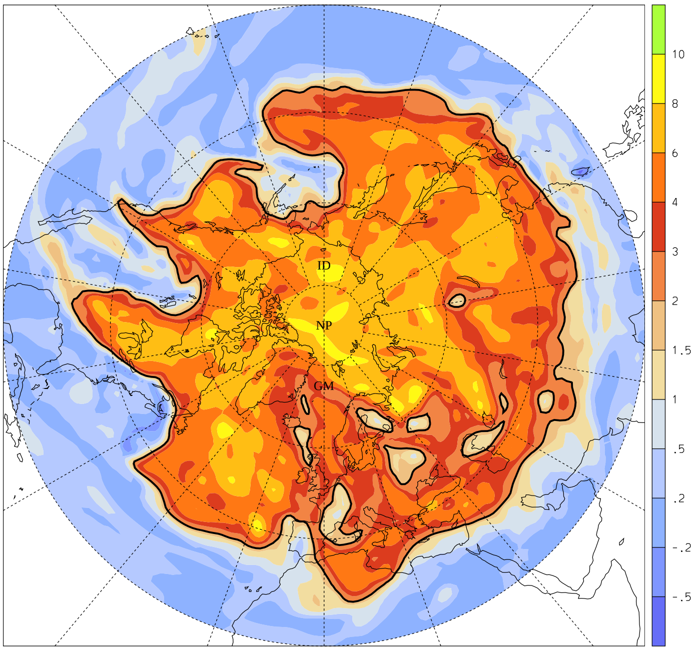
Numerical Piecewise Potential Vorticity Inversion
A user guide for real-case experiments



Michael Sprenger, Ph. D.[1]

Institute for Atmospheric and Climate Science, ETH Zürich, Switzerland

February 18, 2007

Diploma Thesis for the Postgraduate Course in Computer Science, FHSS Schweiz,
under the supervision of Lars Kruse, Ph. D.

[1] *Corresponding author and his address:*
Michael Sprenger
Institute for Climate and Atmospheric Science, *IAC*ETH
ETH Zurich
CH-8092 Zurich, Switzerland
e-mail: michael.sprenger@env.ethz.ch

Title figure: Ertel's potential vorticity at 320 K.

Abstract

Potential vorticity (PV) is a key quantity in atmospheric dynamics. Its value is based upon several points: (a) PV is conserved in purely adiabatic flows, i.e. if radiative and condensational heating and other diabatic processes can be neglected; (b) under suitable balance conditions the specification of PV in the interior of a domain and with boundary values for potential temperature completely determines the flow field, in particular horizontal wind and temperature can be derived; (c) in many respects the atmosphere can be partitioned into several distinct PV features, which then interact and allow the dynamics to be interpreted as the interaction of these PV elements. Points (b) and (c) are known as the invertibility and partitioning principle, respectively.

In this thesis, a program package is presented which allows to isolate PV elements and then to study their impact on the atmospheric flow field and on the temperature distribution. Technically, this is done with a so-called PV inversion, which in turn comprises several different steps, as for example transformation into suitable co-ordinate systems and numerical solution of a poisson equation. With PV inversion, two major problems arise: Firstly, a suitable balance condition must be formulated, secondly the inversion problem for Ertel's PV is nonlinear, and therefore poses a significant challenge. The program presented in this thesis is based upon the so-called quasi-geostrophic balance condition. More specifically, the linear quasi-geostrophic PV equation is solved. Since quasi-geostrophic and Ertel's PV do not exactly coincide, an iterative technique is adopted to approach the nonlinear Ertel-PV inversion by means of successively quasi-geostrophic inversions.

The aim of this work is to present an in-depth discussion of all steps which are needed for a PV inversion. Special focus was given to a clear and user-friendly program package, where all steps are well documented and hence are attractive for further development. In this respect, the complete inversion is controlled by one single Linux Shell script. The new user needs only to adjust some few paths in this script. The main parameters for the inversion itself are specified in a separate parameter file.

Zusammenfassung

Die potentielle Vortizität (PV) ist eine Kerngrösse der dynamischen Meteorologie. Ihre Bedeutung basiert auf mehreren Eigenschaften: (a) In einer adiabatischen Strömung, dh. wenn Strahlung, die Freisetzung von latenter Wärme und andere diabatische Prozesse vernachlässigt werden können, ist PV eine lagrange'sche Erhaltungsgrösse; (b) unter geeigneten Balancebedingungen bestimmt die Verteilung der PV im Innern und der potentiellen Temperatur am Rand eines Gebietes vollständig den Zustand der Atmosphäre, insbesondere die horizontalen Windfelder und das Temperaturfeld; (c) häufig lässt sich die Atmosphäre in mehrere, klar voneinander getrennte PV-Elemente unterteilen, die dann die Dynamik der Atmosphäre durch ihre Wechselwirkung bestimmen. Die Eigenschaften (b) und (c) sind in der Literatur bekannt als das Invertibilitätsprinzip und Partitionierungsprinzip.

In dieser Arbeit wird ein Programmpaket vorgestellt, mit dessen Hilfe sich einzelne PV-Elemente isolieren und ihr Einfluss auf die atmosphärischen Windfelder und Temperaturen studieren lassen. Programmtechnisch besteht diese sogenannte PV-Inversion aus mehreren Schritten. So muss zum Beispiel eine Koordinatentransformation durchgeführt werden und eine Poisson-Gleichung numerisch gelöst werden. Damit die PV-Inversion sinnvoll ist, müssen zwei Probleme gelöst werden. Zunächst muss eine geeignete Balancebedingung vorgegegeben sein. Ausserdem handelt es sich bei der Inversion um einen nichtlinearen Prozess, der numerisch einige Herausforderungen stellt. In dem Programmpaket dieser Arbeit wird die sogenannte quasi-geostrophische Balance verwendet, dh. der mathematische/numerische Inversionsprozess besteht in der Lösung der linearen quasi-geostrophischen PV-Gleichung. Das Problem der Nichtlinearität wird dadurch gelöst, dass die Berechnung des genannten linearen Inversionsproblems mehrmals wiederholt wird. Iterativ ergibt sich somit eine Lösung des nichtlinearen Problems.

Ziel dieser Arbeit ist eine detailierte Darstellung aller Schritte, die bei einer PV-Inversion nötig sind. Besonderes Augenmerk wurde bei der Entwicklung des Programmpakets auf die klare Strukturierung und Anwenderfreundlichkeit gelegt. Dadurch kann das Paket einerseits als Black-Box verwendet werden, zugleich erlaubt es erfahrenen Benutzern eine leichte Einarbeitung in die Programme und damit die Möglichkeit zu deren Erweiterung. Die ganze Inversion wird von einem einzigen Linux Shell Skript kontrolliert, in dem lediglich einige wenige Pfade angepasst werden müssen. Die meisten Parameter, welche das gestellte Problem der PV-Inversion beschreiben, sind in einer separaten Parameterdatei eingetragen.

Contents

Abbreviations

[1]	**DWD**	Deutscher Wetterdienst (German weather service)
[2]	**ECMWF**	European Centre for Medium Range Weather Forecasts
[3]	**ERA-40**	Re-analysis of the ECMWF for the periods 1958-2001
[4]	**ERA-15**	Re-analysis of the ECMWF for the periods 1979-1993
[5]	**ETH**	Eidgenössische Technische Hochschule
[6]	**f**	Coriolis parameter (in s^{-1})
[7]	**IACETH**	Institute for Atmospheric and Climate Science at ETH Zurich
[8]	**METEOSAT**	European geostationary weather satellite
[9]	**MeteoSwiss**	Swiss weather service
[10]	**NaN**	Not a Number (missing data flag)
[11]	**netcdf**	Compact and random-access file format
[12]	**NWP**	Numerical weather prediction
[13]	N^2	Squared Brunt-Vaisälä frequency (in s^-2)
[14]	**PV**	Potential vorticity (if not otherwise stated Ertel's potential vorticity)
[15]	**pvu**	Potential vorticity unit
[16]	**PDE**	Partial differential equation
[17]	**SOR**	Successive over-relaxation
[18]	**T**	Temperature (in °C)
[19]	T_v	Virtual temperature (in °C)
[20]	**U**	Zonal velocity component (in west/east direction)
[21]	**V**	Meridional velocity component (in south/north direction)
[22]	ω	Vertical velocity component (in Pa/s)
[23]	**q**	Specific humidity (in kg/kg)
[24]	θ	Potential temperature (in K)
[25]	ρ	Air density (in kg/m^3)
[26]	$\mathbf{R}_d$	Gas constant for dry air (in kg/m^3)
[27]	ϵ	Ratio of gas constants for dry air and water vapour
[28]	**g**	Earth's gravity (in kg m/s^2)

List of Figures

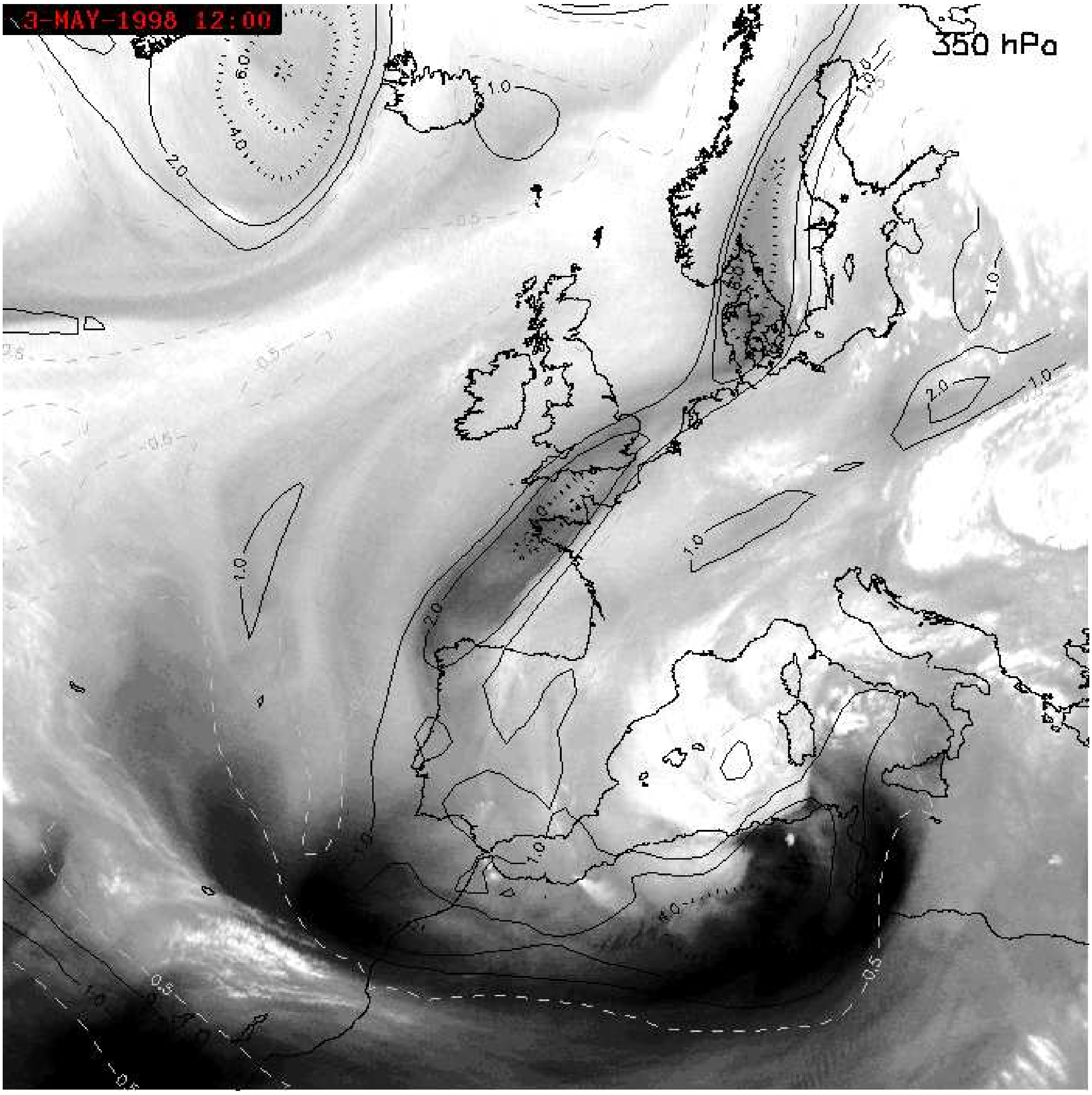


Figure 1: *Water vapour satellite imagery (in gray shading) and isolines of Ertel's potential vorticity (PV) at 350 hPa and for 3 May 1998, 12 UTC. Dark regions indicate a dry upper troposphere, whereas bright and white regions are indicative for a lot of moisture in the upper troposphere. Note that high-PV regions are coincident with dry regions.*

1 Introduction and Motivation

There are different ways how the state of the atmosphere can be described. Traditionally this is done by specification of temperature, horizontal and vertical wind components and of either pressure, if geometrical height is used as the vertical co-ordinate, or geopotential height, if pressure is used as the vertical co-ordinate. This traditional approach has its main advantage in its simplicity. On the other hand, theoretical physics has often experienced the case that new insight can be gained if abstract, but physically more fundamental quantities were introduced. Consider for example the introduction of action and its primary physical parameter, the Planck constant. Similarly, geophysical fluid dynamics and hence atmospheric dynamics made some transitions away from the simple traditional meteorological fields toward more abstract ones.

A long existing concept of engineering fluid dynamics is the concept of vorticity, which mathematically is defined as the curl of the wind vector field. Vorticity is treated in nearly every text book on fluid dynamics, for instance in Acheson (1990). Vorticity is a scalar quantity, but nevertheless, in a divergent-free and two-dimensional flow its specification is sufficient to deduce the complete velocity field. We could call this the invertibility principle for vorticity in such a non-divergent and two-dimensional flow. Moreover, an interesting conservation law hold for vorticity under these assumptions: Following the fluid motion, vorticity is conserved, i.e. its lagrangian derivative with respect to time is zero. Probably, in its most elegant way this conservation principle is expressed in Kelvin's circulation theorem (Acheson, 1990). These concepts of barotropic flow and of conservation of vorticity were indeed the basis for the first numerical weather predictions by Charney in 1950 (for an interesting history of numerical weather prediction, consult either Lynch, 2006, or Nebeker, 1995).

Of course, the atmosphere cannot be treated in an exact way as a barotropic fluid. It constitutes a stratified fluid where density and pressure decreases with increasing height. Therefore, a suitable generalisation of the barotropic-vorticity equation for the real atmosphere is by far not trivial, but would be highly desirable. Potential vorticity goes into this direction. The fundamental work by Rossby (1940) and Ertel (1942) showed that potential vorticity is conserved in adiabatic (no latent heat release, no radiative heating or cooling) and frictionless flow. The generalisation of the "invertibility principle for vorticity" in two-dimensional flow was first stated in a rough way by Kleinschmidt (1950). He was able to attribute some low-level flow features to an upper-level PV anomaly, in his words to a "Zyklonenkörper". Figure 1 is a nice illustration of an upper-level PV anomaly. It shows some isolines of Ertel's PV on 350 hPa, which form an elongated and narrow filament of high PV extending from Scandinavia south to the north-western edge of Spain. Remarkably, this anomaly of high PV is also discernible as a dark band in the water vapour imagery of the geostationary METEOSAT weather satellite. Hence, the high PV band is associated with a very dry upper troposphere.

A very influential set of equations was introduced by Charney by means of a scale analysis of the dynamical equations (1948). These so called quasi-geostrophic equations are well suitable to describe synoptic and planetary-scale processes, whilst neglecting smaller-scale features. Although these equations nowadays are considered no longer of sufficient accuracy for numerical weather prediction, they nevertheless still are of great importance in theoretical dynamic meteorology due to their simplicity and elegance (Holton, 1992). Along with this set of equations came a new version of potential vorticity. This quasi-geostrophic potential vorticity is linearly "linked" to the flow and temperature field, and -particularly interesting for this study- an invertibility principle can be formulated. Indeed, the linear relationship between quasi-geostrophic potential vorticity and the flow

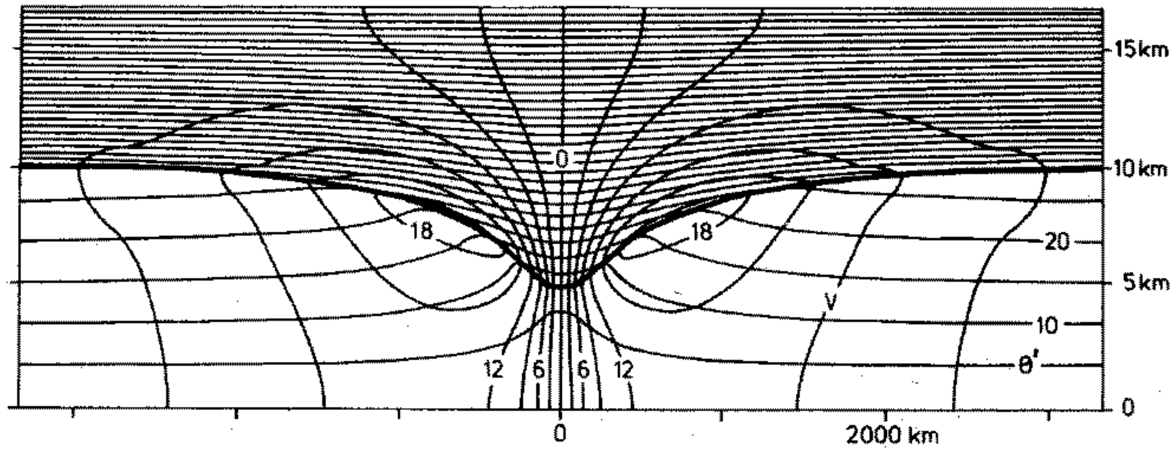


Figure 2: *Section across the circularly symmetric structure induced by an isolated, circularly symmetric, cyclonic potential vorticity anomaly near the model tropopause across which the Rossby-Ertel potential vorticity has a strong discontinuity, by a factor of 6. The bold line is the tropopause, the horizontal thin lines are potential temperature, plotted at 5 K intervals, and the circular/elliptical lines denote tangential velocity, at 3 m/s intervals. The structure is typical for middle latitudes; the Coriolis parameter is* $10^{-4}s^{-1}$ *(as at geographical latitude* 43.3° *N). The domain shown has a radius of 2500 km (taken from McIntyre, 1997).*

field makes it very interesting since sophisticated numerical techniques exist for the solution of this kind of problem. An example of such an inversion is shown in Fig. 2 where a cyclonic potential-vorticity anomaly near a model tropopause is shown. The potential vorticity is high in the stratosphere and low in the troposphere. Therefore, the downward excursion of the tropopause is associated locally with anomalously high potential vorticity. This anomaly is marked in the figure with stippling. The "horizontal" lines correspond to the potential temperature, the "circular/elliptical" contours to the tangential velocity. Note that the isolines of potential temperature (the so-called isentropes) are pulled upwards below the PV anomaly, and pulled downward within the PV anomaly. An interesting aspect of the upward pulling of the isentropes is the associated reduction in atmospheric stability. Due to the reduced vertical gradient of potential temperature, the atmosphere is more prone to convective instability. In fact, in a recent study Martius et al. (2006) showed that many heavy precipitation events in the south of the Alps are associated with such upper-level distortions of the tropopause, and hence with a PV anomaly. Moreover, note that the upper-level PV anomaly is not only associated with a deformation of the isentropes, but is also linked with a wind field. The induced wind field in this idealised setting reaches 21 m/s at its maximum and is cyclonically (anti-clockwise) oriented. The wind field is strongest at the tropopause level, but is even discernible at the surface far below the anomaly. In this respect, the PV field exhibits a far-field effect, and this in turn is the basic idea behind the invertibility principle. If this principle is taken together with the partitioning principle, its explaining power becomes particularly attractive. This latter principles states that the atmospheric state can be expressed as the

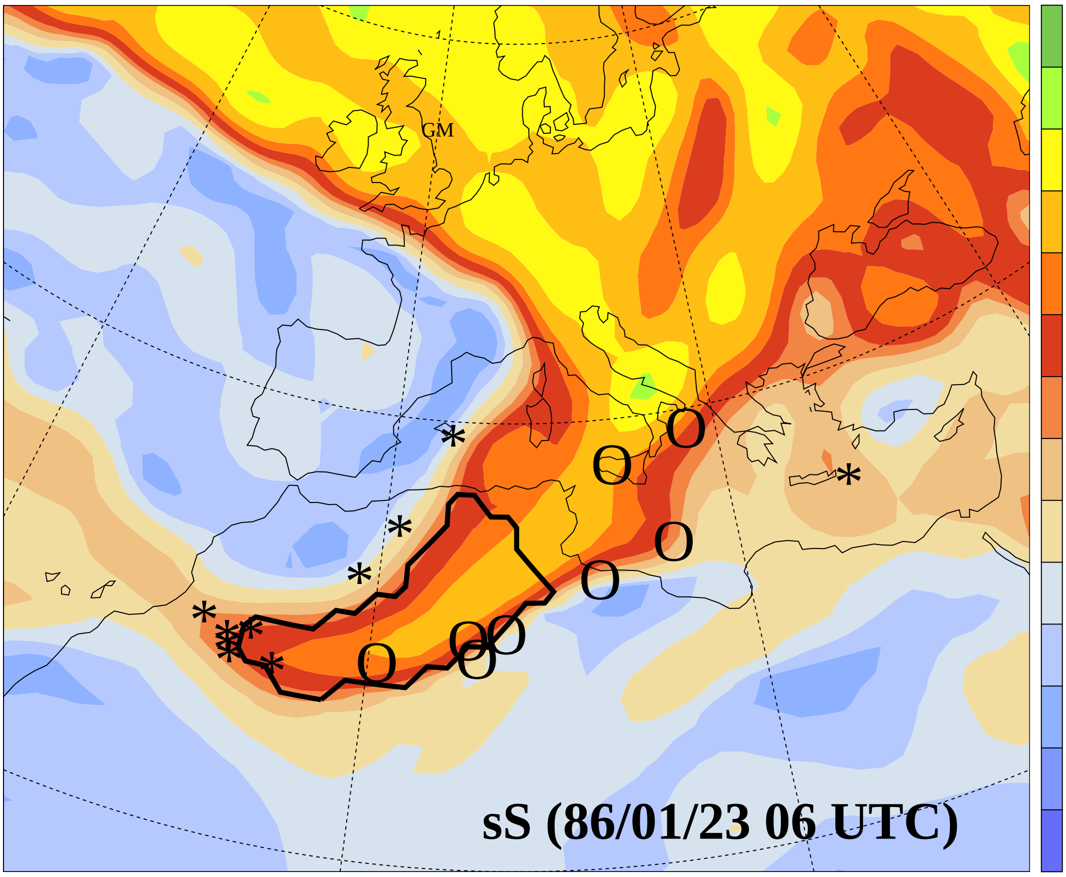


Figure 3: *Stratospheric PV streamer and positions with stratosphere-troposphere mass exchange. The section is for 1 January 1986, 06 UTC and on the 320 K isentrope. It is taken from the ERA-15 data set of the ECMWF. In color Ertel's potential vorticity is shown in pvu. Mass exchange from the stratosphere to the troposphere is marked with stars, transport in the opposite direction by open circles (taken from Sprenger et al., 2007)*

interaction of isolated PV elements. This immediately invokes a research strategy, which is: to enter, modify or remove some of these PV elements, and to see how the flow evolves in the thus changed state. A very influential review article on PV and PV thinking was presented by Hoskins et al. (1985). Here the key elements of PV thinking are discussed and applied to many atmospheric dynamical problems.

