DOI 33-2, 1996) and ensure widest platform compatibility, the developed algorithms will be implemented as American National Standards Institute (ANSI)-compliant FORTRAN routines (FORTRAN 77). These routines will be written for use and tested with AFGWC isobaric atmospheric model output from the Navy Operational Global Atmosphere Prediction System (NOGAPS) model and the NCEP Medium Range Forecast (MRF) model.

Development will employ an analysis of various existing IPV calculation methods. The method outlined by Hoskins *et al.* (1985), Davis and Emanuel (1991), and later by Davis (1992) refer to calculations of P on isobaric surfaces, P_p, using a centered finite difference method derived from equation (6), then a transformation of P to isentropic coordinates, P_θ, *via* interpolation from the isobaric fields. Although this method may slightly minimize computational time, it will be compared to an alternative method where isentropic interpolation of wind and pressure data from isobaric coordinates precedes P_θ calculations (see Fig. 1). The comparison will be performed later as part of determining the most accurate application of the algorithms. The method of calculating P, whether isobarically or isentropically, will also be addressed. GEMPAK (desJardins *et al.*, 1996) calculates P using a layered average, whether between isobaric or isentropic surfaces. This thesis proposes P calculations valid at a specified level as performed by Hoskins *et al.* (1985) rather than for a layer. During early program development, the output grids created from earlier versions of the routines contained in Appendices A-M were compared with those produced by routines from GEMPAK[1] version 5.4 for initial accuracy. From there, modifications were made. Careful attention was paid to develop code that would eliminate floating-point calculation overflows, underflows, and divisions by zero.

Current operational plans indicate that AFGWC personnel are likely to use NOGAPS for formulation of near-term (out to 72 hours) forecasts, and employ the MRF for longer range forecasting (beyond 72 hours). Developed routines could also be easily tailored to other models such as the Relocatable Window Model (RWM), which uses a terrain-following vertical coordinate, σ, or the Mesoscale Model 5 (MM5). However, due to the

---

[1] GEMPAK calculates both P_p and P_θ

nature of IPV, the algorithm is best suited to diagnose synoptic scale features in the absence of local diabatic effects and friction, and may not be suited for use in conjunction with a mesoscale model. The potential migration of AFGWC systems from the RWM to MM5 also posed an implementation risk in tailoring software programs to these models. The MRF data was also chosen as a supplement to the NOGAPS data due to its wide availability, global coverage, and the gridded binary (GRIB) data format (Dey, 1996) already used at the AFGWC. IPV algorithm development and visualization employs use of both models.

The programs developed by this effort must be as portable and modular as possible to allow flexibility in integration into AFGWC computer systems, and potentially into other weather computer systems. Algorithm coding techniques adhered to AFGWC FORTRAN coding standards (AFGWC/SY DOI 33-2, 1996) as closely as possible. The reformatting of data output produced by the programs is left to existing packing and storage methods employed at the AFGWC and is not specifically addressed as part of the program development.

Following development, this thesis demonstrates use of the developed code by producing some standard isentropic and IPV products. Samples of these products are visualized using the Grid Analysis and Display System (GrADS), version 1.5, software packages as a visualization tool. Output from the program at Appendix A is tailored to visualization by GrADS (Doty, 1995). It is assumed that AFGWC has the capability to produce visual products from the expected gridded fields using any of their software visualization products, such as PV-WAVE®.

Since different methods of calculating IPV clearly exist, this thesis focuses on verification of certain approaches and techniques for calculating IPV from specified data sources, and transitioning the results to personnel at AFGWC for implementation. Expansion of algorithms to include equivalent potential vorticity, EPV, products as implied by Zapotocny and Runk (1995) will not specifically be addressed by this thesis, but will be an opportunity for further development.

b. Development

The FORTRAN code development consisted mainly of three separate efforts: 1) data retrieval, 2) IPV calculations, and 3) isentropic interpolation. The following sections describe the implicit decisions made during algorithm development for each effort.

1) DATA RETRIEVAL

Data for this thesis included output from the Navy's NOGAPS model valid through the 72-hour forecast period, every 3 hours, and NCEP's MRF model valid through the 384-hour forecast, every 12 hours. AFGWC personnel provided data from both models from the 0000 UTC model runs on 13 September 1996. Periodically, routines were run with current data from the NCEP's Aviation (AVN) model obtained from a local GEMPAK data feed via Unidata. This allowed a comparison of program output with GEMPAK data fields for general correctness and as a basis for troubleshooting programming errors within the routines.

Both the MRF and NOGAPS data are in GRIB data format with grid populations as specified in Table 1. The NOGAPS grids obtained were originally at a one-degree resolution, but were reduced to a 2.5-degree resolution by AFGWC-Navy computer

TABLE 1. Grid resolutions.

Model	Projection	Grid size (i x j)	Resolution (longitude x latitude)
MRF	Cylindrical Equidistant	360 x 181	1.0° x 1.0°
NOGAPS	Cylindrical Equidistant	144 x 73	2.5° x 2.5°
AVN (via GEMPAK)	Cylindrical Equidistant	73 x 73	5.0° x 2.5°

systems (AFGWC, 1995). The MRF model is a global spectral model run once per day (0000 UTC) out to 16 days (384 hours). The same global spectral model that is used for the AVN run is used for the MRF (T126[1] horizontal spectral resolution, 28 vertical layers), with the exception that the horizontal resolution is reduced to T62 after day 7.

A C-Shell script that writes GEMPAK data and grid information to a data file, combined with a FORTRAN subroutine, allowed the integration of GEMPAK AVN model data. Model data parameter arrays could also be read directly from the NOGAPS GRIB files containing a separate file for each grid (AFGWC, 1995), or from the MRF GRIB files containing all parameters for specified time period using similar FORTRAN subroutines to unpack the original GRIB-formatted data files. These data arrays were passed directly to IPV calculation or isentropic interpolation subroutines. Table 2 specifies the parameters used in this thesis from each of the models. The u and v-wind

[1] T indicates that the spectral model uses a triangular truncation method. The suffix is the truncation number for the spherical harmonics. T106 reflects a latitude/longitude resolution of approximately 1.21° (Holton, 1992).

TABLE 2. Model output used.

Parameter	MRF	NOGAPS	AVN (via GEMPAK)
u	Mandatory isobaric levels 100-1 kPa, 10 m	Mandatory isobaric levels 100-1 kPa, 10 m	Mandatory isobaric levels 100-10 kPa, 10 m,
v	Mandatory isobaric levels 100-1 kPa, 10 m	Mandatory isobaric levels 100-1 kPa, 10 m	Mandatory isobaric levels 100-10 kPa, 10 m
T	Mandatory isobaric levels 100-1 kPa, 2 m	Mandatory isobaric levels 100-1 kPa, 2 m	Mandatory isobaric levels 100-10 kPa, 2 m
p	Surface	Mean Sea Level	Surface
Z	Mandatory isobaric levels 100-1 kPa, Surface	Mandatory isobaric levels 100-1 kPa, Mean Sea Level	
RH	Mandatory isobaric levels 100-30 kPa, 2 m	Mandatory isobaric levels 100-30 kPa, 2 m	
Other		Terrain Height	

components are grid-relative east and north wind components, respectively. *RH* and *Z* parameters refer to relative humidity and geopotential height, respectively, and are not required in IPV calculations. *Z* is used to calculate the Montgomery streamfunction, ψ, and *RH* is simply interpolated to isentropic coordinates to allow incorporation of a moisture parameter along with analysis of the other isentropic variables. IPV calculations from NOGAPS model output require a surface terrain database to allow derivation of surface pressure fields used to determine where isentropic surfaces intersect with the ground.

After specification of the desired model output by the forecast time, *t*, the data retrieval module returns several arrays of model data to the main program (Appendix A) for calculation of *P*. FORTRAN programs use an *i* (column), *j* (row) grid numbering convention where (column = 1, row = 1) represents the upper left corner of the grid. This convention is typically standard for AFGWC applications (Hoke *et al.*, 1981), and is the same numbering convention used by FORTRAN array structures. However, both GEMPAK and NOGAPS begin with the lower-left grid corner as (1, 1) and j increasing northward. A third array dimension represents increasing vertical directions with surface data in the first element, if present. The subroutines that unpack the GRIB data are modifications of freely-available NCEP programs. Since AFGWC personnel have packing and unpacking programs already available, these are not discussed in detail here and are omitted from Appendix A.

2) IPV CALCULATIONS

P is calculated either on each mandatory-level isobaric surface or on each isentropic surface via a series of subroutines that determine the parameters from equation (6).

First, from the grid information, a subroutine generates latitude and longitude information (Appendix C) corresponding to the desired grid. Since all the grids used are cylindrical equidistant projections (a.k.a. latitude/longitude grids), the navigation information can be stored in a latitude vector corresponding to each grid row, and a longitude vector corresponding to each grid column. Conventions according to Hoke *et al.* (1981) assign negative values to longitudes in the Western Hemisphere and to latitudes in the Southern Hemisphere. This differs slightly from WMO representation (Dey, 1996) where longitude values range from 0 to 360 (East). Other projections may require a two dimensional grid if latitude and longitude both vary across grid rows and/or columns. The latitude and longitude information is required for Coriolis parameter and finite difference calculations.

ζ is calculated from the wind field, where the horizontal wind, **V**, is broken into eastward and northward wind components, u and v, respectively:

$$\mathbf{V} = u\mathbf{i} + v\mathbf{j}. \tag{7}$$

In general terms, the Cartesian form of relative vorticity is expressed as:

$$\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \tag{8}$$

where x and y are in orthogonal directions. However, since the data used is latitude-longitude oriented, x and y will be chosen to represent distances (in meters) in the longitudinal and latitudinal directions. Separate subroutines take the partial derivative with respect to the x and y-directions (Appendix H and Appendix I, respectively). These same subroutines can also be employed to calculate the gradient of a scalar. When calculating the first term in equation (8), the subroutine accounts for the decreasing x distance between grid points as you approach the poles where the circumference of the latitude circle decreases. Distance between grid points is calculated along latitudinal and longitudinal paths, and assumes a spherical Earth with an effective radius of 6,371,221.3 m (Hoke *et al.*, 1981). The partial derivatives are calculated using a second order centered finite difference scheme (Haltiner and Williams, 1980) with a few exceptions:

(a) at the poles $\partial v/\partial x$ is set to 0, where the entire row of grid points theoretically represent the same point;

(b) for $\partial u / \partial y$, the grid points on the first and last column look for the possibility of a worldwide grid and calculate a second order centered difference if possible, otherwise a first order forward or backward difference is calculated, as appropriate;

(c) since a large number of isentropic surfaces intersect the surface, routines account for missing data (represented as -9999.0) by performing a first order forward or backward difference near these boundaries, as appropriate; and

(d) first order forward and backward differences are calculated at the poles for $\partial u / \partial y$, as appropriate.

Finally, when using a latitude-longitude relative grid, equation (8) must include a correction to account for the decreasing x distance between grid points as latitude, ϕ, increases. Therefore the natural form of the relative vorticity in equation (8), when expressed in spherical coordinates, becomes:

$$\zeta = \frac{1}{a\cos\phi}\frac{\partial v}{\partial \lambda} - \frac{1}{a}\frac{\partial u}{\partial \phi} + \frac{u}{a}\tan\phi; \text{ or, } \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + \frac{u}{a}\tan\phi \qquad (9)$$

where, a is the radius of the Earth and λ represents longitude (see Appendix J). Because of the choice of coordinate system, equation (9) becomes undefined at the poles. To eliminate this singularity and the floating point calculation errors that may accompany it, the circulation theorem is applied at the poles using the data at the nearest latitude circle:

$$\zeta_{\text{Pole}} = \frac{\oint_A u_{\phi \pm 1} \bullet dx}{A} \qquad (10)$$

where, A is the surface area of the polar cap to the nearest latitude circle, $\phi \pm 1$. The surface area, A, can be expressed as:

$$A = 2\pi a^2 \left[1 - \sin\left(\phi \pm 1\right)\right]. \tag{11}$$

Using equation (11), equation (10) simplifies to:

$$\zeta_{\text{Pole}} = \overline{u}_{\phi \pm 1} \frac{\cos\left(\phi \pm 1\right)}{a\left[1 - \sin\left(\phi \pm 1\right)\right]} \tag{12}$$

where, $\overline{u}_{\phi \pm 1}$ is the average wind at the nearest latitude circle to the pole. Consideration was also given to the possibility that the input data may not be from a global grid or from an overlapping grid (such as the AVN data via GEMPAK) when determining the value from equation (10), i.e., the grid distance between the first and last points may differ from grid distances between the rest of the points in the row. To obtain a grid of absolute vorticity, planetary vorticity values are added to the relative vorticity values using the subroutine found at Appendix K.

P can now be calculated at constant p. On such an isobaric surface, a correction must be added to the absolute vorticity to account for changes in θ along the isobaric surface. From equation (6) we find:

$$P = -g\left[\zeta_a + \left(\frac{\partial u}{\partial \theta}\right)\left(\frac{\partial \theta}{\partial y}\right)_p - \left(\frac{\partial v}{\partial \theta}\right)\left(\frac{\partial \theta}{\partial x}\right)_p\right]\frac{\partial \theta}{\partial p} \tag{13}$$

As mentioned earlier, GEMPAK (desJardins *et al.*, 1996) approaches a calculation of equation (13), valid for a given isobaric layer, by calculating a linear average of u, v, and θ parameters between two isobaric levels. A simplistic analysis of interpolation methods of the u and v wind component parameters using AVN 00-hour forecast data valid at 0000 UTC, 3 January 1997 (Table 3), indicates that when dealing with mandatory-level isobaric data, such as NOGAPS and the MRF model output, a linear average of these parameters with respect to p, is most likely not the best choice. Physically, this supports the thermal wind relation where u and v-wind components are proportional to $\ln p$ when the temperature gradient is constant:

$$\frac{\partial v_g}{\partial \ln p} = -\frac{R_d}{f}\left(\frac{\partial T}{\partial x}\right)_p \text{ and } \frac{\partial u_g}{\partial \ln p} = \frac{R_d}{f}\left(\frac{\partial T}{\partial y}\right)_p \tag{14}$$

where, u_g and v_g represent the geostrophic wind.

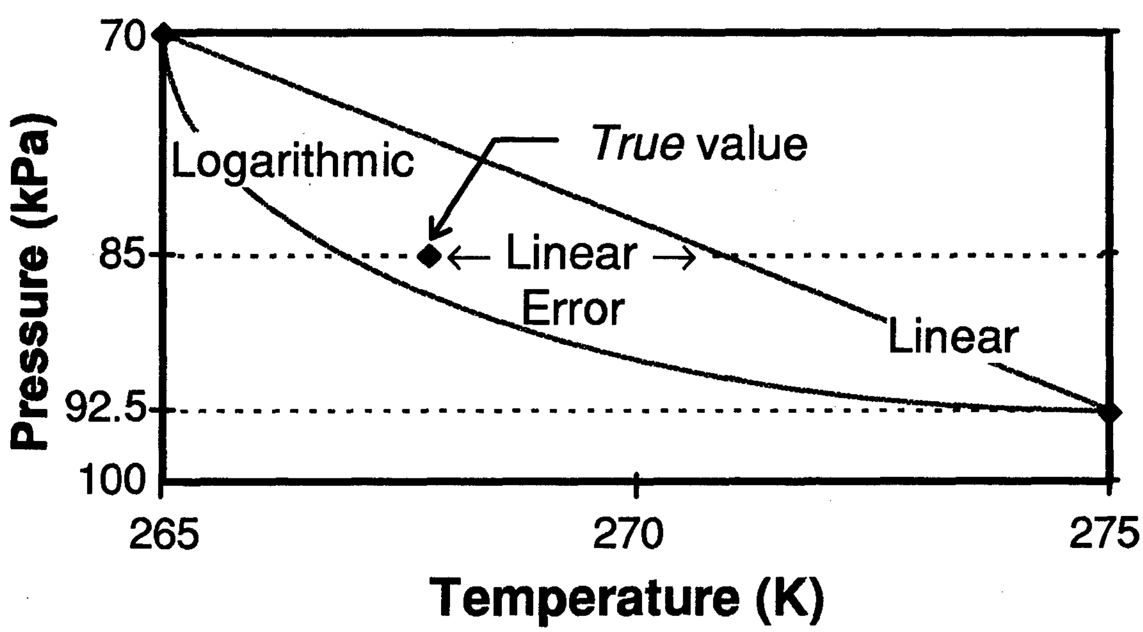


FIG. 3. Method used to determine error between vertical interpolation methods from initialized model data.

Root-mean square errors (RMSE) in Table 3 were calculated by comparing initialized data from the AVN model. Initialized fields (00-hour forecast) were selected to decrease the amount of smoothing later that may occur later in the forecast cycle. Values were interpolated using both a linear and logarithmic method from data values from the nearest mandatory isobaric level below and above a given *true* point as shown in Fig. 3. Linear

TABLE 3. Comparison of vertical interpolation methods for u and v-wind components, and T using initialized model data from AVN, valid 0000 UTC, 3 January 1997.

		vs p		vs $\ln p$	
Level (kPa)	Global Mean	Global RMSE*	Max Error	Global RMSE*	Max Error
	u wind (m s^{-1})				
85	1.33	2.52	12.36	1.47	11.91
70	3.67	2.87	15.44	2.82	15.64
50	7.28	2.61	17.26	2.48	16.51
40	9.93	2.53	15.45	2.49	16.16
30	12.91	2.33	11.89	2.33	12.01
25	14.12	2.17	13.69	2.16	13.59
20	14.81	2.75	21.10	2.72	19.8
15	13.87	3.63	24.92	3.35	25.0
Mean		2.71		2.63	
	v wind (m s^{-1})				
85	-0.15	2.22	11.79	2.20	11.24
70	0.05	2.65	20.52	2.62	21.20
50	0.10	2.40	16.35	2.35	14.76
40	-0.04	2.40	14.19	2.38	14.20
30	-0.09	2.40	15.12	2.39	14.91
25	0.05	1.10	14.04	1.06	13.58
20	0.24	2.45	13.41	2.43	13.99
15	0.51	3.25	21.45	3.25	21.66
Mean		2.51		2.48	
	T (K)				
85	273.30	2.62	9.87	2.71	9.48
70	266.67	3.26	13.71	2.56	13.57
50	251.33	2.57	7.66	1.53	6.92
40	240.67	2.11	7.85	1.83	6.38
30	228.30	2.33	10.58	2.56	10.83
25	223.29	1.67	6.57	1.71	6.75
20	219.25	2.10	11.05	2.02	11.41
15	214.87	1.88	8.14	2.61	8.65
Mean		2.36		2.23	

*RMSE = Root Mean Square Error between actual model output and interpolated model output from above and below a mandatory isobaric level logarithmic interpolation of u and v wind components in the vertical, a method also recommended by Bergman (1979), seems to be a better estimator than the linear method used in existing GEMPAK routines. However, if computational time is a problem, the cost of calculating the logarithms may not justify the improvement. A cubic or quadratic interpolation, as discussed later during isentropic interpolation methods, may warrant future consideration. Additionally a consideration may be given to interpolating the wind direction and speed instead of u and v components.

To determine how potential temperature changes with pressure, temperature variations with pressure were analyzed. Although Bergman (1979) and the U.S. Standard Atmosphere definition (NOAA, 1976) both suggest that temperature also varies linearly with $\ln p$ (approximating geometric height), in an adiabatic atmosphere where $d\theta = 0$ application of Poisson's equation, equation (1), leads to an atmosphere where logarithmic changes in T vary linearly with $\ln p$:

$$d \ln T = \frac{R_d}{c_p} d \ln p. \tag{15}$$

Performing an analysis of $\ln T$ and T variations with respect to $\ln p$, gives the results presented in Table 4. Although interpolation of T against $\ln p$, as presented in Table 3, is also an improvement over strict linear interpolation of T vs p, $\ln T$ varying linearly with $\ln p$ may be physically more meaningful. Therefore, both temperature and potential temperature will be interpolated assuming that $\ln T$ varies linearly with $\ln p$. Again, if computational time is important, logarithmic calculations may not warrant the

TABLE 4. Comparison of vertical interpolation of T and $\ln T$ against $\ln p$ from initialized AVN, valid 0000 UTC 25 December 1996.

		T vs $\ln p$		$\ln T$ vs $\ln p$	
Level (kPa)	Global Mean (K)	RMSE* (K)	Max Error (K)	RMSE* (K)	Max Error (K)
85	274.36	2.47	8.96	2.44	8.96
70	267.03	2.29	11.99	2.35	12.03
50	252.29	1.42	6.43	1.52	6.21
40	241.81	1.51	6.02	1.52	5.90
30	229.03	1.88	6.30	1.83	6.20
25	222.86	1.33	5.58	1.30	5.58
1.75	217.02	1.75	6.59	1.68	6.53
15	211.77	2.74	7.42	2.61	7.34
Mean		1.99		1.96	

*RMSE = Root Mean Square Error between actual model output and interpolated model output from above and below a mandatory level

consideration especially when considering the small advantage gained in interpolating $\ln T$ instead of T against $\ln p$.

Next, the layered method (a modified version of GEMPAK's method using a logarithmic-weighted pressure average) was compared to a method where P_p was calculated directly for a given mandatory isobaric level. This mandatory-level method employs u, v, and θ components for the level in question and assumes they change with $\ln p$ as previously determined. According to these relations the wind components still change linearly with θ. Therefore, $\partial u / \partial \theta$, and $\partial v / \partial \theta$ remain linear differences. However, the stability, $\partial \theta / \partial p$, becomes:

$$\frac{\partial \theta}{\partial p} = \frac{\theta}{p}\left(\frac{d \ln T}{d \ln p} - \frac{R_d}{c_p}\right) \tag{16}$$

where, p and θ are the pressure and potential temperature where the stability is valid, respectively.

The layered method is valid at the ln p-weighted mean between the mandatory isobaric levels and the mandatory-level method is valid at the mandatory isobaric level, itself. For the layered method, p is the pressure at the ln p-weighted mean, and θ is determined from the both this pressure and the temperature determined from ln T at the ln p-weighted mean in the layer (See Appendix M).

A comparison between the two methods was performed by defining a known analytic wave function representing geopotential, Φ, where the amplitude of the wave varied with pressure. P could then be analytically calculated and compared to calculations from each method to determine which method had the largest source of error. For this comparison, all winds were assumed geostrophic.

The geopotential was defined with longitudinal and latitudinal wave numbers of k and l, respectively, as follows:

$$\Phi\left(\lambda,\ \phi,\ p\right) = \Phi_0(p) + \frac{\Phi_0(p)}{14} \sin k\lambda\ \sin l\phi, \tag{17}$$

where, and $\Phi_0\,(p)$ represents the mean geopotential at a given isobaric level as determined assuming a hydrostatic atmosphere with a pre-defined lapse rate, Γ. $\Phi_0\,(p)\,/14$ depicts a scaling of the geopotential amplitude to obtain reasonable values in the deformation field. Using the lapse rate definition, $\Gamma = -dT\,/dz$, temperature can be defined as:

$$T(p) = T_0\left(\frac{p}{p_0}\right)^{\frac{\Gamma R_d}{g}} \tag{18}$$

where T_0 is the temperature at some reference pressure, p_0. For our purposes, the U.S. standard atmospheric value of 287.43 K at 100 kPa (NOAA, 1976) was selected. Also using the U.S. standard atmosphere tropospheric lapse rate of 0.0065 K m^{-1} (NOAA, 1976), equation (18) describes a uniform temperature field with pressure. After combining equation (18) with the hydrostatic approximation, the mean geopotential field can be expressed as:

$$\Phi_0(p) = \frac{gT_{MSL}}{\Gamma}\left[1 - \left(\frac{p}{p_{MSL}}\right)^{\frac{\Gamma R_d}{g}}\right] \tag{19}$$

where, T_{MSL} and p_{MSL} represent the mean sea-level temperature and pressure, respectively. T_{MSL} and p_{MSL} values were chosen to represent U.S. standard atmospheric values (NOAA, 1976) of 288.15 K and 101.325 kPa, respectively. Therefore, equation (19) satisfies the boundary condition where $\Phi_0\,(p_{MSL}) = 0$. Fig. 4 depicts the theoretical geopotential height field at 50 kPa described by equations (17) through (19).

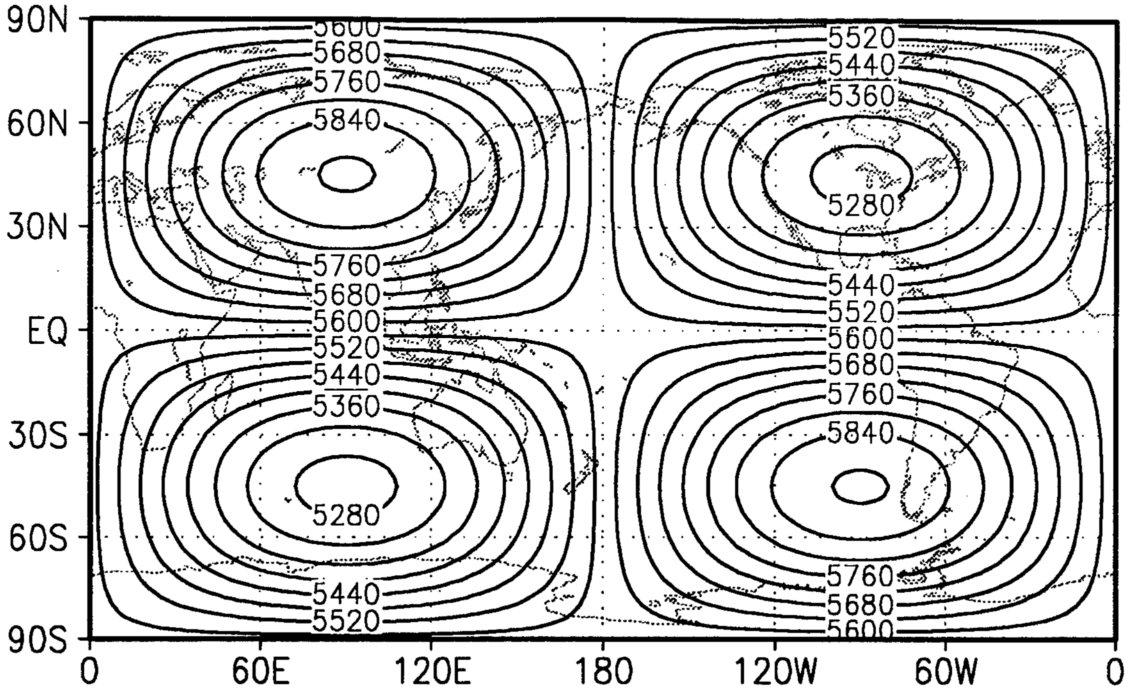


FIG. 4. Analytic 50 kPa geopotential height field (m) where $k = 1$ and $l = 2$.

The calculations of P from each of the subroutines were compared to the analytical value assuming geostrophic balance. The geostrophic wind field is described by the u and v wind components derived from equation (17), where:

$$u = -\frac{1}{f}\frac{\partial \Phi}{\partial y} = -\frac{\Phi_0(p)\,l}{14af}\sin k\lambda\,\cos l\phi\text{; and,} \tag{20a}$$

$$v = \frac{1}{f}\frac{\partial \Phi}{\partial x} = \frac{\Phi_0(p)\,k}{14af\,\cos\phi}\cos k\lambda\,\sin l\phi\,. \tag{20b}$$

This leads to a relative vorticity calculation from equation (11) where:

$$\zeta_\theta = \zeta_p$$
$$= -\frac{\Phi_0(p)}{14fa^2}\sin k\lambda\left[2l\Omega\cos\phi\,\cos l\phi + \left(\frac{k^2}{\cos^2\phi} + l^2\right)\sin l\phi\right] + \frac{u}{a}\tan\phi \tag{21}$$

The stability derived from equation (18) and Poisson's equation becomes:

$$\frac{\partial \theta}{\partial p} = \frac{R_d T_0}{p} \left(\frac{\Gamma}{g} - \frac{1}{c_p} \right) \left(\frac{p}{p_0} \right)^{\left(\frac{\Gamma R_d}{g} - \frac{R_d}{c_p} \right)} . \tag{22}$$

Substituting equations (21) and (22) back into equation (16), we obtain an analytic calculation of P_p. The magnitude of P increases away from the equator owing mostly to planetary vorticity. Fig. 5 depicts the analytic representation of P at 15 kPa.

Describing the temperature field as uniform with respect to pressure simplifies P calculations since the isentropic and isobaric surfaces are parallel eliminating the correction terms in (16). However, the rigor of the test becomes limited since calculations inherent in the isentropic relative vorticity correction terms are zero. The rigor of the test is also limited due to the well-behaved nature of the function chosen.