Many studies relate to this PV thinking perspective. For instance, Davis and Emanuel (1991) looked at the potential vorticity diagnostics of cyclogenesis; Fehlmann and Davies (1997) investigated the impact of PV structures on precipitation events over Europe. In addition to this methodology, PV has gained a lot of interest in recent years as a diagnostic tool. Indeed, from a dynamicist's point of view it is highly attractive to use potential vorticity as the defining component of the extra-tropical tropopause (Stohl et al. 2003). Typically, this value is set to 2 pvu, and every air parcel with larger PV is treated as stratospheric and every air parcel with lower PV as tropospheric. This definition allows to discuss the dynamics of the extra-tropical tropopause, as it would not be feasible with the more common thermal (or lapse-rate) definition of the tropopause. In Fig. 3 a prominent feature of the extra-tropical tropopause is shown: a stratospheric PV streamer. It corresponds to a pronounced excursion of stratospheric (high-PV) air towards the south. These features are quite common in the extra-tropics and are important for the mass exchange between the stratosphere and the troposphere, amongst other impacts (for example they are often related to cyclogenesis). The positions where stratospheric mass (and chemical constituents, as for example ozone) crosses the tropopause are marked in the figure by stars. They are predominantly found on the upstream (western) side of the streamer. Crossing in the opposite direction, i.e. from the troposphere to the stratosphere, on the other hand occurs on the downstream (eastern) side of the streamer. Interesting questions emerge from the inspection of a such a figure: If the intensity, or the southward excursion of the PV streamer was reduced, would there still be a significant mass flux across the tropopause? These, and similar questions are at the heart of the PV inversion, which is introduced in this study.

The study is organised in the following way: In chapter 2 the mathematical problem of PV inversion is formulated and some essential aspects of the numerical algorithm are discussed. Chapter 3 introduces some key aspects for the re-structuring of the program package. Then, in chapter 4 a detailed example of PV inversion is discussed. This part is intended to be a practical user guide for PV inversion, and therefore rather detailed technical aspects are presented. Chapter 5 follows with some helpful diagnostic tools which allow to quality and impact of the PV inversion. Finally, chapter 6 concludes with some general remarks and wishes regarding the PV inversion tool.

2 Mathematical Framework and Numerical Aspects

In this study we use the quasi-geostrophic approximation for the real-case inversion problem. In this limit the relative (quasi-geostrophic) potential vorticity q is given by:

$$q = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z}(\frac{\rho_0}{N_0^2} \cdot \frac{\partial \psi}{\partial z})$$

with the boundary values of potential temperature at the upper and lower lid and of zonal and meridional wind components at the lateral boundaries:

$$g \cdot \frac{\theta^*}{\theta_0} = f \cdot \frac{\partial \psi}{\partial z} \qquad u = -\frac{\partial \psi}{\partial y} \qquad v = -\frac{\partial \psi}{\partial x}$$

Here N_0^2, ρ_0 and θ_0 denote the squared Brunt-Väsäla frequency, the density and the potential temperature of a reference state depending only on the vertical co-ordinate z. ψ is the streamfunction from which the horizontal wind components can be derived. Finally, f is the Coriolis parameter which measures the Earth's rotation rate (typically $10^{-4}s^{-1}$ in the mid-latitudes and 0 at the equator), and g is the Earth's gravity.

The above equations constitute a so-called von Neumann boundary problem for the elliptic differential equation relating the potential vorticity q to the streamfunction ψ. Hence, the elliptic partial differential equation (PDE) can be transformed into the general form:

$$\frac{\partial}{\partial x}(\alpha \cdot \frac{\partial \psi}{\partial x}) + \frac{\partial}{\partial y}(\beta \cdot \frac{\partial \psi}{\partial y}) + \frac{\partial}{\partial z}(\gamma \cdot \frac{\partial \psi}{\partial z}) + \sigma = 0$$

where the coefficients α, β, γ and σ are easily expressed in terms of the density ρ_0, the Coriolis parameter f and the squared Brunt-Vaisälä frequency N^2:

$$\alpha = \rho_0(z) \qquad \beta = \rho_0(z) \qquad \gamma = \frac{f^2(y) \cdot \rho_0(z)}{N_0^2(z)} \qquad \sigma = -\rho_0(z) \cdot q(x, y, z)$$

The next step is to discretise the above PDE by means of finite differences. We assume that all fields (quasi-geostrophic PV q, streamfunction ψ,...) are defined on a three-dimensional grid, whose grid points can be addressed by the three indices i, j, and k in the x-, y- and z-direction (see Fig. refgrid). With this convention, the discretised PDE can be written as:

$$\Delta_x(A \cdot \Delta_x \psi) + \Delta_y(B \cdot \Delta_y \psi) + \Delta_z(C \cdot \Delta_z \psi) + S = 0$$

Here the operators $\Delta_x(A \cdot \Delta_x \psi)$, $\Delta_y(B \cdot \Delta_y \psi)$ and $\Delta_z(C \cdot \Delta_z \psi)$ at the grid position i, j, k are defined by:

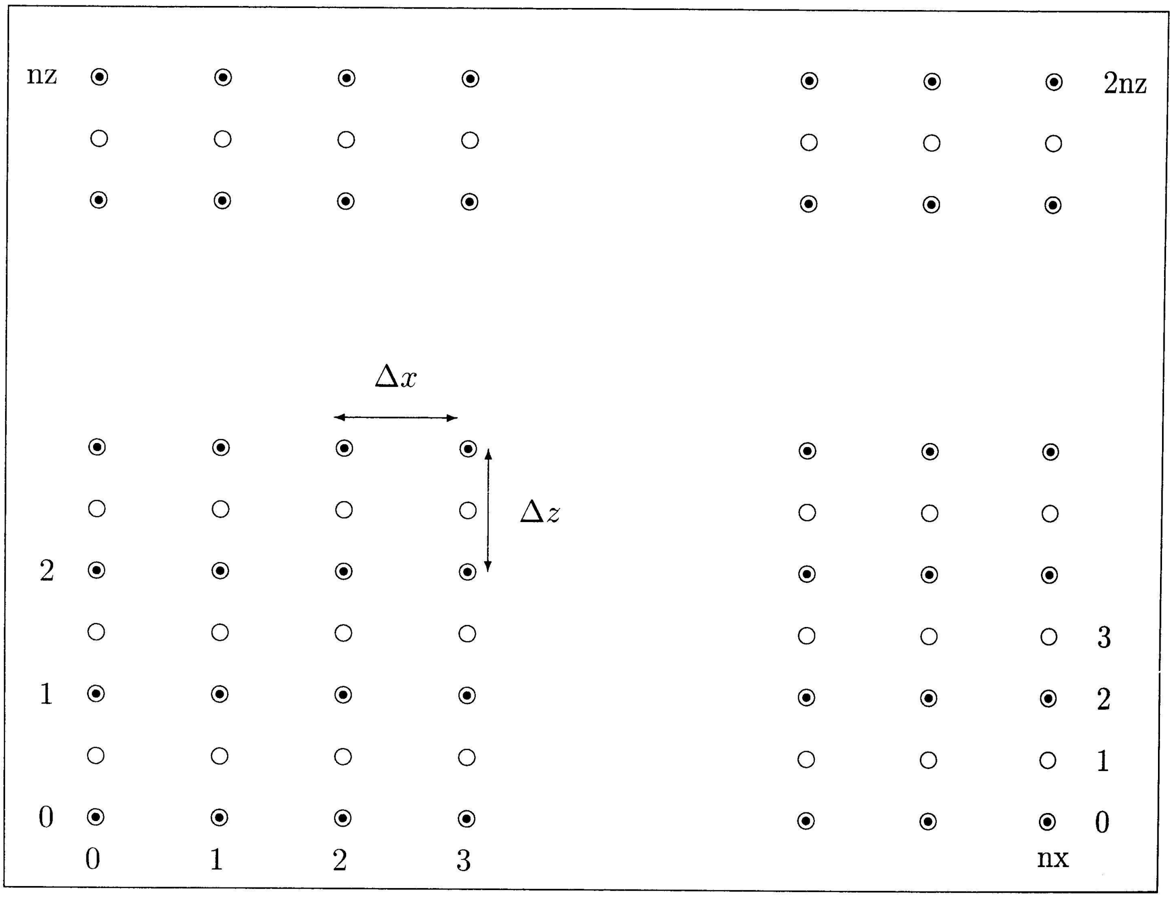


Figure 4: *The numerical grid for the PV inversion in an xz cross-section. The horizontal grid spacing is* Δx *in the x-direction (and correspondingly* Δy *in the y-direction), the vertical grid spacing* Δz*. Note that in the vertical, a staggered grid at half-levels is also needed (taken from Fehlmann, 1997).*

$$\begin{aligned}
\Delta_x(A \cdot \Delta_x \psi)_{i,j,k} &= A_{i+1/2,j,k} \cdot (\psi_{i+1,j,k} - \psi_{i,j,k}) - A_{i-1/2,j,k} \cdot (\psi_{i,j,k} - \psi_{i-1,j,k}) \\
\Delta_y(B \cdot \Delta_y \psi)_{i,j,k} &= B_{i,j+1/2,k} \cdot (\psi_{i,j+1,k} - \psi_{i,j,k}) - B_{i,j-1/2,k} \cdot (\psi_{i,j,k} - \psi_{i,j-1,k}) \\
\Delta_z(C \cdot \Delta_z \psi)_{i,j,k} &= C_{i,j,k+1/2} \cdot (\psi_{i,j,k+1} - \psi_{i,j,k}) - C_{i,j,k-1/2} \cdot (\psi_{i,j,k} - \psi_{i,j,k-1})
\end{aligned}$$

Some simple algebra yields the following expressions for A, B and C:

$$\begin{aligned}
A_{i,j,k} &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \alpha_{i,j,k} \\
B_{i,j,k} &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \beta_{i,j,k} \\
C_{i,j,k} &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \gamma_{i,j,k}
\end{aligned}$$

Finally, the additive operator S is expressed as:

$$S_{i,j,k} = \Delta x \cdot \Delta y \cdot \Delta z \cdot \sigma_{i,j,k}$$

It turns out to be advantageous to have the coefficients α, β, γ and σ not only on the 3d grid where the quasi-geostrophic PV and the streamfunction are defined, but to have them also on intermediate points. These leads to the grid structure which is shown in Fig. 4. On the left side the indices for the ψ-grid is shown, i.e. for the grid where the quasi-geostrophic PV and the streamfunction is defined. The right side gives the indices for the coefficients α, β, γ and σ, i.e. for the intermediate layers.

Let $\rho(k) = \rho_{k/2}$ and $N^2(k) = N^2_{k/2}$, hence expressing the vertical grid with intermediate layers. With these definitions the coefficients A, B and C for the inversion problem are readily obtained (note that these coefficients are dependent only on the vertical index z):

$$\begin{aligned}
A_k &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \rho(2k) \quad (k = 0, ..., nz) \\
B_k &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \rho(2k) \quad (k = 0, ..., nz) \\
C_k &= \frac{\Delta y \cdot \Delta z}{\Delta x} \cdot \frac{\rho(k) \cdot f^2}{N^2(k)} \quad (k = 0, ...2 \cdot nz)
\end{aligned}$$

The additive operator S depends on all three grid indices. In its discretised form it is:

$$S_{i,j,k} = -\Delta x \cdot \Delta y \cdot \Delta z \cdot \rho(2k) \cdot q_{i,j,k} \quad (i = 0, ..., nx, j = 0, ..., ny, k = 0, ..., nz)$$

It is convenient to additionally define the coefficients $C_{-1} = C_0$ and $C_{2nz+1} = C_{2nz}$. With all these definitions the fully discretised quasi-geostrophic PV equation can be written as:

$$\begin{aligned}
S_{i,j,k} = &\quad (\psi_{i-1,j,k} - 2\psi_{i,j,k} + \psi_{i+1,j,k}) \cdot A_k \\
&+ (\psi_{i,j-1,k} - 2\psi_{i,j,k} + \psi_{i,j+1,k}) \cdot B_k \\
&+ (\psi_{i,j,k-1} - \psi_{i,j,k}) \cdot C_{2k-1} \\
&+ (\psi_{i,j,k+1} - \psi_{i,j,k}) \cdot C_{2k+1}
\end{aligned}$$

with the indices ranges as follows: $i = 0, ...nx$ and $k = 0, ...nz$. In addition to this discretised PV equation, additional equations must be specified for terms like $\psi_{-1,j,k}$. These discretised von Neumann boundary conditions are:

$$\begin{aligned}
A_k \cdot (\psi_{0,j,k} - \psi_{-1,j,k}) &= \Delta y \cdot \Delta z \cdot \rho(2k) \cdot v_a(j,k) \\
A_k \cdot (\psi_{nx+1,j,k} - \psi_{nx,j,k}) &= \Delta y \cdot \Delta z \cdot \rho(2k) \cdot v_b(j,k) \\
B_k \cdot (\psi_{i,0,k} - \psi_{i,-1,k}) &= -\Delta x \cdot \Delta z \cdot \rho(2k) \cdot u_a(i,k) \\
B_k \cdot (\psi_{i,ny+1,k} - \psi_{i,ny,k}) &= -\Delta x \cdot \Delta z \cdot \rho(2k) \cdot u_b(i,k)
\end{aligned}$$

where u_a, u_b denote the velocity components in x direction at the left (western) and right (eastern) lateral boundary, and correspondingly v_a, v_b denote the velocity components in y direction at the front (southern) and back (northern) lateral boundary. Additional boundary values are specified at the lower and upper lid of the domain:

$$\begin{aligned} C_{-1} \cdot (\psi_{i,j,0} - \psi_{i,j,-1}) &= \Delta x \cdot \Delta z \cdot \frac{f \cdot g \cdot \rho(0) \cdot \theta_{bot}(i,j)}{N^2(0) \cdot \theta_0(0)} \\ C_{2nz+1} \cdot (\psi_{i,j,nz+1} - \psi_{i,j,nz}) &= \Delta x \cdot \Delta z \cdot \frac{f \cdot g \cdot \rho(2nz) \cdot \theta_{top}(i,j)}{N^2(2nz) \cdot \theta_0(2nz)} \end{aligned}$$

Here, θ_{bot} and θ_{top} denote the boundary conditions for potential temperature at the lower and upper lid of the domain.

This is a system of linear equations which can be expressed as

$$B\psi = b$$

where B is a $m \times m$ matrix with in total $m = (nx+1) \cdot (ny+1) \cdot (nz+1)$ elements (not to be confused with the above coefficients B_k). This linear system has a solution if the following condition is fulfilled:

$$\sum_{i,j,k} b_{i,j,k} = 0$$

This is exactly the discrete version of the compatibility condition in section 5.1. The derivation of the condition starts with the observation that the null space, i.e. the kernel of the system is at least one-dimensional because the non-trivial vector $\psi_{i,j,k} = 1 \forall i,j,k$ is element of this kernel. Physically, this expresses the fact that the streamfunction is determined only up to an additive constant. Since the kernel of the linear system is at least one-dimensional, the image of the operator B is at most (m-1) dimensional. Moreover, the operator B is normal, and therefore the image is orthogonal to the kernel. Because b is in the image of the operator B and $\psi_{i,j,k} = 1$ is in the kernel, the two vectors are orthogonal. This leads immediately to the necessary consistency condition $\sum_{i,j,k} b_{i,j,k} = 0$. Note that this expresses a complicated relationship which must be fulfilled by the interior PV distribution and the boundary values of potential temperature and horizontal wind components, because b includes all these forcing terms.

The consistency condition is not met if vanishing boundary values are specified. This automatically leads to an inconsistency in the numerical solution of the equation. In practice, the fulfillment of the consistency condition can be enforced if a potential temperature "correction" is added at the lower and upper lid, this correction being uniform and of opposite sign on the two boundaries. For reasonable PV and temperature distributions this additive temperature shift remains smaller than about 2 K.

There exist several techniques how to solve linear systems of equations. Here we adopt the successive over-relaxation (SOR) method: Let A be a linear operator which is represented by an $m \times m$ matrix, and let ω be a real number (the relaxation parameter) such that $|1 - \omega(1 + A)| < 1$. Then a solution of the equation can be obtained iteratively, starting with an arbitrary first guess $\psi_i^{(0)}$. The iterations

$$\psi_i^{(n+1)} = \omega \cdot (b_i - \sum_{j=1}^{i-1} A_{i,j} \cdot \psi_j^{(n+1)} - \sum_{j=i}^{m} A_{i,j} \cdot \psi_j^{(n)}) + (1 - \omega) \cdot \psi_i^{(n)}$$

converge toward the solution of the system $A\psi + \psi = b$. If we choose $A = B - 1$, the iterations converge toward a solution of the quasi-geostrophic PV equation.

Note that the above outlined algorithm allows to overwrite the variable $\psi_i^{(n)}$ with the updated variable $\psi_i^{(n+1)}$, and therefore needs a minimum of computational memory. In the Fortran program *invcart.f* the number of iterations and the SOR parameter ω are specified. The are set to 500 iterations and 1.81, respectively.

3 Re-Structuring of the Code

In this section some key concepts of the program package are described. In fact, an earlier version of the package already existed (developed originally by Rene Fehlmann and later modified by Sebastien Dirren) and the present work is based upon this pre-existing package. So, the question arises what added value this re-development and re-structuring of the code is associated with. The basic idea is to provide a code which fulfills the following three requirements:

a) **Transparency and Modularity**

b) **Universality and Model Independence**

c) **User friendliness**

In the following parts, a detailed discussion is intended to illustrate how this requirements were missing in the existing code and how they could be incorporated into the new version. Before doing so, it is worthwhile to consider why the original code was lacking these requirements. A short review of the relevant literature illustrates that the situation is actually quite common. Any software which is really used has to be adapted to the changing demands. It must be maintained. Unfortunately this maintenance often leads to a degeneration of the code. The changes introduced into the code then make necessary some further adaptions, and so on. Finally, in the end the code becomes so "distorted" and "chaotic" that often a complete re-coding is easier to be done than a re-structuring of the existing version. The remedy against this code degeneration is a continuous re-structuring. Hence, if changes in the code are obligatory due to new user demands, it is important not only to introduce the needed changes short-sightedly. The overall structure of the program should be kept in mind, and if possible long-term perspectives in code maintenance should be allowed, although such a long-term perspective momentarily leads to an additional effort.

Computer science, in particular software engineering, has defined the term "re-structuring" (or "re-factoring" for object-oriented programming languages) for the process how code should be maintained in order to avoid severe degeneration. Fowler defines the term refactoring in the following way (taken from URL *www.refactoring.com*):

> "Refactoring is a disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior. Its heart is a series of small behavior preserving transformations. Each transformation (called a 'refactoring') does little, but a sequence of transformations can produce a significant restructuring. Since each refactoring is small, it's less likely to go wrong. The system is also kept fully working after each small refactoring, reducing the chances that a system can get seriously broken during the restructuring."

Hence, the key aspect of re-structuring is to continuously maintain the code, not only in its performance, but also in its readability, documentation and adaptabilty to needed changes.

Scientific programming in particular is very prone to code degeneration. This is certainly due to the fact that cutting-edge science is, by "definition", investigating what is not known. Therefore, if for instance a computer program is written to numerically simulate a physical process, it might well turn out during the development of the code that complete new aspects must be included. This naturally leads to chaotically structured code. In the present case, i.e. for the PV inversion program, this certainly explains to a large part why the code became so un-organised. Several works have been done with the program, several researchers included into the code what they needed for their specific task, without considering that other users might have no use of their changes. The results was a program package which, in principle, was still working, but the knowledge of how to apply it went lost.

A first rough inspection immediately made clear that major changes were necessary to make the code available to a broader community again. Would it have been possible to only re-organise the code and write a new user guide? In a first try, this was in fact considered. But then the advantages of a more or less complete re-coding turned out to be more suitable. Hence, the original software package joined the sad fate of so many degenerated codes: a complete re-coding. On the other hand, such a re-coding must be seen as a great chance. It allows to improve the code in such a way, as it would never be possible if only "slight" changes at the existing code were made.

In the following three subsections, some key aspects of this re-coding will be presented. They are by no means exhausting, but should give an impression of the new "philosophy". Hopefully, they also motivate future users of the program package to "successfully" maintain it.

3.1 Transparency and Modularity

What makes a computer program readable? Probably one of the most important points is "modularity". The problem to be solved numerically can and should be split into several distinct steps. For instance, for the PV inversion a classic three step splitting is appropriate: pre-processing, PV inversion, and post-processing. Moreover, each of these three primary steps can further be split into several secondary sub-steps. In the existing code, the splitting of the problem into distinct sub-problems was not clearly discernible. Indeed, the main Fortran program included not only the numerical inversion of the quasi-geostrophic PV equation, but also some of the preparatory steps and some of the post-processing steps.

A major improvement was gained by a very strict separation of the distinct primary steps. So, pre-processing, inversion and post-processing are done by three completely separated program packages. This strict separation is supported by the fact that three separated directories are used: There is one directory where pre-processing is done, on directory where inversion is done, and finally one directory where post-processing is done. A flow of data between the three directories is allowed only at well-defined steps within the whole process. At the end of pre-processing the relevant files, and only those, are moved to the inversion (or run) directory. Similarly, at the end of the inversion, the relevant files are moved from the run directory to the post-processing directory.