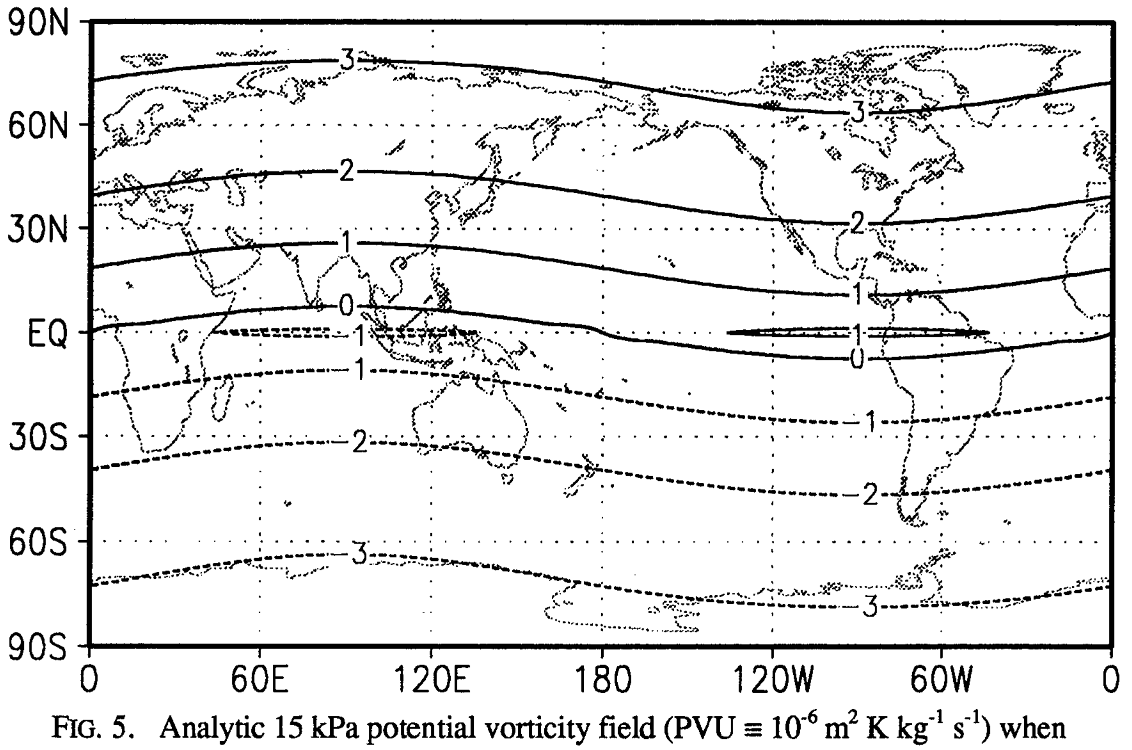


FIG. 5. Analytic 15 kPa potential vorticity field (PVU ≡ 10^{-6} m^2 K kg^{-1} s^{-1}) when $k = 1$ and $l = 2$.

An analysis of the two methods indicates that both the layered and the mandatory-level calculation methods are very accurate. Of course, near the equator the geostrophic assumption breaks down and larger errors occur. Fig. 6 indicates very small errors for both methods when compared to the analytic solution. The mandatory-level method appears better behaved, and has slightly smaller errors at most latitudes as shown in Fig. 7. However, these differences are small compared to the overall errors. The deviation in the layered method from the mandatory level method is directly correlated to the depth of the layer which influences interpolation accuracy when calculating θ in equation (16). Both methods see an increase in error as the wind speeds approach maximums at 90E and 90W. Interestingly, the layered-method has a continuously negative error bias when compared to the mandatory-level method. For instance, at 90E, deviations from the mandatory-level method are the same magnitude as 90W, but result in a larger error rather than smaller. Fig. 8 indicates these error variations in the longitudinal direction. The mean latitudinal error in the longitudinal direction is zero for the mandatory-level method.

Since both the layered and direct surface methods are comparable, there is a clear advantage to calculating P valid directly on mandatory pressure levels where other data is routinely collected, vice describing a different set of vertical coordinates where P is valid. Both methods appear very accurate, with errors on the order of one percent. At high altitudes, the amount of error is comparable to errors due to gravitational variations with height and is therefore acceptable. The algorithm used for isobaric P calculations is at Appendix L, the layered method is at Appendix M.

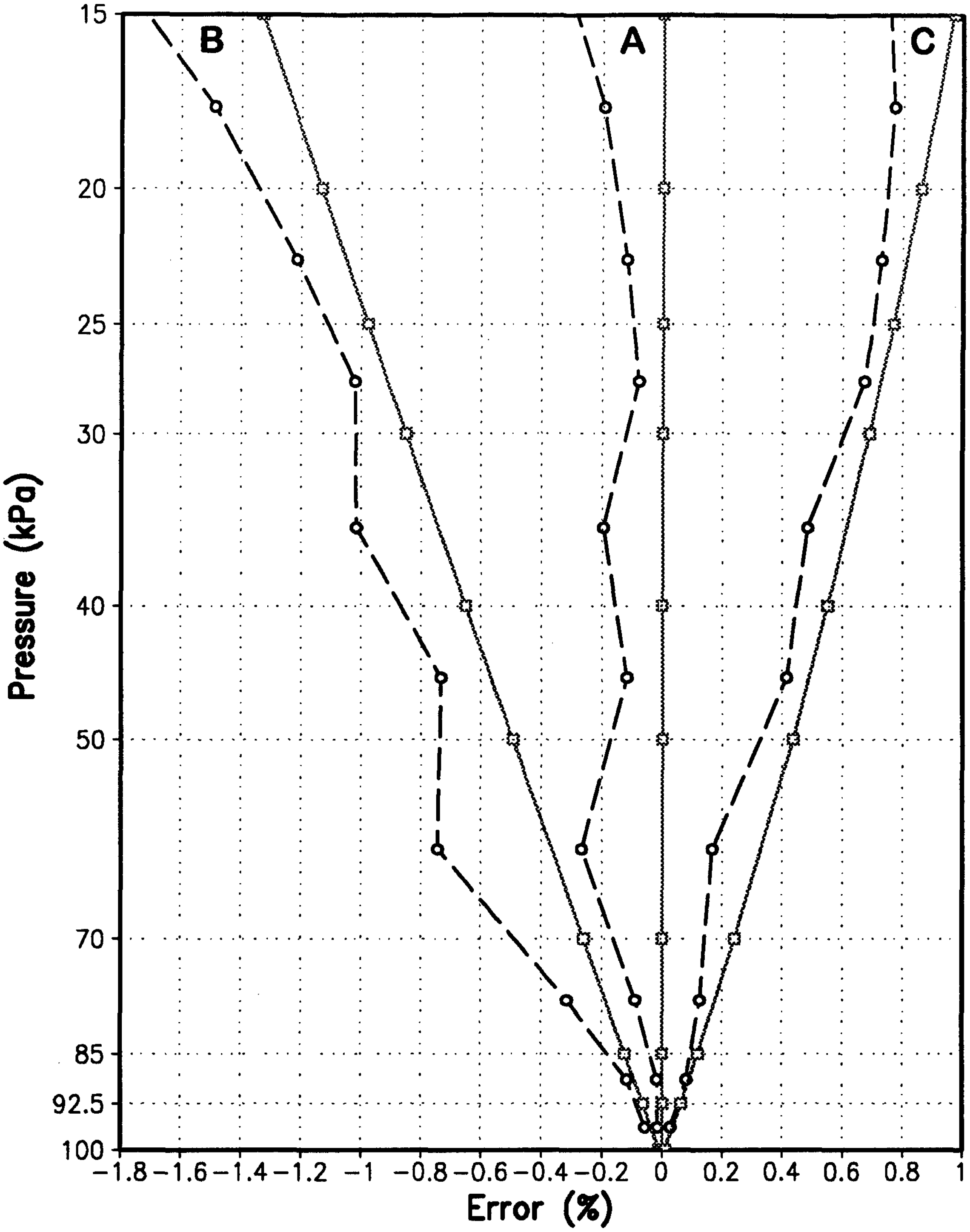


FIG. 6. Vertical *P* calculation errors against analytical solutions for a layered method valid between isobaric levels (dashed line, open circles) and a mandatory-level method valid on a given isobaric surface (solid line, open squares) at longitudes of A) 0 and 180, B) 90E, and C) 90W. All plots valid at 40N.

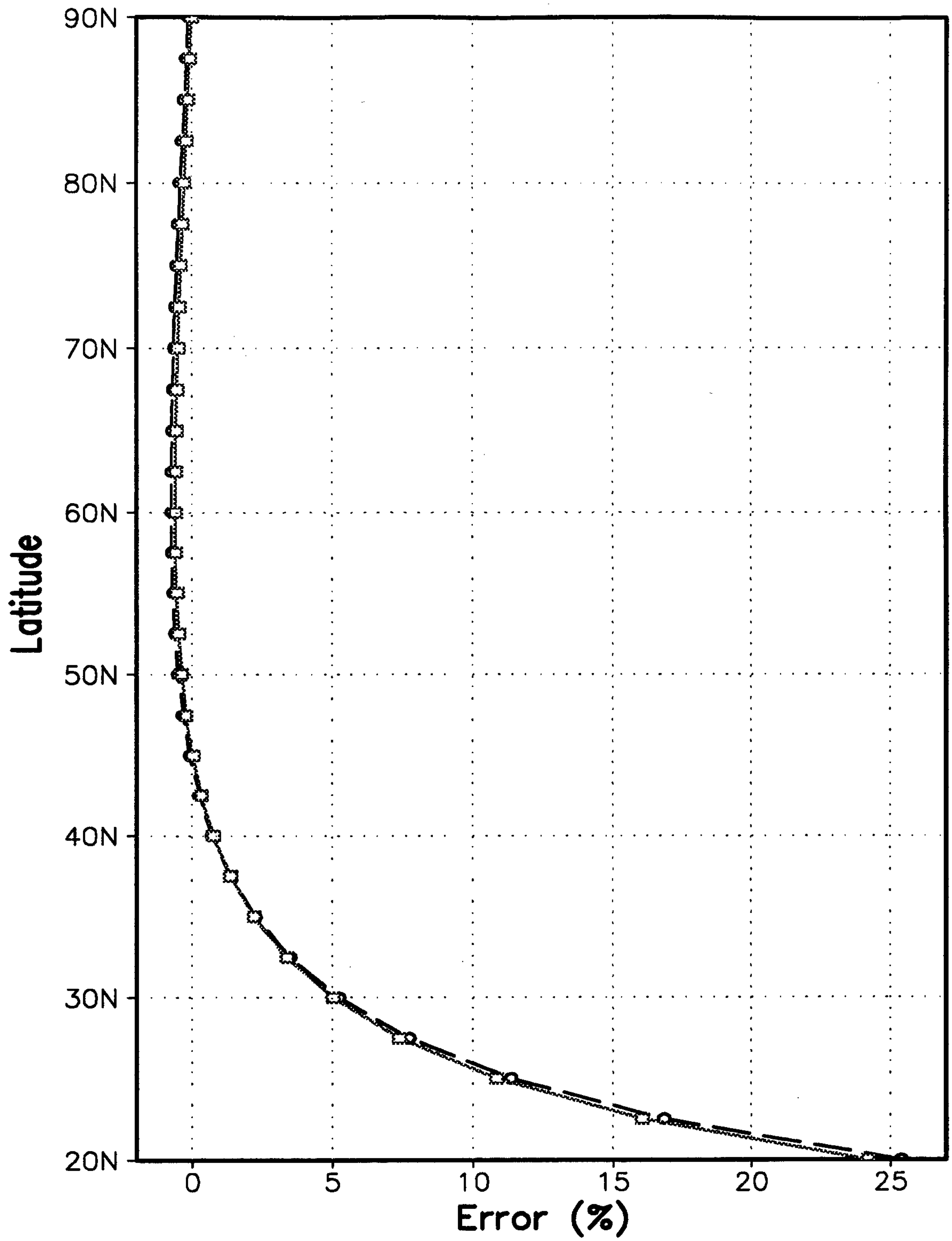


FIG. 7. Latitudinal *P* calculation errors against analytical solutions for a layered method valid between 30 and 25 kPa (dashed line, open circles) and a mandatory-level method valid at 25 kPa (solid line, open squares). Valid at 90W.

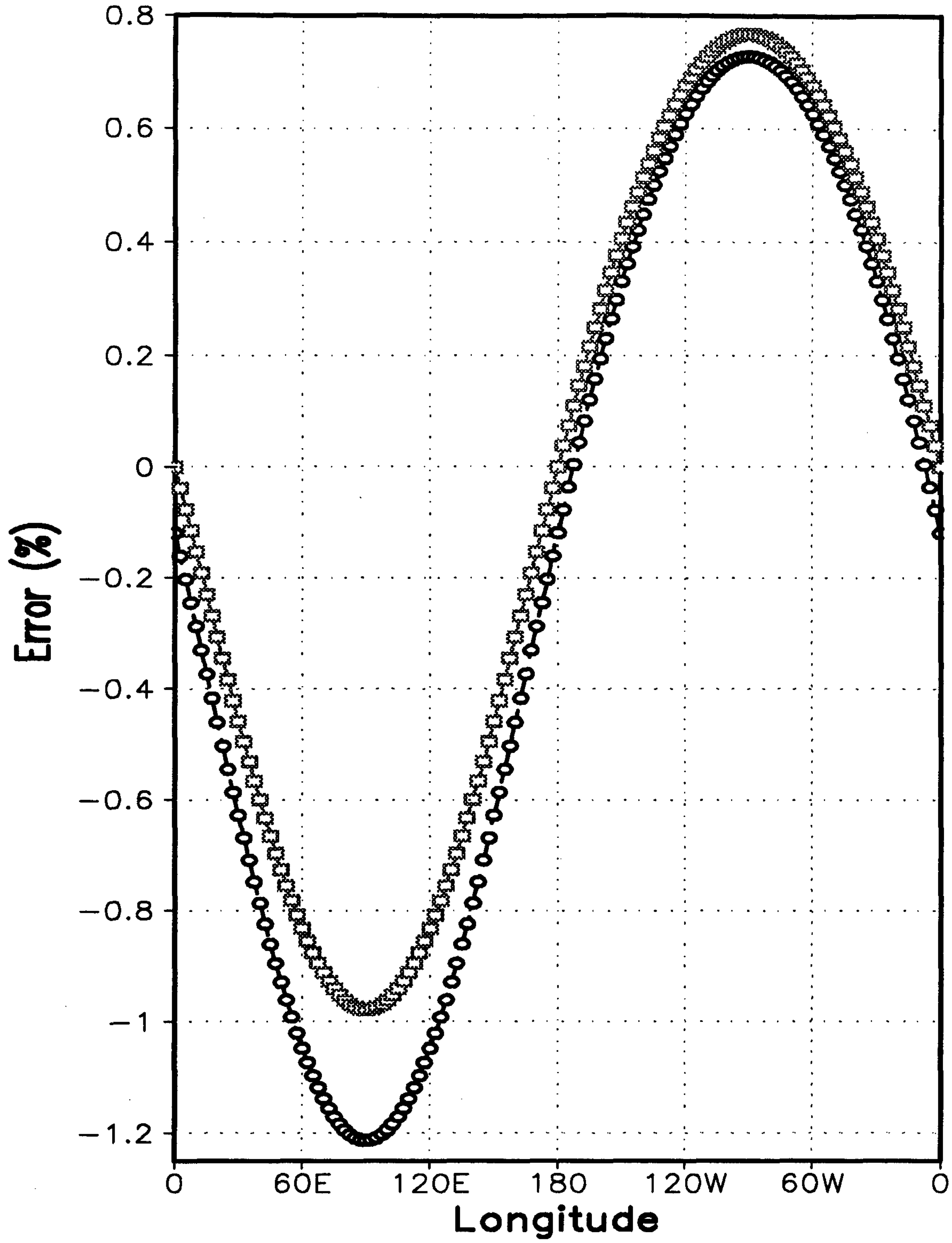


FIG. 8. Longitudinal *P* calculation errors against analytical solutions for a layered method valid between 30 and 25 kPa (dashed line, open circles) and a mandatory-level method valid at 25 kPa (solid line, open squares). Valid at 40N.

For our purposes, gravity will be assumed constant with height and latitude at a value of 9.80665 m s^{-2} (NOAA., 1976). An analysis of tropospheric gravity values indicate there is only about a three percent decrease in gravitational acceleration from 0 to 18,000 m above mean sea level (MSL). When comparing IPV values on an isentropic or isobaric surface, gravity is merely a constant weighted equally across the surface. However, gravitational variations could play a larger role in upper-atmospheric locations where isentropic surfaces are steeply sloped and actual gravitational changes may be more significant. Gravitational changes may also become more important when considering IPV variations at a mesoscale level where local gravitational variations can be resolved.

3) ISENTROPIC INTERPOLATION

An interpolation scheme to convert from pressure coordinates to isentropic coordinates was developed. In essence this approach is a two-step process following desJardins *et al.* (1996), where pressure (thus, temperature using Poisson's equation) is interpolated to isentropic coordinates, and then any other isobaric parameter can be interpolated using the determined pressure-isentropic correlation.

Since potential temperature does not always increase with height in the real atmosphere (is not always monotonic), consideration was first given to handling these superadiabatic (unstable) and neutral layers. Annual global and zonal-mean vertical lapse rates (Fig. 9a) suggest that potential temperature monotonically increases with height everywhere except near the surface at high latitudes in the Southern Hemisphere where lapse rates are negative. Therefore, superadiabatic(unstable) layers can typically be

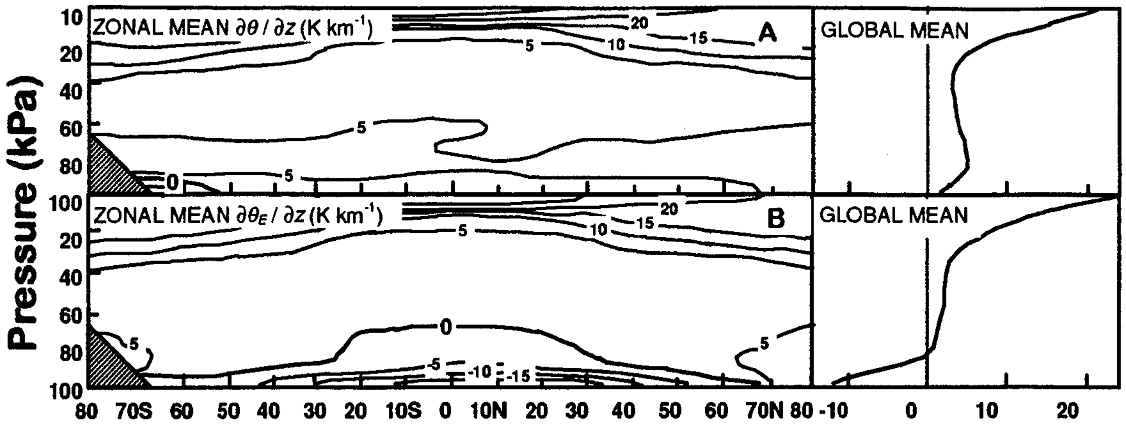


FIG. 9. Zonal-mean cross sections of the A) vertical gradient of potential temperature and B) the vertical gradient of equivalent potential temperature, θ_E, (K km^{-1}) for annual mean conditions. vertical profiles of the global mean values are shown on the right (after Peixoto and Oort, 1992).

classified as short time scale features. This data supports the validity of using potential temperature as a vertical coordinate, even at low latitudes where solar radiation is large.

If AFGWC desires to transition to EPV analysis as suggested by Zapotocny and Runk (1995), careful attention should be paid to the deviations from monotonic behavior in the vertical profile of equivalent potential temperature (or saturated potential temperature) and its validity as a vertical coordinate. Even in climatological means equivalent potential temperature exhibits a tendency to not increase monotonically. This is illustrated in Fig. 9b by the negative lapse rates. Mean equivalent potential temperature values are only monotonic above approximately 70 kPa (Peixoto and Oort, 1992). In these cases, more thought would have to be given to the validity of these variables as a vertical coordinate system.

The interpolation scheme used (Appendix D) begins with potential temperature values at the surface. The scheme analyzes successive potential temperature values vertically until it encounters a higher potential temperature value. Once found, if any lower levels were ignored because they were neutral or superadiabatic, they are assigned a potential temperature value only slightly less than the value just encountered. In essence, the scheme redefines the temperature profile in these layers to make them slightly stable by warming the uppermost layer. Therefore, this method adds potential energy to unstable or neutral layers. An alternative method (Moore, 1993) applies a cooling of the bottom layer in conjunction with a warming of the top layer. That method was not investigated in this thesis but is superior because it preserves the potential energy of the layer. Since the routines developed are not used in a prognostic manner, the overall energy balance is still maintained by the original model (MRF or NOGAPS) between forecast periods. Also, the energy balance is changed in areas where isentropic resolution is poor and diabatic effects or friction taint the adiabatic assumption--near the surface. The selected method is also not as computation intensive. To maintain the potential energy, PE, in a given layer at least two vertical iterations need to be performed changing the temperature profile in unstable and neutral layers. The second iteration is needed to adjust layers that may not begin at the surface. The energy balance can be maintained according to Haltiner and Williams (1990) by maintaining the PE in a layer, where:

$$PE = \frac{c_p}{g} \int_{p_{lower}}^{p_{upper}} T \, dp . \tag{23}$$

In either method, P is near zero in these layers due to near neutral static stability.

Fig. 10 and Table 5 indicate where superadiabatic layers for a given forecast may be found. Surprisingly, models such as the MRF appear to maintain superadiabatic lapse rates (despite their instability) well into the forecast cycle, probably to parameterize convection cycles. Generally, these layers exist in warm boundary layers near the surface, such as daytime deserts, or above warm ocean waters. Most superadiabatic layers dissipate once the effects of surface heating are diminished. This results in potential temperature being a very good vertical coordinate (monotonic) at pressures less than 70 kPa.

TABLE 5. Superadiabatic layers identified from mandatory-level data from the MRF 108-hour forecast valid 1200 UTC 17 September 1996.

Pressure Layer (kPa)	No. of grid points with superadiabatic lapse rates (65,160 at each level)	Percent of grid points
Surface – 100	15,303	23.5
100* – 92.5	8,453	13.0
92.5* – 85	3,661	5.6
85* – 70	1,902	2.9
70* – 50	157	0.2
50* – 40	61	0.1
40* – 30	0	0.0
30 – 25	14	0.0**
Above 25	0	0.0

*Lower layer boundary may be the surface if lower level indicated lies below the surface.

**Less than 5 /100th of one percent.

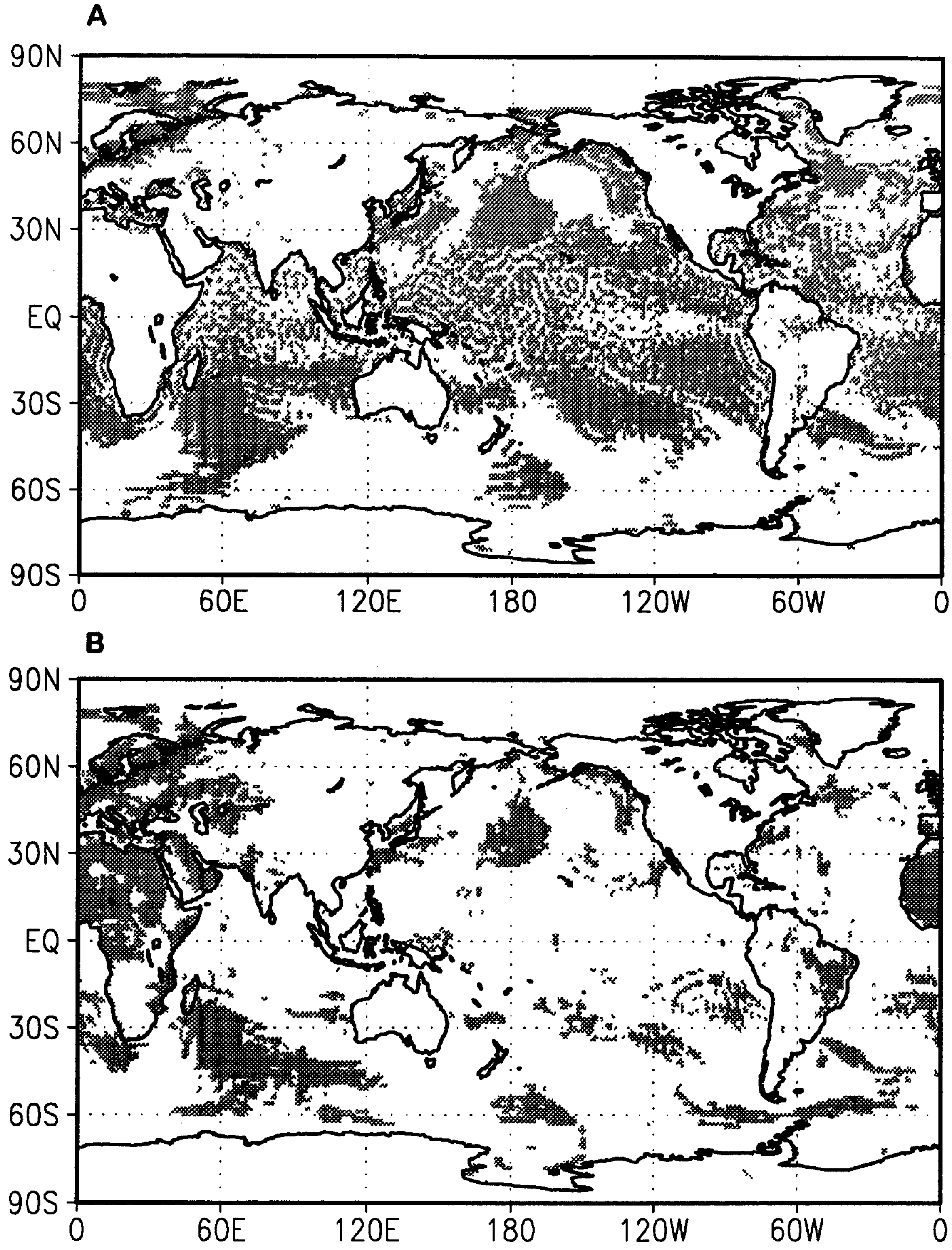
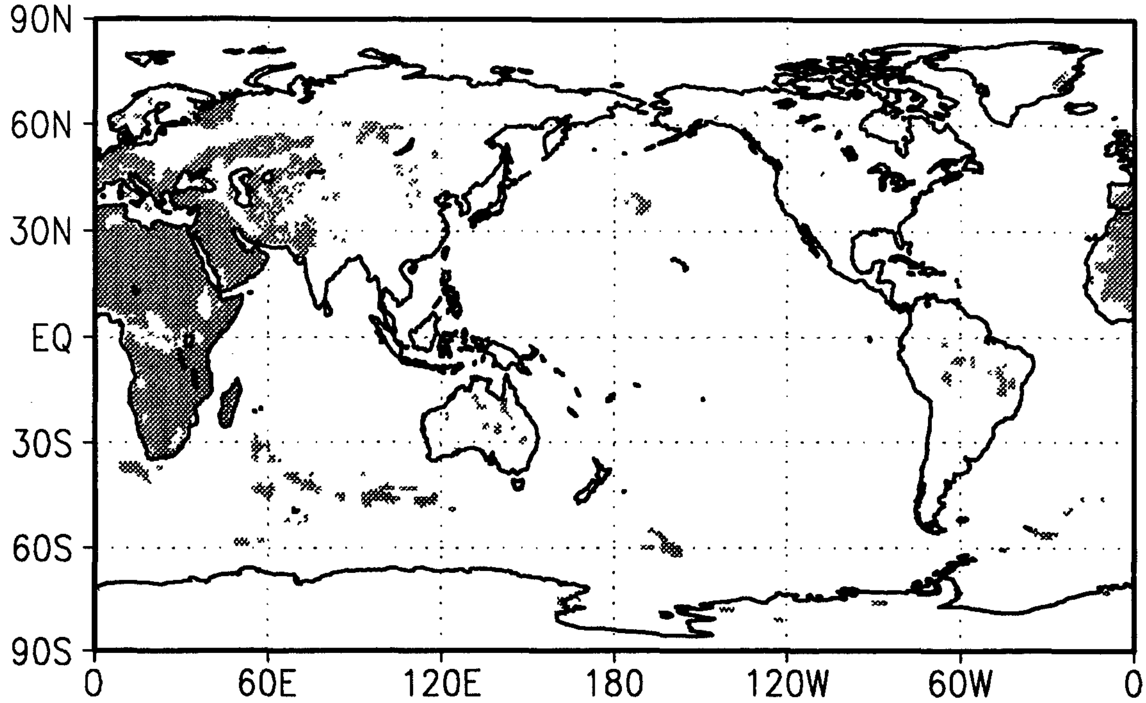


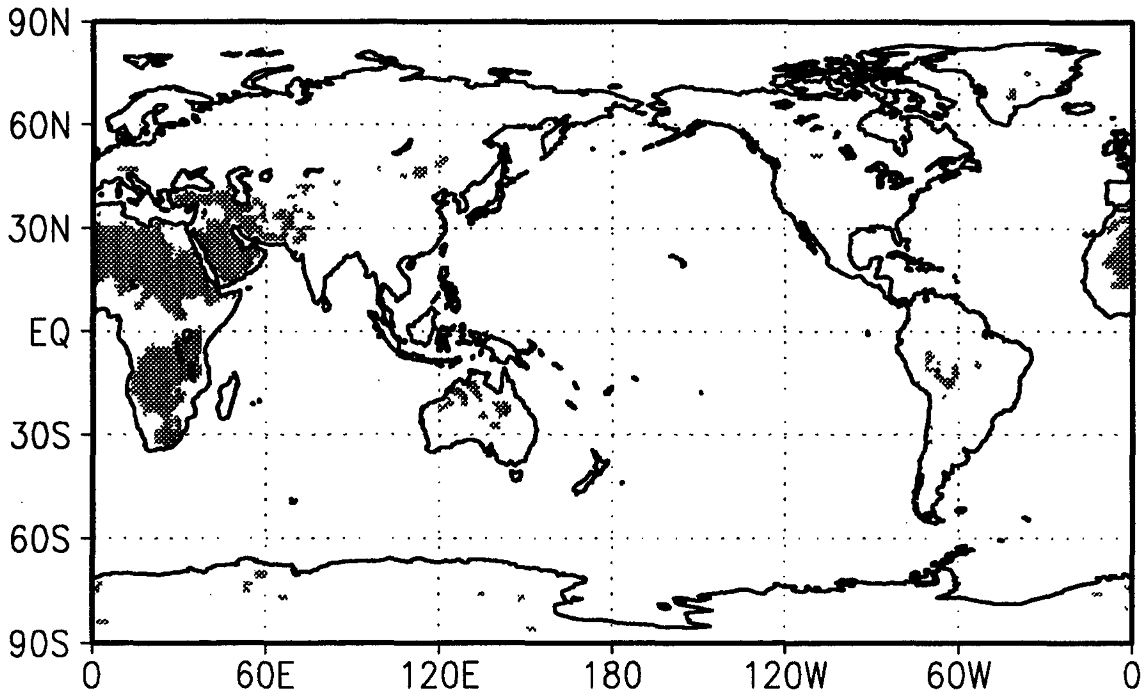
FIG. 10A-E. Grid points (shaded) indicating existence of a superadiabatic layer between A) Surface – 100 kPa, B) 100 – 92.5 kPa, C) 92.5 – 85 kPa, D) 85 – 70 kPa, and E) 70 – 50 kPa. Data is from MRF 108-hour forecast valid 1200 UTC 17 September 1996.