Additional improvement resulted from a very clear separation of the sub-processes in the three main-processes (pre-processing, inversion, post-processing). In fact, it was not tried to incorporate all preparatory steps into one large Fortran program, but instead each well-defined task (as for example transformation into a new co-ordinate system) constitutes a separate Fortran program. This modularity is very similar to the "philosophy" of the Linux operating system: Keep flexibility and clarity by offering not one program which handles everything (and thereby becomes a "monstrosity"), but offer many flexible and simple tools which the user only has to combine in order to perform complex tasks.

What else except for modularity can be done to improve the readability of computer programs? It is obvious, and probably the most neglected aspect of good computer code generation: in-code documentation. Every small sub-section of a computer program should be documented. It should be possible to gain insight into an algorithm only by looking at the in-code documentation. It is definitely not sufficient only to document for every subroutine what its aim is and what its interface is, although quite often even this most basic principle is not fulfilled. Hence, in the re-coding of the PV inversion focus was led to good documentation: Where are files opened, where variables initialised, what is a subroutine call meaning?

Finally a remark concerning the used programming language. A complete re-coding of a software package might also be a chance to switch to a "better" programming language. In the present case, the original code was written in Fortran 77, and this language was also used for the re-coding. Fortran, and in particular its older version Fortran 77, is by far not a modern language. Comparing its vocabulary and its structural elements to languages like C++ or Java, its power is very limited. Moreover, Fortran includes jump statements like *Goto* which could easily make any code un-readable. So why do so many research codes still rely on Fortran? In fact, most numerical weather prediction models are written in Fortran. There are two reasons why the PV inversion was written in Fortran: Firstly, the small vocabulary and the intuitive naming of the inherent Fortran commands makes it quite easy to read a code. Many researches never had a profound introduction into programming. Nevertheless they should be able to implement algorithms for their daily work. This is easily done with Fortran, and certainly much easier done than using object-oriented concepts offered by C++ and other modern languages. The PV inversion should be available to a research community which has excellent algorithmic thinking, but only a ”weak” background in computer languages. Fortran is an optimal choice in this respect. The second reason why the re-coding was done in Fortran is its high performance. In fact, PV inversion is very resource demanding, and compilers must create fast-running codes in order the PV inversion tool to be helpful.

Regarding the ”unholy” *Goto* statements, it can be immediately said that good programming style is not restricted to languages which do not offer a *Goto* statement. A well-structured and carefully developed Fortran program is certainly preferable to a ”bad” algorithm in a ”good” computer language, ”good” meaning that this language offers many controlling structures. Probably, a key concept of well-written computer code is a good balance between the ”power” of a (Fortran) subroutine and its length (expressed in the number of code lines). Two extreme examples might illustrate this point: If no subroutines at all are used, i.e. if the program consists only in one single main program, the algorithm might get ”lost” in two many ”technical” details. On the other hand, if every single and minor step is a separate subroutine, the code lacks clarity due to the large number of subroutines. In the present re-coding of the PV inversion, focus was given to a good balance of functionality (power) of a subroutine and its length.

3.2 Universality and Model Independence

Universality was another aspect which was important in re-coding the PV inversion. In the present example universality might be better named ”model independence”. Model in this respect refers either to different numerical weather prediction (NWP) models or to idealised experiments. In the former case, the variety of NWP models is very large. The PV inversion tool was originally written for the Europa model (EM) of the German weather service (DWD). Later it was applied to the higher-resolution model (HRM and CHRM) which was run by the DWD and the Swiss weather service (MeteoSwiss). In recent years, the non-hydrostatic local model (LM) developed by the DWD replaced the HRM in regional weather forecasting. So, an adaption of the existing PV inversion code would have been necessary. Moreover, PV inversion is particularly attractive for global-scale data sets. So, the method and the programs were adapted to conform with the global model of the European Centre for Medium Range Forecasts (ECMWF). To make things worse, idealised experiments should also be performed. What actually makes the situation so difficult is that each model has its own horizontal grid structure and its own vertical grid structure. For instance, ECMWF uses a terrain-following co-ordinate system which in higher levels becomes a pressure-level co-ordinate system, quite in contrast what is used by the regional LM model (a variant of geometrical height).

Model independence cannot completely avoided. It is a fact which has to be adopted and cleverly dealt with. In the re-coding, model aspects were removed in the first two preparatory steps (out of eight different steps, see section 4.4) and the way back to the model is done in the last post-processing steps. Essentially the preparatory steps interpolate the needed meteorological fields onto a stack of height levels and transforms the fields into a local quasi-cartesian co-ordinate system. After completing these steps, the other preparatory steps and the PV inversion itself is completely independent of the model from which the input data is retrieved. As a particular example consider the PV inversion of a structure over the North Pole. In the existing code such an inversion would not have been feasible due to the convergence of the longitude circles at the North Pole. The new code, on the other hand, elegantly circumvents this problem by introducing a new quasi-cartesian co-ordinate system centered over the North Pole. In this respect the North Pole is not different from any other region on the globe.

With respect to idealised experiments, the following strategy was adopted. This kind of experiments is generally so different from what is needed for real-case studies that a complete separation is preferable. So, in contrast to the existing code, this type of experiments is now handled by a completely separated program package. Of course, many programs "overlap" or are even identical, but it nevertheless makes sense to treat the two kind of experiments (real-case versus idealised) separately.

3.3 User Friendliness

Computer programs are not written for a computer's joy, but because they should solve a problem which cannot be solved analytically by any person. So, computer programs are tools which should make "life" easier for their users, or at least they should make some problems tractable. If the user-interface of a computer program is chaotic and badly structured, the computer does not complain. The user, on the other hand, might give up using the tool simply because it needs too much effort to understand how the program must be handled. Scientific programs are especially in danger of having bad user-interfaces because these kind of programs are always "in the flow". This in turn means that often it is not worthwhile to develop a graphical user-interface, as it is an absolute must for commercial software.

In the existing code for the PV inversion the user-interface was highly "chaotic". It consisted of several hundred lines of Linux Shell scripts which were difficult to read -partly due to lack of documentation, partly due to the unnecessarily complex structure. The behaviour of the inversion was controlled by many variables set in the beginning of the Linux Shell script, set without knowing exactly where and when these variables are used. This gave the user a "bad feeling" about his experiments, simply because it was not always clear what changing parameters really meant.

Although a graphical user-interface is not provided for the new version, an attractive command-based interface is now available. This interface essentially consists of two parts. Firstly, a Linux Shell script is provided which handles all calls to the computing Fortran programs. For instance, *inversion.sh prep* runs through all preparatory steps. User friendliness is reached by the fact the the user needs to make no changes to this Linux Shell script, although its limited size and documentation would make it quite simple. Probably the best user-interface is provided by a parameter file. Such a file is provided in the new version, and all parameters describing the inversion problem can be entered into this file. In fact, this kind of user-interface was motivated by sophisticated numerical weather prediction (NWP) models. There too, the specifications characterising a model run are passed to the NWP model by means of a parameter file.

What else made the existing user-interface chaotic? A close inspection revealed that the flow of data and information was absolutely non-transparent. Meteorological fields were moved from one file to another, only to be renamed there and subsequently being moved again to another file. For a part-time user it would have been impossible to keep track of the flow of information. In the new code the flow of information is very strictly controlled (see also the above subsection 3.1). In fact, there are many different files involved in the PV inversion: input files, several intermediate files and finally the output files. It is not the number of files which makes a process difficult to understand. If the flow of information is well controlled, it remains tractable. For the case of the PV inversion, this information flow can be kept clear, and this was indeed an important aspect of the re-coding.

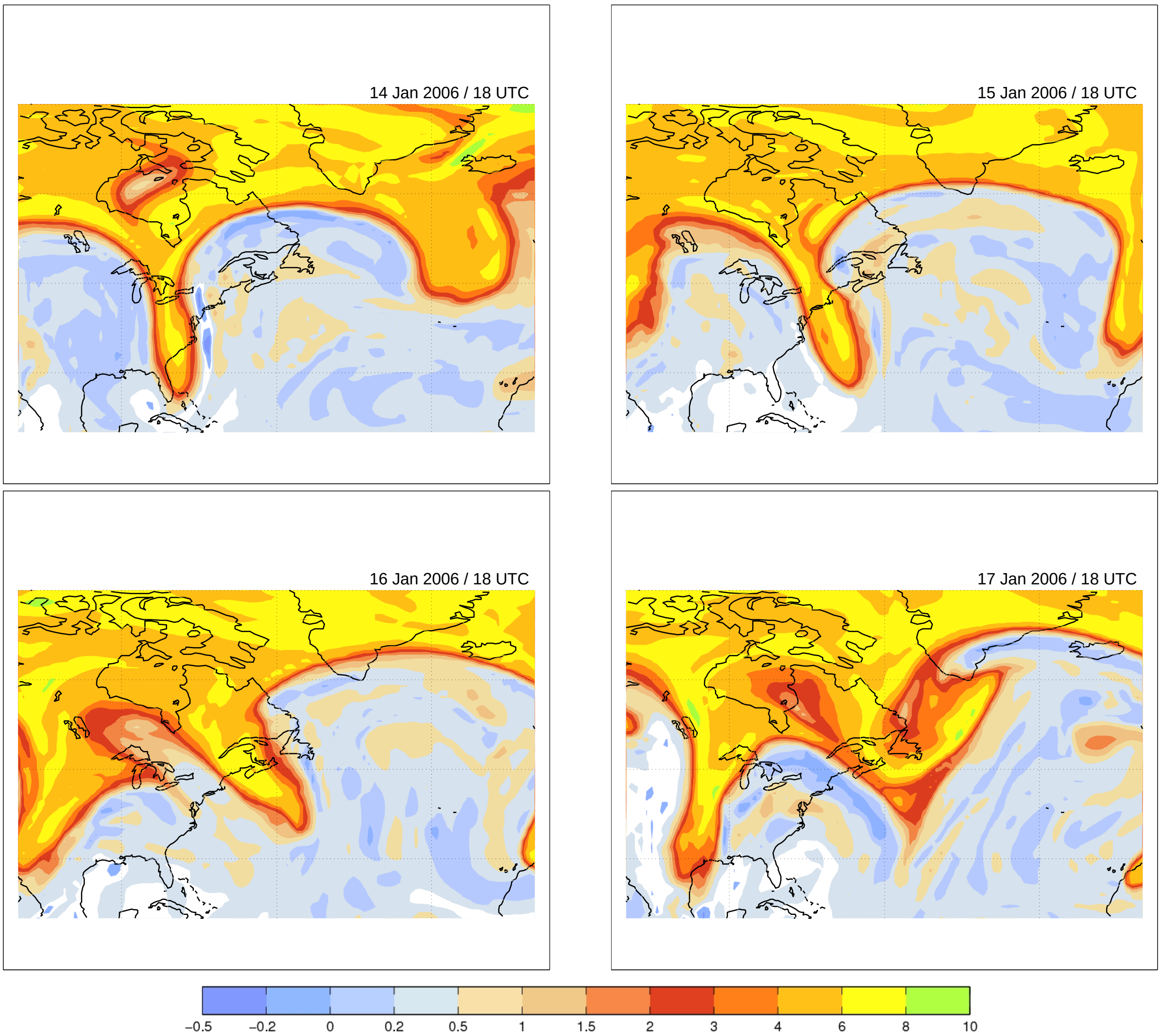


Figure 5: *Time evolution of Ertel's potential vorticity (PV) on the 315 K isentropic surface. The four plots are for 14, 15, 16 and 17 January 2006, 18 UTC. The PV inversion will be exemplified in this user guide for 16 January 2006, 18 UTC.*

4 Inversion of an Extra-tropical PV Streamer

4.1 Scientific question

The PV inversion starts with the selection of a distinct PV anomaly which should be removed or added to the original PV distribution. It is then the aim to analyse and compare the meteorological fields of horizontal wind and potential temperature associated with the modified PV distribution with the corresponding fields associated with the original PV distribution.

The time evolution of potential vorticity (PV) on the 315 K isentropic surface shows a distinct feature over the Eastern United States (Fig. 5). This so-called stratospheric PV streamer is indicative for a deep intrusion of stratospheric air towards the equator. In the course of the four days shown, the PV streamer moves towards the east, and in the later stages exhibits an cyclonic rolling-up. An eminent question is how this stratospheric

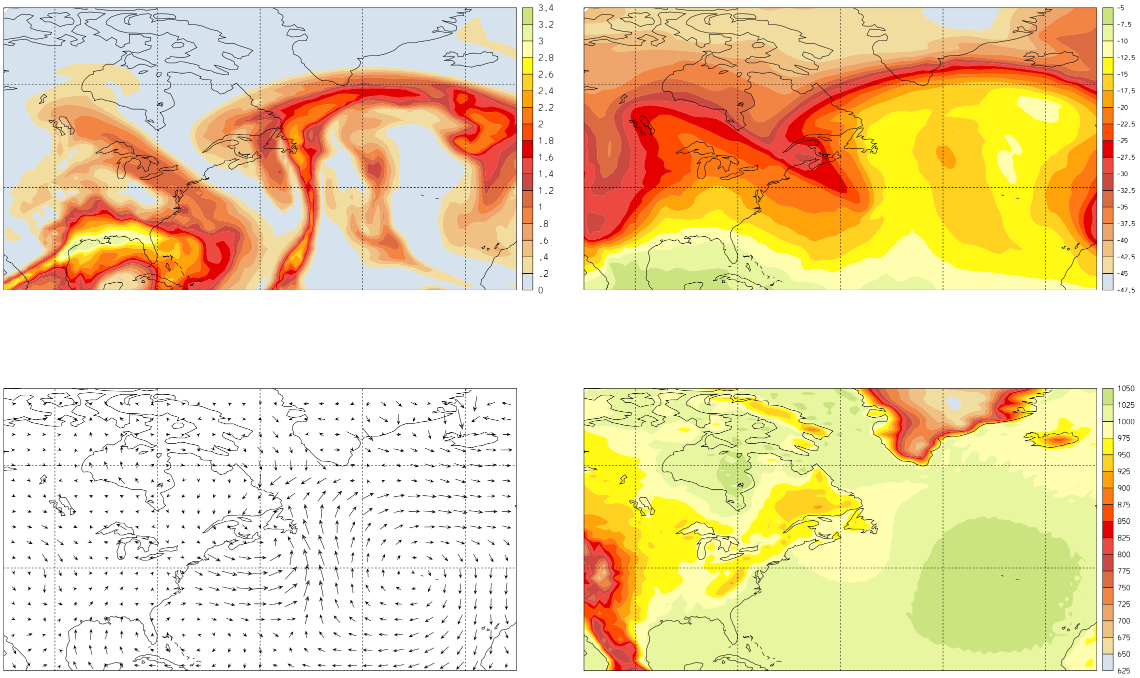


Figure 6: *Specific humidity (g/kg), temperature (in °C), wind vectors at 500 hPa and surface pressure for 16 January 2006, 18 UTC. These field are needed as input for the PV inversion, and are available on the so-called P file.*

PV streamer influences the atmospheric flow in its neighbourhood. From theoretical considerations, it can be expected that the impact on the horizontal velocity and on the static stability is quite large (see introduction, Fig. 2). By means of the so-called quasi-geostrophic omega equation (Holton, 1992), it is also expected that pronounced vertical motions are directly associated with the passage of the PV streamer. It is the aim of this study -and of PV inversion in general- to surmount the qualitative argumentation, and to quantitatively assess the impact of such a structure on the atmospheric flow.

In the following chapters, all steps are discussed which are needed to quantify this impact. Here, we exemplarily consider the date of 16 January 2006, 18 UTC, and determine explicitly for this date the atmospheric response the upper-level (near tropopause level) PV structure.

4.2 Necessary input files

Several preparatory steps are necessary before the PV inversion itself can be performed. Here each of these steps is described in detail and -where possible- illustrated with suitable figures. Starting point of the analysis is the P and Z file of the ECMWF (re-)analysis:

- **P20060116_18** with temperature T (in °C), horizontal wind components U,V (in m/s) in zonal and meridional direction, respectively, specific humidity Q (in kg/kg) and surface pressure PS (in hPa).

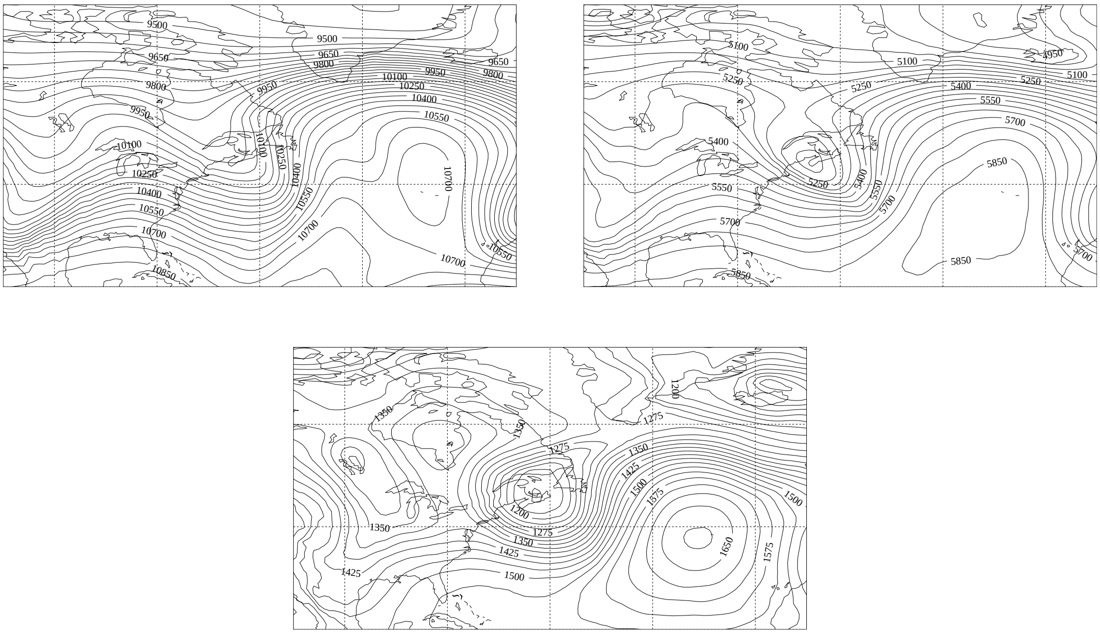


Figure 7: *Geopotential height (in m) on 250, 500 and 850 hPa for 16 January 2006, 18 UTC. These field are needed as input for the PV inversion, and are available on the Z file.*

- **Z20060116_18** with geopotential height Z (in m) on a stack of pressure levels (for instance on 850, 500 and 250 hPa).

Figure 6 shows temperature T, specific humidity Q and wind vectors (U,V) at 500 hPa and the surface pressure for 16 January 2006, 18 UTC. A narrow band of high specific humidity is discernible on the downstream side (to the east) of the PV streamer (upper-left panel). Most probably, this extended band is directly associated with the southerly winds which prevail to the east of the PV streamer, and which are able to transport moist air from the warm subtropics to the north. In the temperature field (upper-right panel), the PV streamer's impact is also discernible. It is associated -at this level- by a local warm anomaly. On theoretical grounds, we would expect a local cold anomaly below the streamer and a local warm anomaly within and above the streamer (see introduction, Fig. 2). The present signal indicates that the streamer reaches far down into the troposphere, and enforces a local warm anomaly at the low level of 500 hPa. As already mentioned before, and explicitly shown by the wind vectors (lower-left panel), the PV streamer is also associated with a cyclonic flow. Besides the impact on the temperature field, this PV induced flow field is the most eminent impact of a PV streamer. Indeed, it is the wind field where the far-field effect of a PV streamer becomes most evident, because the streamer's impact on temperature and stratification is essentially confined to the regions just below or above. Finally, the P file contains the surface pressure PS (lower-right panel), which of course to the largest extent simply reflects the surface topography. So for instance, the high topography of the Rocky Mountains or of the Greenland ice shield, is associated with low surface pressure, 700 hPa corresponding to about 3 km height above

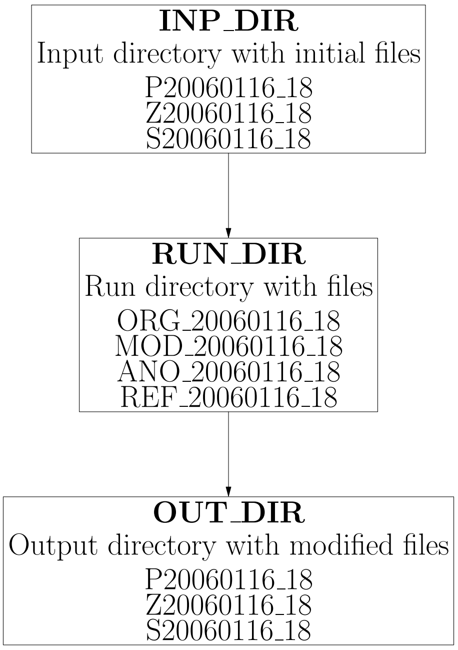


Figure 8: *Data flow diagram for the three steps of the PV inversion. The preparatory steps are essentially performed in the input directory, the inversion in the run directory and the output is finally written to the output directory.*

sea level.

In addition to the P file, a so-called Z file must be provided as input for the PV inversion. This file contains the geopotential height on a distinct set of pressure levels. Typically, these levels are 850, 500 and 250 hPa. A plot of these levels is shown in Fig. 7. The need for this reference levels will become evident in a subsequent section. In short: They are needed to integrate the hydrostatic equation and thereby to transform from pressure to geometrical height as vertical co-ordinate. Note how the PV streamer over the west Atlantic is discernible in the geopotential as a deep trough.

4.3 An overview of the inversion

The PV inversion can be split into three well-separated steps. The first step is associated with preparations, in particular with the definition of the modified Ertel-PV field. In addition, some preparatory steps have to be performed in order to adapt the ECMWF grid to the cartesian grid which is assumed in the inversion algorithm. In the second step the atmospheric state is iteratively adjusted to the modified PV field, as it was specified in the previous preparatory step. This second part comprises the "core" of the algorithm, i.e. the numerical inversion of the quasi-geostrophic potential vorticity equation by means of a successive over-relaxation (SOR) technique. Finally, in the third step the modified atmospheric state is brought back from the cartesian grid of the inversion to the original latitude/longitude grid of the ECMWF, including its hybrid vertical co-ordinate. After this step a direct comparison between input and output fields is feasible due to the equivalent grid structure of input and output fields. The basic three steps are shown in Fig. 8, and will subsequently be described in greater detail in the following sections. Before doing so, a note has to be made on the user-interface for the PV inversion.