C



90N
60N
30N
EQ
30S
60S
90S
0
60E
120E
180
120W
60W
0

D



90N
60N
30N
EQ
30S
60S
90S
0
60E
120E
180
120W
60W
0

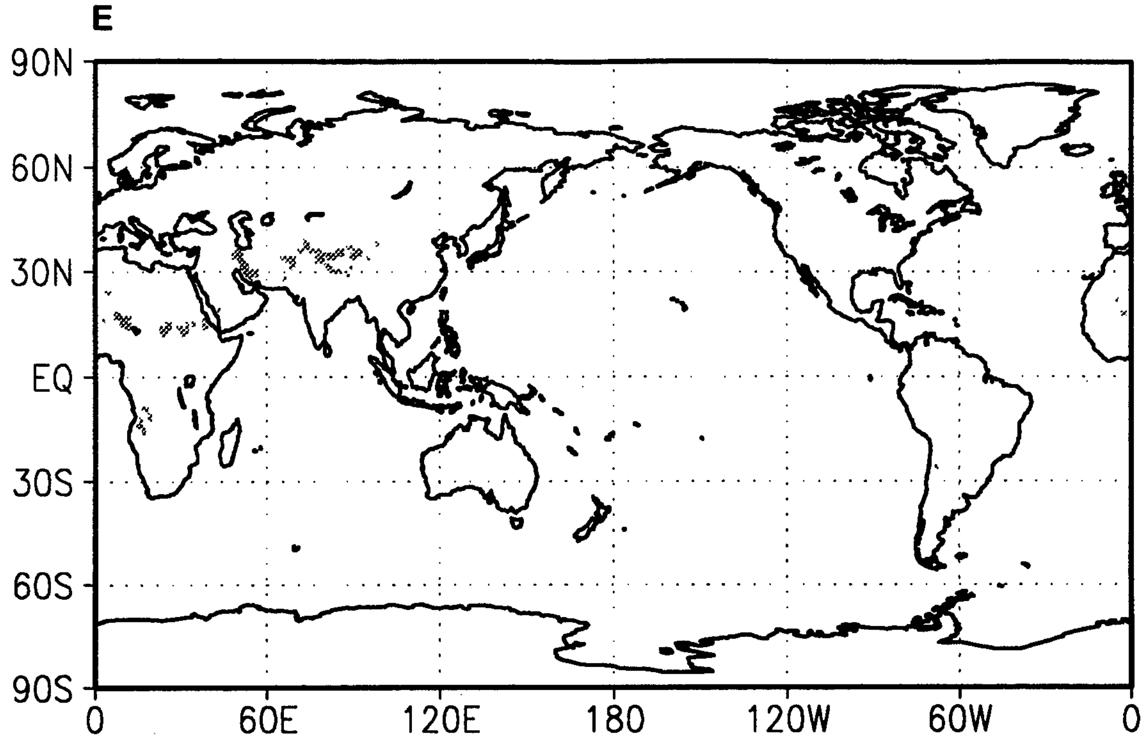


Fig. 10c clearly shows an example of the effects of daytime heating over the African continent. You can expect that 0000 UTC model data would exhibit an increase in superadiabatic layers over the Americas and a decrease over Africa due to diurnal heating effects. This choice of handling superadiabatic layers should not result in problems since resolution on isentropic surfaces near the surface isn't nearly as good as in the upper troposphere. In addition, the effects of friction near the surface invalidate conservation of IPV here, and the intersection of isentropes with the Earth's surface further complicate analysis. Superadiabatic areas are also found occasionally just below the tropopause inversion. This is indicated by the 14 grid points between 30 and 25 kPa and also appears in Fig. 15.

Because of potential operational use, values for all data below the surface is depicted as missing (-9999.0). This provides a feeling for where the effects of friction and the Earth's surface may need to be taken into account, because the surface is not hidden from the data.

When interpolating pressure to isentropic coordinates, we implicitly use temperature through Poisson's equation, equation (1). GEMPAK (desJardins *et al.*, 1996) performs an interpolation assuming that T varies linearly with $\ln p$. However, since Table 3 and Table 4 previously indicated there may be a slight advantage gained by modifying the GEMPAK interpolation scheme, the routines were developed under the assumption that $\ln T$ varies linearly with $\ln p$.

Using this temperature interpolation assumption, the pressure value at a point valid for a given isentropic level was narrowed using a Newton iteration method (Kreyszig, 1993) in conjunction with the definition of potential temperature. Using this method, the pressure value for the nth iteration is given by:

$$p_n = p_{n-1} - \frac{T_{n-1} - \theta\left(\frac{p_{n-1}}{p_0}\right)^{\frac{R_d}{c_p}}}{\left(\frac{dT}{dp} + \frac{R_d T}{c_p p_{n-1}}\right)}. \tag{24}$$

As the approximation approaches the actual value, the numerator approaches zero and physically satisfies Poisson's equation. The denominator is simply the derivative of the numerator. In some cases the restraints put on p and T by the presupposed relation result in pressure values that don't converge within 1.0 Pa.

Assuming ln T is a linear function of ln p in equation (24), $T = \exp(b)\, p^m$, where m represents the slope and b is the intercept of the aforementioned linear relationship when given data at two points. In order to preserve the vertical coordinate system, temperature values in equation (24) may deviate from the actual observed data in superadiabatic or neutral layers. Temperature values used in the interpolation are described using Poisson's equation with actual pressure values and revised potential temperature values. The revised temperature profile is a result of making all superadiabatic and neutral layers slightly stable (see Appendix D).

To determine the optimum number of Newton iterations to perform, an analysis of actual data and the associated residual errors were performed. Using the MRF 84-hour forecast data valid 1200 UTC on 16 September 1996, Table 6 shows that to an accuracy of 1.0 Pa, no further convergence of p_n occurs after $n = 2$ iterations. The maximum

TABLE 6. Convergence of p to within 1.0 Pa for grid points from the 84-hour MRF forecast valid 1200 UTC 16 September 1996. Interpolation resolution set to 5 K

n	Points converging	Percent	Max Residual Error (Pa)
0	1,817,700	99.99	10151.11
1	65	0.00*	31.67
2	3	0.00*	32.48
3	0	0.00	67.97
4	0	0.00	70.73
5	0	0.00	64.39
6	0	0.00	32.47
7	0	0.00	31.70
8	0	0.00	67.96
9	0	0.00	64.39
Did not converge	70	0.00*	

*Less than 5 /1000th of one percent

residual error shows only oscillatory effects after the first iteration, with maximum residual errors less than 100 Pa. Since the original model data is only reported to the nearest 10 Pa, these residual errors are well within tolerable limits. Since processing time was not a factor in development *n* is set to a maximum of 5 iterations in Appendix D. But, if processing time is at a premium, one iteration seems to converge over 99.99 percent of the data points and obtain reasonable accuracy.

If *n* is decreased, the possibility of obtaining pressure values that increase with isentropic heights in areas of high stability (such as above the tropopause) exists. If this occurs, the code at Appendix D decrements pressure vertically by 0.1 Pa between isentropic surfaces in order to ensure that pressure does not increase with geometric height. Therefore, a non-convergent grid point could potentially perturb data points above it, by ensuring that pressure values continue to decrease vertically. However, a vertical *ripple* will usually only occur if the isentropic resolution is very high (not recommended if originating from isobaric data) or in areas of very high stability such as in the stratosphere.

Once pressure data has been interpolated, an interpolation of any other scalar to isentropic coordinates can be performed. However, temperature data is implied from Poisson's equation and should not be interpolated because of the adiabatic assumptions made in superadiabatic layers. A separate interpolation of temperature was part of the historical reason for an original degradation of the validity of isentropic analysis until the error was discovered by Danielson in 1959 (Moore, 1993).

The routine for the scalar conversion (Appendix F) extracts pressure and scalar data at the three nearest mandatory levels (including the surface) and performs a quadratic interpolation of the scalar (Kreyszig, 1993) versus ln p from the previously interpolated isentropic pressure, p_θ. The equation to interpolate any given scalar, s, at a given isentropic grid point is:

$$s_\theta = s_{p_1} \frac{\ln\frac{p_\theta}{p_2}\ln\frac{p_\theta}{p_3}}{\ln\frac{p_1}{p_2}\ln\frac{p_1}{p_3}} + s_{p_2} \frac{\ln\frac{p_\theta}{p_1}\ln\frac{p_\theta}{p_3}}{\ln\frac{p_2}{p_1}\ln\frac{p_2}{p_3}} + s_{p_3} \frac{\ln\frac{p_\theta}{p_1}\ln\frac{p_\theta}{p_2}}{\ln\frac{p_3}{p_1}\ln\frac{p_3}{p_2}} \qquad (25)$$

where p_1, p_2, and p_3 are pressures at a lower, middle and upper isobaric level, respectively, in relation to the isentropic surface. Furthermore, s_{p_1}, s_{p_2}, and s_{p_3} correspond to the mandatory-level isobaric scalar values at the pressure level of the respective subscript. Except at the uppermost levels, the levels used for interpolation are the nearest lower level and the nearest two upper levels. Therefore, there is a slightly larger influence by data above rather than below a given point. For this reason a cubic interpolation using two levels above and two levels below a given point may need to be considered further. A method similar to the one performed in obtaining the results from Table 3 and Table 4 is recommended. A crude visual analysis interpolation of u wind data from the AVN 24-hour forecast valid 0000 UTC 15 November 1996 at 90N between 25 and 10 kPa (Fig. 11) indicates that a cubic interpolation using the nearest four levels of data may not result in any appreciably significant smoothing, considering we know nothing about the true vertical distribution between mandatory levels. The data chosen purposely

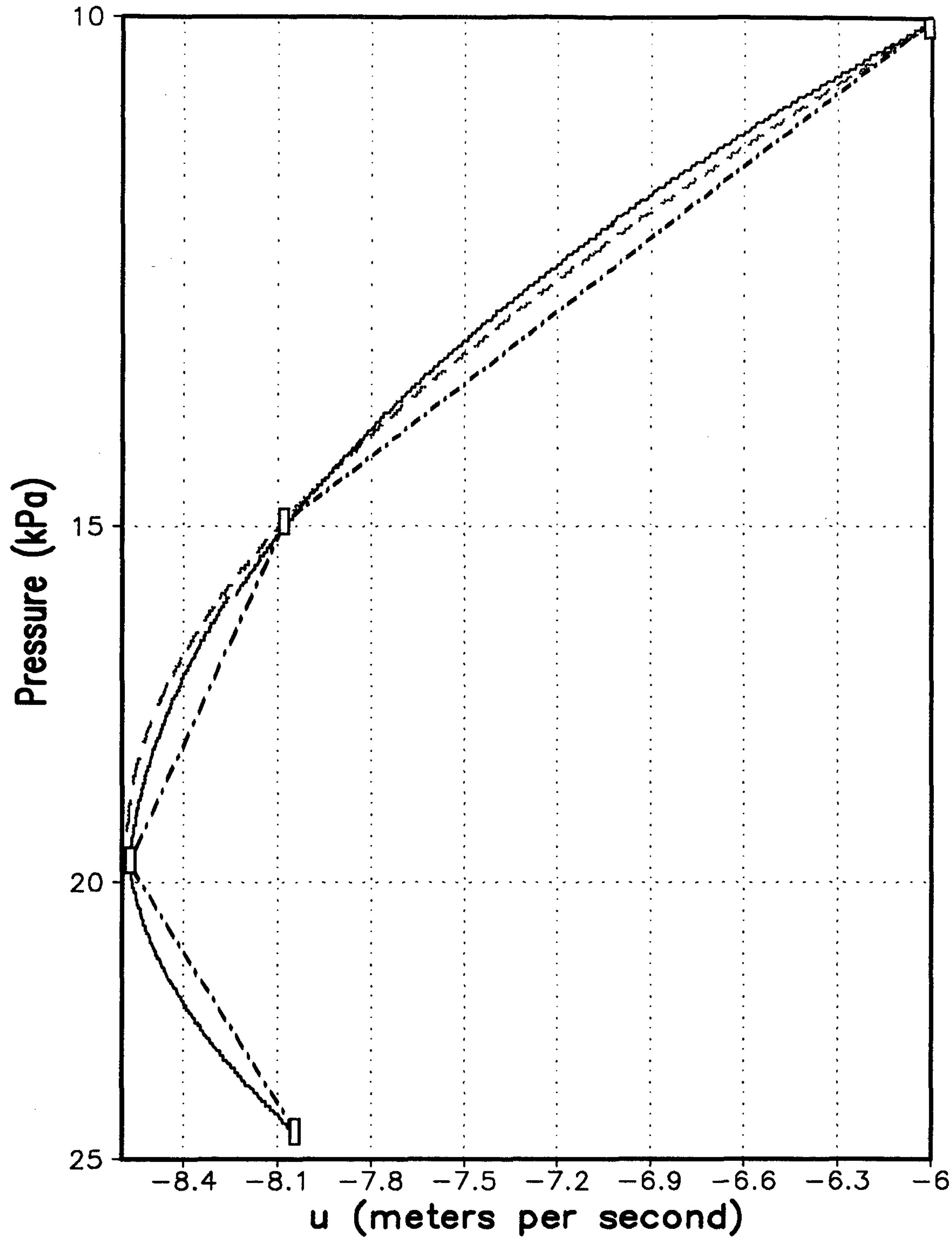


FIG. 11. Vertical interpolation of *u* wind component at 90N from 25 to 10 kPa using linear interpolation (dash-dot), quadratic interpolation from nearest lower level and nearest two upper levels, or uppermost three levels (15 and 10 kPa) (solid line), and cubic interpolation using all four data levels (dotted line). Data from AVN 24-hour forecast valid 0000 UTC 15 November 1996.

has a relative minimum to try to exaggerate the interpolation effects. Although smoothing through vertical discontinuities may not be physically representative of real-world data where inversions may result in strong discontinuities, it should be more representative of the more well-behaved model output in use. In Fig. 11, the method used has a vaguely noticeable discontinuity at 20 kPa. The cubic interpolation shown is only valid above 20 kPa.

Part of the consideration when performing a vertical interpolation from isobaric or sigma coordinates to isentropic coordinates includes determining the optimum isentropic thickness. As shown earlier with the hypothetical potential vorticity field in isobaric coordinates, vertical resolution can become relevant when computing the static stability. Past work has referenced isentropic vertical resolution anywhere from 40 K (Platzman, 1949) to 4 K (Starr and Neiburger, 1940). Of course, if computing power and time were not factors, the finer the resolution the better. An analysis using various resolutions was performed in order to determine an optimum point where reducing the isentropic thickness results in no additional information when generated from mandatory-level pressure data.

Table 7 shows the results of an analysis of actual isentropic resolution from mandatory-level data in the middle to upper troposphere from 50 to 10 kPa. The median isentropic thickness between these isobaric levels appears to be near 10 K. However, in order to have a vertical resolution at least comparable to the original mandatory-level data (at least one isentropic surface between mandatory pressure levels) between 90 percent of these grid points, the preferred isentropic thickness would be roughly 4 K. Based on this, a 5 K resolution is used in the program at Appendix D. As a result of this selection, a

TABLE 7. Analysis of isentropic thicknesses between six layers of mandatory-level pressure data from 50 to 10 kPa. Data from 84-hour MRF forecast valid 1200 UTC 16 September 1996.

Isentropic layer thickness between mandatory levels (K)	No. of isobaric grid points (percent)	No. of isentropic levels from 300 to 400 K if interpolated at given thickness
< 1	341 (0.1%)	101
< 2	2,494 (0.6%)	50
< 3	12,426 (3.2%)	33
< 4	37,975 (9.7%)	25
< 5	70,260 (18.0%)	20
< 6	104,951 (26.8%)	16
< 7	135,983 (34.8%)	14
< 8	163,699 (41.9%)	12
< 9	185,988 (47.6%)	11
< 10	202,146 (51.7%)	10
< 20	275,957 (70.6%)	5

*Less than 5 /100th of one percent

typical analysis from 300 to 400 K, using data from 85 to 10 kPa, requires an increase from the original 9 isobaric levels to 20 isentropic levels, approximately doubling the original database. This confirms the Hoskins *et al.* (1985) revelation that an isentropic analysis from mandatory-level isobaric data is a coarse-grain resolution, at best. Using the approximate median thickness previously mentioned (assuming a normal distribution of thicknesses), we can assess that the actual vertical resolution of our analysis is probably on the order of 10 K, despite a selected isentropic separation of 5 K.

A visual analysis of the pressure interpolation is shown in the cross section at Fig. 12. The adequacy of choosing the 5 K resolution can be seen near 60N along 30 kPa, near 10N along 15 kPa, and at 35S along 20 kPa. Despite the 5-fold increase in processing the 1 K resolution, very little change is noticed from the 5 K resolution. For our purposes the

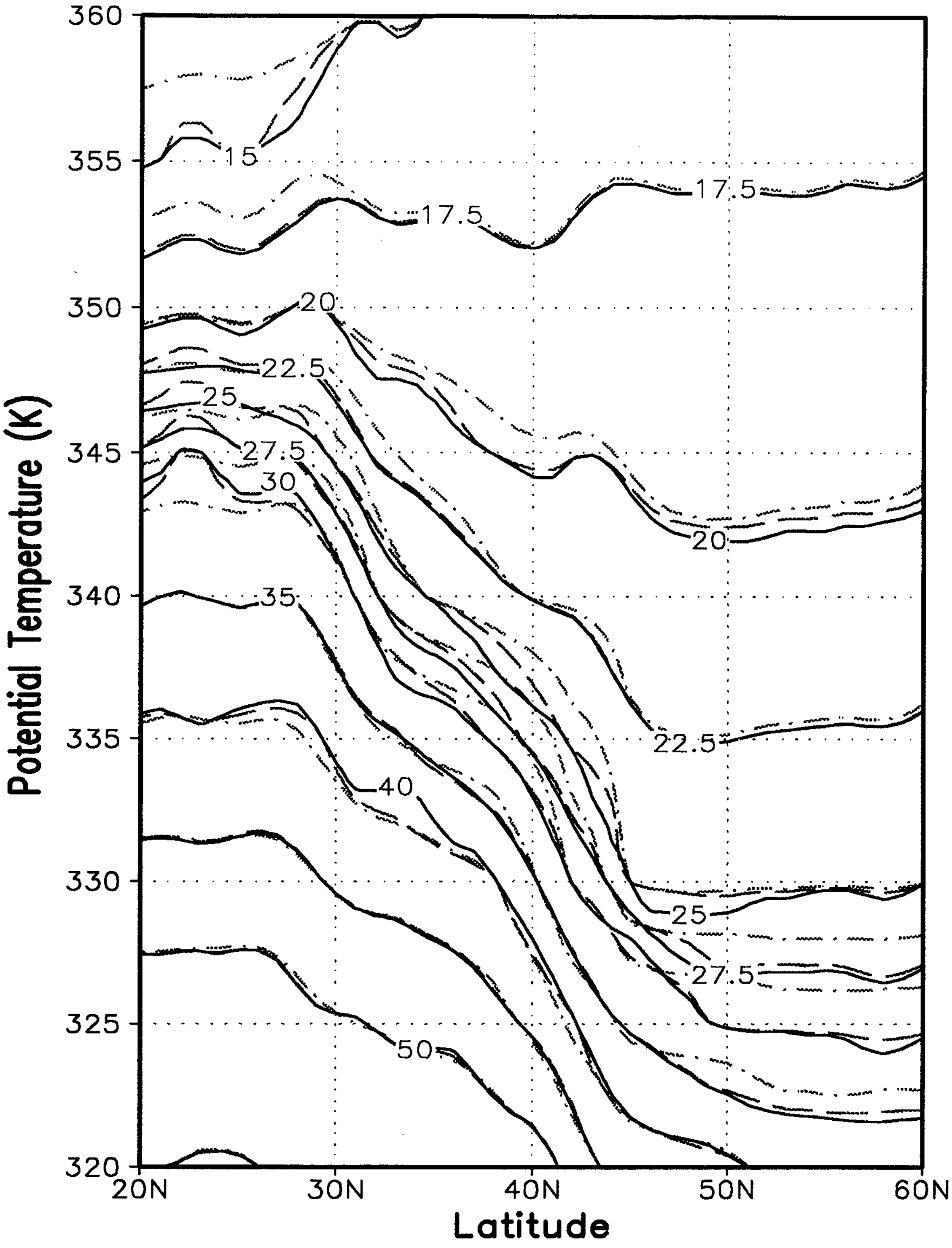


FIG. 12. Isentropic pressure (kPa) interpolation at 10 K (dash-dot line), 5 K (dashed line) and 1 K (solid line) resolutions. Data from 84-hour MRF forecast valid 1200 UTC 16 September 1996. Valid at 95W.

1 K resolution can be considered truth since a similar construction at 2 through 10 K showed continual convergence toward the 1 K resolution. Fig. 13 shows how different isentropic resolutions may effect the IPV field when calculated after wind and pressure data are interpolated to isentropic coordinates. The higher vertical pressure gradient on the 10 K interpolation at 60N between 30 and 20 kPa in Fig. 12 is evident in the higher IPV value shown in Fig. 13 at the 3.0 PVU contour.

To determine an optimum implementation sequence in calculating IPV, an investigation was performed to look at the differences between calculating IPV on pressure surfaces followed by an interpolation to isentropic coordinates as done by Hoskins *et al.* (1985), Davis and Emanuel (1991), and later by Davis (1992), and interpolating wind and pressure to isentropic coordinates and then calculating IPV. To do this, a comparison was performed using the previously mentioned analytic function.

With results similar to the comparison of the layered to mandatory-level IPV calculation, Fig. 14 shows that it is preferable to interpolate the pressure and wind variables to isentropic coordinate prior to calculating IPV valid on an isentropic surface. The errors for the preferred method were again more well-behaved with a lower mean latitudinal error. These results were used in determining the order of interpolation operations shown in the main program at Appendix A.

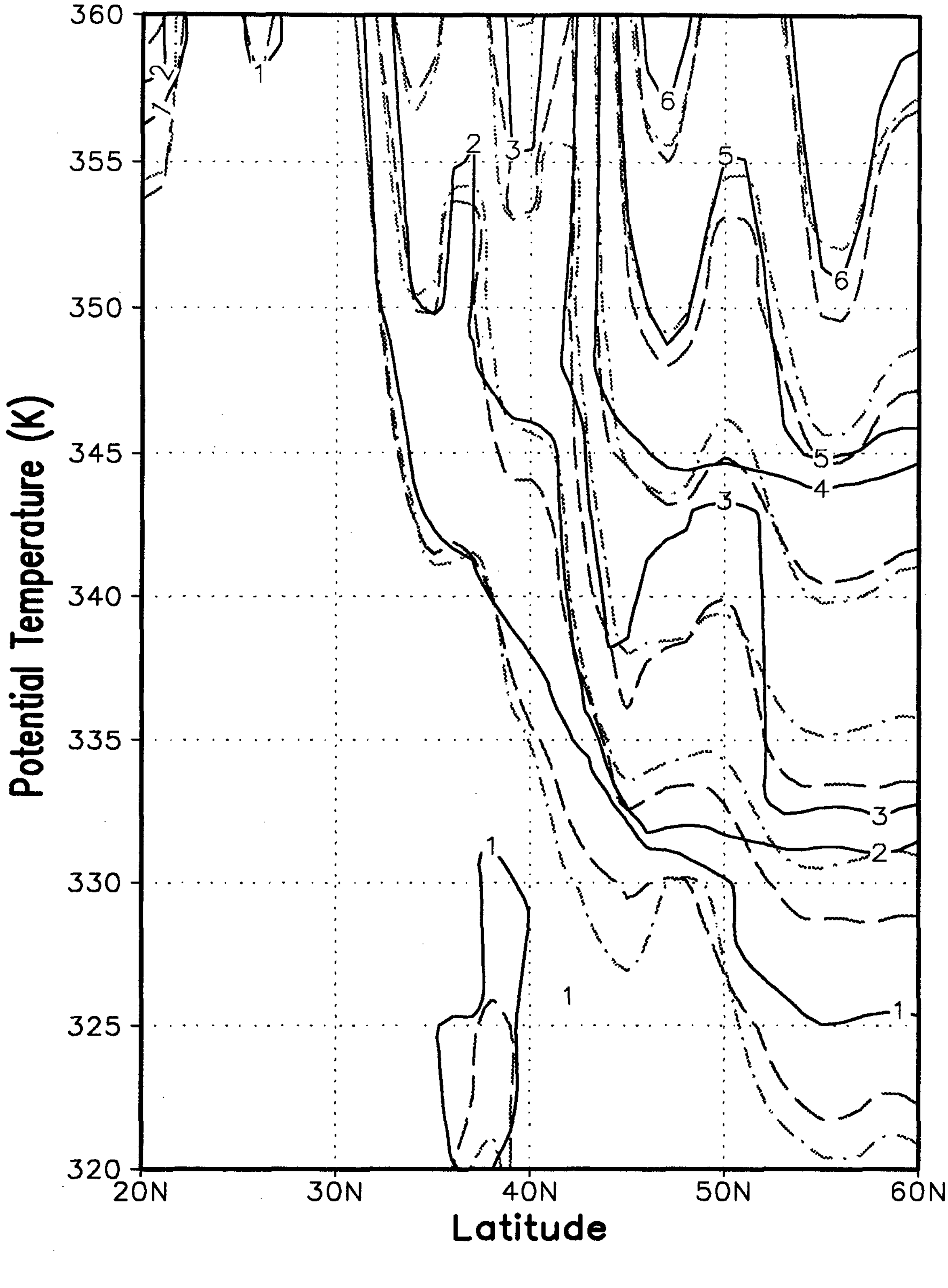


FIG. 13. Isentropic potential vorticity (PVU ≡ 10^{-6} m^2 K kg^{-1} s^{-1}) analysis from data interpolated at 10 K (dash-dot line), 5 K (dashed line) and 1 K (solid line) resolutions. Data from 84-hour MRF forecast valid 1200 UTC 16 September 1996. Valid at 95W.

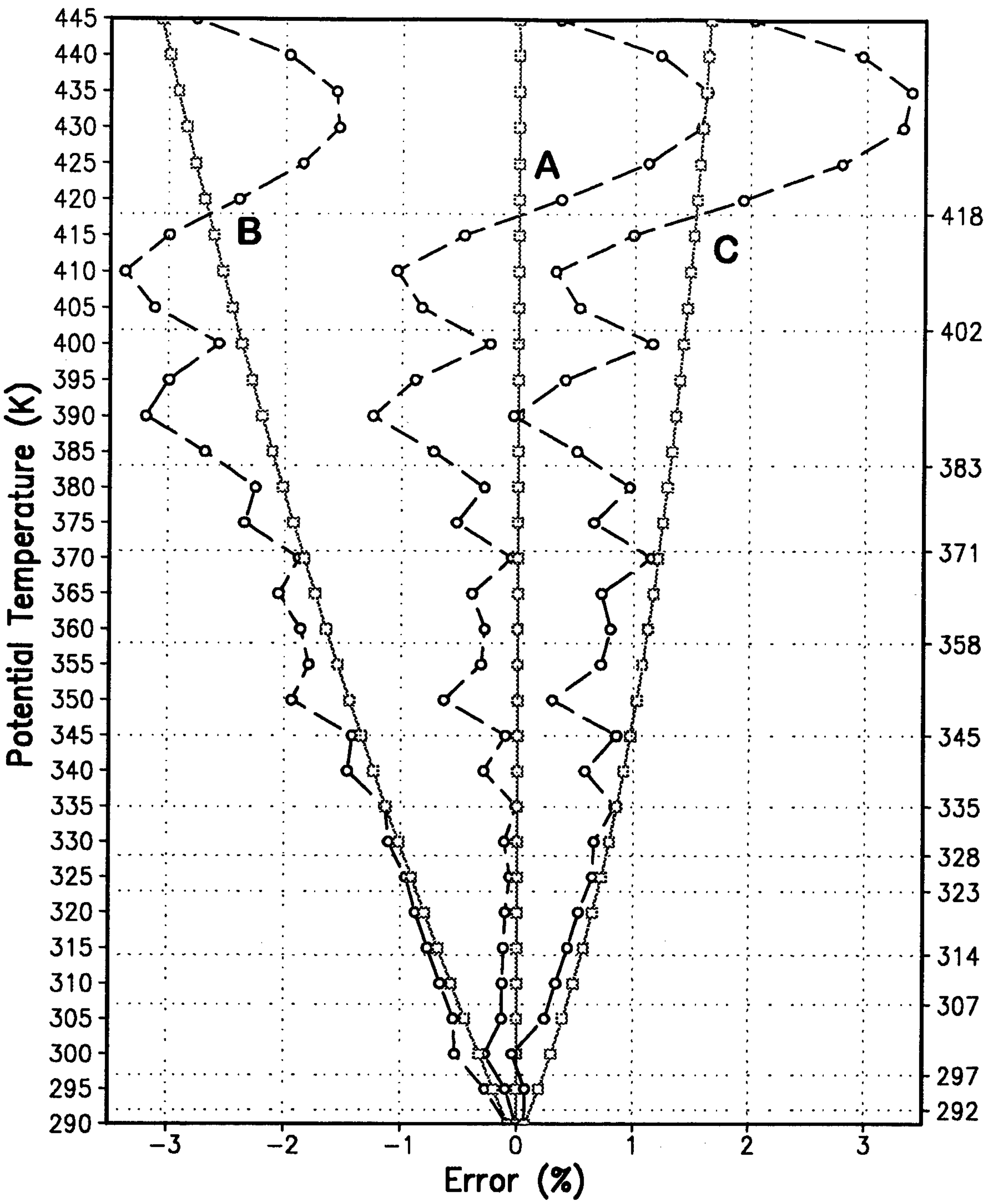


FIG. 14. Calculation error against analytical solutions for values of P_θ interpolated from isobaric to isentropic coordinates (dashed line, open circles) and values of P_θ calculated from pressure and wind data interpolated from isobaric to isentropic coordinates (solid line, open squares) at longitudes of A) 0 and 180, B) 90E, and C) 90W. All plots valid at 40N. Potential temperature values on the right are representative of the mandatory-level pressure surfaces.