The inversion is controlled by one master Linux Shell script (inversion.sh), which calls the performing Fortran programs (for a complete listing of the Linux Shell script consider Appendix 9.2). A major aim of this work was to make the application of the inversion as easy as possible. Therefore, the typical user is not forced to study the master script itself, but can provide all needed parameters by means of a so-called "parameter file" (inversion.param). The contents of this file is parsed by a Perl script, which extracts all needed information. In the description of the subsequent steps, the relevant parts of this parameter file will be reproduced and described. A complete listing of the parameter file can be found in Appendix 9.3.

Nevertheless, it will be necessary for an advanced usage of the inversion package to go into some greater details. For instance, if particular PV structures should be removed or added, the implemented simple filtering approach (to be discussed in section 4.4.4) might not be sufficient. In this case, the advanced user is encouraged to modify the Fortran code in order to fulfill his particular tasks.

In a first step some directories must be specified. This is done in the parameter file *inversion.param* in the section DATA. For the example of this study this section looks like:

```
BEGIN DATA
  DATE    = 20060116_18;
  INP_DIR = /lhome/sprenger/PV_Inversion_Tool/real/inp;
  RUN_DIR = /lhome/sprenger/PV_Inversion_Tool/real/run;
  OUT_DIR = /lhome/sprenger/PV_Inversion_Tool/real/out;
END DATA
```

The first argument (DATE) gives the date of the case. Then the following three entries specify the input (INP_DIR), the run (RUN_DIR) and the output (OUT_DIR) directory, respectively. These directories can be created by the call

```
inversion.sh inst
```

For test reasons a set of sample files is provided. These sample P20060115_18 and Z20060115_18 files can be copied to the input directory with the call

```
inversion.sh sample
```

The fields available on the two files are:

Q	Specific humidity (kg/kg)	P20060116_18
U	Zonal velocity (m/s)	P20060116_18
V	Meridional velocity (m/s)	P20060116_18
T	Temperature ($°C$)	P20060116_18
OMEGA	Vertical velocity (Pa/s)	P20060116_18
PS	Surface pressure (hPa)	P20060116_18
Z	Geopotential height (m) on 850, 500 and 250 hPa	Z20060115_18

With these preparatory steps, the inversion can be performed. The following sections discuss the three basic step, i.e. the preparation of the input files for the inversion, the inversion itself and the post processing of the output files.

4.4 Pre-processing: Defining the inversion problem [prep]

Firstly, the vertical co-ordinate has to be changed from pressure to geometrical height (section 4.4.1) and the fields have to be transformed from the geographical latitude/longitude grid to a cartesian grid (section 4.4.2). Then additional meteorological fields have to be computed on this new cartesian grid (section 4.4.3). With these two preparations the Ertel-PV anomaly can be defined (section 4.4.4), and the input files for the quasi-geostrophic PV inversion be written (section 4.4.5). Some additional steps are then: definition of a reference profile (in section 4.4.6) and the transformation of the coastlines into the cartesian grid (section 4.4.6).

These single steps have to be done once for every inversion problem. A complete flowchart of the preparatory steps is shown in Fig. 9. In the following description every single step will be launched and discussed individually. If desired, the call

```
inversion.sh prep
```

will run through all steps without waiting for intermediate check. This is particularly practical if all settings remain essentially unchanged, and therefore a detailed check after each step is not necessary.

4.4.1 Transformation to height levels [prep1]

The input data are available on a latitude/longitude grid in the horizontal and on a so-called hybrid σ-grid on the vertical. Whereas the former is well known, the latter needs some explanation. The basic idea is illustrated in Fig. 10. The model levels are terrain-following, and the pressure p(i,j,k) at a specific grid point (with grid indices i,j,k) is given by the expression:

$$p(i,j,k) = ak(k) + bk(k) * ps(i,j)$$

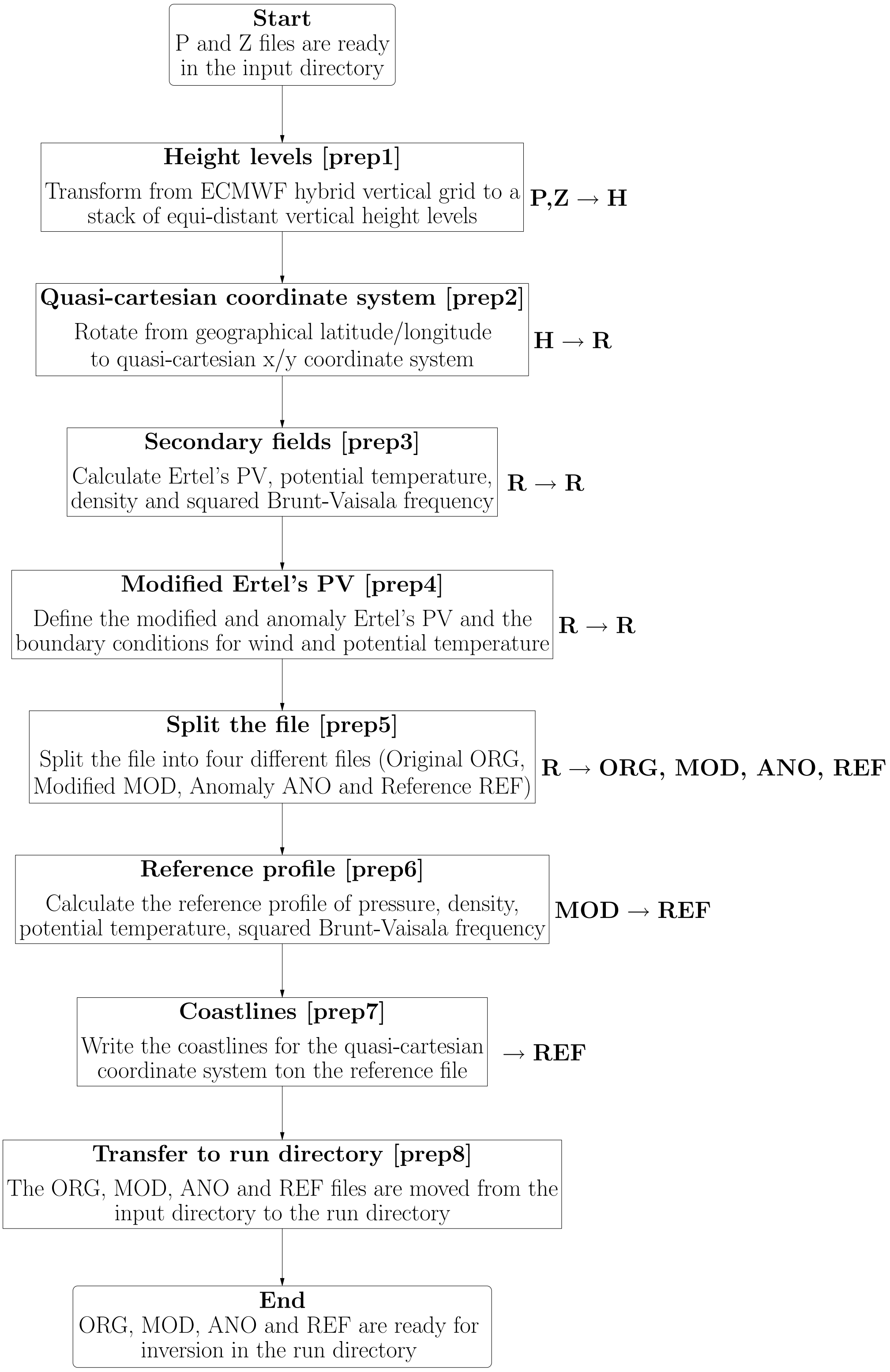

Start
P and Z files are ready in the input directory
Height levels [prep1]
Transform from ECMWF hybrid vertical grid to a stack of equi-distant vertical height levels
P,Z → H
Quasi-cartesian coordinate system [prep2]
Rotate from geographical latitude/longitude to quasi-cartesian x/y coordinate system
H → R
Secondary fields [prep3]
Calculate Ertel's PV, potential temperature, density and squared Brunt-Vaisala frequency
R → R
Modified Ertel's PV [prep4]
Define the modified and anomaly Ertel's PV and the boundary conditions for wind and potential temperature
R → R
Split the file [prep5]
Split the file into four different files (Original ORG, Modified MOD, Anomaly ANO and Reference REF)
R → ORG, MOD, ANO, REF
Reference profile [prep6]
Calculate the reference profile of pressure, density, potential temperature, squared Brunt-Vaisala frequency
MOD → REF
Coastlines [prep7]
Write the coastlines for the quasi-cartesian coordinate system ton the reference file
→ REF
Transfer to run directory [prep8]
The ORG, MOD, ANO and REF files are moved from the input directory to the run directory
End
ORG, MOD, ANO and REF are ready for inversion in the run directory

Figure 9: *Flowchart of preparatory steps. Additionally, to the left of the single steps the essential flow of information is shown. For instance, $H \longrightarrow R$ means that the input is taken from the file with prefix H and the output is then written to the file with prefix R.*

where $ps(i,j)$ is the pressure at the surface at the specified grid position. The two one-dimensional arrays ak and bk are part of the P file, more specifically of the constants file associated with the P file. The main advantage of this hybrid grid is its non-intersection with the ground, which facilitates many numerical calculations.

The ECMWF fields are easily plotted on pressure surfaces since the vertical co-ordinate

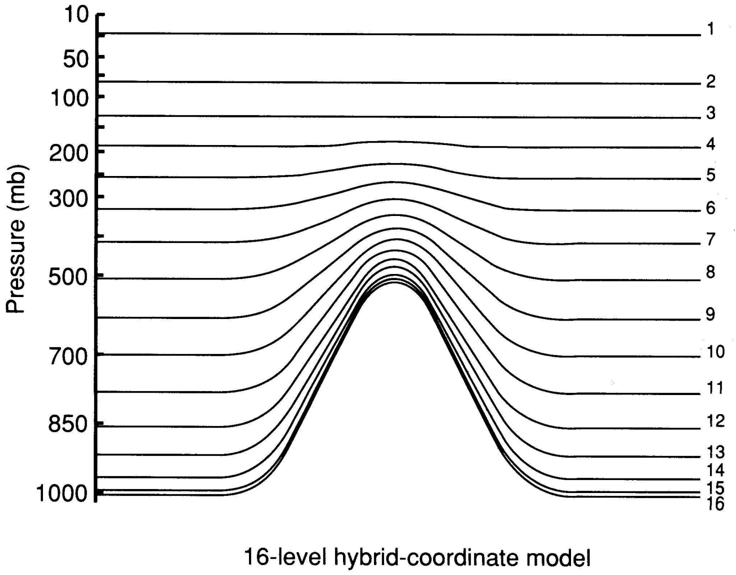


Figure 10: *Illustration of the terrain-following hybrid co-ordinate system of the ECMWF grid. Here it is shown with 15 levels, recent versions of the ECMWF model contain 60 hybrid levels for the ERA-40 re-analysis and 91 levels for the operational analysis and deterministic 10 day forecast [illustration taken from "An Introduction to dynamic meteorology" by Holton, 1992].*

of the underlying data set is essentially based upon pressure. On the other hand, the inversion program assumes a cartesian grid with geometrical height (in m) as vertical co-ordinate. Therefore, all needed fields from the original P file must be interpolated onto a stack of height levels. This transformation is easily done by means of the hydrostatic equation:

$$\frac{\partial p}{\partial z} = -\rho \cdot g$$

where ρ denotes the density of moist air and g is the Earth's gravity. Note that this equation can be re-formulated to one including temperature if the ideal gas equation

$$p = \rho \cdot R_d \cdot T_v$$

is used and ρ replaced. Here R_d is the ideal gas constant of dry air and T_v is the virtual temperature, i.e. the temperature corrected for the water vapour contents of the air. The definition of virtual temperature takes into account the specific humidity of an air parcel and its (sensible) temperature (see for instance Wallace and Hobbs, 1977):

$$T_v = T \cdot \{1 - \frac{q}{\epsilon} \cdot (1 - \epsilon)\}^{-1} \approx T \cdot (1 + \epsilon \cdot q)$$

where T is the temperature (in K), q the specific humidity (in kg/kg), $\epsilon = R_d/R_v$ is the ratio of the gas constant for dry (R_d) air and water vapour (R_v).

With the previous definitions, the hydrostatic equation can be integrated vertically to get the height as a function of the pressure p:

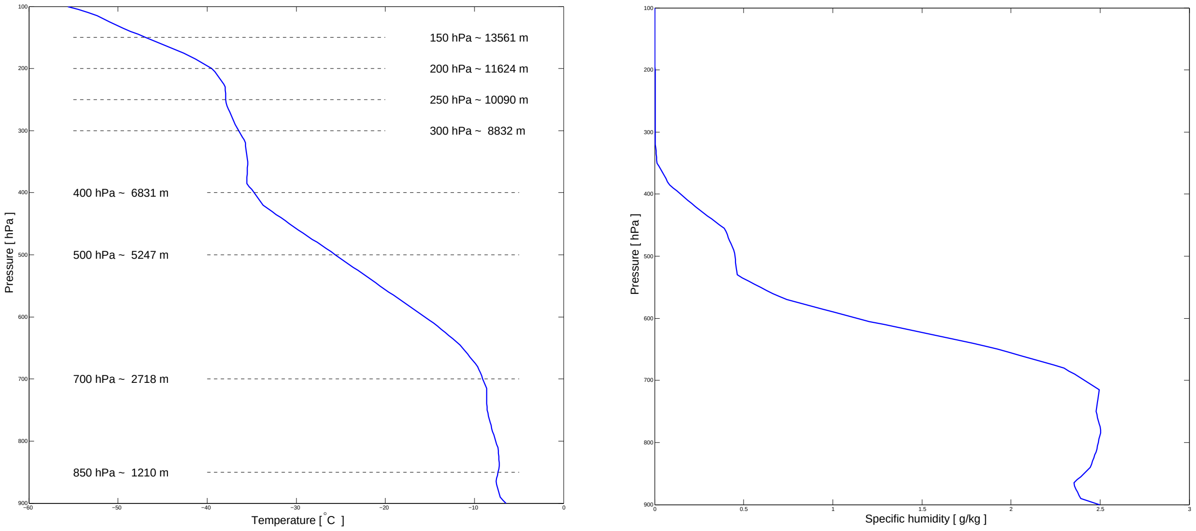


Figure 11: *Left: Vertical profile of temperature (in °C) in dependence of pressure at 70° W and 40° for 15 January 2006, 18 UTC. Additionally the reference pressure levels are included with the corresponding geopotential height (in m). Right: Vertical profile of specific humidity (in g/kg).*

$$z(p) = z_{ref} - \frac{R_d}{g} \cdot \int_{\log(p_{ref})}^{\log(p)} T_v \cdot d\log(p)$$

where z_{ref} and p_{ref} denote the reference level. If several reference levels are given (for example at 850, 500 and 250 hPa, see Fig. 7) the integration can be performed from the nearest reference level to the specified pressure level. Note that at least one reference level must be given, i.e. at one pressure level the geometrical height must be known. Typically, this correspondence is known for the 500 hPa level (see Fig. 11). As a result of the integration, the geometrical height is given as a function of the pressure at all points of the model grid.

In addition to the geometrical height at each grid point, the hydrostatic equation can be integrated downward to obtain a consistent estimate of topography height. In fact, if the surface pressure p_s is given, the height of the topography is readily determined by

$$z(p_s) = z_{ref} - \frac{R_d}{g} \cdot \int_{\log(p_{ref})}^{\log(p_s)} T_v \cdot d\log(p)$$

The advantage of the thus determined topography (instead of using the ECMWF topography) is its consistency.

Having determined the geometrical height $z(i,j,k)$ at each grid point i,j,k in addition to the already known pressure $p(i,j,k)$, it is straightforward to interpolate a field (temperature, velocity,...) onto an arbitrary height level. The method adopted here is to lie a natural cubic spline through the grid points along a vertical profile and then to evaluate the spline at the pre-specified height levels. In detail: The cubic spline at a horizontal grid position i, j is based upon the values $[x_k = z(i, j, k), y_k = T(i, j, k)]$ if temperature T is to be interpolated onto a stack of height levels. Having determined the associated cubic spline T_{sp}, temperature at an arbitrary height z can be calculated by evaluating the cubic spline at this level: $T_{sp}(z)$. The algorithm for the cubic spline is taken from Press et al. (1992).

The interpolation onto height levels is done with the call

```
inversion.sh prep1
```

which numerically integrates the hydrostatic equation and therefore attributes to each grid point not only its pressure (in hPa) but also its geometrical height (in m). The relevant numerical parameters are taken from the parameter file *inversion.param*, where the following section is essential:

```
BEGIN GRID
  GEO_ZMIN =    0.;
  GEO_NZ   =  125 ;
  GEO_DZ   =  200.;
END GRID
```

The three parameters describe the new vertical co-ordinate. The lowest levels is assumed to be at ground (GEO_ZMIN), the number of vertical levels is 125 (GEO_NZ) and the vertical resolution is equidistantly 200 m (GEO_DZ). Hence, the new vertical grid spans the range from ground up to 25 km.

The output file H20060115_18 is a new netcdf file which has now geometrical height as vertical co-ordinate and the variables (U,V,T,Q) interpolated onto the height levels from surface up to 25 km. Note that the height of the topography (ORO) is also determined. If a model level is below the topography, all field values are set to missing data. The following table gives all the variables which are available on the new H file:

Q	Specific humidity (kg/kg)	H20060116_18
U	Zonal velocity (m/s)	H20060116_18
V	Meridional velocity (m/s)	H20060116_18
T	Temperature $(°C)$	H20060116_18
P	Pressure (hPa)	H20060116_18
ORO	Height of topography (m)	H20060116_18

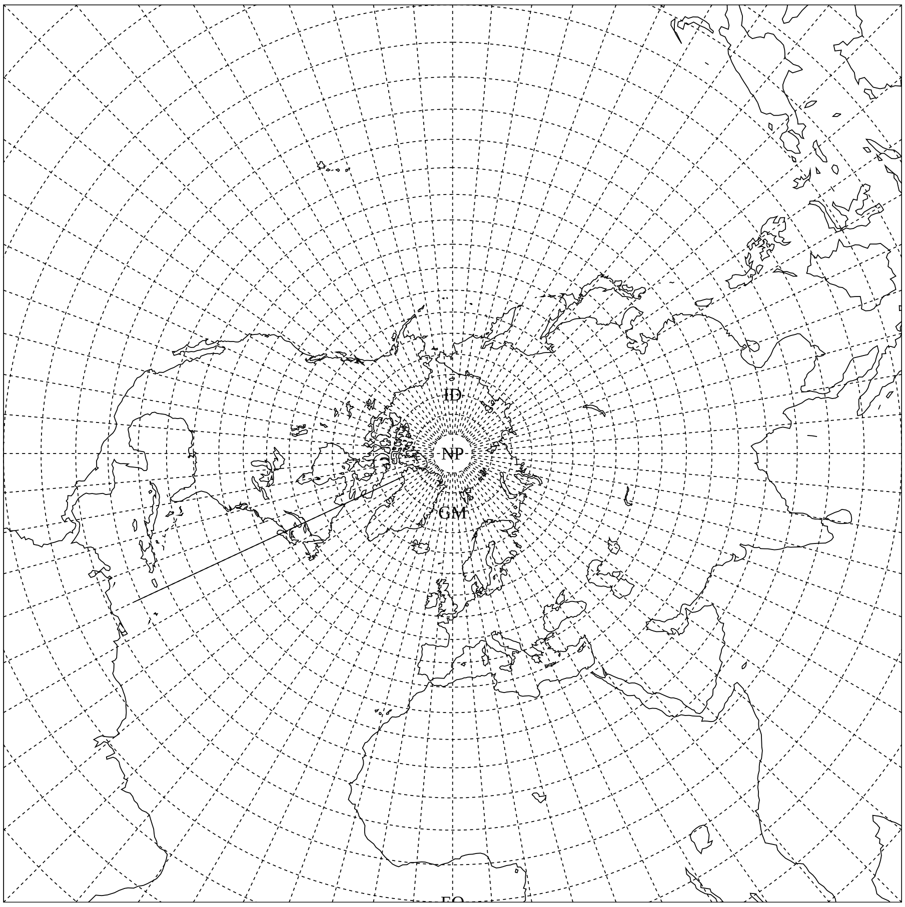


Figure 12: *Latitude/longitude grid of the ECMWF P and Z files. For clarity, the grid spacing was reduced to 5 degrees, instead of the 1 degree in the files. Meteorological fields -as for instance temperature, wind vectors, pressure and geopotential- are given at the intersection of the latitude and longitude circles. Note how this grid becomes singular toward the north pole.*

4.4.2 Rotating to a quasi-cartesian co-ordinate system [prep2]

The input fields are given on a latitude/longitude grid, with -after the previous step- the fields on an equi-distant grid of height levels. The inversion itself assumes a cartesian grid, hence a uniform grid spacing is also assumed in the horizontal directions. This is clearly not the case for the latitude/longitude grid, as illustrated in Fig. 12. Due to the sphericity of the Earth, the zonal grid spacing decreases with increasing geographical latitude. As a remedy to this singularity, all input fields are transformed to a "quasi-cartesian" grid with its center on the PV structure of interest. The transformation between the two grids is illustrated in Fig. 13. It becomes evident that the grid distortion is significantly reduced by this step. The formula which relates the two grids is based upon the co-ordinate transformation given in the Appendix A (Fortran subroutines *lmstolm* and *phstoph*). The geographical latitude/longitude corresponding to a grid point in the rotated (quasi-cartesian) co-ordinate system is obtained in the following way. Firstly, the rotated latitude/longitude $\phi_{rot}, \lambda_{rot}$ is transformed into