4. Applications of IPV and other isentropic products

The isentropic products and application techniques shown in this section are recommended to complement, not replace, existing AFGWC products and techniques. It is recommended that the routines at the appendices be used to create isentropic, isobaric, and vertical cross sections. From these charts forecasters can identify tropopause features, cyclogenesis regions, and upper-level and surface fronts. Also, visualization of synoptic-scale vertical motions can provide additional information for forecasting events pertinent to air operations, such as freezing rain and icing. These routines produce gridded fields that can allow AFGWC to produce and view products and apply techniques and theory similar to those recommended for use by National Weather Service forecasters (Moore, 1993).

a. Limitations and considerations

In order to effectively use IPV data produced from the developed routines, it is important to understand not only the inherent advantages already mentioned earlier; but, to also understand the weaknesses of isentropic analysis and the developed IPV algorithm. The largest inherent problems with isentropic charts (Carlson, 1991; Moore, 1993) include:

(a) the atmosphere is not completely adiabatic, especially in the boundary layer and in the vicinity of strong vertical mixing or convection;

(b) Strong diurnal radiational changes in the boundary layer disrupt the continuity of analysis;

(c) isentropic surfaces may intersect the ground;

(d) isentropic surfaces extend from low to high levels in the atmosphere and thereby do not represent a horizontal surface; and,

(e) meteorologists are unaccustomed to interpreting isentropic weather maps.

To diminish the effects of diurnal oscillations, consideration should be given to maintaining isentropic continuity on a 24-hour cycle instead of the typical 12-hour cycle. This will inherently be done with the MRF since the model only produces output on a daily cycle.

As previously discussed, Hoskins *et al.* (1985) notes that the largest problem inherent in these IPV calculations is the fact that the data was originally analyzed isobarically rather than isentropically. Therefore the data is, at best, a coarse-grain approximation. These inherent weaknesses require a conscientious choice of appropriate isentropic surfaces depending on the types of analysis to be performed or the features of interest.

b. Application

Before isentropic analysis is performed, appropriate isentropic levels need to be chosen. Some of the guess work in selecting proper isentropic levels to analyze has been automatically eliminated by the interpolation routine. The routine begins performing interpolation from isobaric to isentropic coordinates once ten percent (by grid point count) of an isentropic surface is above the surface. Generally this lower potential temperature surface is near 260 K. This value should remain fairly consistent during a global analysis. However, annual and diurnal effects may change the value of this bottom level. Seasonal climatology in specific areas of interest may also be used to aid in determining changes in isentropic levels. The routine then interpolates for 50 levels at 5 K increments. The result is isentropic grids roughly up to 500 K. Namias suggests the lowest isentropic levels to use for analysis by season as shown in Table 8 (Moore, 1993).

A vertical cross section as shown in Fig. 15 is initially recommended to aid in identifying the best isentropic levels to contour in a region, or to identify tropopause positions. This is similar to Fig. 13, but uses an isobaric vertical scale and data from isobaric IPV algorithm at Appendix L. This product can easily be incorporated into isobaric analyses. In addition, this type of product (produced from constant pressure data), may easily be implemented locally using the Air Force's Automated Weather Distribution System (AWDS).

Fig. 16 and Fig. 17 represent an isobaric analysis at 50 kPa of absolute vorticity and potential vorticity, respectively. Since both are initially derived from the absolute vorticity field they are almost identical; however, the potential vorticity field also carries with it information about the static stability and thus the depth of a disturbance (Bluestein, 1993). Most interestingly, the features at 110W, 55N and 110W, 43N have higher relative values of absolute vorticity than those on the potential vorticity chart, suggesting the vertical extent of these disturbances may be limited. Conversely, the feature near the Gulf Coast

TABLE 8. Suggested lowest isentropic analysis level by season

Season	Lowest isentropic level (K)
Winter	290
Spring	295
Summer	310
Fall	300

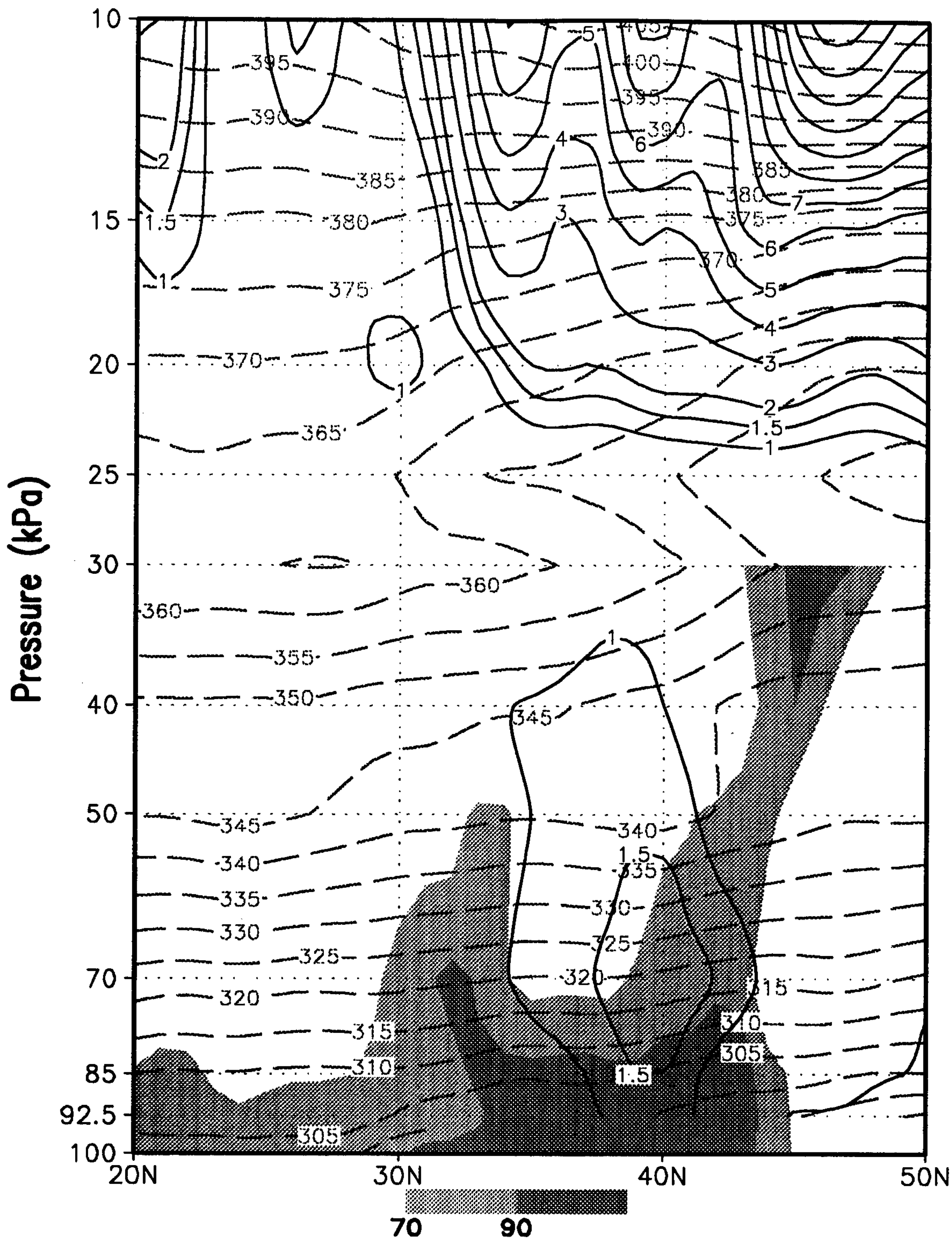


FIG. 15. Potential vorticity cross section (solid lines, PVU ≡ 10^{-6} m^2 K kg^{-1} s^{-1}), potential temperature (dashed lines, K), and relative humidity (shaded at 70 and 90 percent). Cross section valid at 95W from MRF 84-hour forecast valid 1200 UTC 16 September 1996.

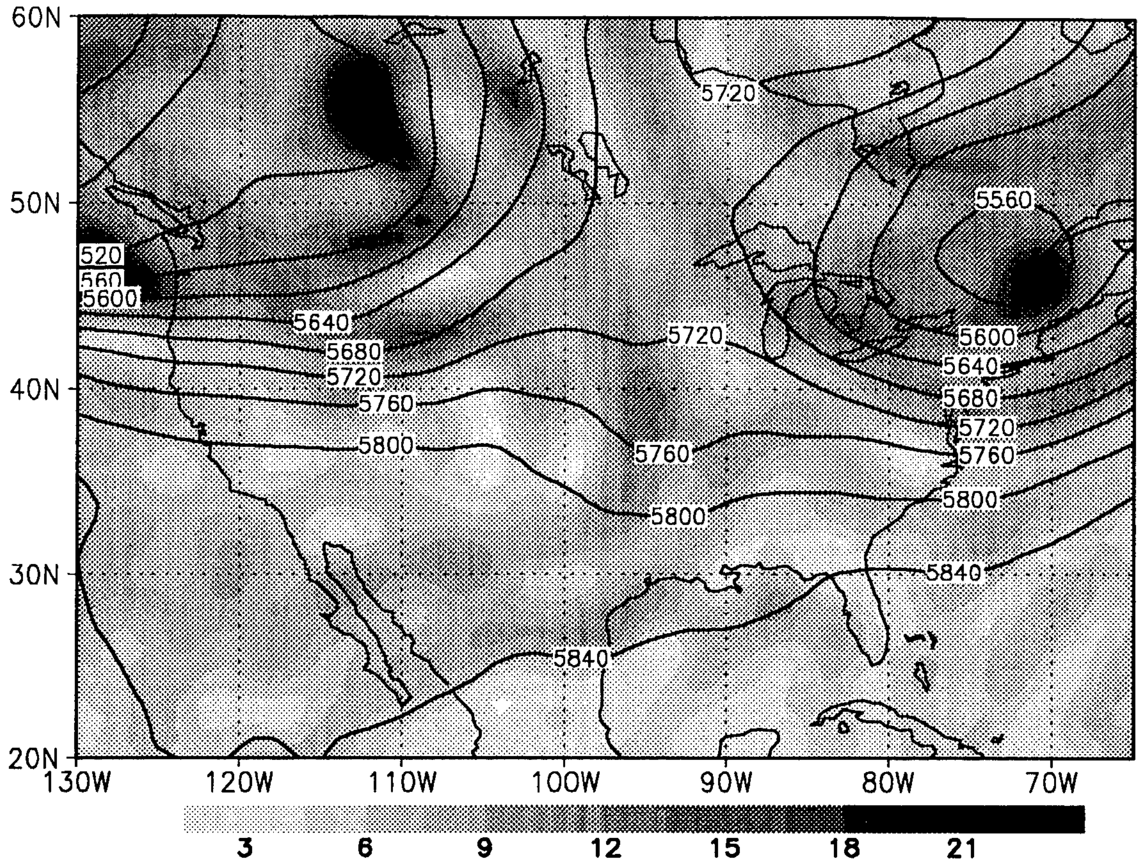


FIG. 16. Absolute vorticity field (shaded, 10^{-5} s^{-1}) and geopotential height (geopotential meters, gpm) at 50 kPa from MRF 84-hour forecast valid 1200 UTC 16 September 1996.

at 88W, 28N has higher relative values of potential vorticity indicating that low static stability may increase the vertical extent of this disturbance.

The Montgomery streamfunction, ψ, is analogous to geopotential in isobaric coordinates; pure adiabatic, frictionless, geostrophic flow on an isentropic surface runs parallel to the streamfunction. The Montgomery streamfunction is defined as:

$$\psi = c_p T + \Phi . \qquad (26)$$

Ageostrophic motions in the vicinity of the entrance and exit regions of jet streaks can

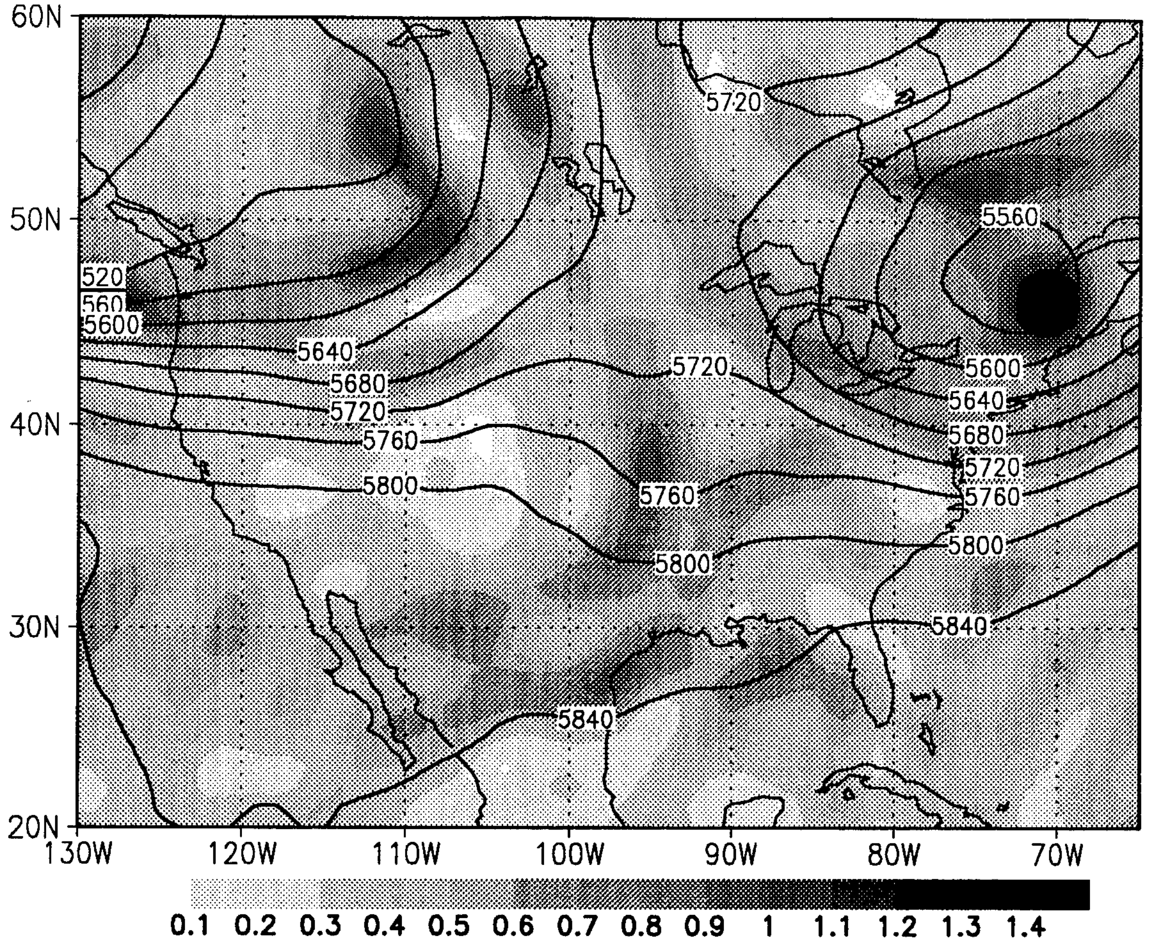


FIG. 17. Potential vorticity field (shaded, PVU ≡ 10^{-6} m^2 K kg^{-1} s^{-1}) and geopotential height (gpm) at 50 kPa from MRF 84-hour forecast valid 1200 UTC 16 September 1996.

easily be spotted and used to help identify regions of probable cyclogenesis or cyclolysis. Fig. 18 shows the relation between the wind field and the Montgomery streamfunction.

Fig. 19 represents a typical isentropic product, often referred to as a *psi* chart (psi refers to ψ). When an isentropic analysis includes pressure (synonymous with temperature on isentropic surfaces) information, vertical motion (and temperature advection) can easily be deduced. Standard analysis increments for psi charts are given by Moore (1993). When accompanied by moisture fields, it becomes easy to deduce areas of precipitation,

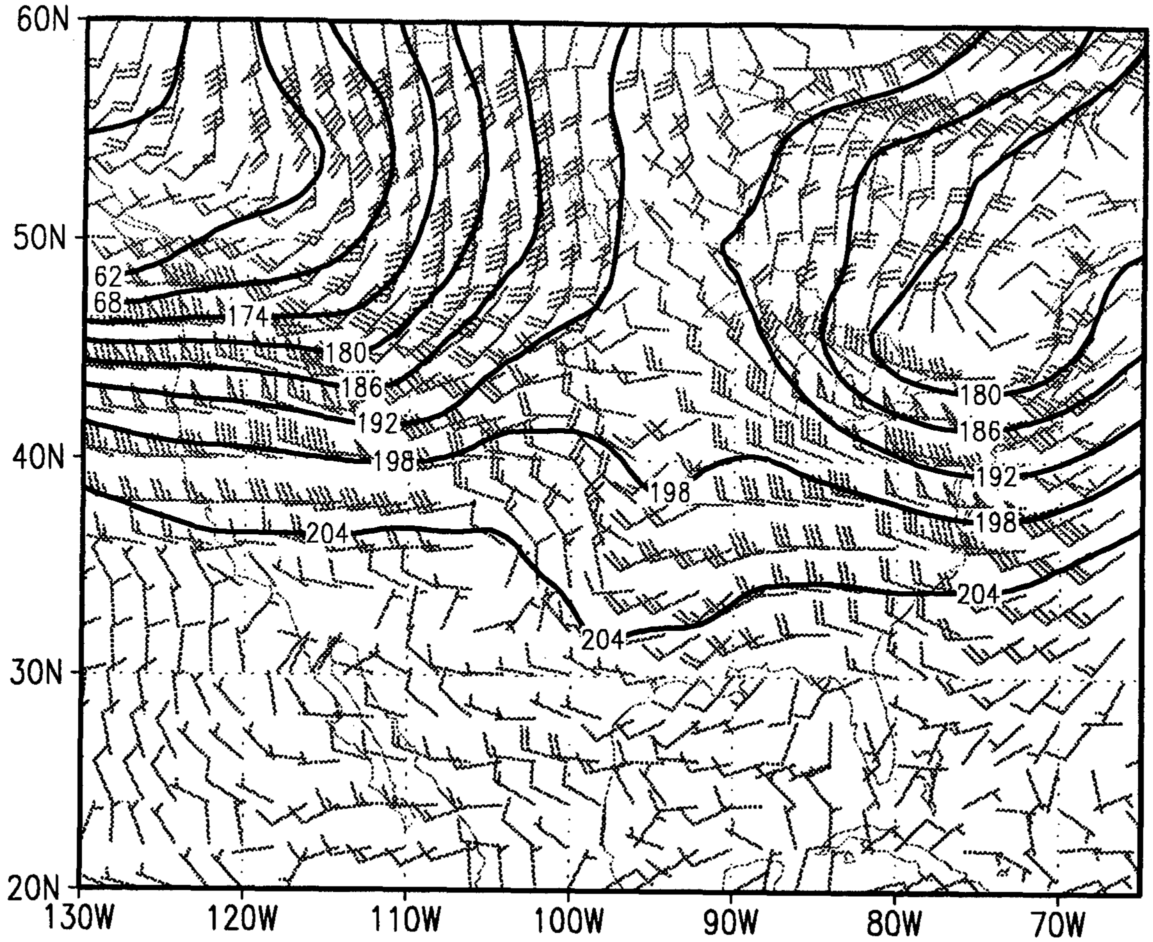


FIG. 18. Montgomery streamfunction (sold lines, 10 m^2 s^{-2} – $3{\times}10^5$) and wind barbs (knots) for a 320 K 84-hour forecast from the MRF valid 1200 UTC 16 September 1996.

dry slots, and the traditional warm and cold conveyor belts. When compared to Fig. 20 we can see the correlation of IPV advection and vertical motions. It is also easy to identify the stratospheric air marked by high IPV values in the upper left corner of the chart.

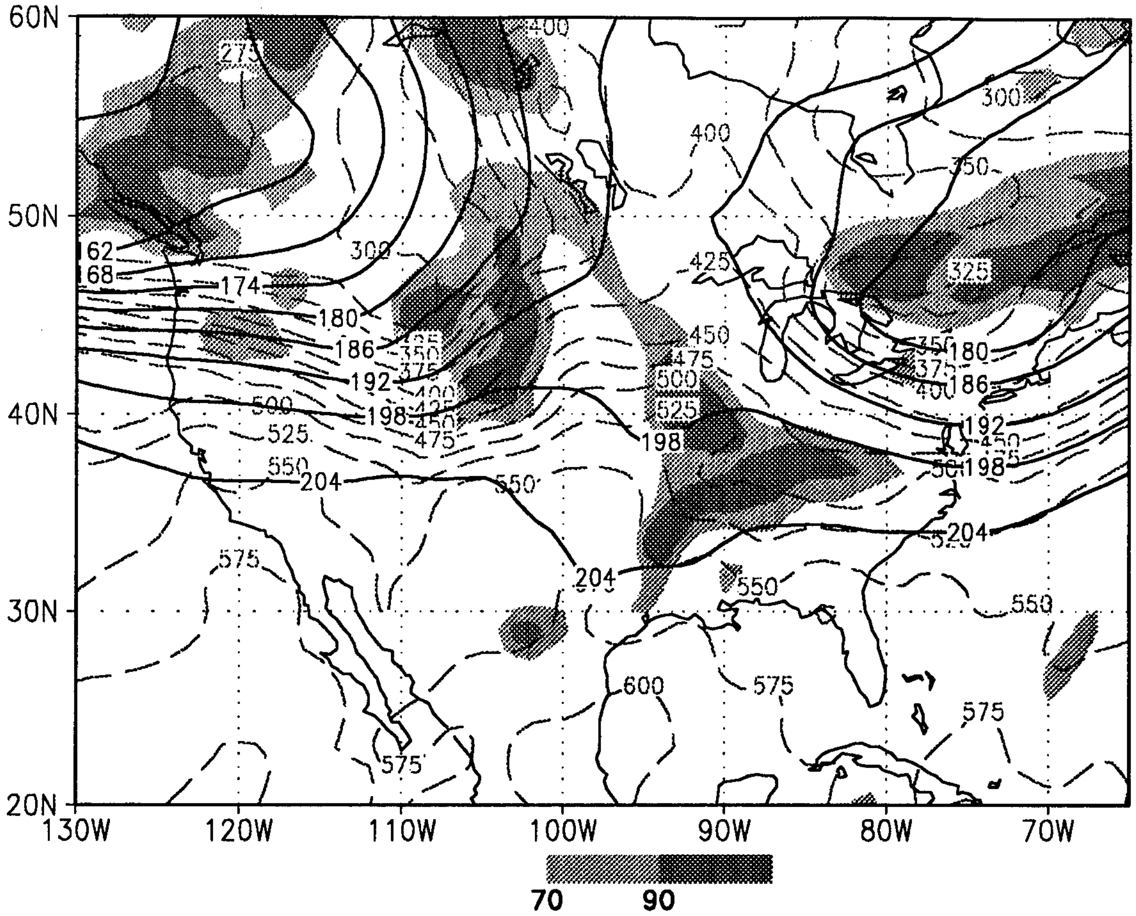


FIG. 19. Montgomery streamfunction (sold lines, 10 $m^2\ s^{-2} - 3\times10^5$), pressure (dashed lines, 10^{-1} kPa) and relative humidity (shaded) for a 320 K 84-hour forecast from the MRF valid 1200 UTC 16 September 1996.

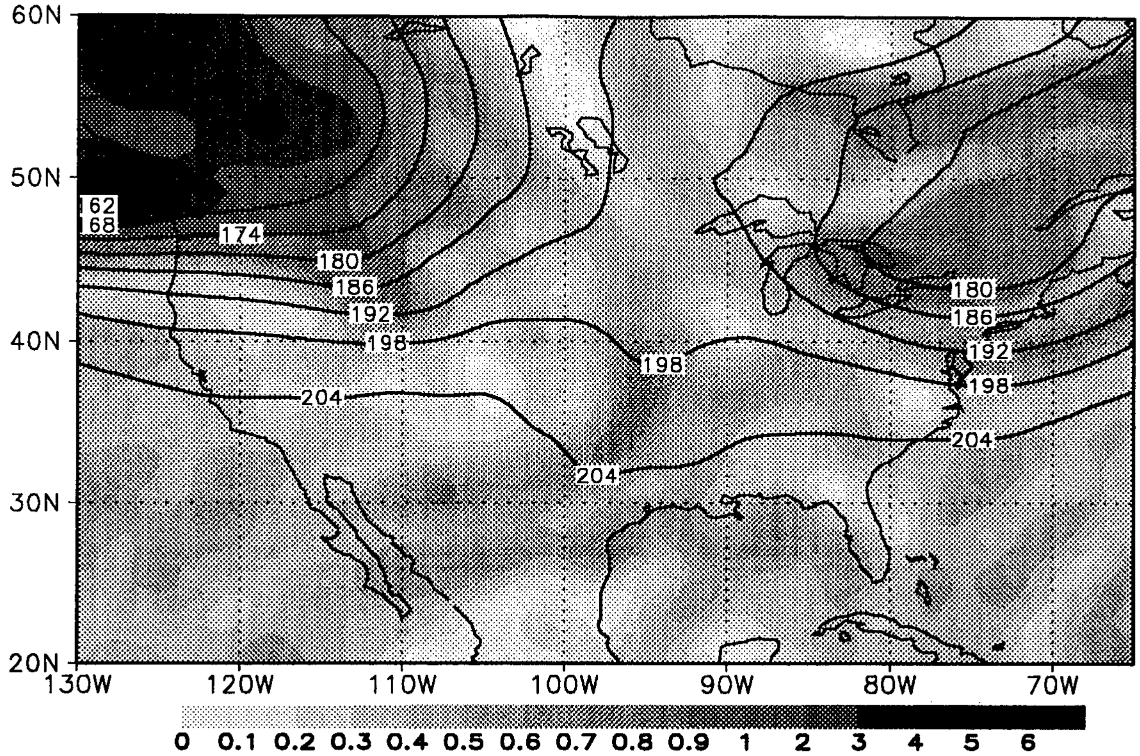


FIG. 20. Montgomery streamfunction (sold lines, 10 $m^2 s^{-2} - 3\times10^5$) and potential vorticity (shaded, PVU $\equiv 10^{-6} m^2 K kg^{-1} s^{-1}$) for a 320 K 84-hour forecast from the MRF valid 1200 UTC 16 September 1996.

5. Conclusions

The FORTRAN routines listed in the appendices are suitable for implementation of an initial isentropic analysis, especially using isentropic potential vorticity. Fig. 2 indicates the flow pattern and logic used by the programs to create the IPV and isentropic data fields. The fields generated can be used to supplement existing forecasting products in use at AFGWC, and potentially even reach individual forecasting units for local use. Careful attention was paid to programming choices in order to avoid the floating-point overflows, underfows, or divisions by zero frequently obtained from GEMPAK. The code at Appendix A-M adheres to common AFGWC coding practices and is ANSI-compliant except for a universal INCLUDE statement used to declare array sizes (Appendix B).

Because of the vertical resolution of the grids used in this thesis (mandatory pressure level data), the analyses produced by the algorithm are incapable of resolving most interesting mesoscale structures, including small areas of banded precipitation. However, the algorithm is sufficient to examine the features typical of synoptic-scale cyclone development (Davis, 1992). Because of this resolution problem, interpolating data to isentropic levels at a resolution less than 5 K will most likely be futile, only resulting in larger databases. A rough analysis suggests that an isentropic interpolation generated from mandatory-level data may truly offer no better than a 10 K resolution, on average.

Analysis of IPV algorithms from GEMPAK (desJardins *et al.*, 1996) indicates that improvements in their interpolation techniques could be made. Specifically, a linear interpolation of u and v wind components was replaced by a linear relation with $\ln p$ as suggested by Bergman (1979). This wind relation is supported by the thermal wind relation from geostrophic theory. Although Bergman also suggests that temperature can also be interpolated linearly against ln p, an adiabatic assumption may be slightly more accurate and is physically more meaningful. This assumption results in a lapse rate where ln T increases linearly with ln p.

Next, an investigation was performed to determine if IPV values could efficiently be calculated from mandatory-level data valid at mandatory levels. In a comparison against a method where P is calculated as a layered average, the IPV values at mandatory levels were shown to be at least comparable to the layered method, and somewhat better behaved. Therefore, the inherent advantages over calculating a layered average IPV field as performed by GEMPAK can be overcome, producing an IPV values that could easily be used in conjunction with other data valid at the same levels. This will allow implementation of IPV analysis even in conjunction with isobaric analysis performed locally by most AFW units.

An isentropic interpolation scheme was developed that first interpolated pressure to isentropic coordinates, then a second program was created that is able to interpolate any other isobaric scalar (except temperature which is inherent in the pressure field) to isentropic coordinates using the pressure data. A Newton iteration scheme was used in conjunction with Poisson's equation to precisely determine the pressure value. For most points only two iterations needed to be performed to reach an accuracy within 1.0 Pa of satisfying Poisson's equation. For interpolation of other scalars, a quadratic interpolation is performed using data from three nearby mandatory levels.

To maintain validity of potential temperature as a vertical coordinate, temperature profiles of superadiabatic (decreasing potential temperature with decreasing pressure) were modified to be adiabatic. This often creates poor vertical resolution near the surface. When complicated by friction and intersection of isentropes with the ground, analysis is best suited for middle and upper tropospheric levels. For this reason, fields below the surface are identified as missing.