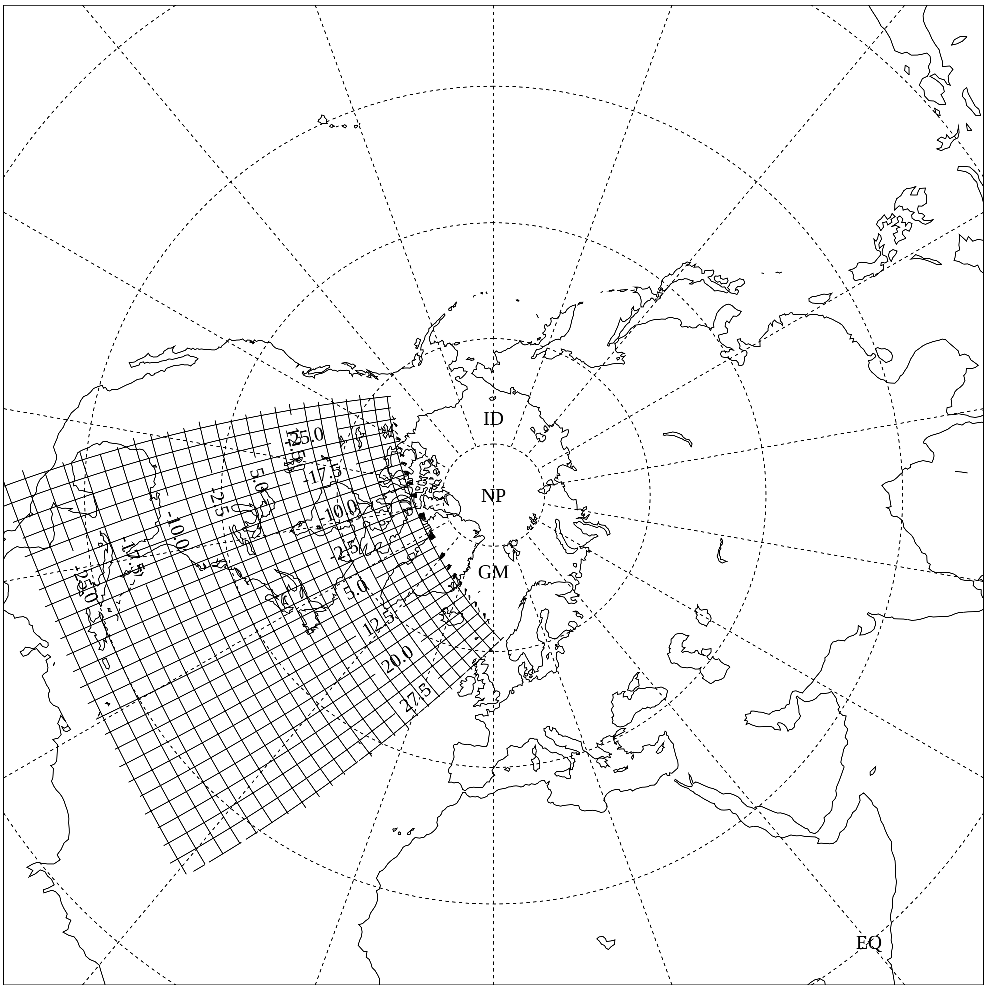


Figure 13: *Local cartesian grid which is centered at the PV streamer's position and which shows much less distortion due to the sphericity of the Earth. The center of the new grid corresponds to the rotated latitude and longitude (0,0). The boundaries are at* -31° *and* $+31^\circ$ *rotated longitude, and at* -31° *and* $+31^\circ$ *rotated latitude.*

$$\begin{aligned}
\lambda'_{rot} &= 90 + lmstolm(\phi_{rot}, \lambda_{rot} - 90, 90 + \alpha, -180) \\
\phi'_{rot} &= phstoph(\phi_{rot}, \lambda_{rot} - 90, 90 + \alpha, -180)
\end{aligned}$$

where α is the rotation angle given in the parameter file (CROT, see below). These two values are then, in a second rotation, transformed into geographical latitude and longitude:

$$\begin{aligned}
\lambda_{geo} &= lmstolm(\phi'_{rot}, \lambda'_{rot}, 90 - \phi_{cen}, \lambda_{cen} - 180) \\
\phi_{geo} &= phstoph(\phi'_{rot}, \lambda'_{rot}, 90 - \phi_{cen}, \lambda_{cen} - 180))
\end{aligned}$$

where the centre of the PV anomaly (the centre of the quasi-cartesian co-ordinate system in geogrpahical latitude/longitude co-ordinates) is given by ϕ_{cen} and λ_{cen} (CLAT and

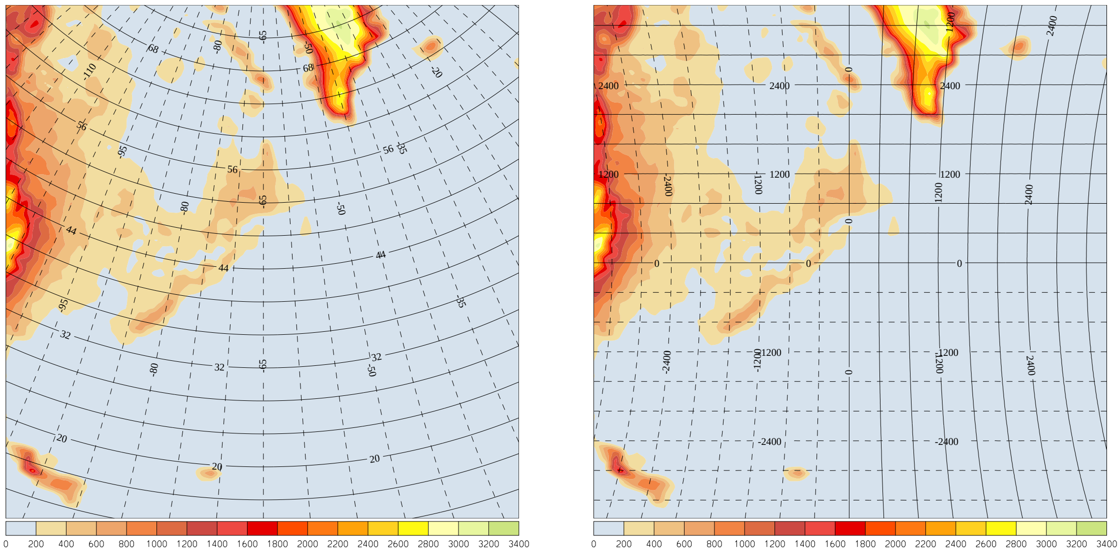


Figure 14: *Quasi-cartesian grid which is centered at the PV streamer's position and which shows much less distortion due to the sphericity of the Earth. On the left panel the latitude and longitude on the Earth surface is shown, on the right panel the x and y co-ordinates in the new co-ordinate system. Note how the distortion is considerably reduced with the new x,y co-ordinates (right), as compared with the original latitude/longitude co-ordinates (left). Additionally, the topography (in m above sea level) is shown in color: Clearly discernible is Greenland and the east coast of North America.*

CLON in the parameter file, see below).

The transformation looks quite complicated, but has an easy geometrical interpretation. In fact, we create a new latitude/longitude grid, but now longitude and latitude are defined not relative to the Earth's north pole, but relative to a rotated imaginary "north pole". Therefore, we call the new latitudes/longitudes rotated co-ordinates. The link between the geographical latitude/longitude and the rotated ones is shown in Fig. 14 for the case study. Here, the left panel gives the latitude and longitude at the grid positions of the quasi-cartesian grid. The right panel gives the quasi-cartesian co-ordinates x and y, which correspond to the grand-circle distance on the Earth surface. While the transformation of a scalar quantity (like for instance temperature, pressure, wind speed) is straightforward with the above formulas, the transformation of a vector field is much more complicated. This is easily illustrated with the following example: Consider a perfectly zonal flow in the geographical grid, i.e. there is only a wind component along the geographical latitude circles, but none along the geographical longitude circles (hence, in the left panel of Fig. 14 the velocity vector is parallel to the latitude circles). In the rotated grid -the quasi-cartesian grid- this pure zonal velocity is transformed into a wind vector which has components both along the rotated longitude and latitude circles (i.e. in Fig. 14 the vector has a component along the horizontal and the vertical axis). In principle, a complicated formula could be derived which gives an explicit expression how the

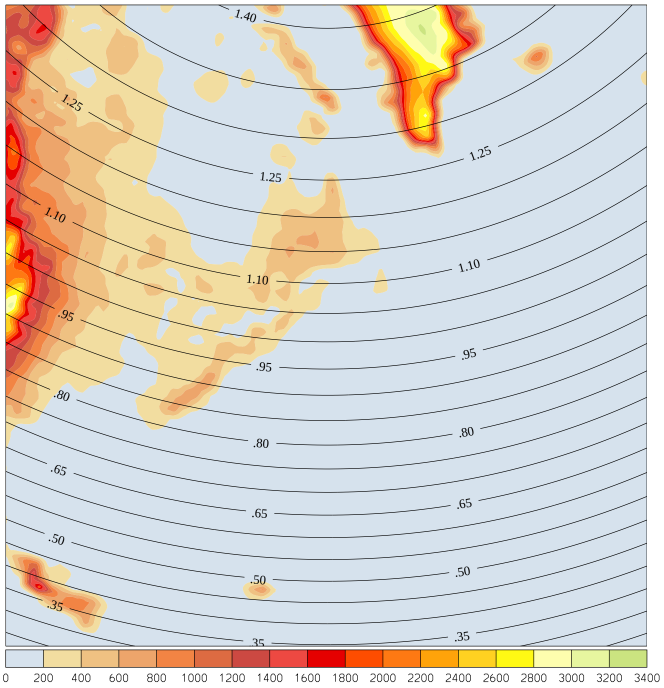


Figure 15: *Coriolis parameter (in $10^{-4}s^{-1}$) in the quasi-cartesian co-ordinate frame. The Coriolis parameter is a local measure for the Earth's rotation. It is constant along latitude circles. Note that f becomes maximal at the pole and vanishes at the equator. No PV inversion can be performed if the cartesian grid crosses the equator and hence crosses the zero isoline of the Coriolis parameter.*

individual velocity components transform. Here, we adopt a more pragmatic approach: We numerically determine the local rotation angle between the two orthogonal co-ordinate systems, and subsequently use these angles to get the transformed velocity components.

Of course, the Coriolis parameter must be kept in the transformation to the quasi-cartesian co-ordinate frame. Tis parameter is given by:

$$f = 2 \cdot \Omega \cdot sin(\phi)$$

where Ω is the angular velocity of the Earth rotation and ϕ is the geographical latitude. From the definition it is clear that the Coriolis parameter is constant along latitude circles. Furthermore, it takes its maximum value at the pole and vanishes at the equator. A typical value for the mid-latitudes is $f = 10^{-4}s^{-1}$. For the case study the Coriolis parameter is shown in Fig. 15.

Numerically, the rotation of the fields to the quasi-cartesian co-ordinate frame is done with the call

```
inversion.sh prep2
```

The program needs some information from the parameter file. The relevant section is:

```
BEGIN GRID
  ROT_NX   =  250 ;
  ROT_NY   =  250 ;
  ROT_DX   =  0.25;
  ROT_DY   =  0.25;
  CLON     =  -65.;
  CLAT     =   45.;
  CROT     =    0.;
END GRID
```

The first two parameters (ROT_NX and ROT_NY) give the number of grid points in the horizontal directions for the rotated co-ordinate system. Here it is assumed that the grid has 250 grid points in x and in y direction. Note that the number of grid points in the z direction needs not to be specified because this information is already available on the file H20060115_18. The new horizontal resolution of the rotated grid is given by the next two values. They give the resolution in x (ROT_DX) and in y (ROT_DY) direction. In the present case these resolutions are 0.25 degrees, which corresponds to approximately 28 km, according to the formula

$$0.25 \cdot \frac{2\pi}{360} \cdot R_E \qquad \text{with the Earth's radius} \quad R_E = 6370\, km$$

Note that the resolution of the rotated grid is higher than the resolution of the input grid, where a value of 1 degree is given. Finally, the last three parameters specify where the new co-ordinate system is centered on the globe. Here, this position is at 65 W (CLON) and 45 N (CLAT), i.e. approximately in the centre of the PV streamer shown in Fig. 5. The last parameter (CROT) allows the new grid to be rotated. At the end of this step, the following fields are available on the R20060115_18 file:

Q	Specific humidity (kg/kg)	R20060116_18
U	(Rotated) Zonal velocity (m/s)	R20060116_18
V	(Rotated) Meridional velocity (m/s)	R20060116_18
T	Temperature ($°C$)	R20060116_18
P	Pressure (hPa)	R20060116_18
ORO	Height of topography (m)	R20060116_18
LAT	Geographical latitude	R20060116_18
LON	Geographical longitude	R20060116_18
CORIOL	Coriolis parameter ($1/s$)	R20060116_18
X	Cartesian distance from centre along x axis (km)	R20060116_18
Y	Cartesian distance from centre along x axis (km)	R20060116_18

4.4.3 Calculating secondary fields [prep3]

Several additional fields are needed for the PV inversion, the most prominent of course being Ertel's PV itself. This field can easily derived from the horizontal wind components, the potential temperature and the density. Firstly, the potential temperature θ is calculated from pressure p (in hPa) and temperature T (in K) according to its definition:

$$\theta = T \cdot (\frac{p_0}{p})^{\kappa}$$

where the reference pressure is fixed as 1000 hPa and $\kappa = R/C_p$ is the ratio of the gas constant for dry air (R) and the specific heat at constant pressure (c_p). Numerically, this value is in good approximation: $\kappa = 0.286$. Note that an inherent property of the atmosphere is its stability. This is expressed physically by a potential temperature which increases with height. Regions where potential temperature decreases with height are hydrostatically unstable and the pre-requisites for a PV inversion are not fulfilled there.

The density ρ (in kg/m^3) of the air is determined by the ideal gas equation for air.

$$\rho = \frac{p}{R_D \cdot T}$$

where R_d is the gas constant for dry air and p and T are the pressure (in Pa) and temperature (in K), respectively.

With this definition of potential temperature and density, Ertel's PV is readily calculated according to:

$$PV = -\frac{1}{\rho} \cdot (\vec{\xi} + f \cdot \hat{k}) \cdot \vec{\nabla}\theta$$

Here, $\hat{k}$ is a unit vector pointing in the vertical direction, f is the planetary vorticity (typically $10^{-4}s^{-1}$ in the extra-tropics), i.e. the vorticity due to the Earth's rotation, and $\vec{\xi}$ is the curl $\vec{\nabla} \times \vec{u}$ of the wind vector $\vec{u}$. In large-scale dynamics vertical wind components can be neglected in the above formula. The final expression for Ertel's PV then reduces to:

$$PV = -\frac{1}{\rho} \cdot [(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + f) \cdot \frac{\partial \theta}{\partial z} + \frac{\partial u}{\partial z} \cdot \frac{\partial \theta}{\partial y} - \frac{\partial v}{\partial z} \cdot \frac{\partial \theta}{\partial x}]$$

In many cases the latter two terms (involving vertical derivatives of horizontal velocity) can also be neglected. Here, we keep these terms in order to have a higher-order approximation to Ertel's PV. Note that the first term on the right side includes the aforementioned vertical derivative of potential temperature. The stability criterion forces this derivative to be positive. The theory of atmospheric turbulence leads to a still stronger constraint. Indeed, the whole PV must be positive (on the northern hemisphere), since otherwise symmetric instability will set in and turbulently adjust the atmosphere to a

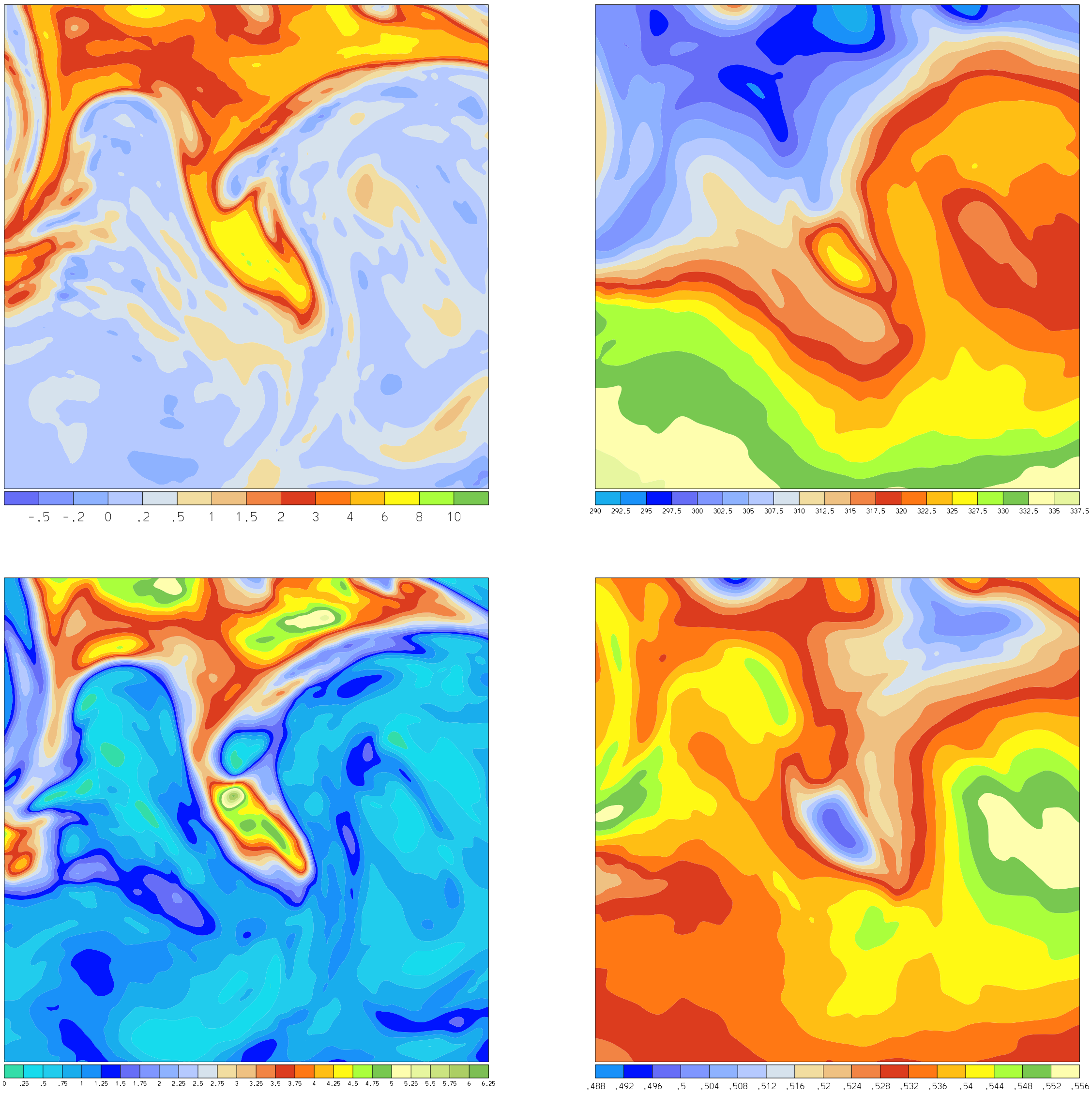


Figure 16: *Secondary fields: Ertel's potential vorticity (upper-left, in pvu), potential temperature (upper-right, in K), squared Brunt-Väisälä frequency (lower-left, in* $10^{-4}s^{-2}$*) and density (lower-right, in* kgm^{-3}*). The stratification is a measure of the vertical stratification of the atmosphere. High values correspond to strong stratification and suppression of vertical motions. Note how the stratospheric PV streamer is associated with enhanced stratification. The density, on the other hand, is slightly decreased within the streamer's region. All fields are plotted on model level 40, corresponding to a geometrical height of approximately 8 km.*

state where PV is non-negative.

The last field which is needed for the PV inversion is strongly related to the vertical derivative of potential temperature. It is the so-called squared Brunt-Väsala frequency (or stratification), defined by

$$N^2 = \frac{g}{\theta} \cdot \frac{\partial \theta}{\partial z}$$

where g is the Earth's gravity. Following the above discussion, a necessary condition for atmospheric stability is that N^2 is non-negative. Note that strong stratifications counteract vertical motions. Particularly strong stratifications are typically found in the stratosphere, ($N^2 \approx 10^{-4}s^{-2}$) whereas the troposphere is considerably weaker stratified ($N^2 \approx 10^{-4}s^{-2}$).

All these secondary fields are calculated with the call to

```
inversion.sh prep3
```

After this step, four additional variables are available on the file R20060116_18. They are listed in the following table and are shown in Fig. 16:

TH	Potential temperature (K)	R20060116_18
PV	Ertel's potential vorticity (pvu)	R20060116_18
NSQ	Squared Brunt-Vaiälä frequency ($1/s^2$)	R20060116_18
RHO	Air density (kg/m^3)	R20060116_18

4.4.4 Identification of a PV anomaly [prep4]

Having calculated Ertel's PV on height levels, it is now time to specify the modified PV distribution which should be obtained by the PV inversion. In short, a PV anomaly has to be identified from the original PV field and then the effect of this anomaly on the potential temperature and velocity has to be determined.

There is no unique and fool-proof method how to specify an anomaly. Indeed, this step strongly depends on the features which should be studied. In the present example, we look at a so-called stratospheric PV streamer and the aim is to remove this streamer from the PV field and thereby to quantify its influence on the stability and wind fields in the middle and lower troposphere. Figure refbox shows the PV in a horizontal cross-section at 8 km height. The PV streamer is clearly discernible as a southward intrusion of high-PV, i.e. stratospheric air toward the south. In the vertical the streamer reaches down to levels of 6 km (not shown).

The extraction of the streamer is performed by filtering the original PV field. This filtering is done by the call

```
inversion.sh prep4
```

Essentially, the program performs the following two steps: Choose a grid point within the filtering box (see Fig refbox), and replace the PV value at this grid point by the zonal mean of all PV values. Formally this might be written as:

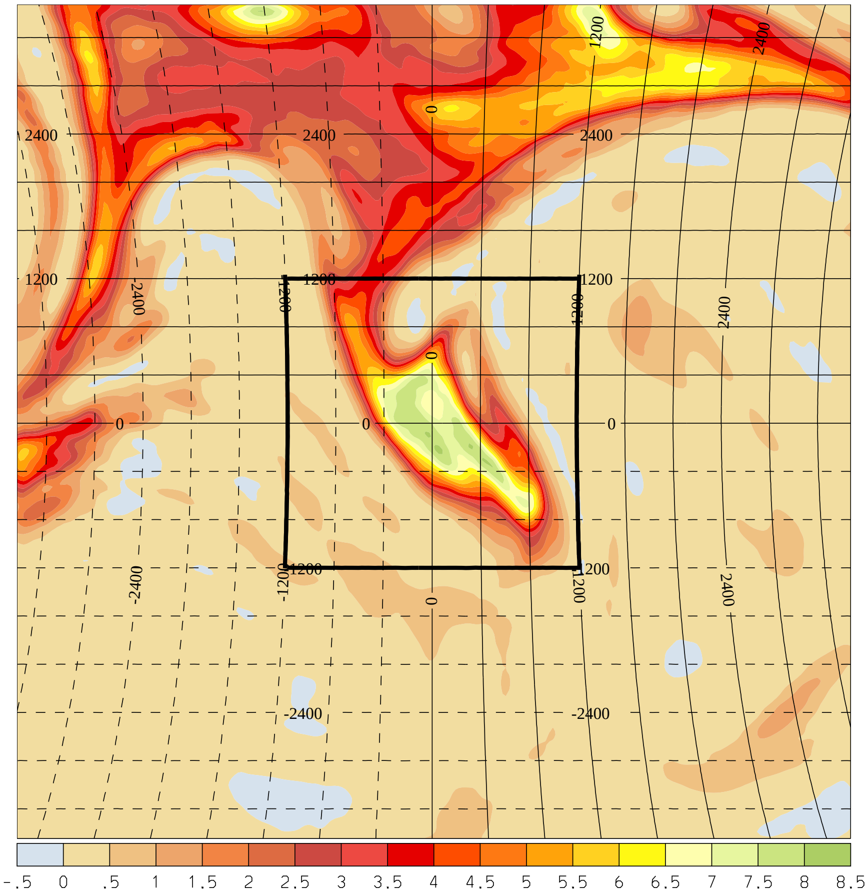


Figure 17: *Ertel's PV on level $k = 40$ (corresponding to a height of 8 km). Additionally the horizontal grid is shown, and the filtering box is marked with the bold black lines.*

$$PV \quad \rightarrow \quad < PV >$$

where $<>$ denotes the filtering operator which applies only within the filtering box. The anomaly could now be determined as the difference of the original and filtered PV: It turns out that this simple difference $AN = PV - < PV >$ would produce not only the desired positive anomaly associated with the streamer, but also surrounding negative PV anomalies. We neglect all these negative anomalies by simply setting them to zero.

$$AN = PV - < PV > \quad \text{and} \quad AN \rightarrow AN^+.$$

The outcome AN^+ is a "well-behaved" PV anomaly, which in turn is used to finally define the modified PV^M, i.e. the PV field which should be obtained by the inversion:

$$PV^M = PV - AN^+$$

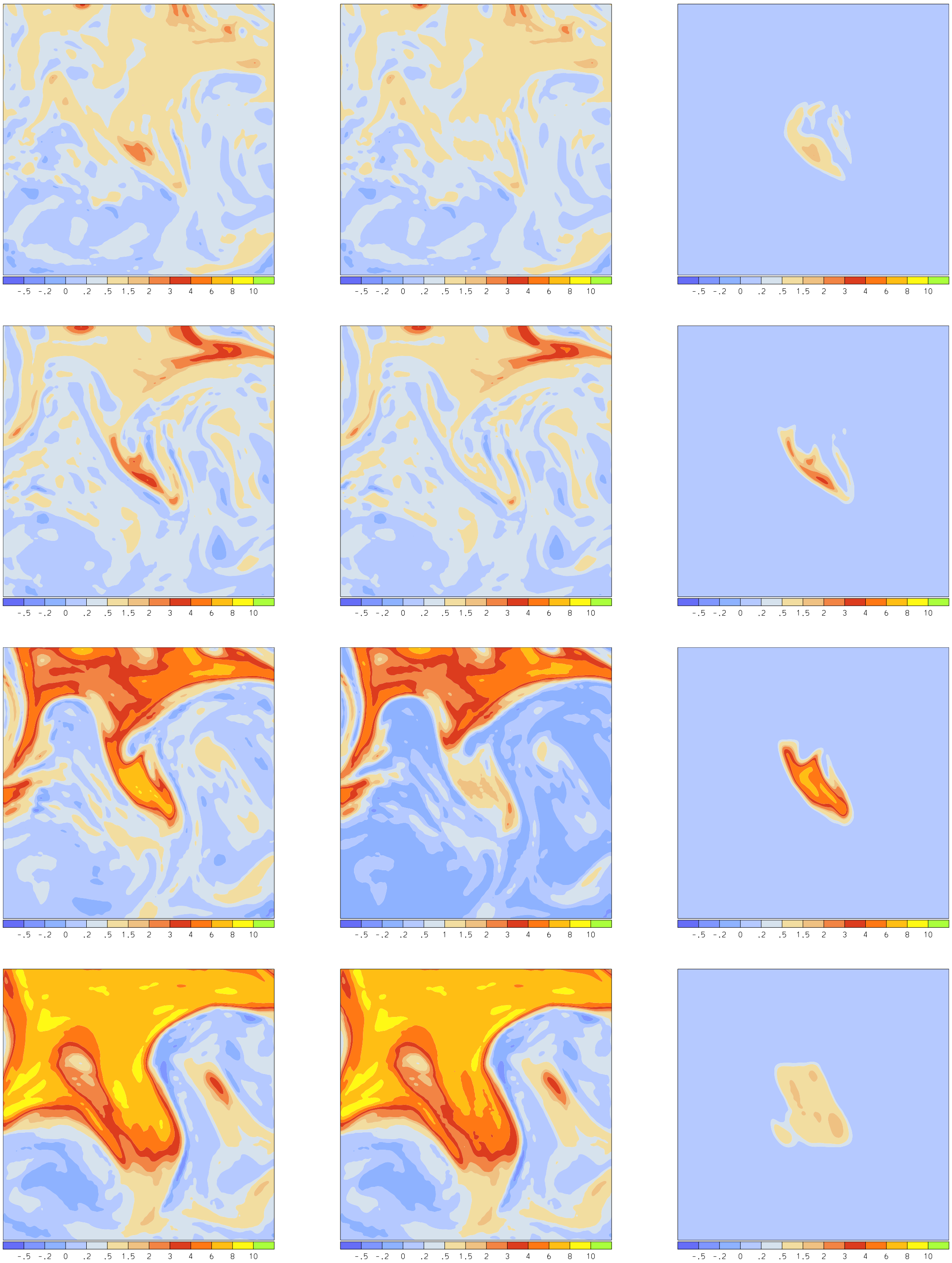


Figure 18: *Original (left), modified (middle) and anomaly (right) Ertel's PV for 16 January 2006, 18 UTC. The different rows correspond to the levels (from top to bottom) 20, 30 40 50 corresponding to the heights 4, 6, 8 and 10 km.*

The original PV, modified PV^M and the anomaly AN^+ is shown in Fig. 18 at several model levels. Note how the PV streamer of Fig. 5 is well captured by the PV anomaly The filtering is characterised by the values in six lines of the parameter file.

```
BEGIN ANOMALY
  BOX_XMIN = -1200.;
  BOX_XMAX =  1200.;
  BOX_YMIN = -1200.;
  BOX_YMAX =  1200.;
  BOX_ZMIN =  2000.;
  BOX_ZMAX = 10000.;
  NFILTER  =     5 ;
  BOUND_XY =   500.;
  BOUND_Z  =   500.;
END ANOMALY
```

The lines specify the filtering box in the west-east direction (from -1200 km to 1200 km, from BOX_XMIN to BOX_XMAX), in the south-north direction (from -1200 km to 1200 km, from BOX_YMIN to BOX_YMAX), and in the vertical (from 2000 m to 10000 m, from BOX_ZMIN to BOX_ZMAX). The filtering is not only applied once, but (as given by NFILTER) $n = 5$ times. Finally the last two parameters describe the handling at the boundaries of the filtering box. In fact, it is advantageous to force the anomaly PV to zero at the boundaries of the box. This is accomplished by multiplying the PV anomaly with a function $\mathrm{f}_{\partial F}$ which is 0 without the box, then monotonically increases to 1 within a boundary zone, and then is constant 1 in the interior of the box. The width of the boundary zone is given as 500 km in the horizontal direction (BOUND_XY) and 500 m in the vertical direction (BOUND_Z).

The modified PV field and the anomaly are written to the R20060116_18 file. Additionally the boundary values are written. The velocity fields must be specified on the lateral boundaries, whereas the potential temperature anomaly must be specified on the lower and upper boundary. In the present settings, all boundary values are zero. The following table lists all fields which are additionally written to the R file.

PV_FILT	Modified Ertel's PV (pvu)	R20060116_18
PV_ANOM	Anomaly in Ertel' PV (pvu)	R20060116_18
UU_ANOM	Lateral boundary condition of zonal wind(m/s)	R20060116_18
VV_ANOM	Lateral boundary condition of meridional wind (m/s)	R20060116_18
TH_ANOM	Upper and lower boundary of potential temperature (K)	R20060116_18

4.4.5 Splitting into different files [prep5]

So far, all new fields are written to the R20060116_18 file. In the further steps it is advantageous to split this file into several files. One file should get all original fields, one only the modified fields and one only anomaly fields. Additionally, some "static" fields, such as orography and Coriolis parameter, should be written to a special file, from here

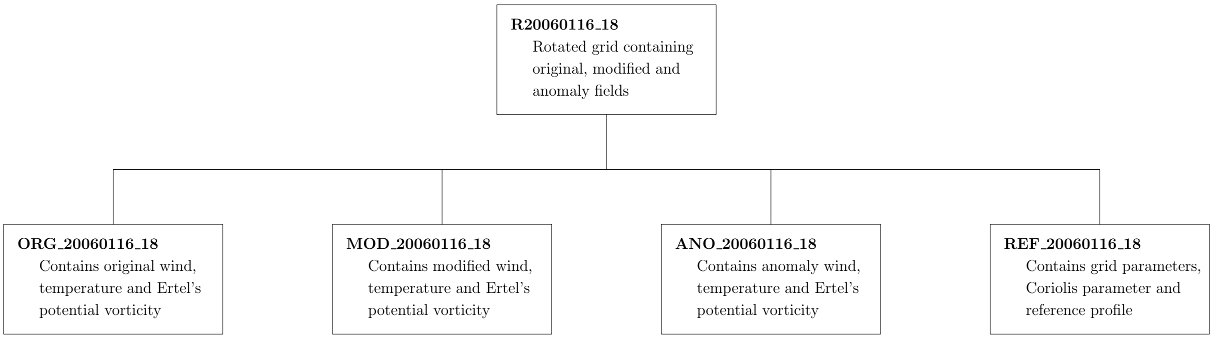

R20060116_18
Rotated grid containing original, modified and anomaly fields
ORG_20060116_18
Contains original wind, temperature and Ertel's potential vorticity
MOD_20060116_18
Contains modified wind, temperature and Ertel's potential vorticity
ANO_20060116_18
Contains anomaly wind, temperature and Ertel's potential vorticity
REF_20060116_18
Contains grid parameters, Coriolis parameter and reference profile

Figure 19: *Splitting of the initial R20060116_18 file into four different files with prefix ORG, MOD, ANO and REF.*

on called the reference file.

The splitting is done with the call

```
inversion.sh prep4
```

The result of this step is a set of new netcdf files with prefix ORG, MOD, ANO and REF. The following table (and Fig. 19) illustrates which variables are written to which file. Note that some variables are renamed in this process.

Old variable name	New variable name	Source file	Destination file
PV	PV	R20060115_18	ORG_20060116_18
U	U	R20060115_18	ORG_20060116_18
V	V	R20060115_18	ORG_20060116_18
TH	TH	R20060115_18	ORG_20060116_18
Q	Q	R20060115_18	ORG_20060116_18
P	P	R20060115_18	ORG_20060116_18
T	T	R20060115_18	ORG_20060116_18
PV_FILT	PV_AIM	R20060115_18	MOD_20060116_18
U	U	R20060115_18	MOD_20060116_18
V	V	R20060115_18	MOD_20060116_18
Q	Q	R20060115_18	MOD_20060116_18
P	P	R20060115_18	MOD_20060116_18
T	T	R20060115_18	MOD_20060116_18
NSQ	NSQ	R20060115_18	MOD_20060116_18
RHO	RHO	R20060115_18	MOD_20060116_18
TH	TH	R20060115_18	MOD_20060116_18

Old variable name	New variable name	Source file	Destination file
PV_ANOM	PV	R20060115_18	ANO_20060115_18
TH_ANOM	TH	R20060115_18	ANO_20060115_18
UU_ANOM	U	R20060115_18	ANO_20060115_18
VV_ANOM	V	R20060115_18	ANO_20060115_18
ORO	ORO	R20060115_18	REF_20060115_18
X	X	R20060115_18	REF_20060115_18
Y	Y	R20060115_18	REF_20060115_18
LAT	LAT	R20060115_18	REF_20060115_18
LON	LON	R20060115_18	REF_20060115_18
CORIOL	CORIOL	R20060115_18	REF_20060115_18

4.4.6 Specifying the reference profile [prep6]

A reference vertical profile must be defined for the PV inversion. The profile should be as near as possible to the real profile. Any quantity is defined in the inversion as a deviation from this profile. Hence if the profile is well chosen only small deviations can be expected and the assumed linearisation (see later) is better. Physically we determine a good reference profile by taking the area mean over the subdomain, i.e at every model level the mean over potential temperature θ, squared Brunt Vaisälä frequency N^2 and density ρ is calculated. The resulting profiles are shown in Fig. 20 as a function of height, and additionally as a function of pressure.

Note how the density of the reference profile exponentially decreases decreases with increasing height. If pressure is used as vertical co-ordinate, the decrease in density is essentially linear with decreasing pressure. The profile of potential temperature increases with increasing height: This is a necessary condition for the convective stability of the air column. In fact, any region where potential temperature decreases with increasing height is unstable and characterised by high levels of turbulence. The stability expressed by this potential temperature increase with height is numerically determined by the so-called (squared) Brunt-Vaisälä frequency, shown in the left panels. Since this field directly goes into the solution of the quasi-geostrophic PV equation (see section 3.5), it must be guaranteed that it always remains positive.

The reference profile is calculated with the call to

```
inversion.sh prep5
```

After his step some additional variables are available on the REF file. These are:

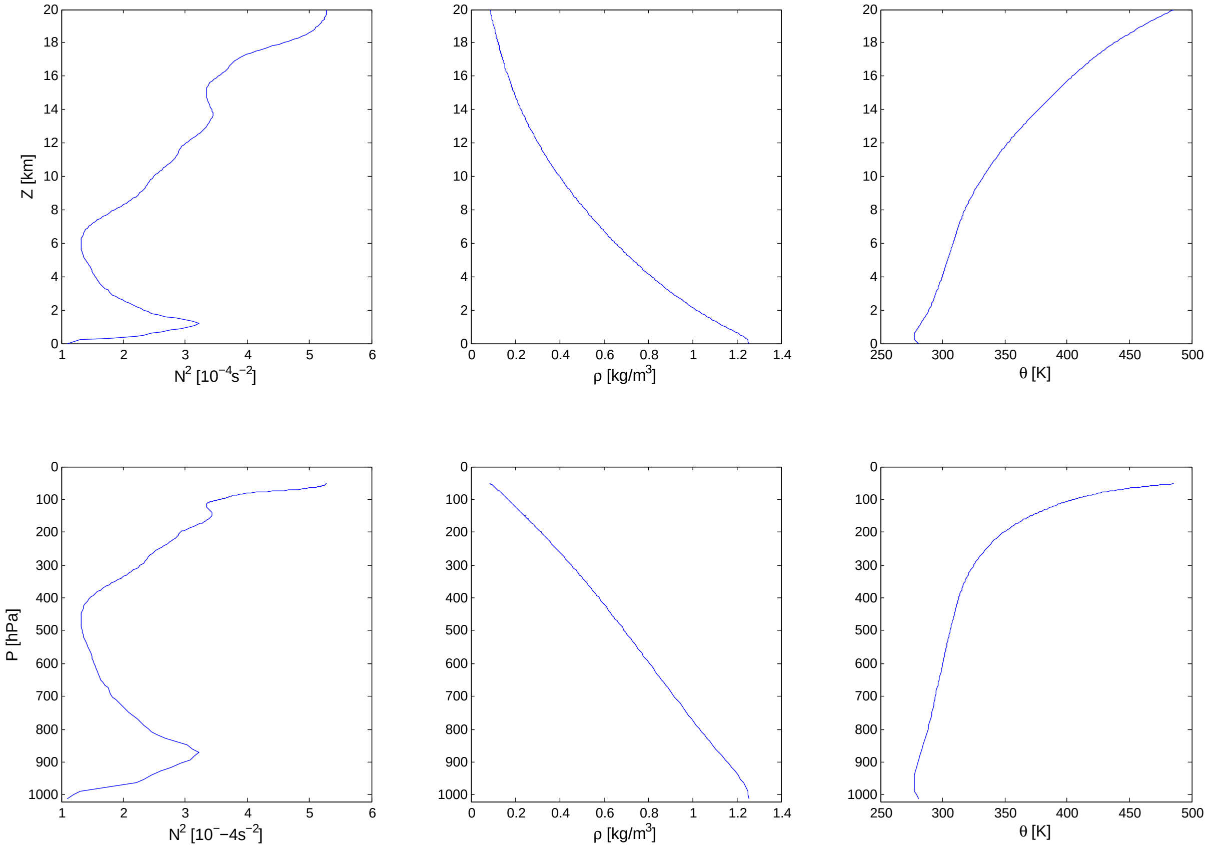


Figure 20: *Reference profiles of potential temperature θ, squared Brunt Vaisälä frequency N^2 and density ρ as a function of height (upper panels) and pressure (lower panels). The reference profiles correspond to the area-averaged values over the inversion subdomain.*

NSQREF	Reference squared Brunt-Vaisälä frequency $(1/s^2)$	MOD_20060116_18
RHOREF	Reference air density (kg/m^3)	MOD_20060116_18
PREREF	Reference pressure (hPa)	MOD_20060115_18
THETAREF	Reference potential temperature (K)	MOD_20060116_18
ZREF	Reference geopotential height (m)	MOD_20060116_18

4.4.7 Adding the coastlines [prep7]

At this stage it is not possible to plot the continental coastlines in the quasi-cartesian co-ordinate frame. In fact, this becomes only feasible if the geographical coastlines (as given in latitude/longitude co-ordinates) are transformed into the new quasi-cartesian co-ordinate system. This is done with the call

```
inversion.sh prep7
```

In Fig. 21 the transformed co-ordinates are shown. On the left panel, the co-ordinate axis are the rotated longitude and latitude and on the left panel they are given by the distance (in km) from the co-ordinate's center (see Fig. 14). In this step the following additional fields are written to the REF_20060116_18 file:

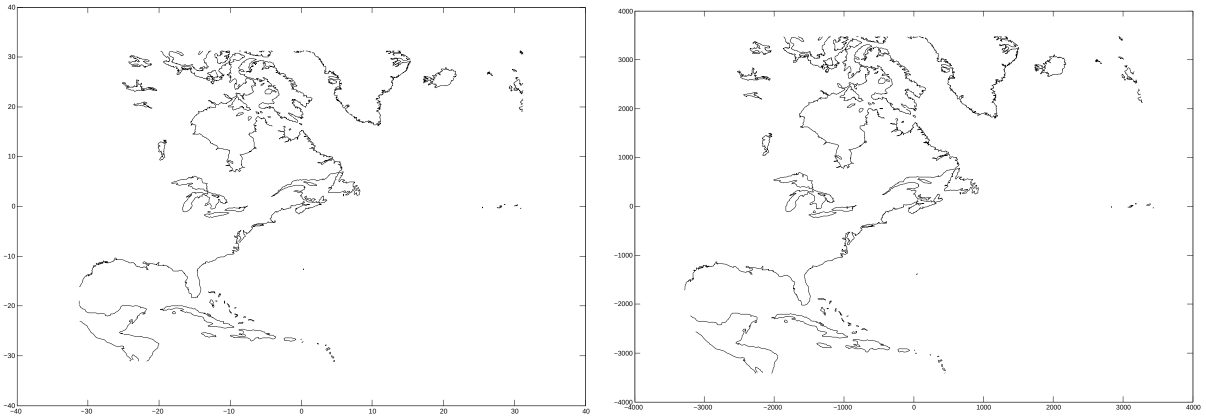


Figure 21: *Coastlines transformed into the quasi-cartesian co-ordinate system. In the left panel, the coastlines are shown in a system with rotated longitude and latitude as axes, in the right panel they are shown in a x/y co-ordinate system.*

COAST_LON	Longitude of coastline	REF_20060115_18
COAST_LAT	Latitude of coastline	REF_20060115_18
COAST_RLON	Rotated longitude of coastline	REF_20060115_18
COAST_RLAT	Rotated latitude of coastline	REF_20060115_18
COAST_X	X co-ordinate of coastline	REF_20060115_18
COAST_Y	Y co-ordinate of coastline	REF_20060115_18

The coastline in geographical co-ordinates is taken from a global data set which can be retrieved from the National Geophysical Data Center (rimmer.ngdc.noaa.gov/coast/). Only, the part of the coastlines inside the rotated co-ordinate system are specified. This allows an efficient plotting of the coastlines. If Matlab is used as a visualisation tool, the plotting can be done with the following few lines, depending whether rotated longitude/latitude or X/Y is used as co-ordinate axes.

```
%Read input file
inp = ncget('REF_20060116_18');

% Plot continental boundaries (rotated latitude/longitude)
figure;
plot(inp.COAST_RLON.data,inp.COAST_RLAT.data,'k');

% Plot continental boundaries (x/y)
figure;
plot(inp.COAST_X.data,inp.COAST_Y.data,'k');
```

The first command reads the reference file. All variables and parameters are then available in the Matlab structure inp. The COAST subfields of this structure can then immediately be plotted. Sometimes it turns out to be better to set all co-ordinates outside the plotting domain to NaN.

4.4.8 Moving the files to the run directory [prep8]

At this stage, all essential preparatory steps for the PV inversion are done: The modified PV field is defined, the boundary conditions are written and the reference profile is available. In this last step, the files are moved from the input directory to the run directory, where all iterations of the PV inversions are performed. The moving of the files is done with

```
        inversion.sh prep8
```

The ORG, MOD, ANO and REF files are now available in the run directory (see Fig. 8). No further processes are run in the input directory.