Analysis against a known analytic function indicated that it was proper to interpolate wind and pressure data to isentropic coordinates then determine IPV. The alternative method used by Hoskins *et al.* (1985), Davis and Emanuel (1991), and later by Davis (1992) calculates IPV at constant pressure then interpolates the values to isentropic coordinates. Although this method may not be as calculation intensive and valuable if using IPV alone, to fully exploit IPV products they must be used in conjunction with other isentropic parameters—so computational time is most likely not lost for the true isentropic analyst.

Other improvements over the GEMPAK routines included accounting for the possibility of a worldwide grid, performing forward or backward differences near missing data points, ensuring continuity of grid points at the poles, and calculating relative vorticity values at the poles using the circulation theorem.

6. Further work

The largest future consideration is inherent in actual isobaric model output. Since many of the models actually perform calculations using σ as the vertical coordinate, which is then interpolated to isobaric fields for output; there is a large source for error by performing yet another vertical interpolation from isobaric to isentropic coordinates. Performing a single translation from σ coordinates to isentropic coordinates may significantly reduce errors. Obtaining higher vertical resolution data from the spectral coefficients should also be considered. This may aid in reducing the "coarse-grain approximation" problem mentioned by Hoskins *et al.* (1985). If model data directly interpolated to isentropic fields is not easily available, the AFGWC programmers could also tailor the interpolation routines to exploit all available model data (at least for the MRF). This would include tropopause data, maximum wind level data, 0.995σ level data, etc.

Handling of superadiabatic layers could be improved by implementing the method suggested by Moore (1993) and Haltiner and Williams (1980). This would minimize perturbations to the potential energy profile in order to obtain continuously increasing potential temperature values with height and preserve the potential energy profile in the column. This method would include cooling at the lower level in conjunction with warming at the upper level. The current method only warms the upper level.

Furthermore, a more in depth analysis of a potential transition to a cubic interpolation method for the both the P_p, and s_θ calculations should be explored. This could be in conjunction with exploring if there significant value added to existing interpolation methods for *T*, *u*, and *v*.

Analysts may experience a continuity problem, or nuisance, due to missing data below the surface. A Lorenz condition (desJardins *et al.*, 1996; Davis, 1992) could be added to hydrostatically extrapolate below the surface. This feature could aid in tracking movement of isentropic features near the surface. Data below the surface could be represented by dashes, or lighter shading.

During development of the IPV programs, several questions and other areas of potential improvement came to mind. Some of these include an analysis of dynamic tropopause seasonal and geographical variations, or the employment of the algorithm with a mesoscale model. An assessment of the actual effects of gravitation variations could also be investigated. As mentioned earlier, further development may also include the employment of EPV products. With an algorithm available to calculate IPV and available moisture fields, EPV cross sections and analysis could be easily developed as proposed for AFGWC by Zapotocny and Runk (1995). These products could be very useful in cross sections or on isentropic surfaces to depict conditional symmetric instability leading to banded precipitation events as discussed by Moore and Lambert (1993). In addition, since this thesis is an introduction to IPV use at the AFGWC, actual application will likely spawn additional research and questions.

REFERENCES

Air Force Global Weather Center, 1996: *Software Development Lifecycle Standards.* Systems Division Operating Instruction 33-2 (AFGWC/SY DOI 33-2). Offutt Air Force Base NE, 109 pp.

Air Force Global Weather Central, and Fleet Numerical Meteorology and Oceanography Center, 1995: *Interface Specification for Air Force Global Weather Central and Fleet Numerical Meteorology and Oceanography Center*, Offutt Air Force Base NE, 48 pp.

Bergman, K. H., 1979: Multivariate analysis of temperatures and winds using optimum interpolation. *Mon. Wea. Rev.*, **107,** 1423-1444.

Bluestein, H. B., 1993: *Synoptic-Dynamic Meteorology in Midlatitudes, Vol II*, Oxford University Press, 594 pp.

Carlson, T. N., 1991: *Mid-Latitude Weather Systems*, Routledge, 507 pp.

Danielsen, E. F., 1968: Stratospheric-tropospheric exchange based on radioactivity, ozone and potential vorticity. *J. Atmos. Sci.*, **25,** 502-518.

Davis, C. A., 1992: A potential-vorticity diagnosis of the importance of initial structure and condensational heating in observed extratropical cyclogenesis. *Mon. Wea Rev.*, **120**, 2409-2428.

Davis, C. A., and K. A. Emanuel, 1991: Potential vorticity diagnostics of cyclogenesis. *Mon. Wea. Rev.*, **119,** 1929-1953.

Dey, C. H., 1996: *The WMO Format for the Storage of Weather Product Information and the Exchange of Weather Product Messages in Gridded Binary Form.* Office Note 388 GRIB (Edition 1). National Centers for Environmental Prediction, National Weather Service, National Oceanic and Atmospheric Administration, Department of Commerce, Washington DC, 91 pp.

desJardins, M. L., K. F. Brill, S. Jacobs, S. S. Schotz, P. Bruehl, R. Schneider, B. Colman, and D.W. Plummer, 1996: General Meteorological Package (GEMPAK), Software Version 5.4, National Centers for Environmental Prediction, Washington DC.

Doty, B., 1995: The Grid Analysis and Display System (GrADS), Software Version 1.5, Center for Ocean-Land-Atmospheric Studies, Calverton MD.

Haltiner G. J., and R. T. Williams, 1980: *Numerical Prediction and Dynamical Meteorology*, John Wiley and Sons, New York NY, 477 pp.

Hoke, J. E., J. L. Hayes, and L. G. Renninger, 1981: *Map Projections and Grid Systems for Meteorological Applications.* AFGWC/TN-79/003 (Revised November 1983, June 1985), Air Force Global Weather Central, Offutt Air Force Base NE, 87 pp.

Holton, J. R., 1992: *An Introduction to Dynamic Meteorology*, 3rd ed., Academic Press, 509 pp.

Hoskins, B. J., M. E. McIntyre, M. E., and A. W. Robertson, 1985: On the use and significance of isentropic potential vorticity maps. *Quart. J. Roy. Meteor. Soc.*, **111,** 877-946.

Kreyszig, E., 1993: *Advanced Engineering Mathematics*, 7th Edition, John Wiley & Sons, New York NY, 1271 pp.

Moore, J. T., 1993: "Isentropic analysis and interpretation: Operational applications to synoptic and mesoscale forecast problems," National Weather Service Training Center, Kansas City MO.

Moore, J. T., and T. E. Lambert, 1993: The use of equivalent potential vorticity to diagnose regions of conditional symmetric instability. *Wea. Forecasting,* **8,** 301-308.

National Oceanic and Atmospheric Administration, National Aeronautic and Space Administration, and U.S. Air Force, 1976: *U.S. Standard Atmosphere*, Washington DC, 227 pp.

Peixoto, J. P., and A. H. Oort, 1992: *Physics of Climate*, American Institute of Physics, New York NY, 520 pp.

Spaete, P., D. R. Johnson, and T. K. Schaack, 1994: Stratospheric-tropospheric mass exchange during the President's Day storm. *Mon. Wea. Rev.,* **122**, 424-439.

Zapotocny, J. V., and K. J. Runk, 1995: Applications of isentropic analysis and satellite data visualizations for aviation forecasting at the AFGWC. *6th Conf. on Aviation Weather Systems,* Dallas TX, Amer. Meteor. Soc., 419-420.

APPENDIX A

Main Program[1]

```
      PROGRAM IPVGRD
************************************************************************
************************************************************************
**  NAME:  IPVGRD - Interpolates isobaric data to isentropic coordinates
**                  and calculates isentropic potential vorticity from
**                  isobaric model data.
**
**  ROUTINE NARRATIVE:  This program transforms u, v, p, and RH to
**  isentropic coordinates, then calculates isentropic potential
**  vorticity (IPV).  Isentropic output also includes Montgomery
**  streamfunction.  Isobaric IPV values are also exported for use with
**  isobaric analysis or cross-sections.  This program begins with the
**  12-hr forecast and creates data at 12-hr increments out to 384
**  hours.  Output files are unformatted for reading by GrADS (Doty
**  1995).  This code was created as part of thesis work by
**  Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*  REFERENCES:
*    desJARDINS, M.L, K.F. Brill, S. Jacobs, S.S. Schotz, P. Bruehl,
*      R. Schneider, B. Colman, D.W. Plummer, 1996:  General
*      Meteorological Package (GEMPAK), Software Version 5.4, National
*      Centers for Environmental Prediction, Washington D.C.
*    DOTY, B., 1995:  The Grid Analysis and Display System (GrADS),
*      Software Version 1.5, Center for Ocean-Land-Atmospheric Studies,
*      Calverton, MD.
*
*  SUBROUTINES CALLED:
*    GETGRB, LATLON, PVONP (CALLS DDX, DDY, DORELV, DOABSV), P2THTA,
*    S2THTA, DOIPV (CALLS DDX, DDY, DORELV, DOABSV)
*
*  FUNCTION USED:
*    POT - Calculates potential temperature from pressure and
*          temperature (used by SUBROUTINES P2THTA, PVONP).
*
*  REQUIRED STARTING CONDITIONS:
*    GRIB files from 12-hr to 384-hr forecast.  Degribber must pass
*    arrays of surface pressure, temperature, geopotential heights,
*    u and v wind components, and relative humidity.  For NOGAPS data
*    surface pressure must be derived from isobaric pressure and terrain
*    information.  A system time function can be added to skip the MRF
*    cycle when processing the 12Z model run (MRF is only available at
```

[1] Program GETGRB not included

```
*    00Z run).
*
*  OUTPUT:
*    isentropicF<hh>.dat - Data file containing isentropic grids
*              calculated from subroutines (formatted for GrADS).
*              Where <hh> is the forecast hour of the model data.
*    isobaricF<hh>.dat - Data file containing isobaric values of IPV
*              (formatted for GrADS).  Where <hh> is the forecast hour
*              of the model data.
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PAREMETER, Maximum number of grid rows.
*
*  PARAMETERS:
*    CP     - Specific Heat of dry air at constant pressure
*             (J K-1 kg-1).
*    GRAVTY - Earth's gravitational acceleration (m s-2) (NOAA, NASA,
*             USAF, 1976).
*    KAPPA  - RD / CP.
*    KMAX   - Maximum number of input or output levels.  50 is based on
*             the value set by GEMPAK (desJardins et al., 1996).  (MRF
*             has 29 different levels including miscellaneous levels,
*             NOGAPS essentially has 16 mandatory levels, where MSL
*             represents several different levels near the surface
*             depending on the parameter).
*    MD     - Average molecular mass of dry air at sea level (kg)
*             (NOAA, NASA, USAF, 1976).
*    R      - Gas constant (J K-1 kg-1) (International Council of
*             Scientific Unions, CODATA Bulletin No. 11, Dec 1973).
*    RD     - Gas constasnt for dry air (J K-1 kg-1).
*
*  VARIABLES:
*    FHR    - Forecast hour of model data to retrieve.
*    I      - Column marker.
*    IPV    - 3D IPV grid (m2 K kg-1 s-1).
*    IREC   - Record number for writing to GrADS file.
*    J      - Row marker.
*    K      - Vertical level marker.
*    KTHTA  - Number of isentropic levels output.
*    LAT    - Array containing latitudes of grid rows (degrees).
*    LAT1   - Starting latitude of grid point (1, 1) (degrees).
*    LAT2   - Ending latitude of grid point (NI, NJ) (degrees).
*    LON    - Array containing longitudes of grid columns (degrees).
*    LON1   - Starting longitude of grid point (1, 1) (degrees).
*    LON2   - Ending longitude of grdi point (NI, NJ) (degrees).
*    MERR   - I/O Error code.
*    MSTRM  - 3D grid of isentropic Montgomery streamfunction (m2 s-2).
*    NI     - Number of columns in grid.
*    NJ     - Number of rows in grid.
*    PRES   - Vector of mandatory isobaric levels (Pa).
*    PSFC   - Grid of surface pressure (Pa).
*    PTHTA  - 3D grid of isentropic pressures (Pa).
*    PVP    - 3D grid of isobaric IPV (m2 K kg-1 s-1).
*    RELV   - Relative vorticity grid at 500mb (s-1).
*    RH     - 3D grid of unpacked floating point data of relative
*             humidities (2-meter height and 1000 through 300mb) (%).
*    THTA   - Vector of isentropic surfaces (K).
```

```
*     TMP     - 3D grid containing temperature (2-meter height and 1000
*               through 10mb) (K).
*     UGRD    - 3D grid of unpacked floating point data for grid relative
*               u wind (East) component (10-meter height and 1000 through
*               10mb) (m s-1).
*     UTHTA   - 3D grid of isentropic U wind (m s-1).
*     VGRD    - 3D grid of unpacked floating point data for grid relative
*               v wind (North) component (10-meter height and 1000 through
*               10mb) (m s-1).
*     VTHTA   - 3D grid of isentropic V wind (m s-1).
*************************************************************************************
      INCLUDE     'grdsiz.inc'

      INTEGER         KMAX
        PARAMETER     (KMAX = 50)

      REAL          CP
        PARAMETER (CP = 1004.)
      REAL          GRAVTY
        PARAMETER (GRAVTY = 9.80665)
      REAL          MD
        PARAMETER (MD = 28.9644)
      REAL          R
        PARAMETER (R = 8314.41)
      REAL          RD
        PARAMETER (RD = R / MD)
      REAL          KAPPA
        PARAMETER (KAPPA = RD / CP)

      CHARACTER * 18  OUTPUT

      INTEGER         FHR
      INTEGER         I
      INTEGER         IREC
      INTEGER         J
      INTEGER         K
      INTEGER         KTHTA
      INTEGER         MERR
      INTEGER         NI
      INTEGER         NJ

      REAL            HGT (NIMAX, NJMAX, 17)
      REAL            IPV (NIMAX, NJMAX, KMAX)
      REAL            LAT (NJMAX)
      REAL            LAT1
      REAL            LAT2
      REAL            LON (NIMAX)
      REAL            LON1
      REAL            LON2
      REAL            MSTRM (NIMAX, NJMAX, KMAX)
      REAL            PVP (NIMAX, NJMAX, 16)
      REAL            PRES (16)
      REAL            PSFC (NIMAX, NJMAX)
      REAL            PTHTA (NIMAX, NJMAX, KMAX)
      REAL            RH (NIMAX, NJMAX, 17)
      REAL            RHTHTA (NIMAX, NJMAX, KMAX)
      REAL            THTA (KMAX)
      REAL            TMP (NIMAX, NJMAX, 17)
```

```
      REAL            VGRD (NIMAX, NJMAX, 17)
      REAL            UGRD (NIMAX, NJMAX, 17)
      REAL            UTHTA (NIMAX, NJMAX, KMAX)
      REAL            VTHTA (NIMAX, NJMAX, KMAX)

      DATA PRES    /100000., 92500., 85000., 70000., 50000., 40000.,
     &               30000., 25000., 20000., 15000., 10000.,  7000.,
     &                5000.,  3000.,  2000.,  1000./
      DATA DTHTA /5./

      DO 1400 FHR = 12, 384, 12
        CALL GETGRB (FHR, NI, NJ, LAT1, LON1, LAT2, LON2, PSFC, HGT,
     &               TMP, UGRD, VGRD, RH)
        CALL LATLON (NI, NJ, LAT1, LON1, LAT2, LON2, LAT, LON)

        OUTPUT (1: 11) = 'isobaricF'
        IF (FHR .LT. 100) THEN
          WRITE (OUTPUT (10: 11), '(I2)'), FHR
          OUTPUT (12: 15) = '.dat'
        ELSE
          WRITE (OUTPUT (10: 12), '(I3)'), FHR
          OUTPUT (13: 16) = '.dat'
        END IF

        OPEN (UNIT = 11, FILE = OUTPUT, STATUS = 'unknown',
     &        FORM = 'UNFORMATTED', ACCESS = 'DIRECT',
     &        RECL = NI * NJ * 4, IOSTAT = MERR)
        IF (MERR .NE. 0) GO TO 1500

        DO 300 K = 1, 16
          IF (K .EQ. 1) THEN
            CALL PVONP (NI, NJ, LAT, LON, PRES (K), PRES (K + 1),
     &                  PRES (K), TMP (1, 1, K), TMP (1, 1, K + 1),
     &                  TMP (1, 1, K), UGRD (1, 1, K),
     &                  UGRD (1, 1, K + 1), UGRD (1, 1, K),
     &                  VGRD (1, 1, K), VGRD (1, 1, K + 1),
     &                  VGRD (1, 1, K), PVP (1, 1, K) )
          ELSE IF (K .EQ. 16) THEN
            CALL PVONP (NI, NJ, LAT, LON, PRES (K), PRES (K),
     &                  PRES (K - 1), TMP (1, 1, K), TMP (1, 1, K),
     &                  TMP (1, 1, K - 1), UGRD (1, 1, K),
     &                  UGRD (1, 1, K), UGRD (1, 1, K - 1),
     &                  VGRD (1, 1, K), VGRD (1, 1, K),
     &                  VGRD (1, 1, K - 1), PVP (1, 1, K) )
          ELSE
            CALL PVONP (NI, NJ, LAT, LON, PRES (K), PRES (K + 1),
     &                  PRES (K - 1), TMP (1, 1, K), TMP (1, 1, K + 1),
     &                  TMP (1, 1, K - 1), UGRD (1, 1, K),
     &                  UGRD (1, 1, K + 1), UGRD (1, 1, K - 1),
     &                  VGRD (1, 1, K), VGRD (1, 1, K + 1),
     &                  VGRD (1, 1, K - 1), PVP (1, 1, K) )
          END IF

*         ---------------------------------------------------------------
*         Since RH only goes to 300mb, copy the 300mb values to the
*         250mb in order to diminish influence on interpolated values
*         between 400 and 300mb.  The RH values will also gracefully go
*         to 0. above 300 mb.
```

```
*         ----------------------------------------------------------------
          IF (K .EQ. 9) THEN
            DO 200 J = 1, NJ
              DO 100 I = 1, NI
                RH (I, J, K) = RH (I, J, K - 1)
100           CONTINUE
200         CONTINUE
          END IF
300     CONTINUE

        IREC = 1
        DO 400 K = 1, 16
          WRITE (11, REC=IREC) ((PVP (I, J, K), I = 1, NI), J = 1, NJ)
          IREC = IREC + 1
400     CONTINUE
        CLOSE (11)

        CALL P2THTA (NI, NJ, LAT, LON, TMP (1, 1, 1), PSFC,
     &               TMP (1, 1, 2), KTHTA, THTA, PTHTA)
        CALL S2THTA (NI, NJ, KTHTA, UGRD (1, 1, 1), PSFC,
     &               UGRD (1, 1, 2), THTA, PTHTA, UTHTA)
        CALL S2THTA (NI, NJ, KTHTA, VGRD (1, 1, 1), PSFC,
     &               VGRD (1, 1, 2), THTA, PTHTA, VTHTA)
        CALL DOIPV (NI, NJ, LAT, LON, KTHTA, THTA, PTHTA, UTHTA, VTHTA,
     &              IPV)
        CALL S2THTA (NI, NJ, KTHTA, RH (1, 1, 1), PSFC, RH (1, 1, 2),
     &               THTA, PTHTA, RHTHTA)

        OUTPUT (1: 11) = 'isentropicF'
        IF (FHR .LT. 100) THEN
          WRITE (OUTPUT (12: 13), '(I2)'), FHR
          OUTPUT (14: 17) = '.dat'
        ELSE
          WRITE (OUTPUT (12: 14), '(I3)'), FHR
          OUTPUT (15: 18) = '.dat'
        END IF

        OPEN (UNIT = 21, FILE = OUTPUT, STATUS = 'unknown',
     &        FORM = 'UNFORMATTED', ACCESS = 'DIRECT',
     &        RECL = NI * NJ * 4, IOSTAT = MERR)
        IF (MERR .NE. 0) GO TO 1500

        IREC = 1
        DO 500 K = 1, KTHTA
          WRITE (21, REC=IREC) ((PTHTA (I, J, K), I = 1, NI), J = 1, NJ)
          IREC = IREC + 1
500     CONTINUE

        DO 600 K = 1, KTHTA
          WRITE (21, REC=IREC) ((UTHTA (I, J, K), I = 1, NI), J = 1, NJ)
          IREC = IREC + 1
600     CONTINUE

        DO 700 K = 1, KTHTA
          WRITE (21, REC=IREC) ((VTHTA (I, J, K), I = 1, NI), J = 1, NJ)
          IREC = IREC + 1
700     CONTINUE
```

```
      DO 800 K = 1, KTHTA
        WRITE (21, REC=IREC) ((RH (I, J, K), I = 1, NI), J = 1, NJ)
        IREC = IREC + 1
800   CONTINUE

      DO 900 K = 1, KTHTA
        WRITE (21, REC=IREC) ((IPV (I, J, K), I = 1, NI), J = 1, NJ)
        IREC = IREC + 1
900   CONTINUE

*     ---------------------------------------------------------------
*     Calculate the Montgomery streamfunction from isentropic pressure
*     and isentropic geopotential height.
*     ---------------------------------------------------------------

      DO 1200 K = 1, KTHTA
        DO 1100 J = 1, NJ
          DO 1000 I = 1, NI
            IF (PTHTHA (I, J, K) .GT. 0.) THEN
              MSTRM (I, J, K) = CP * THTA (K) *
     &                          (PTHTA (I, J, K) / PRES (1) )**KAPPA +
     &                          GRAVTY * HGTTHTA (I, J, K)
            ELSE
              MSTRM (I, J, K) = -9999.
            END IF
1000      CONTINUE
1100    CONTINUE
1200  CONTINUE

      DO 1300 K = 1, KTHTA
        WRITE (21, REC=IREC) ((MSTRM (I, J, K), I = 1, NI), J = 1, NJ)
        IREC = IREC + 1
1300  CONTINUE

      CLOSE (21)
1400 CONTINUE
     STOP

1500 CONTINUE
     PRINT *, 'IPVGRD:  OPEN OUTPUT FILE ERROR ON FILE = ', OUTPUT,
     &          '. MERR = ', MERR
     STOP

     END
```

APPENDIX B

Grid Size Inclusion Statements

```
************************************************************************
************************************************************************
**  NARRATIVE:  These parameter statements are included by grid
**  subroutines to consistently define the maximum grid size, and
**  eleviate errors when passing data back and forth between routines.
**  This code was developed as part of thesis work by
**  Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Jan 97
************************************************************************
************************************************************************
*  REFERENCES:
*    Dey, C.H., 1996:  The WMO format for the storage of weather product
*      information and the exchange of weather product messages in
*      gridded binary form, Office Note 388 GRIB (Edition 1).  U.S.
*      Department of Commerce, National Oceanic and Atmospheric
*      Administration, National Weather Service, National Centers for
*      Environmental Prediction. 91 pp.
*
*  CALLED BY:
*    DDX, DDY, DOABSV, DOPV, DORELV
*
*  PARAMETER VARIABLES:
*    NIMAX - Maximum number of grid columns based on WMO grid type 3
*            (Dey, 1996).
*    NJMAX - Maximum number of grid rows based on WMO grid type 3 (Dey,
*            1996).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER          NIMAX
        PARAMETER     (NIMAX = 360)
      INTEGER          NJMAX
        PARAMETER     (NJMAX = 181)
```

APPENDIX C

Latitude/Longitude Subroutine

```
      SUBROUTINE LATLON (NI, NJ, LAT1, LON1, LAT2, LON2, LAT, LON)
***********************************************************************
***********************************************************************
**   NAME:  LATLON - DETERMINES THE LAT/LON FOR GRID POINTS
**
**   ROUTINE NARRATIVE:  This subroutine calculates the latitude and
**   longitude of a Cylindrical Equidistant (Latitude-Longitude).  This
**   subroutine uses the indexing convention common to most grids at
**   AFGWC (Hoke et al, 1981) with (1, 1) in the upper left corner.  It
**   returns two grids of values, LAT and LON, representing the latitudes
**   and longitudes of the grid points in degrees, respectively.  The
**   routine requires grid description information.  This code was
**   created as part of thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**   LAST MODIFICATION DATE:  12 Dec 96
***********************************************************************
***********************************************************************
*   REFERENCES:
*     GEMPAK V5.2.1, 1995.
*     Hoke, J.E., J.L. Hayes, L.G. Renninger, 1981:  Map projections and
*       grid systems for meteorological applications.  AFGWC/TN-79/003
*       (Revised Nov 83, Jun 85), Air Force Global Weather Central,
*       Offutt Air Force Base, NE.  87 pp.
*
*   INPUT VARIABLES:
*     LAT1 - Upper left J grid latitude (degrees) (90. for MRF & NOGAPS).
*     LAT2 - Lower right J grid latitude (degrees)
*            (-90. for MRF & NOGAPS).
*     LON1 - Upper left I grid longitude (degrees) (0. for MRF & NOGAPS).
*     LON2 - Lower right I grid longitude (degrees)
*            (-1 or 359. for MRF, -2.5 or 357.5 for NOGAPS).
*     NI   - Number of data points in longitudinal direction (columns)
*            (360 for MRF, 144 for NOGAPS)
*     NJ   - Number of data points in latitudinal direction (rows)
*            (181 for MRF, 73 for NOGAPS)
*     PROJ - Projection type.
*               MRF/NOGAPS: 'CED' for cylindrical equidistant (lat/lon)
*   OUTPUT:
*     LAT - Grid array containing the latitudes of corresponding grid
*           row (degrees).  Southern Hemisphere values are negative (Hoke
*           et al, 1981).
*     LON - Grid array containing the longitudes of corresponding grid
*           column (degrees).  Western Hemisphere values are negative
*           (Hoke et al, 1981)
*
*   VARIABLES:
*     I      - Increments grid columns.
*     J      - Increments grid rows.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       INTEGER        I
       INTEGER        J
```

```
      INTEGER         NI
      INTEGER         NJ

      REAL            LAT (NJ)
      REAL            LAT1
      REAL            LAT2
      REAL            LON (NI)
      REAL            LON1
      REAL            LON2

*     -----------------------------------------------------------
*     Initialize LAT/LON arrays in degrees.
*     -----------------------------------------------------------

      LON (1) = LON1
      LON (NI) = LON2
      IF (LON1 .LT. 0.) LON (1) = LON (1) + 360.
      IF (LON2 .LE. 0.) LON (NI) = LON (NI) + 360.

      DO 100 I = 1, NI
        LON (I) = LON (1) + FLOAT (I - 1) * (LON (NI) - LON (1)) /
     &            FLOAT (NI)
        IF (LON (I) .GT. 180.) LON (I) = LON (I) - 360.
  100 CONTINUE

      LAT (1) = LAT1
      LAT (NJ) = LAT2
      IF (LAT2 .GT.  90.) LAT (NJ) =  180. - LAT (NJ)
      IF (LAT2 .LT. -90.) LAT (NJ) = -180. - LAT (NJ)
      IF (LAT1 .GT.  90.) LAT (1) =  180. - LAT (1)
      IF (LAT1 .LT. -90.) LAT (1) = -180. - LAT (1)

      DO 200 J = 2, NJ - 1
        LAT (J) = LAT (1) + FLOAT (J - 1) * (LAT (NJ) - LAT (1)) /
     &            FLOAT (NJ - 1)
        IF (LAT (J) .GT.  90.) LAT (J) =  180. - LAT (J)
        IF (LAT (J) .LT. -90.) LAT (J) = -180. - LAT (J)
  200 CONTINUE

      RETURN
      END
```

APPENDIX D

Isentropic Pressure (Temperature) Interpolation Subroutine

```
      SUBROUTINE P2THTA (NI, NJ, LAT, LON, TSFC, PSFC, TPRES, KTHTA,
     &                   THTA, PTHTA)
************************************************************************
************************************************************************
**  NAME:  P2THTA - INTERPOLATES PRESSURE DATA TO ISENTROPIC VERTICAL
**                  COORDINATES (CONSTANT POTENTIAL TEMPERATURE)
**
**  ROUTINE NARRATIVE:  This subroutine calculates and returns an array
**  of scalar grids for pressure interpolated to isentropic surfaces.
**  This subroutine doesn't perform any extrapolation below the surface;
**  instead values are depicted as missing (-9999.) below the surface.
**  Interpolation begins at the first isentropic level where 10% of the
**  data is above the surface.  Following desJardins et al. (1996), a
**  Newton interation method (Kreyszig, 1993) is used to refine the
**  interpolation in balance with Poisson's equation.  The code ignores
**  convectively unstable (decreasing potential temperatures with
**  height) and neutral layers and makes these layers slightly stable.
**  This code was developed as part of thesis work by
**  Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*  REFERENCES:
*    desJARDINS, M.L, K.F. Brill, S. Jacobs, S.S. Schotz, P. Bruehl,
*      R. Schneider, B. Colman, D.W. Plummer, 1996:  General
*      Meteorological Package (GEMPAK), Software Version 5.4, National
*      Centers for Environmental Prediction, Washington D.C.
*    KREYSZIG, E., 1993:  Advanced Engineering Mathematics, 7th Edition.
*      John Wiley & Sons, 1271 pp.
*    NOAA, NASA, USAF, 1976:  U.S. Standard Atmosphere.  Washington DC,
*      227 pp.
*
*  INPUT VARIABLES:
*    NI    - Number of data points in longitudinal direction (columns).
*    NJ    - Number of data points in latitudinal direction (rows).
*    PSFC  - Grid of surface pressures (Pa).
*    TPRES - 3D grid of temperatures on mandatory isobaric levels (K).
*    TSFC  - Grid of surface temperatures (K).
*
*  SUBROUTINES CALLED
*    NONE
*
*  FUNCTIONS USED
*    POT - Calculates potential temperature given temperature and
*          pressure.
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PAREMETER, Maximum number of grid rows.
*
```

```
*  OUTPUT:
*    KTHTA  - Number of isentropic levels output.
*    PTHTA  - 3D grid of pressure values calculated on isentropic
*             surfaces (Pa).
*    THTA   - Vector of isentropic surfaces data is valid for (K).
*
*  PARAMETER VARIABLES:
*    CP     - Specific Heat of dry air at constant pressure
*             (J K-1 kg-1).
*    KAPPA  - RD / CP.
*    MAXLVL - Maximum number of input or output levels.  50 is based on
*             the value set by GEMPAK (desJardins et al., 1996).  (MRF
*             has 29 different levels including miscellaneous levels,
*             NOGAPS essentially has 16 mandatory levels, where MSL
*             represents several different levels near the surface
*             depending on the parameter).
*    MD     - Average molecular mass of dry air at sea level (kg) (NOAA,
*             NASA, USAF, 1976).
*    PLVLS  - Number of mandatory isobaric surfaces represented in PRES
*             based on mandatory levels from 1000 to 10 mb.
*    R      - Gas constant (J K-1 kg-1) (International Council of
*             Scientific Unions, CODATA Bulletin No. 11, Dec 1973).
*    RD     - Gas constant for dry air (J K-1 kg-1).
*
*  VARIABLES
*    ALOGP  - Natural logarithm of the mandatory pressure levels.
*    ALOGPD - Natural logarithm of nearest lower mandatory pressure
*             level.
*    ALOGPU - Natural logarithm of nearest upper mandatory pressure
*             level.
*    DFDP   - Derivative of F with respect to pressure, P.
*    DLTDLP - Linear change of temperature with respect to ln (p)
*             between two known levels.
*    DTHTA  - Desired isentopic increment bewteen layers (K).
*    EPSLN  - Used to determine accuracy of Newton iteration (Pa).
*    F      - Function to determine the root of in the Newton iteration;
*             specifically, THTA - T*(Po/P)**(Rd/Cp); where T varies
*             linearly with ln (P).
*    I      - Increments grid columns.
*    INTERC - ln T value (intercept) where pressure is 1 Pa,
*             assuming a linear relation with ln (p) bewteen two known
*             pressure and temperature values.
*    J      - Increments grid rows.
*    KIN    - Increments vertical isobaric levels.
*    KOUT   - Increments vertical isentropic levels.
*    MAXIT  - Number of pressure levels that did not converge to EPSLN.
*    N      - Increments Newton iteration scheme or through parameters.
*    NMAX   - Maximum number of times to perform Newton iteration.
*    NPTS   - Number of points on grid where the isentropic surface is
*             above ground.
*    P1     - New pressure guess on isentropic surface from Newton
*             iteration (Pa).
*    PDWN   - Known pressure at lower level (Pa).
*    POTDWN - Known potential temperature at lower level (K).
*    POTSFC - Grid of potential temperature values at the surface (K).
*    POTUP  - Known potential temperature at upper level (K).
*    PRES   - Vector of mandatory isobaric levels (Pa).
*    PUP    - Known pressure at upper level (Pa).
```

```
*     RESID  - Residual error when calculating Newton iteration (Pa).
*     RESMAX - Maximum residual error for non-convergent pressures (Pa).
*     T1     - 1st guess of temperature at intermediate pressure level
*              (K).
*     TDWN   - Known temperature at lower level (K).
*     THTA1  - 1st guess of potential temperature given T1 and
*              intermediate pressure level (K).
*     THTAHI - Maximum potential temperature value found (K).
*     THTALO - Minimum potential temperature value found (K).
*     TUP    - Known temperature at upper level (K).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE     'grdsiz.inc'

      INTEGER        MAXLVL
        PARAMETER (MAXLVL = 50)
      INTEGER        PLVLS
        PARAMETER (PLVLS = 16)

      REAL           CP
        PARAMETER (CP = 1004.)
      REAL           MD
        PARAMETER (MD = 28.9644)
      REAL           R
        PARAMETER (R  = 8314.41)
      REAL           RD
        PARAMETER (RD = R / MD)
      REAL           KAPPA
        PARAMETER (KAPPA = RD / CP)

      INTEGER        I
      INTEGER        J
      INTEGER        KIN
      INTEGER        KOUT
      INTEGER        KTHTA
      INTEGER        MAXIT
      INTEGER        N
      INTEGER        NI
      INTEGER        NJ
      INTEGER        NMAX
      INTEGER        NPTS

      REAL           ALOGP  (PLVLS)
      REAL           ALOGPD
      REAL           ALOGPU
      REAL           DFDP
      REAL           DLTDLP
      REAL           DTHTA
      REAL           EPSLN
      REAL           F
      REAL           INTERC
      REAL           LAT  (*)
      REAL           LON  (*)
      REAL           P1
      REAL           PDWN
      REAL           POT
      REAL           POTDWN
      REAL           POTSFC  (NIMAX, NJMAX)
      REAL           POTUP
```

```
      REAL          PRES (PLVLS)
      REAL          PSFC (NIMAX, NJMAX)
      REAL          PTHTA (NIMAX, NJMAX, MAXLVL)
      REAL          PUP
      REAL          RESID
      REAL          RESMAX
      REAL          T1
      REAL          TDWN
      REAL          THTA (MAXLVL)
      REAL          THTAHI
      REAL          THTALO
      REAL          THTAP (NIMAX, NJMAX, PLVLS)
      REAL          TPRES (NIMAX, NJMAX, *)
      REAL          TSFC (NIMAX, NJMAX)
      REAL          TUP

      DATA DTHTA /5./
      DATA EPSLN /1./
      DATA NMAX  /5/
      DATA PRES  /100000., 92500., 85000., 70000., 50000., 40000.,
     &             30000., 25000., 20000., 15000., 10000.,  7000.,
     &              5000.,  3000.,  2000.,  1000./

*     ---------------------------------------------------------------
*     Calculate potential temperatures at the surface.  Keep track of
*     lowest value.
*     ---------------------------------------------------------------

      THTALO = POT (TSFC (1, 1), PSFC (1, 1))
      DO 200 J = 1, NJ
        DO 100 I = 1, NI
          POTSFC (I, J) = POT (TSFC (I, J), PSFC (I, J))
          IF (POTSFC (I, J) .LT. THTALO) THTALO = POTSFC (I, J)
  100   CONTINUE
  200 CONTINUE

*     ---------------------------------------------------------------
*     Compute the potential temperatures for each mandatory isobaric
*     level, eliminating superadiabatic or neutral layers.  Derive any
*     future temperatures from the new profile.
*     ---------------------------------------------------------------

      THTAHI = POT (TPRES (1, 1, 10), PRES (10))
      DO 500 KIN = 1, PLVLS
        DO 400 J = 1, NJ
          DO 300 I = 1, NI
            THTAP (I, J, KIN) = POT (TPRES (I, J, KIN), PRES (KIN) )
            IF (PSFC (I, J) .GT. PRES (KIN) ) THEN
              IF (KIN .GT. 1) THEN
                IF (PSFC (I, J) .LT. PRES (KIN - 1) ) THEN
                  IF (THTAP (I, J, KIN) .LE. POTSFC (I, J) ) THEN
                    THTAP (I, J, KIN) = POTSFC (I, J) + 0.01
                  END IF
                ELSE IF (THTAP (I, J, KIN) .LE. THTAP (I, J, KIN - 1) )
     &                THEN
                  THTAP (I, J, KIN) = THTAP (I, J, KIN - 1) + 0.01
                END IF
              ELSE
```

```
              IF (THTAP (I, J, 1) .LE. POTSFC (I, J) ) THEN
                THTAP (I, J, 1) = POTSFC (I, J) + 0.01
              END IF
            END IF
          END IF

*         ------------------------------------------------------------
*         Keep track of highest potential temperature value starting
*         at level 10 (just in case only using data to 100mb).
*         ------------------------------------------------------------

          IF (KIN .GE. 10 .AND. THTAP (I, J, KIN) .GT. THTAHI) THEN
            THTAHI = THTAP (I, J, KIN)
          ENDIF
  300     CONTINUE
  400   CONTINUE
  500 CONTINUE

*     ------------------------------------------------------------------
*     Identify isentropic levels to interpolate to (at least 10% grid
*     coverage)
*     ------------------------------------------------------------------

      KOUT = 0
  600 CONTINUE
      KOUT = KOUT + 1
      THTA (1) = 200. + FLOAT (KOUT - 1) * DTHTA
      IF (THTA (1) + DTHTA .GE. THTALO) GO TO 700
      GO TO 600

  700 CONTINUE
      THTA (1) = THTA (1) + DTHTA
      NPTS = 0
      J = 0
  800 CONTINUE
      J = J + 1
      IF (J .LE. NJ) THEN
        I = 0
  900   CONTINUE
        I = I + 1
        IF (I .LE. NI) THEN
          IF (POTSFC (I, J) .LE. THTA (1)) NPTS = NPTS + 1
          IF (NPTS .GE. FLOAT (NI * NJ) / 10.) GO TO 1000
          GO TO 900
        ELSE
          GO TO 800
        END IF
      ELSE
        GO TO 700
      END IF

 1000 CONTINUE
      PRINT *, 'FIRST ISENTROPIC LEVEL IS ', THTA (1), ' K.'
      KTHTA = 1
 1100 CONTINUE
      IF (KTHTA .LE. MAXLVL) THEN
        IF (THTA (KTHTA) .LE. THTAHI) THEN
          THTA (KTHTA + 1) = THTA (KTHTA) + DTHTA
```

```
          KTHTA = KTHTA + 1
          GO TO 1100
        END IF
      END IF

 1300 CONTINUE
      KTHTA = KTHTA - 1
      NPTS = 0
      J = 0
 1400 CONTINUE
      J = J + 1
      IF (J .LE. NJ) THEN
        I = 0
 1500   CONTINUE
        I = I + 1
        IF (I .LE. NI) THEN
          IF (THTAP (I, J, PLVLS) .GE. THTA (KTHTA) ) NPTS = NPTS + 1
          IF (FLOAT (NPTS) .GE. FLOAT (NI * NJ) / 10. ) GO TO 1600
          GO TO 1500
        ELSE
          GO TO 1400
        END IF
      ELSE
        GO TO 1300
      END IF

 1600 CONTINUE
      IF (KTHTA .GE. MAXLVL) THEN
        PRINT *, 'P2THTA:  ONLY THE FIRST ', MAXLVL,
     &           ' ISENTROPIC LEVELS WILL BE CALCULATED.  INCREASE ',
     &           'MAXLVL PARAMETER OR DTHTA TO OBTAIN DATA ABOVE ',
     &           THTA (MAXLVL), 'K.'
      ELSE
        PRINT *, 'TOP ISENTROPIC LEVEL ', THTA (KTHTA)
      END IF
      PRINT *, 'INTERPOLATING PRESSURE TO', KTHTA, ' LEVELS NOW...'

*     ------------------------------------------------------------------
*     Calculate pressure on isentropic surfaces. The method solves an
*     implicit equation derived by combining the definition of potential
*     temperature and the assumption that ln (T) varies linearly with
*     ln (p).  Newton iteration is used to solve for pressure.
*     ------------------------------------------------------------------

      DO 1650 KIN = 1, PLVLS
        ALOGP (KIN) = ALOG (PRES (KIN) )
 1650 CONTINUE

      MAXIT = 0
      RESMAX = 1.
      DO 2600 KOUT = 1, KTHTA
        DO 2500 J = 1, NJ
          DO 2200 I = 1, NI
            IF (THTA (KOUT) .LT. POTSFC (I, J) ) THEN

*           ------------------------------------------------------------
*           Theta level is below surface at this (i, j) location.
*           ------------------------------------------------------------
```

```
            PTHTA (I, J, KOUT) = -9999.
          ELSE IF (THTA (KOUT) .GT. THTAP (I, J, PLVLS) ) THEN
*         -----------------------------------------------------------
*         Theta level is above top pressure level.
*         -----------------------------------------------------------

            PTHTA (I, J, KOUT) = -9999.
          ELSE IF (ABS (THTA (KOUT) - POTSFC (I, J) ) .LT. 0.001) THEN
*         -----------------------------------------------------------
*         Theta level at the surface.
*         -----------------------------------------------------------

            PTHTA (I, J, KOUT) = PSFC (I, J)
          ELSE
            KIN = 0
 1700       CONTINUE
            KIN = KIN + 1
            IF (KIN .LE. PLVLS) THEN
              IF (THTA (KOUT) .LT. THTAP (I, J, KIN) ) THEN
                IF (KIN .EQ. 1) THEN
*               -----------------------------------------------------
*               Theta level is between surface and 1000 mb level.
*               -----------------------------------------------------

                  POTDWN = POTSFC (I, J)
                  PDWN = PSFC (I, J)
                  ALOGPD = ALOG (PSFC (I, J) )
                  IF (ABS (PSFC (I, J) - PRES (KIN) ) .LT. 0.001) THEN
                    POTUP = THTAP (I, J, KIN + 1)
                    PUP = PRES (KIN + 1)
                    ALOGPU = ALOGP (KIN + 1)
                  ELSE
                    POTUP = THTAP (I, J, KIN)
                    PUP = PRES (KIN)
                    ALOGPU = ALOGP (KIN)
                  END IF
                ELSE IF (POTSFC (I, J) .GT. THTAP (I, J, KIN - 1) )
     &                  THEN
*               -----------------------------------------------------
*               Theta level is between surface and other mandatory
*               level.
*               -----------------------------------------------------

                  POTDWN = POTSFC (I, J)
                  PDWN = PSFC (I, J)
                  ALOGPD = ALOG (PSFC (I, J) )
                  IF (ABS (PSFC (I, J) - PRES (KIN) ) .LT. 0.001) THEN
                    POTUP = THTAP (I, J, KIN + 1)
                    PUP = PRES (KIN + 1)
                    ALOGPU = ALOGP (KIN + 1)
                  ELSE
                    POTUP = THTAP (I, J, KIN)
                    PUP = PRES (KIN)
```

```
                ALOGPU = ALOGP (KIN)
              END IF
            ELSE
*
*           ---------------------------------------------------
*           Theta level is between two mandatory levels.
*           ---------------------------------------------------
*
              POTUP = THTAP (I, J, KIN)
              PUP = PRES (KIN)
              ALOGPU = ALOGP (KIN)
              POTDWN = THTAP (I, J, KIN - 1)
              PDWN = PRES (KIN - 1)
              ALOGPD = ALOGP (KIN - 1)
            END IF
            GO TO 1800
          ELSE
            GO TO 1700
          END IF
        END IF
*
*       -------------------------------------------------------------
*       Perform Newton iteration.
*       -------------------------------------------------------------
*
 1800   CONTINUE
        TDWN = POTDWN * (PDWN / 100000.)**KAPPA
        TUP = POTUP * (PUP / 100000.)**KAPPA
        DLTDLP = ALOG (TUP / TDWN) / (ALOGPU - ALOGPD)
        INTERC = ALOG (TUP) - DLTDLP * ALOGPU
        PTHTA (I, J, KOUT) = EXP ( (ALOG (THTA (KOUT) ) - INTERC -
     &                           KAPPA * ALOGP (1) ) /
     &                           (DLTDLP - KAPPA) )
        N = 0
 1900   CONTINUE
*
*       -------------------------------------------------------------
*       Note:  Use EXP (DLTDLP * ALOG (P) ) vice P**DLTDLP to
*       eliminate IEEE floating point overflow error.
*       -------------------------------------------------------------
*
        T1 = EXP (DLTDLP * ALOG (PTHTA (I, J, KOUT)) + INTERC)
        RESID = PTHTA (I, J, KOUT) -
     &          100000. * (T1 / THTA (KOUT) )**(1. / KAPPA)
        IF (ABS (RESID) .GT. EPSLN) THEN
          N = N + 1
          IF (N .LE. NMAX) THEN
            THTA1 = T1 * (100000. / PTHTA (I, J, KOUT) )**KAPPA
            F = THTA (KOUT) - THTA1
            DFDP = (KAPPA - DLTDLP) *
     &             (100000. / PTHTA (I, J, KOUT))**KAPPA *
     &             EXP (INTERC + (DLTDLP - 1.) *
     &                  ALOG (PTHTA (I, J, KOUT) ) )
            P1 = PTHTA (I, J, KOUT) - F / DFDP
            IF (P1 .LE. PDWN) THEN
              IF (P1 .GE. PUP) THEN
                PTHTA (I, J, KOUT) = P1
                GO TO 1900
```

```
                 ELSE
                   N = NMAX
                 END IF
               END IF
             ELSE
*            ---------------------------------------------------------
*            Keep track of pressures that don't converge.
*            ---------------------------------------------------------

               IF (RESID .GT. RESMAX) RESMAX = RESID
               MAXIT = MAXIT + 1
               GO TO 2100
             END IF
           END IF
 2100      CONTINUE
*          ------------------------------------------------------------
*          Make sure pressure decreases as potential temperature
*          increases.
*          ------------------------------------------------------------

           IF (PTHTA (I, J, KOUT - 1) .GT. 0.) THEN
             IF (PTHTA (I, J, KOUT) .GT. PTHTA (I, J, KOUT - 1))
     &         THEN
               PTHTA (I, J, KOUT) = PTHTA (I, J, KOUT - 1) + 0.001
             END IF
           END IF
         END IF
 2200  CONTINUE
*      ------------------------------------------------------------------
*      Assign values at the pole the average of the row representing the
*      point.
*      ------------------------------------------------------------------

       NPTS = 0
       IF (ABS (LAT (J)) .GE. 90.) THEN
         IF (PTHTA (1, J, KOUT) .GT. 0.) NPTS = 1
         DO 2300 I = 2, NI - 1
           IF (PTHTA (1, J, KOUT) .GT. 0.) THEN
             IF (PTHTA (I, J, KOUT) .GT. 0.) THEN
               PTHTA (1, J, KOUT) = PTHTA (1, J, KOUT) +
     &                              PTHTA (I, J, KOUT)
               NPTS = NPTS + 1
             END IF
           ELSE
             IF (PTHTA (I, J, KOUT) .GT. 0.) THEN
               PTHTA (1, J, KOUT) = PTHTA (I, J, KOUT)
               NPTS = NPTS + 1
             END IF
           END IF
 2300    CONTINUE
         IF (NPTS .EQ. 0) GO TO 2500
         IF (ABS (LON (1) - LON (NI) ) .LT. 0.001) THEN
           PTHTA (1, J, KOUT) = PTHTA (1, J, KOUT) / FLOAT (NPTS)
         ELSE IF (PTHTA (NI, J, KOUT) .GT. 0.) THEN
           PTHTA (1, J, KOUT) = PTHTA (1, J, KOUT) +
```

```
     &                               PTHTA (NI, J, KOUT)
                PTHTA (1, J, KOUT) = PTHTA (1, J, KOUT) / FLOAT (NPTS + 1)
              ELSE
                PTHTA (1, J, KOUT) = PTHTA (1, J, KOUT) / FLOAT (NPTS)
              END IF
              DO 2400 I = 2, NI
                PTHTA (I, J, KOUT) = PTHTA (1, J, KOUT)
 2400         CONTINUE
            END IF
 2500     CONTINUE
 2600 CONTINUE
      PRINT *, 'P2THTA:  ', MAXIT, ' POINTS REACHED ', NMAX,
     &         ' ITERATIONS WITHOUT CONVERGING TO ', EPSLN, ' PA.  MAX',
     &         ' RESIDUAL ERROR = ', RESMAX, ' PA.'
      RETURN
      END
```

APPENDIX E

Potential Temperature Function

```
      FUNCTION POT (TMP, PRES)
**********************************************************************
**********************************************************************
**  NAME:  POT - CALCULATES POTENTIAL TEMPERATURE
**
**  ROUTINE NARRATIVE:  This function calculates potential temperature
**  given the temperature (K), and pressure using Poisson's equation.
**  If virtual temperature is input instead of temperature, POT returns
**  virtual potential temperature.  This code was developed as part of
**  thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  19 Jan 97
**********************************************************************
**********************************************************************
*  POT = TMP * ( 100000. / PRES ) ** (Rd / Cp)
*
*  REFERENCES:
*    NOAA, NASA, USAF, 1976:  U.S. Standard Atmosphere.  Washington DC,
*      227 pp.
*
*  INPUT VARIABLES:
*    TMP  - Temperature (K).
*    PRES - Pressure (Pa).
*
*  OUTPUT:
*    POT - Potential temperature (K).
*
*  PARAMETER VARIABLES:
*    CP - Specific Heat of dry air at constant pressure (J K-1 kg-1).
*    MD - Average molecular mass of dry air at sea level (kg) (NOAA,
*         NASA, USAF, 1976).
*    R  - Gas constant (J K-1 kg-1) (International Council of Scientific
*         Unions, CODATA Bulletin No. 11, Dec 1973).
*    RD - Gas constasnt for dry air (J K-1 kg-1).
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      REAL         CP
        PARAMETER (CP = 1004.)
      REAL         MD
        PARAMETER (MD = 28.9644)
      REAL         R
        PARAMETER (R = 8314.41)
      REAL         RD
        PARAMETER (RD = R / MD)

      REAL         POT
      REAL         PRES
      REAL         TMP

      IF (PRES .LE. 0.) GO TO 100

      POT = TMP * (100000. / PRES) ** (RD / CP)
```

```
      RETURN

100   CONTINUE
      POT = -9999.0
      RETURN

      END
```

APPENDIX F

Isentropic Scalar Interpolation Subroutine

```
      SUBROUTINE S2THTA (NI, NJ, KTHTA, SSFC, PSFC, SPRES, THTA, PTHTA,
     &                   STHTA)
************************************************************************
************************************************************************
**   NAME:   S2THTA - INTERPOLATES A SCALAR GRID (OTHER THAN PRESSURE OR
**                    TEMPERATURE) FROM ISOBARIC VERTICAL COORDINATES TO
**                    ISENTROPIC VERTICAL COORDINATES (CONSTANT POTENTIAL
**                    TEMPERATURE)
**
**   ROUTINE NARRATIVE:  This subroutine calculates and returns an array
**   of scalar grids for a scalar value interpolated from isobaric
**   surfaces to isentropic surfaces.  Subroutine P2THTA must be run
**   prior to S2THTA to obtain pressure values on the isentropic
**   surfaces.  Therefore, this subroutine can NOT be used for pressure
**   interpolation.  Likewise, for consistency, temperature data should
**   be derived from the Poisson's equation where,
**     T = THTA * (P / Po)**(Rd / Cp).
**   This routine does not perform any extrapolation below the surface;
**   instead values are depicted as missing (-9999.) if they lie below
**   the surface.  This routine uses a quadratic interpolation
**   following desJardins et al. (1996) for S vs. ln p using the nearest
**   upper two mandatory levels and nearest lower mandatory level data
**   according to Kreyszig, (1993).  This code was developed as part of
**   thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**   LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*   REFERENCES:
*     KREYSZIG, E., 1993:  Advanced Engineering Mathematics, 7th Edition.
*       John Wiley & Sons, 1271 pp.
*     desJARDINS, M.L, K.F. Brill, S. Jacobs, S.S. Schotz, P. Bruehl,
*       R. Schneider, B. Colman, D.W. Plummer, 1996:  General
*       Meteorological Package (GEMPAK), Software Version 5.4, National
*       Centers for Environmental Prediction, Washington D.C.
*
*   INPUT VARIABLES:
*     KTHTA - Number of isentropic levels output.
*     NI    - Number of data points in longitudinal direction (columns).
*     NJ    - Number of data points in latitudinal direction (rows).
*     PSFC  - Grid of surface pressures (Pa).
*     PTHTA - 3D grid of pressure values calculated on isentropic
*             surfaces (Pa).
*     SSFC  - Grid of surface values for a given scalar.
*     SPRES - 3D grid of values for a given scalar on mandatory isobaric
*             levels.
*     THTA  - Vector of isentropic surfaces values to interpolate to (K).
*
*   SUBROUTINES CALLED
*     NONE
*
```

```
*  FUNCTIONS USED
*    NONE
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PAREMETER, Maximum number of grid rows.
*
*  OUTPUT:
*    STHTA - 3D grid of values for a given scalar interpolated to
*            isentropic surfaces.
*
*  PARAMETER VARIABLES:
*    MAXLVL - Maximum number of input or output levels.  50 is based on
*             the value set by GEMPAK (desJardins et al., 1996).  (MRF
*             has 29 different levels including miscellaneous levels,
*             NOGAPS essentially has 16 mandatory levels, where MSL
*             represents several different levels near the surface
*             depending on the parameter).
*    PLVLS  - Number of mandatory isobaric surfaces represented in PRES
*             based on mandatory levels from 1000 to 10 mb.
*
*  VARIABLES
*    I      - Increments grid columns.
*    J      - Increments grid rows.
*    KIN    - Increments vertical isobaric levels.
*    KOUT   - Increments vertical isentropic levels.
*    LNP1P2 - LN ( Upper Pressure/Middle Pressure ) for a given point.
*    LNP1P3 - LN ( Upper Pressure/Lower Pressure ) for a given point.
*    LNP2P3 - LN ( Middle Pressure/Lower Pressure ) for a given point.
*    LNPU1P - LN ( Up 1 Pressure Level/ P ) for mandatory levels.
*    LNPU2P - LN ( Up 2 Pressure Levels/ P ) for mandatory levels.
*    PDWN   - Known pressure at lower mandatory level (Pa).
*    PMID   - Known pressure at intermediate mandatory level (Pa).
*    PRES   - Vector of mandatory isobaric levels (Pa).
*    PUP    - Known pressure at upper mandatory level (Pa).
*    QDWN   - Quadratic multiplier of SDWN for interpolation of STHTA.
*    QMID   - Quadratic multiplier of SMID for interpolation of STHTA.
*    QUP    - Quadratic multiplier of SUP for interpolation of STHTA.
*    SDWN   - Scalar value at lower mandatory level.
*    SMID   - Scalar value at intermediate mandatory level.
*    SUP    - Scalar value at upper mandatory level.
************************************
      INCLUDE      'grdsiz.inc'

      INTEGER       MAXLVL
        PARAMETER (MAXLVL = 50)
      INTEGER       PLVLS
        PARAMETER (PLVLS = 16)

      INTEGER       I
      INTEGER       J
      INTEGER       KIN
      INTEGER       KOUT
      INTEGER       KTHTA
      INTEGER       NI
      INTEGER       NJ

      REAL          LNP1P2
```

```
      REAL          LNP1P2
      REAL          LNP2P3
      REAL          LNPU1P (PLVLS - 1)
      REAL          LNPU2P (PLVLS - 2)
      REAL          PDWN
      REAL          PMID
      REAL          PRES (PLVLS)
      REAL          PSFC (NIMAX, NJMAX)
      REAL          PTHTA (NIMAX, NJMAX, *)
      REAL          PUP
      REAL          QDWN
      REAL          QMID
      REAL          QUP
      REAL          SDWN
      REAL          SMID
      REAL          SSFC (NIMAX, NJMAX)
      REAL          SPRES (NIMAX, NJMAX, *)
      REAL          STHTA (NIMAX, NJMAX, MAXLVL)
      REAL          SUP
      REAL          THTA (*)

      DATA PRES     /100000., 92500., 85000., 70000., 50000., 40000.,
     &                30000., 25000., 20000., 15000., 10000.,  7000.,
     &                 5000.,  3000.,  2000.,  1000./

*     -----------------------------------------------------------------
*     Calculate and store log ratios for mandatory levels.
*     -----------------------------------------------------------------

      DO 50 KIN = 1, PLVLS - 2
        LNPU1P (KIN) = ALOG (PRES (KIN + 1) / PRES (KIN) )
        LNPU2P (KIN) = ALOG (PRES (KIN + 2) / PRES (KIN) )
   50 CONTINUE
      LNPU1P (PLVLS - 1) = ALOG (PRES (PLVLS) / PRES (PLVLS - 1) )

      PRINT *, 'INTERPOLATING SCALAR TO ', KTHTA,
     &         ' ISENTROPIC LEVELS...'
      DO 500 KOUT = 1, KTHTA
        DO 400 J = 1, NJ
          DO 300 I = 1, NI
            IF (PTHTA (I, J, KOUT) .LE. 0.) THEN

*           -----------------------------------------------------------
*           Theta level is either below surface or above lowest pressure
*           level at this (i, j) location.
*           -----------------------------------------------------------

              STHTA (I, J, KOUT) = -9999.
            ELSE IF (ABS (PTHTA (I, J, KOUT) - PSFC (I, J) ) .LT. .001)
     &              THEN

*           -----------------------------------------------------
*           Theta level is on surface.
*           -----------------------------------------------------

              STHTA (I, J, KOUT) = SSFC (I, J)
            ELSE
              KIN = 0
```

```
 100        CONTINUE
            KIN = KIN + 1
            IF (KIN .LE. PLVLS) THEN
              IF (ABS (PTHTA (I, J, KOUT) - PRES (KIN) ) .LT. .001)
     &          THEN
*
*             ---------------------------------------------------------
*             Theta level is on a mandatory isobaric level.
*             ---------------------------------------------------------

                STHTA (I, J, KOUT) = SPRES (I, J, KIN)
                GO TO 300
              ELSE IF (PTHTA (I, J, KOUT) .GT. PRES (KIN) ) THEN
                IF (KIN .EQ. 1) THEN
*
*               -------------------------------------------------------
*               Theta level is between surface and 1000 mb level.
*               -------------------------------------------------------

                  PDWN = PSFC (I, J)
                  SDWN = SSFC (I, J)
                  IF (ABS (PSFC (I, J) - PRES (KIN) ) .LT. .001) THEN
                    PMID = PRES (KIN)
                    PUP  = PRES (KIN + 1)
                    SMID = SPRES (I, J, KIN)
                    SUP  = SPRES (I, J, KIN + 1)
                    LNP1P2 = LNPU1P (KIN)
                    LNP1P3 = ALOG (PUP / PDWN)
                    LNP2P3 = ALOG (PMID / PDWN)
                  ELSE
                    PMID = PRES (KIN + 1)
                    PUP  = PRES (KIN + 2)
                    SMID = SPRES (I, J, KIN + 1)
                    SUP  = SPRES (I, J, KIN + 2)
                    LNP1P2 = LNPU1P (KIN + 1)
                    LNP1P3 = LNPU2P (KIN)
                    LNP2P3 = LNPU1P (KIN)
                  END IF
                ELSE IF (KIN .EQ. PLVLS) THEN
*
*               -------------------------------------------------------
*               Theta level is just below uppermost isobaric level.
*               -------------------------------------------------------

                  PDWN = PRES (KIN - 2)
                  PMID = PRES (KIN - 1)
                  PUP  = PRES (KIN)
                  SDWN = SPRES (I, J, KIN - 2)
                  SMID = SPRES (I, J, KIN - 1)
                  SUP  = SPRES (I, J, KIN)
                  LNP1P2 = LNPU1P (KIN - 1)
                  LNP1P3 = LNPU2P (KIN - 2)
                  LNP2P3 = LNPU1P (KIN - 2)
                ELSE IF (PSFC (I, J) .LT. PRES (KIN - 1) ) THEN
*
*               -------------------------------------------------------
*               Theta level is between surface and another mandatory
*               isobaric level.
```

```
*                       ------------------------------------------------------
                          PDWN = PSFC (I, J)
                          SDWN = SSFC (I, J)
                          IF (ABS (PSFC (I, J) - PRES (KIN) ) .GT. .001) THEN
                            PMID = PRES (KIN)
                            PUP  = PRES (KIN + 1)
                            SMID = SPRES (I, J, KIN)
                            SUP  = SPRES (I, J, KIN + 1)
                            LNP1P2 = LNPU1P (KIN)
                            LNP1P3 = ALOG (PUP / PDWN)
                            LNP2P3 = ALOG (PMID / PDWN)
                          ELSE
                            PMID = PRES (KIN + 1)
                            PUP  = PRES (KIN + 2)
                            SMID = SPRES (I, J, KIN + 1)
                            SUP  = SPRES (I, J, KIN + 2)
                            LNP1P2 = LNPU1P (KIN + 1)
                            LNP1P3 = LNPU2P (KIN)
                            LNP2P3 = LNPU1P (KIN)
                          END IF
                        ELSE
*                       ------------------------------------------------------
*                       Theta level is between two other mandatory isobaric
*                       levels.
*                       ------------------------------------------------------
                          PDWN = PRES (KIN - 1)
                          PMID = PRES (KIN)
                          PUP  = PRES (KIN + 1)
                          SDWN = SPRES (I, J, KIN - 1)
                          SMID = SPRES (I, J, KIN)
                          SUP  = SPRES (I, J, KIN + 1)
                          LNP1P2 = LNPU1P (KIN)
                          LNP1P3 = LNPU2P (KIN - 1)
                          LNP2P3 = LNPU1P (KIN - 1)
                        ENDIF
                        GO TO 200
                      END IF
                      GO TO 100
                    END IF
*                   ------------------------------------------------------
*                   Perform quadratic LaGrange interpolation against ln p.
*                   ------------------------------------------------------
  200               CONTINUE
                    QDWN = ALOG (PTHTA (I, J, KOUT) / PMID) *
     &                     ALOG (PTHTA (I, J, KOUT) / PUP) / LNP2P3 / LNP1P3
                    QMID = -ALOG (PTHTA (I, J, KOUT) / PDWN) *
     &                     ALOG (PTHTA (I, J, KOUT) / PUP) / LNP2P3 / LNP1P2
                    QUP = ALOG (PTHTA (I, J, KOUT) / PDWN) *
     &                    ALOG (PTHTA (I, J, KOUT) / PMID) / LNP1P3 / LNP1P2
                    STHTA (I, J, KOUT) = QDWN * SDWN + QMID * SMID + QUP * SUP
                  END IF
  300           CONTINUE
  400         CONTINUE
```

```
500 CONTINUE
    RETURN
    END
```