4.5 Inversion - Iterative steps toward modified PV field [pvin]

The following steps build the core of the PV inversion. An interior anomaly of Ertel's PV and the boundary values for the inversion were specified in the previous preparatory steps. Now the flow anomalies (wind vectors, temperature, pressure and potential temperature) associated with this PV anomaly has to be determined. This is the aim of the inversion. More specifically the process must be split again into several distinct steps. These different steps are shown as a flowchart in Fig. 22.

4.5.1 Calculation of secondary fields [pvin1]

Secondary fields have to be calculated. These fields are potential temperature, potential vorticity density and squared Brunt-Vaisälä frequency. The details of these calculations were already presented in section 4.4.3, and are not repeated here. All the secondary fields are calculated with the call

```
        inversion.sh pvin1
```

The call subsequently calculates potential temperature, squared Brunt-Väsäla frequency, density and Ertel's PV. Where the height levels are below topography, missing data values are written. The following table lists the new fields which are written to the MOD file:

TH	Potential temperature (K)	MOD_20060116_18
PV	Ertel's potential vorticity (pvu)	MOD_20060116_18
NSQ	Squared Brunt-Vaiälä frequency ($1/s^2$)	MOD_20060116_18
RHO	Air density (kg/m^3)	MOD_20060116_18

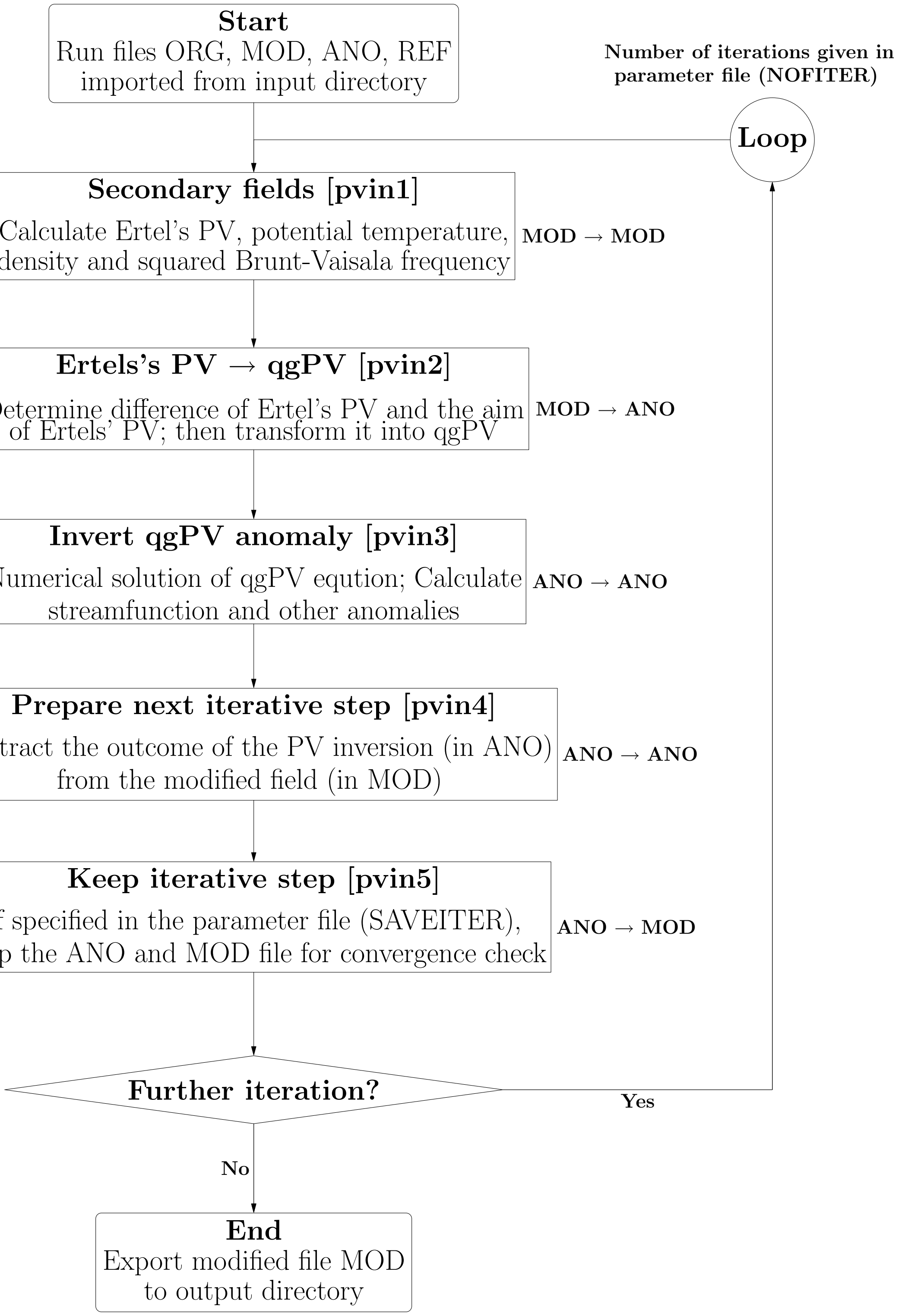

Start
Run files ORG, MOD, ANO, REF imported from input directory
Number of iterations given in parameter file (NOFITER)
Loop
Secondary fields [pvin1]
Calculate Ertel's PV, potential temperature, density and squared Brunt-Vaisala frequency
MOD → MOD
Ertels's PV → qgPV [pvin2]
Determine difference of Ertel's PV and the aim of Ertels' PV; then transform it into qgPV
MOD → ANO
Invert qgPV anomaly [pvin3]
Numerical solution of qgPV eqution; Calculate streamfunction and other anomalies
ANO → ANO
Prepare next iterative step [pvin4]
Subtract the outcome of the PV inversion (in ANO) from the modified field (in MOD)
ANO → ANO
Keep iterative step [pvin5]
If specified in the parameter file (SAVEITER), keep the ANO and MOD file for convergence check
ANO → MOD
Further iteration?
Yes
No
End
Export modified file MOD to output directory

Figure 22: *Flowchart of the different inversion steps. Additionally, to the right of the different blocks, the "flow" of information is illustrated. For instance, MOD → ANO means that the input information is taken from the MOD file and then written to the ANO file.*

4.5.2 Transforming Ertel's PV to quasi-geostrophic PV [pvin2]

So far only Ertel's PV was used. On the other hand, the inverson problem is formulated for the quasi-geostrophic PV. Therefore a conversion between the two has to be done. In a

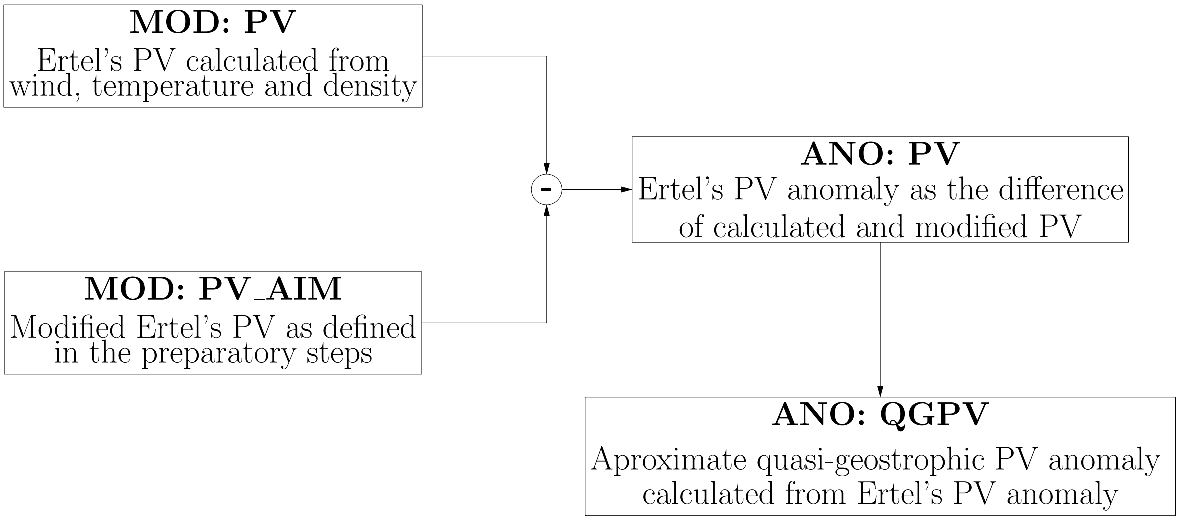

MOD: PV
Ertel's PV calculated from wind, temperature and density
MOD: PV_AIM
Modified Ertel's PV as defined in the preparatory steps
-
ANO: PV
Ertel's PV anomaly as the difference of calculated and modified PV
ANO: QGPV
Aproximate quasi-geostrophic PV anomaly calculated from Ertel's PV anomaly

Figure 23: *Determination of Ertel's PV anomaly and its transformation into an anomaly of quasi-geostrophic PV. The calculated Ertel's PV is subtracted from Ertel's PV which should be reached, and then this difference is transformed into an approximate quasi-geostrophic PV. The input is taken from the MOD file, and the output is written to the ANO file.*

first approximation Ertels's PV is given by (for clarity it is denoted as EPV in this section):

$$EPV = \frac{1}{\rho} \cdot (\xi + f) \cdot \frac{\partial \theta}{\partial z}$$

where ρ is the density, ξ the relative vorticity, f the Coriolis parameter and θ the potential temperature. Ertel's PV can be expressed in terms of the reference profile and deviation thereof. If we set $\theta = \theta_0 + \theta^*$ and $\rho = \rho_0 + \rho^*$, consider the definition of the reference stratification $N_0^2 = g/\theta_0 \cdot \partial\theta_0/\partial z$ and furthermore apply the quasi-geostrophic assumption $\xi \ll f$, the following approximate expression hold in first order (we have neglected the density perturbation ρ^* in the denominator):

$$EPV = \frac{\theta_0 N_0^2}{g\rho_0} \cdot (\xi + \frac{fg}{\theta_0 N_0^2} \cdot \frac{\partial \theta^*}{\partial z})$$

On the other hand, the quasi-geostrophic PV (denoted by qgPV in this section) is defined by the streamfunction ψ

$$qgPV = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z}(\frac{\rho_0}{N_0^2} \cdot \frac{\partial \psi}{\partial z})$$

where ρ_0 and N_0^2 denotes a reference profile of density and stratification (as defined in the preparatory steps and written to the REF file). The following relationship between the streamfunction and the meteorological fields persists:

$$g \cdot \frac{\theta^*}{\theta_0} = f \cdot \frac{\partial \psi}{\partial z} \qquad u = -\frac{\partial \psi}{\partial y} \qquad v = -\frac{\partial \psi}{\partial x}$$

If these expressions are inserted into the above equation for q, one gets:

$$qgPV = \xi + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z}(\frac{\rho_0 g}{N_0^2 f \theta_0} \cdot \theta^*)$$

In a first approximation, we treat the reference profile and the Coriolis parameter as constant. Then, this last expression is further simplified to:

$$qgPV = \xi + \frac{fg}{N_0^2 \theta_0} \cdot \frac{\partial \theta^*}{\partial z}$$

With this form, it is easy to see that the following approximate relationship between Ertel's and the quasi-geostrophic PV holds.

$$qgPV \approx \frac{\rho_0 g}{\theta_0 N_0^2} \cdot EPV - f$$

Note that according to the previous derivation, this equation does not hold exactly. It is based upon an assumption of linearisation and on approximate forms of Ertel's PV. Nevertheless, we can expect it to be a good first-order approximation. In the PV inversion tool, use is made of this relationship, but since it is only approximate, an iterative technique has to be applied. Finally note that this formula can be easily applied to PV anomalies. Indeed, if ΔPV refers to a anomaly in Ertel's PV, the corresponding anomaly in quasi-geostrophic PV is given by:

$$\Delta(qgPV) \approx \frac{\rho_0 g}{\theta_0 N_0^2} \cdot \Delta(EPV)$$

It is in this latter form that the program call

```
inversion.sh pvin2
```

makes the conversion. It takes the Ertel-PV from the MOD file, and subtracts it from the PV_AIM which is also available on the MOD file. The difference between the two PV is the new anomaly in Ertel's PV which has to be inverted in this iterative step. The approximate relationship between quasi-geostrophic PV and Ertel's PV is then used to get a corresponding anomaly of the former one (see Fig. refpvtoqg). Note that this program handles also missing data values. If at some grid point Ertel's PV is not defined, i.e. the missing data flag is set, the corresponding value of quasi-geostrophic PV is set to zero. It is not set to the missing data value since no check will be done by the PV inversion program. Therefore, it is best to treat every missing value as a vanishing anomaly of quasi-geostrophic PV. A vertical cross-section of the anomaly in quasi-geostrophic PV is shown in Fig. 24. The additional or changed fields on the file ANO_20060116_18 are given in the following table:

PV	Ertel's PV anomaly (pvu)	ANO_20060116_18
QGPV	Anomaly in quasi-geostrophic PV (s^{-1})	ANO_20060116_18

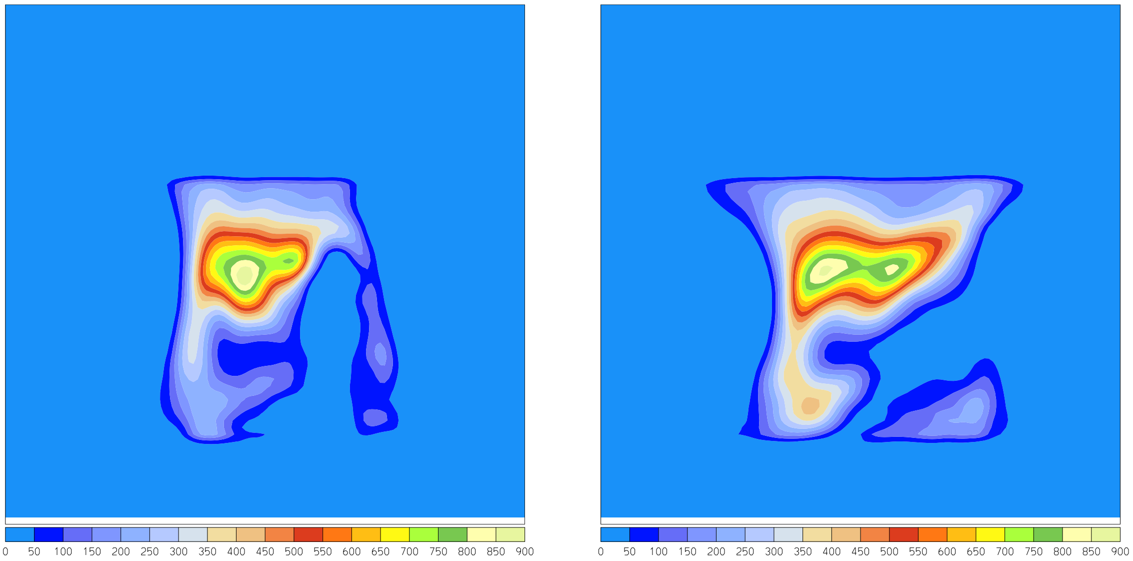


Figure 24: *Quasi-geostrophic PV anomaly in an east/west (left) and an north/south (right) vertical cross-section for 16 January 2006, 18 UTC.*

4.5.3 Inversion of the quasi-geostrophic PV [pvin3]

At this stage the interior distribution of quasi-geostrophic PV and the boundary values are given, and therefore the inversion can be performed in order to get the streamfunction. Numerical aspects of the inversion were already presented in section 2. Here, we focus on the practical aspects. The numerical problem solved in this step is expressed in the following boundary value problem:

$$q = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z}\left(\frac{\rho_0}{N_0^2} \cdot \frac{\partial\psi}{\partial z}\right)$$

with the pre-described values

$$g \cdot \frac{\theta^*}{\theta_0} = f \cdot \frac{\partial\psi}{\partial z} \qquad u = -\frac{\partial\psi}{\partial y} \qquad v = -\frac{\partial\psi}{\partial x}$$

on the upper and lower boundaries (for θ^*) and on the lateral boundaries (for u and v). The inversion program is called by

```
        inversion.sh pvin3
```

Note that the inversion affects only the file ANO_20060116_18 because the inversion is performed for an anomaly of quasi-geostrophic PV. The final aim of this step is to calculate the wind, temperature, pressure and potential temperature perturbations which are associated with the specified anomaly of quasi-geostrophic PV. The call to the program yields some information about the ongoing iterative steps. First of all, the spectral width in the x-, y- and z-direction is written to screen. In the present example this is:

```
Spectrum of the matrix in each direction
Spectrum =     1.028528       0.9136161        1.231250
```

The spectral widths $\lambda_x, \lambda_y, \lambda_z$ (in the same order as in the above program output) are important to decide whether the inversion can be reasonably be performed. The criterion is based upon the matrix elements defining the inversion problem. For a reasonable setting the following inequalities should be fulfilled:

$$\lambda_x > 1/2 \cdot \lambda_y \qquad \text{and} \qquad \lambda_x > 1/2 \cdot \lambda_z$$

$$\lambda_y > 1/2 \cdot \lambda_x \qquad \text{and} \qquad \lambda_y > 1/2 \cdot \lambda_z$$

$$\lambda_z > 1/2 \cdot \lambda_x \qquad \text{and} \qquad \lambda_z > 1/2 \cdot \lambda_y$$

which is evidently fulfilled in our case study (otherwise the program would abort with the error message that the grid dimensions are not large enough). Then some run-time statistics is provided for the iterative solution of the elliptical partial differential equation. As described in section 2 the solution of the quasi-geostrophic PV equation is found by means of an successive over-relaxation (SOR) technique. Starting from an initial estimate of the streamfunction ψ several hundred iterations of

$$\psi_i^{(n+1)} = \omega \cdot (b_i - \sum_{j=1}^{i-1} A_{i,j} \cdot \psi_j^{(n+1)} - \sum_{j=i}^{m} A_{i,j} \cdot \psi_j^{(n)}) + (1-\omega) \cdot \psi_i^{(n)}$$

are performed (see section 2 for details). Essentially two statistical measures are provided: The first one ψ_{gauge} measures the amplitude of the streamfunction which is produced in the course of the iteration, the second one is a direct check whether the iteration solution converges. In fact, after several iterations the so-far reached streamfunction is used to calculate the quasi-geostrophic PV^c according to the formula:

$$PV^c = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z} (\frac{\rho_0}{N_0^2} \cdot \frac{\partial \psi}{\partial z})$$

This value can then be compared to the quasi-geostrophic PV^i distribution which was given as input to the inversion algorithm. The following norm Δ^2 is then calculated as a measure of the convergence:

$$\Delta^2 = \sum_{i,j,k} (\psi_{i,j,k}^i - \psi_{i,j,k}^c)^2$$

Some sample output is reproduced below for the case study, the first column corresponding to ψ_{gauge} and the second one to Δ^2:

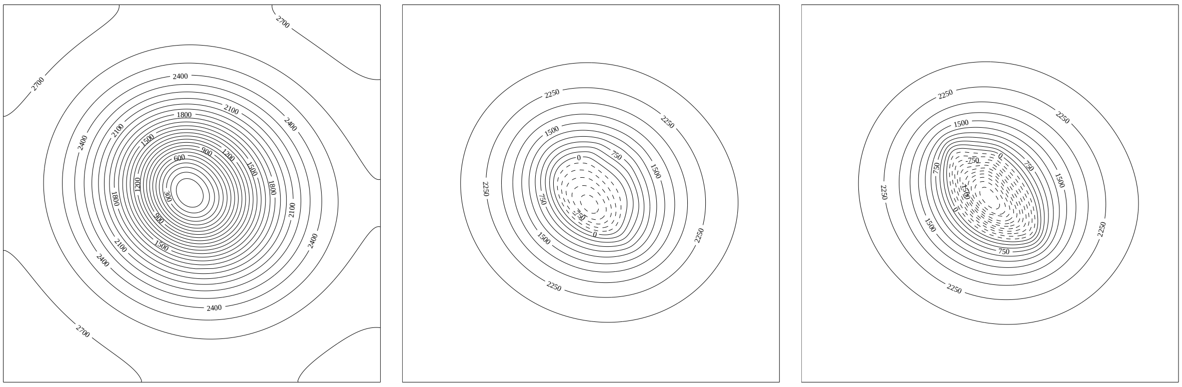


Figure 25: *Streamfunction associated with the quasi-geostrophic PV anomaly shown in Fig. refqgano. The three panels show horizontal cross-sections at 0, 4 km and 8 km above ground.*

```
psigauge =  0.000E+00    deltasq =  0.850E-09
psigauge =  0.000E+00    deltasq =  0.112E-09
psigauge = -0.681E+03    deltasq =  0.643E-10
psigauge = -0.681E+03    deltasq =  0.466E-10
psigauge = -0.806E+03    deltasq =  0.367E-10
psigauge = -0.806E+03    deltasq =  0.303E-10
psigauge = -0.547E+03    deltasq =  0.259E-10
psigauge = -0.547E+03    deltasq =  0.228E-10
psigauge = -0.398E+03    deltasq =  0.206E-10
psigauge = -0.398E+03    deltasq =  0.191E-10
psigauge = -0.309E+03    deltasq =  0.180E-10
```

Note that the gauge streamfunction reaches a value of order -3000 (corresponding to a substantial anomaly of wind and temperature), and that the convergence measure Δ^2 decreases by an order of magnitude from the beginning to the end. At the end of this step, the streamfunction associated with the quasi-geostrophic PV anomaly is determined. It is shown in Fig. 25 at three horizontal cross-sections. The minimum of the streamfunction is found in the interior of the domain, in agreement with the cyclonic flow around the positive anomaly of quasi-geostrophic PV.