APPENDIX G

Isentropic Potential Vorticity Subroutine

```
      SUBROUTINE DOIPV (NI, NJ, LAT, LON, KTHTA, THTA, PTHTA, UTHTA,
     &                  VTHTA, IPV)
*********************************************************************
*********************************************************************
**  NAME:  DOIPV - CALCULATES POTENTIAL VORTICITY VALID ON AN ISENTROPIC
**               SURFACE
**
**  ROUTINE NARRATIVE:  This subroutine calculates and returns a scalar
**  3D grid of potential vorticity values on an isentropic surfaces from
**  isentropic wind and pressure fields.  When calculating the
**  stability, a centered difference is used in the vertical, where
**  possible.  Otherwise, a forward or backward difference is used as
**  appropriate (typically to account for missing data below the surface
**  or above 10mb).  Missing data is represented as -9999.  Calculation
**  of the stability assumes that ln T increases linearly with ln p.
**  This code was developed as part of thesis work by
**  Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
*********************************************************************
*********************************************************************
*  IPV (U, V) = -GRAVTY * ABSV (UTHTA, VTHTA) * DTHTA / DP
*
*  REFERENCE:
*    NOAA, NASA, USAF, 1976:  U.S. Standard Atmosphere.  Washington DC,
*      227 pp.
*
*  INPUT VARIABLES:
*    LAT    - Array containing latitudes (degrees).
*    LON    - Array containing longitudes (degrees).
*    KTHTA  - Number of vertical isentropic levels.
*    NI     - Number of data points in longitudinal direction (columns).
*    NJ     - Number of data points in latitudinal direction (rows).
*    PTHTA  - 3D Grid of isentropic pressures valid (Pa).
*    THTA   - Vector of isentropic surface values to calculate IPV upon
*             (K).
*    UTHTA  - 3D Grid containing grid-relative, U winds on isentropic
*             surface (m s-1).
*    VTHTA  - 3D Grid containing grid-relative, V wind on isentropic
*             surface (m s-1).
*
*  SUBROUTINES CALLED
*    DDX, DDY, DORELV, DOABSV
*
*  FUNCTIONS USED
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PARAMETER, Maximum number of grid rows.
*
*  OUTPUT:
```

```
*     IPV - 3D Scalar grid of isentrtopic potential vorticity
*           (m2 K kg-1 s-1).
*
*  PARAMETER VARIABLES:
*     CP     - Specific Heat of dry air at constant pressure
*              (J K-1 kg-1).
*     GRAVTY - Earth's gravitational acceleration (m s-2) (NOAA, NASA,
*              USAF, 1976).
*     KAPPA  - RD / CP.
*     MD     - Average molecular mass of dry air at sea level (kg)
*              (NOAA, NASA, USAF, 1976).
*     R      - Gas constant (J K-1 kg-1) (International Council of
*              Scientific Unions, CODATA Bulletin No. 11, Dec 1973).
*     RD     - Gas constasnt for dry air (J K-1 kg-1).
*
*  VARIABLES
*     ABSV   - Grid array containing the absolute vorticity (s-1).
*     DUDY   - Grid containing partial derivative of UTHTA with respect
*              to the Y-direction (s-1).
*     DVDX   - Grid containing partial derivative of VTHTA with respect
*              to the X-direction (s-1).
*     I      - Increments columns.
*     J      - Increments rows.
*     K      - Increments vertical isentropic levels.
*     RELV   - Grid of relative vorticity (s-1).
*     STABL  - Stability (change in potential temperature with respect to
*              pressure) (K Pa-1).
*     TDWN   - Temperature of lower layer used to calculate stability
*              (K).
*     TUP    - Temperature of upper layer used to calculate stability
*              (K).
************************************
      INCLUDE      'grdsiz.inc'

      INTEGER      KMAX
        PARAMETER (KMAX = 150)

      REAL         CP
        PARAMETER (CP = 1004.)
      REAL         GRAVTY
        PARAMETER (GRAVTY = 9.80665)
      REAL         MD
        PARAMETER (MD = 28.9644)
      REAL         R
        PARAMETER (R = 8314.41)
      REAL         RD
        PARAMETER (RD = R / MD)
      REAL         KAPPA
        PARAMETER (KAPPA = RD / CP)

      INTEGER      I
      INTEGER      J
      INTEGER      K
      INTEGER      KTHTA
      INTEGER      NI
      INTEGER      NJ

      REAL         ABSV (NIMAX, NJMAX)
```

```
      REAL         DUDY (NIMAX, NJMAX)
      REAL         DVDX (NIMAX, NJMAX)
      REAL         IPV (NIMAX, NJMAX, KMAX)
      REAL         LAT (*)
      REAL         LON (*)
      REAL         PTHTA (NIMAX, NJMAX, *)
      REAL         RELV (NIMAX, NJMAX)
      REAL         STABL
      REAL         TDWN
      REAL         THTA (*)
      REAL         TUP
      REAL         UTHTA (NIMAX, NJMAX, *)
      REAL         VTHTA (NIMAX, NJMAX, *)

      PRINT *, 'CALCULATING PV ON ', KTHTA, ' ISENTROPIC LEVELS...'
      DO 300 K = 1, KTHTA
        CALL DDX (VTHTA (1, 1, K), NI, NJ, LAT, LON, DVDX)
        CALL DDY (UTHTA (1, 1, K), NI, NJ, LAT, DUDY)
        CALL DORELV (NI, NJ, LAT, LON, UTHTA (1, 1, K), DVDX, DUDY,
     &               RELV)
        CALL DOABSV (NI, NJ, LAT, RELV, ABSV)

        DO 200 J = 1, NJ
          DO 100 I = 1, NI
            IF (K .EQ. 1) THEN
              IF (PTHTA (I, J, K) .LE. 0. .OR.
     &            PTHTA (I, J, K + 1) .LE. 0. .OR.
     &            ABSV (I, J) .LT. -9998.) THEN
                IPV (I, J, K) = -9999.
              ELSE
                TDWN = THTA (K) * (PTHTA (I, J, K) / 100000.)**KAPPA
                TUP = THTA (K + 1) *
     &                (PTHTA (I, J, K + 1) / 100000.)**KAPPA
                STABL = THTA (K) / PTHTA (I, J, K) *
     &                  (ALOG (TUP / TDWN) /
     &                   ALOG (PTHTA (I, J, K + 1) / PTHTA (I, J, K) ) -
     &                   KAPPA)
                IPV (I, J, K) = -GRAVTY * ABSV (I, J) * STABL
              END IF
            ELSE IF (K .EQ. KTHTA) THEN
              IF (PTHTA (I, J, K) .LE. 0. .OR.
     &            PTHTA (I, J, K - 1) .LE. 0. .OR.
     &            ABSV (I, J) .LT. -9998.) THEN
                IPV (I, J, K) = -9999.
              ELSE
                TDWN = THTA (K - 1) *
     &                 (PTHTA (I, J, K - 1) / 100000.)**KAPPA
                TUP = THTA (K) * (PTHTA (I, J, K) / 100000.)**KAPPA
                STABL = THTA (K) / PTHTA (I, J, K) *
     &                  (ALOG (TUP / TDWN) /
     &                   ALOG (PTHTA (I, J, K) / PTHTA (I, J, K - 1) ) -
     &                   KAPPA)
                IPV (I, J, K) = -GRAVTY * ABSV (I, J) * STABL
              END IF
            ELSE
              IF (PTHTA (I, J, K + 1) .GT. 0. .AND.
     &            PTHTA (I, J, K - 1) .GT. 0. .AND.
     &            ABSV (I, J) .GT. -9998.) THEN
```

```
              TDWN = THTA (K - 1) *
     &               (PTHTA (I, J, K - 1) / 100000.)**KAPPA
              TUP = THTA (K + 1) *
     &              (PTHTA (I, J, K + 1) / 100000.)**KAPPA
              STABL = THTA (K) / PTHTA (I, J, K) *
     &                (ALOG (TUP / TDWN) /
     &                 ALOG (PTHTA (I, J, K + 1) /
     &                       PTHTA (I, J, K - 1) ) - KAPPA)
              IPV (I, J, K) = -GRAVTY * ABSV (I, J) * STABL
            ELSE IF (PTHTA (I, J, K + 1) .LE. 0. .AND.
     &               PTHTA (I, J, K - 1) .GT. 0. .AND.
     &               PTHTA (I, J, K) .GT. 0. .AND.
     &               ABSV (I, J) .GT. -9998.) THEN
              TDWN = THTA (K - 1) *
     &               (PTHTA (I, J, K - 1) / 100000.)**KAPPA
              TUP = THTA (K) * (PTHTA (I, J, K) / 100000.)**KAPPA
              STABL = THTA (K) / PTHTA (I, J, K) *
     &                (ALOG (TUP / TDWN) /
     &                 ALOG (PTHTA (I, J, K) / PTHTA (I, J, K - 1) ) -
     &                 KAPPA)
              IPV (I, J, K) = -GRAVTY * ABSV (I, J) * STABL
            ELSE IF (PTHTA (I, J, K + 1) .GT. 0. .AND.
     &               PTHTA (I, J, K - 1) .LE. 0. .AND.
     &               PTHTA (I, J, K) .GT. 0. .AND.
     &               ABSV (I, J) .GT. -9998.) THEN
              TDWN = THTA (K) * (PTHTA (I, J, K) / 100000.)**KAPPA
              TUP = THTA (K + 1) *
     &              (PTHTA (I, J, K + 1) / 100000.)**KAPPA
              STABL = THTA (K) / PTHTA (I, J, K) *
     &                (ALOG (TUP / TDWN) /
     &                 ALOG (PTHTA (I, J, K + 1) / PTHTA (I, J, K) ) -
     &                 KAPPA)
              IPV (I, J, K) = -GRAVTY * ABSV (I, J) * STABL
            ELSE
              IPV (I, J, K) = -9999.
            END IF
          END IF
100     CONTINUE
200   CONTINUE
300 CONTINUE
    RETURN
    END
```

APPENDIX H

Partial Derivative with Respect to X-direction Subroutine

```
      SUBROUTINE DDX (S, NI, NJ, LAT, LON, DSDX)
************************************************************************
************************************************************************
**  NAME:  DDX - CALCULATES PARTIAL DERIVATIVE RELATIVE TO X-DIRECTION
**
**  ROUTINE NARRATIVE:  This subroutine calculates the partial
**  derivative of a scalar variable, S, with respect to the X-grid
**  direction (East), on a latitude/longitude-oriented (Cylindrical
**  Equidistant) grid where (1, 1) represents the upper left grid point
**  as typically used by AFGWC (Hoke et al, 1981).  It returns a grid of
**  values, DSDX.  The derivative is calculated using a 2d order
**  centered finite difference scheme, accounting for the possibility of
**  a global grid.  If the grid is not global, a 1st order forward and
**  backward difference are calculated on the starting and ending
**  columns, respectively.  1st order forward and backward difference
**  schemes are also used near missing data (represented as -9999).
**  Therefore, only if two of three successive points are missing is
**  the derivative declared missing.  This code was created as part of
**  thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*  REFERENCES:
*    HOKE, J.E., J.L. Hayes, L.G. Renninger, 1981:  Map projections and
*       grid systems for meteorological applications.  AFGWC/TN-79/003
*       (Revised Nov 83, Jun 85), Air Force Global Weather Central,
*       Offutt Air Force Base, NE, 87 pp.
*
*  INPUT VARIABLES:
*    LAT - Array containing latitudes of grid points (degrees).
*    LON - Array containing longitudes of grid points (degrees).
*    NI  - Number of data points in longitudinal direction (columns).
*    NJ  - Number of data points in latitudinal direction (rows).
*    S   - Grid of variable to compute the derivative of.
*
*  OUTPUT:
*    DSDX - Grid array containing the partial derivatives of S with
*           respect to X.
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER, Maximum number of grid columns.
*    NJMAX - INTEGER, Maximum number of grid rows.
*
*  PARAMETER VARIABLES (available from grid definition):
*    REARTH - Radius of the Earth, meters (Hoke et al, 1981).
*
*  VARIABLES:
*    DI - Longitudinal distance between data points, meters.
*    I  - Increments columns.
*    J  - Increments rows.
```

```
*     PI - Constant 'pi'.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE     'grdsiz.inc'

      REAL          REARTH
        PARAMETER (REARTH = 6371221.3)

      INTEGER       I
      INTEGER       J
      INTEGER       NI
      INTEGER       NJ

      REAL          DI
      REAL          DSDX (NIMAX, NJMAX)
      REAL          LAT (*)
      REAL          LON (*)
      REAL          PI
      REAL          S (NIMAX, NJMAX)

*     ------------------------------------------------------------------
*     Note:  Sun Fortran yields a compile warning when trying to define
*     PI as a parameter with the ASIN function.
*     ------------------------------------------------------------------

      PI  = 2. * ASIN (1.)

*     ------------------------------------------------------------------
*     Compute the partial derivative and place it in the new grid.
*     Loop over all grid rows.  Note that if the grid begins at the
*     pole, DSDX (1, J) = 0 (i.e., all grid points in first row
*     represent the same point).).
*     ------------------------------------------------------------------

      DO 200 J = 1, NJ

*       ----------------------------------------------------------------
*       Compute differential increment along X-direction for the given
*       latitude.
*       ----------------------------------------------------------------

        IF (ABS (LAT (J)) .GE. 90.) THEN
          DI = 2. * PI * COS (LAT (J) * PI / 180.) * REARTH *
     &         (LON (1) - LON (2)) / 360.
          DI = ABS (DI)
        END IF

*       ----------------------------------------------------------------
*       Loop over interior grid points in row J.  Perform forward or
*       backward differences near missing data.
*       ----------------------------------------------------------------

        DO 100 I = 2, NI - 1
          IF (ABS (LAT (J)) .GE. 90) THEN
            DSDX (I, J) = 0.
          ELSE IF (S (I + 1, J) .GT. -9998. .AND.
     &             S (I - 1, J) .GT. -9998.) THEN
            DSDX (I, J) = (S (I + 1, J) - S (I - 1, J)) / (2 * DI)
          ELSE IF (S (I + 1, J) .LT. -9998. .AND.
```

```
     &          S (I - 1, J) .GT. -9998. .AND. S (I, J) .GT. -9998.)
     &         THEN
          DSDX (I, J) = (S (I, J) - S (I - 1, J)) / DI
        ELSE IF (S (I + 1, J) .GT. -9998. .AND.
     &          S (I - 1, J) .LT. -9998. .AND. S (I, J) .GT. -9998.)
     &         THEN
          DSDX (I, J) = (S (I + 1, J) - S (I, J)) / DI
        ELSE
          DSDX (I, J) = -9999.
        END IF
  100   CONTINUE

*       --------------------------------------------------------------
*       Compute difference at the beginning and end of row J, accounting
*       for the possibility of a global grid.
*       --------------------------------------------------------------

        IF (ABS (LAT (J)) .GE. 90) THEN
          DSDX (1, J) = 0.
          DSDX (NI, J) = 0.
        ELSE IF (ABS (2 * LON (1) - LON (NI) - LON (2) ) .LT. .001 .OR.
     &          ABS (2 * LON (1) - LON (NI) - LON (2) + 360.) .LT.
     &          .001) THEN
          IF (S (2, J) .GT. -9998. .AND. S (NI, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (2, J) - S (NI, J)) / (2. * DI)
          ELSE IF (S (2, J) .LT. -9998. .AND. S (NI, J) .GT. -9998.
     &          .AND. S (1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (1, J) - S (NI, J)) / DI
          ELSE IF (S (2, J) .GT. -9998. .AND. S (NI, J) .LT. -9998.
     &          .AND. S (1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (2, J) - S (1, J)) / DI
          ELSE
            DSDX (1, J) = -9999.
          END IF
          IF (S (1, J) .GT. -9998. .AND. S (NI - 1, J) .GT. -9998.)
     &       THEN
            DSDX (NI, J) = (S (1, J) - S (NI - 1, J)) / (2. * DI)
          ELSE IF (S (1, J) .LT. -9998. .AND. S (NI - 1, J) .GT. -9998.
     &          .AND. S (NI, J) .GT. -9998.) THEN
            DSDX (NI, J) = (S (NI, J) - S (NI - 1, J)) / DI
          ELSE IF (S (1, J) .GT. -9998. .AND. S (NI - 1, J) .LT. -9998.
     &          .AND. S (NI, J) .GT. -9998.) THEN
            DSDX (NI, J) = (S (1, J) - S (NI, J)) / DI
          ELSE
            DSDX (NI, J) = -9999.
          END IF
        ELSE IF (ABS (LON (1) - LON (NI) ) .LT. .001) THEN
          IF (S (2, J) .GT. -9998. .AND. S (NI - 1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (2, J) - S (NI - 1, J)) / (2. * DI)
          ELSE IF (S (2, J) .LT. -9998. .AND. S (NI - 1, J) .GT. -9998.
     &          .AND. S (1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (1, J) - S (NI - 1, J)) / DI
          ELSE IF (S (2, J) .GT. -9998. .AND. S (NI - 1, J) .LT. -9998.
     &          .AND. S (1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (2, J) - S (1, J)) / DI
          ELSE
            DSDX (1, J) = -9999.
          END IF
```

```
          DSDX (NI, J) = DSDX (1, J)
        ELSE
          IF (S (2, J) .GT. -9998. .AND. S (1, J) .GT. -9998.) THEN
            DSDX (1, J) = (S (2, J) - S (1, J)) / DI
          ELSE
            DSDX (1, J) = -9999.
          END IF
          IF (S (NI, J) .GT. -9998. .AND. S (NI - 1, J) .GT. -9998.)
     &         THEN
            DSDX (NI, J) = (S (NI, J) - S (NI - 1, J)) / DI
          ELSE
            DSDX (NI, J) = -9999.
          END IF
        END IF
  200 CONTINUE
      RETURN
      END
```

APPENDIX I

Partial Derivative with Respect to Y-direction Subroutine

```
      SUBROUTINE DDY (S, NI, NJ, LAT, DSDY)
************************************************************************
************************************************************************
**  NAME:  DDY - CALCULATES PARTIAL DERIVATIVE RELATIVE TO Y-DIRECTION
**
**  ROUTINE NARRATIVE:  This subroutine calculates the partial
**  derivative of a scalar variable, S, with respect to the Y-grid
**  direction (North), on a latitude/longitude-oriented (Cylindrical
**  Equidistant) grid where (1, 1) represents the upper left grid point
**  as typically used by AFGWC (Hoke et al, 1981).  It returns a grid
**  of values, DSDY.  The derivative is calculated using a 2d order
**  centered finite difference scheme, accounting for the possibility of
**  a global grid.  If the grid is not global, a 1st order forward and
**  backward difference are calculated on the starting and ending
**  columns, respectively.  1st order forward and backward difference
**  schemes are also used near missing data (represented as -9999).
**  Therefore, only if two of three successive points are missing is
**  the derivative declared missing.  This code was created as part of
**  thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*  REFERENCES:
*    HOKE, J.E., J.L. Hayes, L.G. Renninger, 1981:  Map projections and
*      grid systems for meteorological applications.  AFGWC/TN-79/003
*      (Revised Nov 83, Jun 85), Air Force Global Weather Central,
*      Offutt Air Force Base, NE, 87 pp.
*
*  INPUT VARIABLES:
*    LAT    - Array containing latitudes for rows (degrees).
*    NI     - Number of data points in longitudinal direction (columns).
*    NJ     - Number of data points in latitudinal direction (rows).
*    S      - Grid of variable to compute the derivative of.
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER, Maximum number of grid columns.
*    NJMAX - INTEGER, Maximum number of grid rows.
*
*  OUTPUT:
*    DSDY - Grid array containing the partial derivatives of S with
*           respect to Y.
*
*  PARAMETER VARIABLES:
*    REARTH - Radius of the Earth, meters (Hoke et al, 1981).
*
*  VARIABLES:
*    DJ - Latitudinal distance between data points, meters.
*    I  - Increments columns.
*    J  - Increments rows.
*    PI - Constant 'pi'.
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE     'grdsiz.inc'

      REAL          REARTH
        PARAMETER (REARTH = 6371221.3)

      INTEGER       I
      INTEGER       J
      INTEGER       NI
      INTEGER       NJ

      REAL          DJ
      REAL          DSDY (NIMAX, NJMAX)
      REAL          LAT (*)
      REAL          PI
      REAL          S (NIMAX, NJMAX)

*     ------------------------------------------------------------------
*     Note:  Trying to assign PI as a parameter with the function ASIN
*     yields a warning with Sun Fortran.
*     ------------------------------------------------------------------

      PI  = 2. * ASIN (1.)

*     ------------------------------------------------------------------
*     Compute differential increment along Y-direction.
*     ------------------------------------------------------------------

      DJ = ABS ( ( (LAT (1) - LAT (2) ) / 180.) * PI * REARTH)

*     ------------------------------------------------------------------
*     Compute the partial derivative and place it in the new grid.  Loop
*     over interior grid rows, and compute forward and backward dif-
*     -ference along top and bottom rows, respectively.
*     ------------------------------------------------------------------

      DO 200 J = 1, NJ
        DO 100 I = 1, NI
          IF (J .EQ. 1) THEN
            IF (S (I, J) .GT. -9998. .AND. S (I, J + 1) .GT. -9998.)
     &          THEN
              DSDY (I, J) = (S (I, J) - S (I, J + 1) ) / DJ
            ELSE
              DSDY (I, J) = -9999.
            END IF
          ELSE IF (J .EQ. NJ) THEN
            IF (S (I, J) .GT. -9998. .AND. S (I, J - 1) .GT. -9998.)
     &          THEN
              DSDY (I, J) = (S (I, J - 1) - S (I, J) ) / DJ
            ELSE
              DSDY (I, J) = -9999.
            END IF
          ELSE
            IF (S (I, J - 1) .GT. -9998. .AND. S (I, J + 1) .GT. -9998.)
     &          THEN
              DSDY (I, J) = (S (I, J - 1) - S (I, J + 1) ) / (2. * DJ)
            ELSE IF (S (I, J - 1) .LT. -9998. .AND.
     &               S (I, J + 1) .GT. -9998. .AND.
```

```
     &                S (I, J) .GT. -9998.) THEN
              DSDY (I, J) = (S (I, J) - S (I, J + 1) ) / DJ
            ELSE IF (S (I, J - 1) .GT. -9998. .AND.
     &                S (I, J + 1) .LT. -9998. .AND.
     &                S (I, J) .GT. -9998.) THEN
              DSDY (I, J) = (S (I, J - 1) - S (I, J) ) / DJ
            ELSE
              DSDY (I, J) = -9999.
            END IF
          END IF
100     CONTINUE
200   CONTINUE
      RETURN
      END
```

APPENDIX J

Relative Vorticity Subroutine

```
      SUBROUTINE DORELV (NI, NJ, LAT, LON, UGRD, DVDX, DUDY, RELV)
**********************************************************************
**********************************************************************
**   NAME:  DORELV - CALCULATES RELATIVE VORTICITY
**
**   ROUTINE NARRATIVE:  This subroutine calculates relative vorticity
**   across a Cylinrical Equidistant (latitude-longitude) grid array and
**   returns the values in the array RELV.  A correction of
**   U * TAN (LAT) / REARTH accounts for the decreasing X-direction
**   distance as the grid approaches the pole (Bluestein, 1993).
**   Centered finite differences are used, except at the poles, where the
**   integral method is used (Bluestein, 1993).  If either derivative,
**   or UGRD is missing (-9999.) the vorticity is reported as missing.
**   This code was developed as part of thesis work by
**   Capt Jay DesJardins, AFIT/ENP.
**
**   LAST MODIFICATION DATE:  11 Mar 97
**********************************************************************
**********************************************************************
*   REFERENCES:
*     BLUESTEIN, H.B., 1993:  Synoptic-Dynamic Meteorology in
*       Midlatitudes, Vol I.  Oxford University Press, 431 pp.
*     HOKE, J.E., J.L. Hayes, L.G. Renninger, 1981:  Map projections and
*       grid systems for meteorological applications.  AFGWC/TN-79/003
*       (Revised Nov 83, Jun 85), Air Force Global Weather Central,
*       Offutt Air Force Base, NE, 87 pp.
*
*   RELV (U, V) = DDX (VGRD) - DDY (UGRD) + UGRD * TAN (LAT) / REARTH
*
*   where, the correction term on the right accounts for the changing
*   distance between grid points as you approach the pole.
*
*   INPUT VARIABLES:
*     DUDY  - Grid containing partial derivative of UGRD with respect
*             to the Y-direction.
*     DVDX  - Grid containing partial derivative of VGRD with respect
*             to X-direction.
*     LAT   - Array containing latitudes (degrees).
*     LON   - Array containing latitudes (degrees).
*     NI    - Number of gridpoints in longitudinal direction.
*     NJ    - Number of gridpoints in latitudinal direction.
*     UGRD  - Grid containing grid-relative, U-wind (East) component.
*
*   INCLUDE-(grdsiz.inc):
*     NIMAX - INTEGER, Maximum number of grid columns.
*     NJMAX - INTEGER, Maximum number of grid rows.
*
*   OUTPUT:
*     RELV  - Grid array containing the relative vorticity.
*
*   CALLED BY:
```

```
*    DOPV
*
*  PARAMETER VARIABLES:
*    REARTH - Radius of the Earth, meters (Hoke et al, 1981)
*
*  VARIABLES
*    I      - Increments columns.
*    J      - Increments rows.
*    MISSNG - Counter for missing data points at the poles.
*    PI     - Constant 'pi'.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE       'grdsiz.inc'

      REAL          REARTH
        PARAMETER (REARTH = 6371221.3)

      INTEGER       I
      INTEGER       J
      INTEGER       MISSNG
      INTEGER       NI
      INTEGER       NJ

      REAL          DUDY (NIMAX, NJMAX)
      REAL          DVDX (NIMAX, NJMAX)
      REAL          LAT (*)
      REAL          LON (*)
      REAL          PI
      REAL          RELV (NIMAX, NJMAX)
      REAL          UGRD (NIMAX, NJMAX)

*     ------------------------------------------------------------------
*     Note:  Sun Fortran yields a compile warning when trying to define
*     PI as a parmeter with the ASIN function.
*     ------------------------------------------------------------------

      PI = 2. * ASIN (1.)

*     ------------------------------------------------------------------
*     Determine the vorticity, accounting for the poles where vorticity
*     is defined using the circulation theorem around the nearest
*     latitude circle to the pole which eliminates the singularity at
*     the pole.
*     ------------------------------------------------------------------

      DO 1000 J = 1, NJ
        IF (ABS (LAT (J) - 90.) .LT. .001) THEN
          DO 100 I = 1, NI
            IF (UGRD (I, J + 1) .GT. -9998.) THEN
              RELV (1, J) = UGRD (I, J + 1)
              GO TO 200
            ELSE
              MISSNG = I
            END IF
  100     CONTINUE
  200     CONTINUE
          DO 300 I = MISSNG + 2, NI - 1
            IF (UGRD (I, J + 1) .GT. -9998.) THEN
              RELV (1, J) = RELV (1, J) + UGRD (I, J + 1)
```

```
          ELSE
            MISSNG = MISSNG + 1
          END IF
300     CONTINUE
        IF (ABS (LON (1) - LON (NI) ) .GT. .001) THEN
          IF (UGRD (NI, J + 1) .GT. -9998.) THEN
            RELV (1, J) = RELV (1, J) + UGRD (NI, J + 1)
          ELSE
            MISSNG = MISSNG + 1
          END IF
          RELV (1, J) = RELV (1, J) * COS (LAT (J + 1) * PI / 180.) /
     &                  (1. - SIN (LAT (J + 1) * PI / 180.) ) /
     &                  REARTH / FLOAT (NI - MISSNG)
        ELSE
          RELV (1, J) = RELV (1, J) * COS (LAT (J + 1) * PI / 180.) /
     &                  (1. - SIN (LAT (J + 1) * PI / 180.) ) /
     &                  REARTH / FLOAT (NI - 1 - MISSNG)
        ENDIF
        DO 400 I = 2, NI
          RELV (I, J) = RELV (1, J)
400     CONTINUE
      ELSE IF (ABS (LAT (J) + 90.) .LT. .001) THEN
        I = 0
500     CONTINUE
        I = I + 1
        IF (I .LE. NI) THEN
          IF (UGRD (I, J - 1) .GT. -9998.) THEN
            RELV (1, J) = UGRD (I, J - 1)
            GO TO 600
          ELSE
            MISSNG = I
            GO TO 500
          END IF
        END IF
600     CONTINUE
        DO 700 I = MISSNG + 2, NI - 1
          IF (UGRD (I, J + 1) .GT. -9998.) THEN
            RELV (1, J) = RELV (1, J) + UGRD (I, J - 1)
          ELSE
            MISSNG = MISSNG + 1
          END IF
700     CONTINUE
        IF (ABS (LON (1) - LON (NI) ) .GT. .001) THEN
          IF (UGRD (NI, J - 1) .GT. -9998.) THEN
            RELV (1, J) = RELV (1, J) + UGRD (NI, J - 1)
          ELSE
            MISSNG = MISSNG + 1
          END IF
          RELV (1, J) = RELV (1, J) * COS (LAT (J - 1) * PI / 180.) /
     &                  (1. - SIN (LAT (J - 1) * PI / 180.) ) /
     &                  REARTH / FLOAT (NI - MISSNG)
        ELSE
          RELV (1, J) = RELV (1, J) * COS (LAT (J - 1) * PI / 180.) /
     &                  (1. - SIN (LAT (J - 1) * PI / 180.) ) /
     &                  REARTH / FLOAT (NI - 1 - MISSNG)
        ENDIF
        DO 800 I = 2, NI
          RELV (I, J) = RELV (1, J)
```

```
 800       CONTINUE
        ELSE
          DO 900 I = 1, NI
            IF (UGRD (I, J) .LT. -9998. .OR. DVDX (I, J) .LT. -9998.
     &            .OR. DUDY (I, J) .LT. -9998.) THEN
              RELV (I, J) = -9999.
            ELSE
              RELV (I, J) = DVDX (I, J) - DUDY (I, J) + UGRD (I, J) *
     &                      TAN (LAT (J) * PI / 180.) / REARTH
            END IF
 900       CONTINUE
        ENDIF
1000 CONTINUE

      RETURN
      END
```

APPENDIX K

Absolute Vorticity Subroutine

```
      SUBROUTINE DOABSV (NI, NJ, LAT, RELV, ABSV)
************************************************************************
************************************************************************
**   NAME:  DOABSV - CALCULATES ABSOLUTE VORTICITY
**
**   ROUTINE NARRATIVE:  This subroutine calculates absolute vorticity
**   across a Cylinrical Equidistant (Lat/Long) grid array and returns
**   the values in the array ABSV.  If relative vorticity values are
**   missing (-9999.), so are absolute values.  This code was created as
**   part of thesis work by Capt Jay DesJardins, AFIT/ENP.
**
**   LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*    ABSV (U, V) = RELV (U, V) + CORL
*
*   INPUT VARIABLES:
*     LAT  - Array containing latitudes of grid rows (degrees).
*     NI   - Number of gridpoints in longitudinal direction.
*     NJ   - Number of gridpoints in latitudinal direction.
*     RELV - Grid of relative vorticity.
*
*   INCLUDE (grdsiz.inc):
*     NIMAX - INTEGER, Maximum number of grid columns.
*     NJMAX - INTEGER, Maximum number of grid rows.
*
*   OUTPUT:
*     ABSV  - Grid array containing the absolute vorticity.
*
*   PARAMETER VARIABLES:
*     OMEGA - Earth's angular velocity (radians per second).
*
*   VARIABLES
*     CORL - Coriolis parameter for a given latitude (row).
*     I    - Increments columns.
*     J    - Increments rows.
*     PI   - Constant 'pi'.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INCLUDE     'grdsiz.inc'

      REAL          OMEGA
        PARAMETER (OMEGA = 7.29212E-05)

      INTEGER       I
      INTEGER       J
      INTEGER       NI
      INTEGER       NJ

      REAL          ABSV (NIMAX, NJMAX)
      REAL          CORL
      REAL          LAT (*)
```

```
      REAL          PI
      REAL          RELV (NIMAX, NJMAX)

*     -------------------------------------------------------------------
*     Note:  Sun Fortran yields a compile warning when trying to define
*     PI as a parameter with the ASIN function.
*     -------------------------------------------------------------------

      PI = 2. * ASIN (1.)
      DO 200 J = 1, NJ
        CORL = 2. * OMEGA * SIN (LAT (J) * PI / 180.)
        DO 100 I = 1, NI
          IF (RELV (I, J) .GT. -9998.) THEN
            ABSV (I, J) = RELV (I, J) + CORL
          ELSE
            ABSV (I, J) = -9999.
          END IF
  100   CONTINUE
  200 CONTINUE
      RETURN
      END
```

APPENDIX L

Potential Vorticity at Constant Pressure Subroutine

```
      SUBROUTINE PVONP (NI, NJ, LAT, LON, PRES, PRES1, PRES2, TMP, TMP1,
     &                  TMP2, UGRD, UGRD1, UGRD2, VGRD, VGRD1,
     &                  VGRD2, PV)
************************************************************************
************************************************************************
**  NAME:  PVONP - CALCULATES POTENTIAL VORTICITY VALID ON AN ISOBARIC
**                 SURFACE
**
**  ROUTINE NARRATIVE:  This subroutine calculates and returns a scalar
**  grid of potential vorticity values on an isobaric surface from wind
**  and temperature fields.  To account for orientation of surface winds
**  along isentropic surface vice isobaric surfaces, the vorticity is
**  corrected by the addition of (k x partial V w.r.t. POT) dot
**  grad (POT) (Bluestein, 1993).  Calculation of the stability
**  (DTHTA/DP) assumes that ln T increases linearly with ln p.  This
**  code was developed as part of thesis work by Capt Jay DesJardins,
**  AFIT/ENP.
**
**  LAST MODIFICATION DATE:  11 Mar 97
************************************************************************
************************************************************************
*  REFERENCES:
*    BLUESTEIN, H.B., 1993:  Synoptic-Dynamic Meteorology in
*      Midlatitudes, Vol I.  Oxford University Press, 431 pp.
*    NOAA, NASA, USAF, 1976:  U.S. Standard Atmosphere.  Washington DC,
*      227 pp.
*
*  PV (U, V) = -GRAVTY * (ABSV (U, V) + VORCOR) * DTHTA / DP
*
*  where,  VORCOR = DU/D(POT) * DDY (POT) - DV/D(POT) * DDX (POT)
*
*  INPUT VARIABLES:
*    LAT   - Array containing latitudes (degrees).
*    LON   - Array containing longitudes (degrees).
*    NI    - Number of data points in longitudinal direction (columns).
*    NJ    - Number of data points in latitudinal direction (rows).
*    PRES  - Constant pressure level to calculate PV on (Pa).
*    PRES1 - Nearest upper pressure level.  Same as PRES if calculating
*            for top layer (Pa).
*    PRES2 - Nearest lower pressure level.  Same as PRES if calculating
*            for bottom layer (Pa).
*    TMP   - Temperature on constant pressure surface where PV is
*            calculated (K).
*    TMP1  - Nearest upper-level grid containing temperature.  Same
*            as TMP if calculating for upper layer (K).
*    TMP2  - Nearest lower-level grid containing temperature.  Same
*            as TMP if calculating for bottom layer (K).
*    UGRD  - Grid containing grid-relative, U-wind on constant pressure
*            surface where PV is calculated (m s-1).
*    UGRD1 - Nearest upper-level grid containing U-wind.  Same as UGRD
*            if calculating for top layer (m s-1).
```

```
*    UGRD2 - Nearest lower-level grid containing U-wind.  Same as UGRD
*            if calculating for bottom layer (m s-1).
*    VGRD  - Grid containing grid-relative, V-wind on constant pressure
*            surface where PV is calculated (m s-1).
*    VGRD1 - Nearest upper-level grid containing V-wind.  Same as VGRD
*            if calculating for top layer (m s-1).
*    VGRD2 - Nearest lower-level grid containing V-wind.  Same as VGRD
*            if calculating for bottom layer (m s-1).
*
*  SUBROUTINES CALLED
*    DDX, DDY, DORELV, DOABSV
*
*  FUNCTIONS USED
*    POT - Calculates potential temperature from pressure and
*          temperature.
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PARAMETER, Maximum number of grid rows.
*
*  OUTPUT:
*    PV - Scalar grid of potential vorticity for a given isobaric level
*         (s-1).
*
*  PARAMETER VARIABLES:
*    CP     - Specific Heat of dry air at constant pressure
*             (J K-1 kg-1).
*    GRAVTY - Earth's gravitational acceleration (m s-2) (NOAA, NASA,
*             USAF, 1976).
*    KAPPA  - RD / CP.
*    MD     - Average molecular mass of dry air at sea level (kg)
*             (NOAA, NASA, USAF, 1976).
*    R      - Gas constant (J K-1 kg-1) (International Council of
*             Scientific Unions, CODATA Bulletin No. 11, Dec 1973).
*    RD     - Gas constasnt for dry air (J K-1 kg-1).
*
*  VARIABLES
*    ABSV   - Grid array containing the absolute vorticity (s-1).
*    DPOTDX - Grid containing partial derivative of POT with respect
*             to the X-direction (K m-1).
*    DPOTDY - Grid containing partial derivative of POT with respect
*             to the Y-direction (K m-1).
*    DUDY   - Grid containing partial derivative of UGRD with respect to
*             the Y-direction (s-1).
*    DVDX   - Grid containing partial derivative of VGRD with respect to
*             X-direction (s-1).
*    I      - Increments columns.
*    J      - Increments rows.
*    LNP1P2 - Difference of natural logarithms of PRES1 from PRES2.
*    THETA  - Grid of potential temperature (K).
*    RELV   - Grid of relative vorticity (s-1).
*    STABL  - Stability (change in potential temperature with respect to
*             pressure) (K Pa-1).
*    VORCOR - Correction for layer-averaged vorticity: (k x partial V
*             w.r.t. POT) dot grad (POT) (s-1).
************************************
      INCLUDE    'grdsiz.inc'
```

```
      REAL          CP
        PARAMETER (CP = 1004.)
      REAL          GRAVTY
        PARAMETER (GRAVTY = 9.80665)
      REAL          MD
        PARAMETER (MD = 28.9644)
      REAL          R
        PARAMETER (R = 8314.41)
      REAL          RD
        PARAMETER (RD = R / MD)
      REAL          KAPPA
        PARAMETER (KAPPA = RD / CP)

      INTEGER       I
      INTEGER       J
      INTEGER       NI
      INTEGER       NJ

      REAL          ABSV (NIMAX, NJMAX)
      REAL          DPOTDX (NIMAX, NJMAX)
      REAL          DPOTDY (NIMAX, NJMAX)
      REAL          DUDY (NIMAX, NJMAX)
      REAL          DVDX (NIMAX, NJMAX)
      REAL          LAT (*)
      REAL          LNP1P2
      REAL          LON (*)
      REAL          POT
      REAL          PRES
      REAL          PRES1
      REAL          PRES2
      REAL          PV (NIMAX, NJMAX)
      REAL          RELV (NIMAX, NJMAX)
      REAL          STABL
      REAL          THETA (NIMAX, NJMAX)
      REAL          THETA1 (NIMAX, NJMAX)
      REAL          THETA2 (NIMAX, NJMAX)
      REAL          TMP (NIMAX, NJMAX)
      REAL          TMP1 (NIMAX, NJMAX)
      REAL          TMP2 (NIMAX, NJMAX)
      REAL          UGRD (NIMAX, NJMAX)
      REAL          UGRD1 (NIMAX, NJMAX)
      REAL          UGRD2 (NIMAX, NJMAX)
      REAL          VGRD (NIMAX, NJMAX)
      REAL          VGRD1 (NIMAX, NJMAX)
      REAL          VGRD2 (NIMAX, NJMAX)
      REAL          VORCOR

*     -----------------------------------------------------------------
*     Find the absolute vorticity for the given pressure level.
*     -----------------------------------------------------------------

      CALL DDX (VGRD, NI, NJ, LAT, LON, DVDX)
      CALL DDY (UGRD, NI, NJ, LAT, DUDY)
      CALL DORELV (NI, NJ, LAT, LON, UGRD, DVDX, DUDY, RELV)
      CALL DOABSV (NI, NJ, LAT, RELV, ABSV)

*     -----------------------------------------------------------------
*     Calculate vorticity correction due to orientation of isentropic
```

```
*     surface relative to isobaric surface.
*     ------------------------------------------------------------------

      DO 200 J = 1, NJ
        DO 100 I = 1, NI
          THETA (I, J) = POT (TMP (I, J), PRES)
          THETA1 (I, J) = POT (TMP1 (I, J), PRES1)
          THETA2 (I, J) = POT (TMP2 (I, J), PRES2)
  100   CONTINUE
  200 CONTINUE

      CALL DDX (THETA, NI, NJ, LAT, LON, DPOTDX)
      CALL DDY (THETA, NI, NJ, LAT, DPOTDY)

*     ------------------------------------------------------------------
*     Calculate the stability assuming that ln T, u, and v wind
*     components increase linearly with ln p.
*     ------------------------------------------------------------------

      LNP1P2 = ALOG (PRES1 / PRES2)

      DO 400 J = 1, NJ
        DO 300 I = 1, NI
          STABL = THETA (I, J) / PRES *
     &            (ALOG (TMP1 (I, J) / TMP2 (I, J) ) /
     &             LNP1P2 - KAPPA)
          VORCOR = ( (UGRD1 (I, J) - UGRD2 (I, J)) * DPOTDY (I, J) -
     &               (VGRD1 (I, J) - VGRD2 (I, J)) * DPOTDX (I, J) ) /
     &             (THETA1 (I, J) - THETA2 (I, J))
          PV (I, J) = -GRAVTY * (ABSV (I, J) + VORCOR) * STABL
  300   CONTINUE
  400 CONTINUE

*     ------------------------------------------------------------------
*     Assign values at the pole the average of the row representing the
*     point.
*     ------------------------------------------------------------------

      DO 700 J = 1, NJ
        IF (ABS (LAT (J) - 90.) .LT. .001) THEN
          DO 500 I = 2, NI - 1
            PV (1, J) = PV (1, J) + PV (I, J)
  500     CONTINUE
          IF (ABS (LON (1) - LON (NI) ) .LT. .001) THEN
            PV (1, J) = PV (1, J) / FLOAT (NI - 1)
          ELSE
            PV (1, J) = PV (1, J) + PV (NI, J)
            PV (1, J) = PV (1, J) / FLOAT (NI)
          END IF
          DO 600 I = 2, NI
            PV (I, J) = PV (1, J)
  600     CONTINUE
        END IF
  700 CONTINUE

      RETURN
      END
```

APPENDIX M

Potential Vorticity Valid in a Layer Subroutine

```
      SUBROUTINE PVLAYR (NI, NJ, LAT, LON, PRES1, PRES2, TMP1, TMP2,
     &                   UGRD1, UGRD2, VGRD1, VGRD2, PV)
************************************************************************
************************************************************************
**   NAME:  PVLAYR - CALCULATES POTENTIAL VORTICITY IN A LAYER
**
**   ROUTINE NARRATIVE:  This subroutine calculates and returns a scalar
**   grid of potential vorticity values in an isobaric layer (desJardins
**   et al., 1996) from wind and temperature fields.  To account for
**   orientation of surface winds along isentropic surface vice isobaric
**   surfaces, the vorticity of  the layer-averaged wind is corrected by
**   the addition of (k x partial V w.r.t. POT) dot grad (POT)
**   (Bluestein, 1993).  Layer averages are interpolated vertically
**   against ALOG (PRES).  This code was developed as part of thesis work
**   by Capt Jay DesJardins, AFIT/ENP.
**
**   LAST MODIFICATION DATE:  3 Mar 97
************************************************************************
************************************************************************
*   REFERENCES:
*     BLUESTEIN, H.B., 1993:  Synoptic-Dynamic Meteorology in
*       Midlatitudes, Vol I.  Oxford University Press, 431 pp.
*     desJARDINS, M.L, K.F. Brill, S. Jacobs, S.S. Schotz, P. Bruehl,
*       R. Schneider, B. Colman, D.W. Plummer, 1996:  General
*       Meteorological Package (GEMPAK), Software Version 5.4, National
*       Centers for Environmental Prediction, Washington D.C.
*     NOAA, NASA, USAF, 1976:  U.S. Standard Atmosphere.  Washington DC,
*       227 pp.
*
*   PV (U, V) = -GRAVTY * (ABSV (UAV, VAV) + VORCOR) * DPOT / DPRES
*
*   where,  VORCOR = (DU / DPOT) * DDY (POT) - (DV / DPOT) * DDX (POT)
*
*   INPUT VARIABLES:
*     ABSV    - Scalar grid containing the absolute vorticity (s-1).
*     LAT     - Array containing latitudes (degrees).
*     LON     - Array containing longitudes (degrees).
*     NI      - Number of data points in longitudinal direction (columns).
*     NJ      - Number of data points in latitudinal direction (rows).
*     PRES1   - Pressure of top level (Pa).
*     PRES2   - Pressure of bottom level (Pa).
*     TMP1    - Top-level grid containing temperature (K).
*     TMP2    - Bottom-level grid containing temperature (K).
*     UGRD1   - Top-level grid containing grid-relative, U wind (m s-1).
*     UGRD2   - Bottom-level grid containing grid-relative, U wind
*               (m s-1).
*     VGRD1   - Top-level grid containing grid-relative, V wind (m s-1).
*     VGRD2   - Bottom-level grid containing grid-relative, V wind
*               (m s-1).
*
*   SUBROUTINES CALLED
```

```
*    DDX, DDY, DORELV, DOABSV
*
*  FUNCTIONS USED
*    POT - Calculates potential temperature from pressure and
*          temperature
*
*  INCLUDE (grdsiz.inc):
*    NIMAX - INTEGER PARAMETER, Maximum number of grid columns.
*    NJMAX - INTEGER PARAMETER, Maximum number of grid rows.
*
*  OUTPUT:
*    PV - Scalar grid of potential vorticity values for a given layer.
*
*  PARAMETER VARIABLES:
*    CP     - Specific Heat of dry air at constant pressure
*             (J K-1 kg-1).
*    GRAVTY - Earth's gravitational acceleration (m s-2) (NOAA, NASA,
*             USAF, 1976).
*    KAPPA  - RD / CP.
*    MD     - Average molecular mass of dry air at sea level (kg)
*             (NOAA, NASA, USAF, 1976).
*    R      - Gas constant (J K-1 kg-1) (International Council of
*             Scientific Unions, CODATA Bulletin No. 11, Dec 1973).
*    RD     - Gas constasnt for dry air (J K-1 kg-1).
*
*  VARIABLES
*    ABSV   - Grid array containing the absolute vorticity (s-1).
*    DPOTDX - Grid containing partial derivative of POT with respect
*             to the X-direction (K m-1).
*    DPOTDY - Grid containing partial derivative of POT with respect
*             to the Y-direction (K m-1).
*    DUDY   - Grid containing partial derivative of UGRD with respect to
*             the Y-direction (s-1).
*    DVDX   - Grid containing partial derivative of VGRD with respect to
*             X-direction (s-1).
*    I      - Increments columns.
*    J      - Increments rows.
*    LNP1   - Natural logarithm of upper pressure value.
*    LNP2   - Natural logarithm of lower pressure value.
*    PAV    - Layer-averaged pressure (Pa).
*    POTAV  - Grid containing layer-averaged potential temperature (K).
*    PI     - Constant 'pi'.
*    RELV   - Grid of relative vorticity (s-1).
*    STABL  - Stability (change in potential temperature with respect to
*             pressure) (K Pa-1).
*    TAV    - Layer-averaged temperature (K).
*    UAV    - Grid containing layer-averaged U wind (m s-1).
*    VAV    - Grid containing layer-averaged V wind (m s-1).
*    VORCOR - Correction for layer-averaged vorticity: (k x partial V
*             w.r.t. POT) dot grad (POT) (s-1).
************************************
      INCLUDE     'grdsiz.inc'

      REAL         CP
        PARAMETER (CP = 1004.)
      REAL         GRAVTY
        PARAMETER (GRAVTY = 9.80665)
      REAL         MD
```

```
      PARAMETER (MD = 28.9644)
    REAL        R
      PARAMETER (R = 8314.41)
    REAL        RD
      PARAMETER (RD = R / MD)
    REAL        KAPPA
      PARAMETER (KAPPA = RD / CP)

    INTEGER     I
    INTEGER     J
    INTEGER     NI
    INTEGER     NJ

    REAL        ABSV (NIMAX, NJMAX)
    REAL        DPOTDX (NIMAX, NJMAX)
    REAL        DPOTDY (NIMAX, NJMAX)
    REAL        DUDY (NIMAX, NJMAX)
    REAL        DVDX (NIMAX, NJMAX)
    REAL        LAT (*)
    REAL        LNP1
    REAL        LNP2
    REAL        LON (*)
    REAL        PAV
    REAL        POT
    REAL        POTAV (NIMAX, NJMAX)
    REAL        PRES1
    REAL        PRES2
    REAL        PV (NIMAX, NJMAX)
    REAL        RELV (NIMAX, NJMAX)
    REAL        STABL
    REAL        TAV
    REAL        TMP1 (NIMAX, NJMAX)
    REAL        TMP2 (NIMAX, NJMAX)
    REAL        UAV (NIMAX, NJMAX)
    REAL        UGRD1 (NIMAX, NJMAX)
    REAL        UGRD2 (NIMAX, NJMAX)
    REAL        VAV (NIMAX, NJMAX)
    REAL        VGRD1 (NIMAX, NJMAX)
    REAL        VGRD2 (NIMAX, NJMAX)
    REAL        VORCOR

    LNP1 = ALOG (PRES1)
    LNP2 = ALOG (PRES2)

*   ---------------------------------------------------------------
*   Calculate the layer-averaged wind to use for calculating ABSV and
*   a layer-averaged ln (T) to calculate a layer-averaged potential
*   temperature.  These averages are weighted against ln (p) which is
*   more representative than straight linear averages.
*   ---------------------------------------------------------------

    PAV = (PRES1 * LNP1 + PRES2 * LNP2 ) / (LNP1 + LNP2)
    DO 200 J = 1, NJ
      DO 100 I = 1, NI
        UAV (I, J) = (LNP2 * UGRD2 (I, J) + LNP1 * UGRD1 (I, J) ) /
   &                 (LNP1 + LNP2)
        VAV (I, J) = (LNP2 * VGRD2 (I, J) + LNP1 * VGRD1 (I, J) ) /
   &                 (LNP1 + LNP2)
```

```
         TAV = EXP ( (LNP2 * ALOG (TMP2 (I, J) ) +
     &                LNP1 * ALOG (TMP1 (I, J) ) ) / (LNP1 + LNP2) )
         POTAV (I, J) = POT (TAV, PAV)
  100  CONTINUE
  200 CONTINUE

      CALL DDX (VAV, NI, NJ, LAT, LON, DVDX)
      CALL DDY (UAV, NI, NJ, LAT, DUDY)
      CALL DORELV (NI, NJ, LAT, LON, UAV, DVDX, DUDY, RELV)
      CALL DOABSV (NI, NJ, LAT, RELV, ABSV)
      CALL DDX (POTAV, NI, NJ, LAT, LON, DPOTDX)
      CALL DDY (POTAV, NI, NJ, LAT, DPOTDY)

*     -----------------------------------------------------------------
*     Calculate PV valid at pressure-weighted level:
*     PAV = (P1 * LN (P1) + P2 * LN (P2) ) /  LN (P1 * P2)
*     -----------------------------------------------------------------


      DO 400 J = 1, NJ
        DO 300 I = 1, NI
          STABL = POTAV (I, J) / PAV *
     &               (ALOG (TMP1 (I, J) / TMP2 (I, J) ) /
     &                (LNP1 - LNP2) - KAPPA)
          VORCOR = ( (UGRD1 (I, J) - UGRD2 (I, J)) * DPOTDY (I, J) -
     &                  (VGRD1 (I, J) - VGRD2 (I, J)) * DPOTDX (I, J) ) /
     &                (POT (TMP1 (I, J), PRES1) - POT (TMP2 (I, J), PRES2))
          PV (I, J) = -GRAVTY * (ABSV (I, J) + VORCOR) * STABL
  300   CONTINUE
  400 CONTINUE

*     -----------------------------------------------------------------
*     Assign values at the pole the average of the row representing the
*     point.
*     -----------------------------------------------------------------

      DO 700 J = 1, NJ
        IF (ABS (LAT (J)) .EQ. 90.) THEN
          DO 500 I = 2, NI - 1
            PV (1, J) = PV (1, J) + PV (I, J)
  500     CONTINUE
          IF (LON (1) .EQ. LON (NI)) THEN
            PV (1, J) = PV (1, J) / FLOAT (NI - 1)
          ELSE
            PV (1, J) = PV (1, J) + PV (NI, J)
            PV (1, J) = PV (1, J) / FLOAT (NI)
          END IF
          DO 600 I = 2, NI
            PV (I, J) = PV (1, J)
  600     CONTINUE
        END IF
  700 CONTINUE
      RETURN
      END
```

VITA

Capt Jay B DesJardins, Jr., [redacted]. He received a commission through Officer Training School in 1988 following graduation from the University of Wisconsin in 1987 with a Bachelor of Science in meteorology.

First assigned to Travis AFB CA, he held positions as a forecaster, Wing Weather Officer, Current Operations Officer, and Officer-in-Charge of a Weather Support Unit. Jay was the lead Air Force weather officer for airlift operations in Korea for exercise TEAM SPIRIT, and in New Zealand/Antarctica for Operation DEEP FREEZE. Next, assigned to Headquarters, Air Weather Service, Scott AFB IL, he performed acquisition work for the Automated Weather Distribution System program and as Assistant Chief, Manpower and Organization Division. At Scott, Jay completed Squadron Officer School in residence, and was enrolled in the graduate meteorology program at Saint Louis University. In August 1995, he was selected to enter the Air Force Institute of Technology, as the Class Leader for the inaugural class in the meteorology program. Following graduation, Jay will be assigned to Headquarters, Joint Special Operations Command at Fort Bragg NC.

Jay is an active member of the American Meteorological Society and Air Weather Association. Jay married the former Debra J. Murrell in 1989. They have three children: Dylan, [redacted] Désirée, [redacted] and Sydney, [redacted]

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1997	Master's Thesis

4. TITLE AND SUBTITLE

DEVELOPMENT AND IMPLEMENTATION OF AN ISENTROPIC POTENTIAL VORTICITY ALGORITHM FOR USE AT AIR FORCE GLOBAL WEATHER CENTER

5. FUNDING NUMBERS

6. AUTHOR(S)

Jay B DesJardins, Jr., Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
2950 P Street
Wright-Patterson AFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GM/ENP/97M-3

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFGWC/SYS
Maj Randy Lefevre
106 Peacekeeper Drive, Suite 2N3
Offutt AFB, NE 68113-4039

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis presents and validates methods for calculating isentropic potential vorticity (IPV) and applies these methods in software programs planned for implementation at the Air Force Global Weather Center (AFGWC). The IPV programs will provide Air Force Weather forecasters additional tools to diagnose atmospheric kinematics and understand atmospheric dynamics. A FORTRAN program is recommended using mandatory-level isobaric data projected to be available on AFGWC computer systems, specifically, from the Navy Operational Global Atmosphere Prediction System and Medium Range Forecast models. Program development and analysis consists of three main steps: (1) data retrieval; (2) IPV calculations; and, (3) interpolation to an isentropic vertical coordinate system. This thesis recommends performing IPV calculations at constant pressure for comparison with other mandatory-level isobaric parameters, or in routine cross-sectional analysis. Additionally, a recommendation is made to calculate IPV at constant potential temperature from interpolated isentropic state variables instead of interpolating isobaric IPV fields. Applications of the developed programs include visualization of synoptic-scale motions an alternative method of locating the tropopause in cross-sectional analysis. This thesis is a significant effort to move toward operational use of isentropic analysis and the incorporation of IPV analysis into forecasting techniques at AFGWC.

14. SUBJECT TERMS

Potential Vorticity, Isentropic Analysis, Isentropic Interpolation, Isentropic Potential Vorticity

15. NUMBER OF PAGES

132

16. PRICE CODE

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