Now, additional perturbation fields can be computed from this streamfunction: Potential temperature, wind components in x and y direction, pressure, and temperature. The corresponding formula are:

$$
\begin{aligned}
\theta^* &= \frac{f}{g} \cdot \theta_0 \cdot \frac{\partial \psi}{\partial z} \\
u &= -\frac{\partial \psi}{\partial y}
\end{aligned}
$$

$$v = \frac{\partial \psi}{\partial x}$$

The anomaly of potential temperature θ^* and the two horizontal wind components u and v can directly determined from the streamfunction. Moreover, the pressure anomaly is directly proportional to the streamfunction, which is only determined up to an additive constant. In order to obtain a vanishing pressure at the boundary of the inversion domain, we firstly calculate such a constant ψ_0 (essentially the average over the domain boundary) and then set the pressure to:

$$p^* = \rho_0 \cdot f \cdot (\psi - \psi_0)$$

The temperature anomaly is now easily calculated from pressure and potential temperature according the relationship $\theta = T \cdot (p_r ef/p)^\kappa$, where T is the temperature in Kelvin, $p_r ef$ is the reference pressure level (1000 hPa) and κ is the ratio between the gas constant for dry air (R_d) and the specific heat at constant pressure (c_p). Consistent with previous approximations we set:

$$T^* = (\frac{p_0}{p_r ef})^\kappa \cdot (\theta^* + \kappa \cdot \theta_0 \cdot \frac{p^*}{p_0})$$

Figure 26 shows some of these perturbation fields. The positive anomaly of quasi-geostrophic PV is associated with a cyclonic (anti-clockwise) flow field, which reaches down to surface levels. Furthermore, we know from theoretical studies that a positive PV anomaly goes along with an increase of potential temperature above the anomaly, and correspondingly a decrease of potential temperature below. This is nicely confirmed in the perturbation field of the calculation. Finally, the cyclonic flow must be associated with a local pressure minimum, according to the geostrophic wind equation. This is also fulfilled in our calculation. The following table gives all anomaly fields on the ANO_10060116_18 file which are calculated:

STREAM	Streamfunction (m^2/s)	ANO_20060116_18
THETA	Potential temperature (K)	ANO_20060116_18
U	Velocity component in X direction (m/s)	ANO_20060116_18
V	Velocity in Y direction (m/s)	ANO_20060116_18
T	Temperature ($°C$)	ANO_20060116_18
P	Pressure (hPa)	ANO_20060116_18

Finally, the question emerges whether the convergence of the algorithm was sufficient. In order to assess the quality of convergence a diagnostic tool is presented in section 5.2, which allows to calculate the quasi-geostrophic PV according to its definition and then to compare it with the anomaly given at the beginning of the PV inversion.

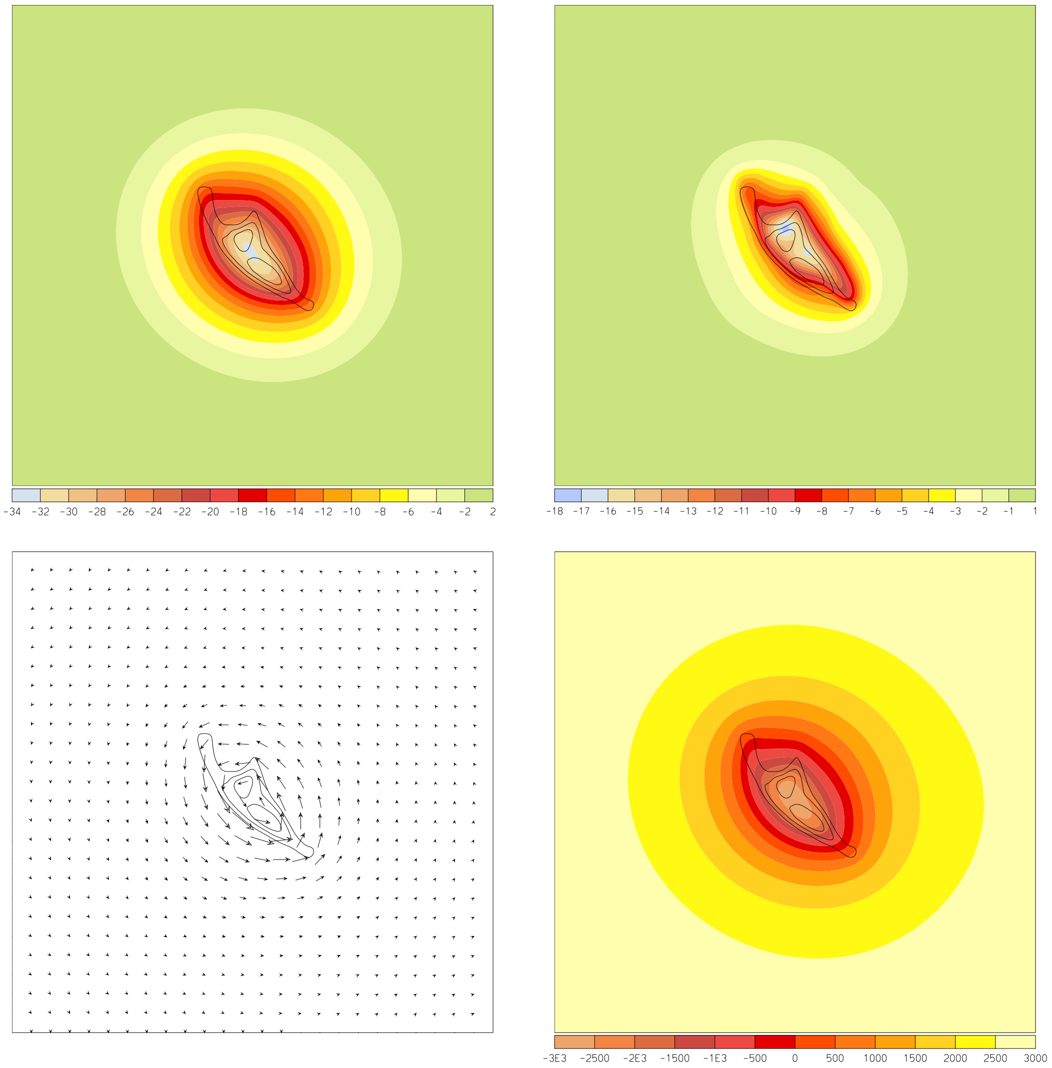


Figure 26: *Pressure, temperature, wind vectors and streamfunction (from upper-left to lower-right) associated with the quasi-geostrophic PV anomaly. The horizontal cross-section is at level 7 km. For clarity, some isolines of quasi-geostrophic PV are also shown.*

4.5.4 Preparing the next step for iteration [pvin4]

At this point, the quasi-geostrophic PV anomaly is inverted and the associated streamfunction and other flow fields have been determined. But the inversion problem is not yet solved. In fact, we specified an anomaly of Ertel's PV, but in the previous step a anomaly of quasi-geostrophic PV was inverted. By the above reasoning, we expect the two anomalies to be closely linked. However, inversion of the quasi-geostrophic PV equation is a linear problem, whereas the inversion of Ertel's PV is inherently non-linear. That is why some further iteration must be performed.

In a first step, we take the basic flow fields (temperature, velocity, pressure) from the anomaly (ANO) file and subtract it from the MOD file (Fig. 27). This is done by call to

```
inversion.sh pvin3
```

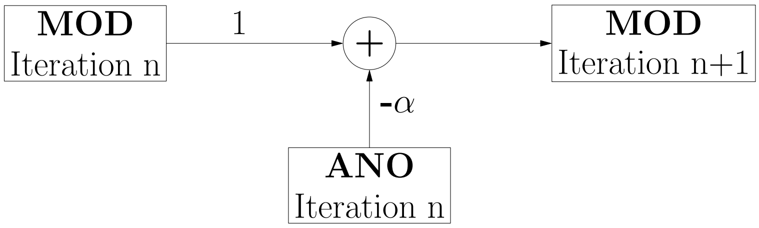


Figure 27: *Preparation for the next iteration: The fields (pressure, temperature, velocity, potential temperature) from iteration n and the anomalies of iteration n are combined to give the fields at iteration n+1.*

There is one external parameter involved in this step. It is found in the NUMERICS section of the parameter file and is called ALPHA. This value determines which fraction of the perturbation has to be subtracted from the MOD fields. Formally, we do the following transformation (see also Fig. 27):

$$\begin{aligned} u_{MOD}^{n+1} &\leftarrow u_{MOD}^{n} - \alpha \cdot u_{ANO}^{n} \\ v_{MOD}^{n+1} &\leftarrow v_{MOD}^{n} - \alpha \cdot v_{ANO}^{n} \\ T_{MOD}^{n+1} &\leftarrow T_{MOD}^{n} - \alpha \cdot T_{ANO}^{n} \\ p_{MOD}^{n+1} &\leftarrow p_{MOD}^{n} - \alpha \cdot p_{ANO}^{n} \end{aligned}$$

A value $\alpha < 1$ slows down the convergence because only part of the inversion result is used. On the other hand, it was found that the stability of the algorithm is enhanced by values $\alpha < 1$. In the case study a value of 0.5 was used, and good convergence reached, whereas a value of 1 resulted in unrealistic small-scale perturbations which diverge with continuing iterative steps.

The variables which are adjusted in the MOD_20060116_18 file are given in the following table and an example for the adjustment is shown in Fig. 28:

THETA	Potential temperature	MOD_20060115_18
U	Velocity component in X direction (m/s)	MOD_20060115_18
V	Velocity in Y direction (m/s)	MOD_20060115_18
T	Temperature ($°C$)	MOD_20060115_18
P	Pressure (hPa)	MOD_20060115_18

It might be valuable to study the convergence of the inversion in greater detail. To this aim, the master script allows the call:

```
inversion.sh pvin5
```

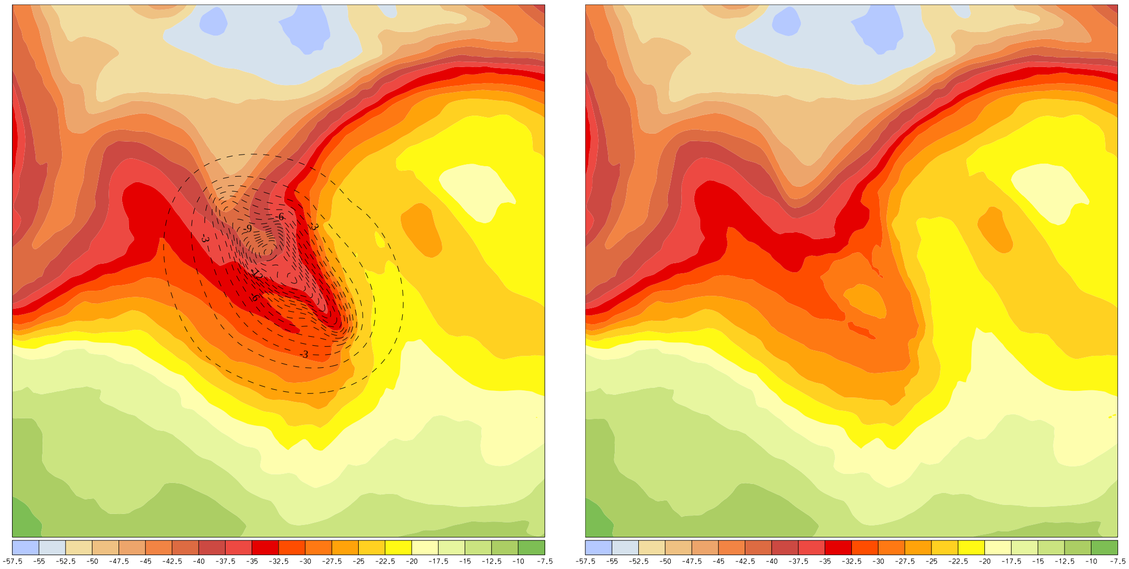


Figure 28: *Left: Temperature in the modified MOD file (in color) and in the anomaly ANO file (with contour lines) at the end of the quasi-geostrophic PV inversion. The temperature anomaly in ANO is subtracted from the initial temperature in MOD. Right: New modified temperature in the MOD file after the subtraction. This new modified field gives the next entry for the PV inversion.*

which makes a copy of the momentarily reached MOD and ANO file. For this save option to work, the SAVEITER flag in the NUMERICS section of the parameter file must be set to ”yes”.

```
BEGIN NUMERICS
  SAVEITER =    yes;
END NUMERICS
```

4.5.5 Convergence after several iterations [pvin5]

At this stage, the next iterative steps can be performed. The MOD file has been adjusted according to the outcome of the quasi-geostrophic PV inversion. The modified wind and temperature fields can be used to calculate a new Ertel-PV field, which in turn can be compared with the pre-specified aim-PV field. Hopefully, after several iterative steps, the re-currently calculated PV of the MOD file converges toward the aim-PV. If so, the non-linear inversion problem for Ertel's PV is solved. Figure 29 illustrates the convergence for the case study. Note that the iterative steps need not be launched by hand, but can be run automatically with call to

```
inversion.sh pvin
```

The script will run through all inversion steps and will loop through the number of iterations specified in NOFITER in the NUMERICS section of the parameter file.

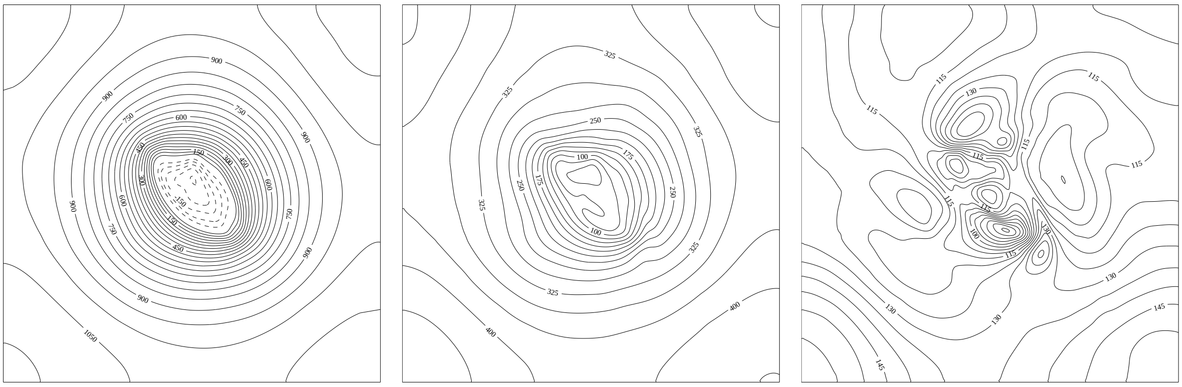


Figure 29: *streamfunction for three iterative steps. Note the different contour intervals for the three sub panels. The decrease of the amplitude clearly indicates a convergence of the PV inversion.*

```
BEGIN NUMERICS
  NOFITER  =      6;
END NUMERICS
```

The panels of Fig. 29 show the streamfunction at three consecutive iterative steps. Note that the amplitiude of the streamfunction clearly decreases, indicating that the iteration process converges. Generally, after six steps good convergence is reached. This is further supported by the standard output during the quasi-geostrophic inversion (see listing on page 38). The final line of this output is shown below for several consecutive iterations (the first one corresponding to the line on page 38):

```
psigauge = -0.309E+03    deltasq =  0.180E-10
psigauge = -0.128E+03    deltasq =  0.548E-11
psigauge = -0.570E+02    deltasq =  0.233E-11
psigauge = -0.265E+02    deltasq =  0.117E-11
```

Note that with each iterative step ”less” streamfunction is produced. Since the basic perturbation fields (temperature, pressure, horizontal wind, potential temperature) are directly proportional to the streamfunction, the corrections to the flow fields also become smaller and smaller.

4.6 Post processing - Changing the P file [post]

The PV inversion was completed in the previous step and the modified temperature, pressure and wind field is available in the run files. It is the aim of the post processing steps to bring these modified fields back to the format of the input fields. The steps are summarised in Fig. 30.

But at first the input field and the link to the modified output field must be made available with a call to:

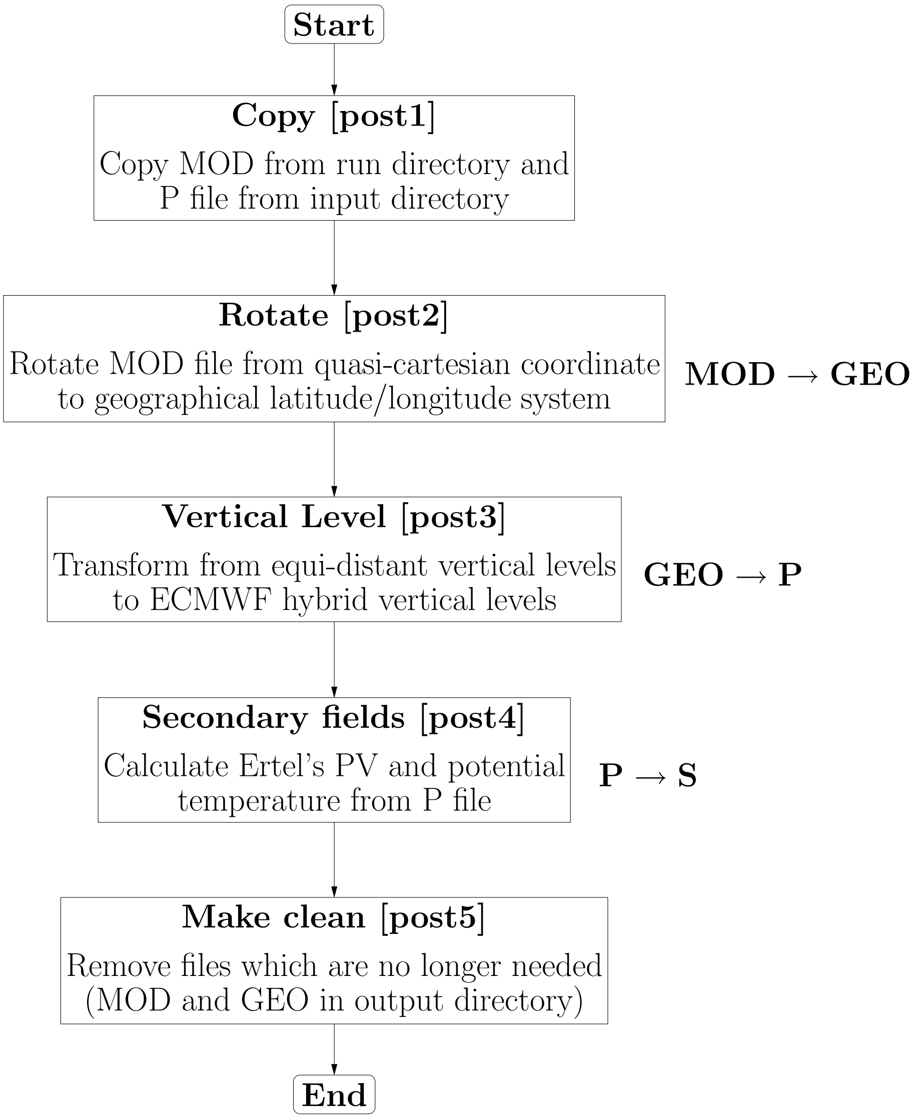

Start
Copy [post1]
Copy MOD from run directory and P file from input directory
Rotate [post2]
Rotate MOD file from quasi-cartesian coordinate to geographical latitude/longitude system
MOD → GEO
Vertical Level [post3]
Transform from equi-distant vertical levels to ECMWF hybrid vertical levels
GEO → P
Secondary fields [post4]
Calculate Ertel's PV and potential temperature from P file
P → S
Make clean [post5]
Remove files which are no longer needed (MOD and GEO in output directory)
End

Figure 30: *Different steps of post processing. Additionally, to the left of the different blocks the main flow of information is indicated. For instance, MOD → GEO means that the input is taken from the MOD file and the output written to the GEO file.*

```
        inversion.sh post1
```

The P file is copied from the input directory to the output directory and a symbolic link is set to the MOD file in the run directory.

4.6.1 Rotating to the geographical latitude/longitude grid [post2]

Figure 31 illustrates the step which is necessary to transform the meteorological fields to a geographical co-ordinate system. The left panel shows the temperature at 2000 m above sea level in the local-cartesian co-ordinate system, the right panel is the same field, but projected back to a geographical grid. Note that in this latter frame only a small section is filled by temperature values, whereas most are undefined.

The rotation itself is the reverse transformation performed in the preparatory steps (sec-

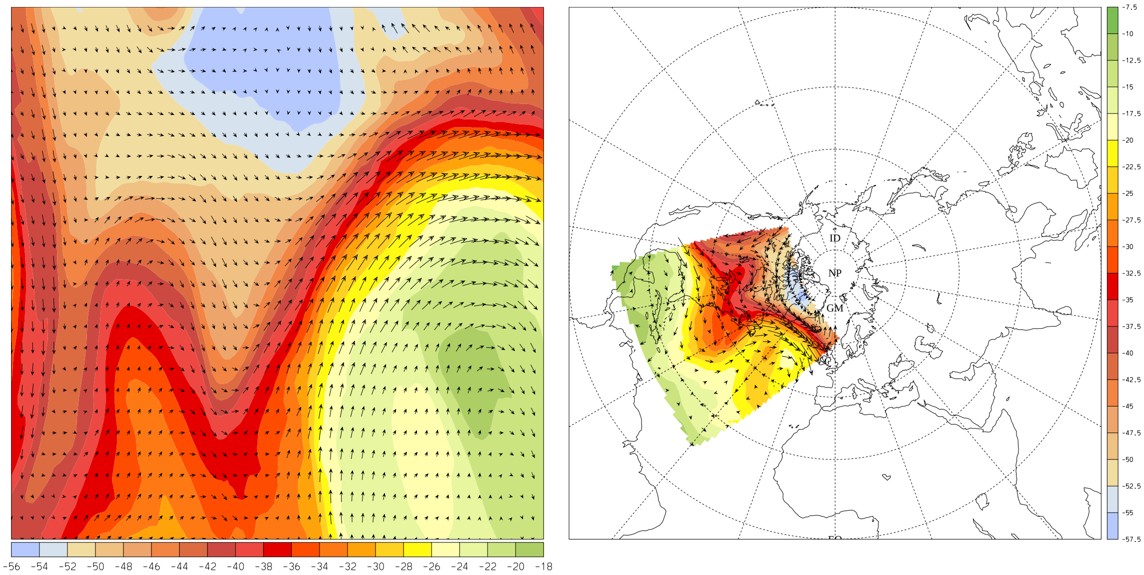


Figure 31: *Modified temperature field at 7 km above ground in the rotated co-ordinate system (left) and transformed back into the geographical latitude/longitude grid (right). Additionally the wind vectors are shown in both co-ordinate systems. Note that the transformation of the vectorial wind is much more complicated than the transformation of the scalar temperature.*

tion 4.4.2). It is called by (and its effect shown in Fig. 31)

```
inversion.sh post2
```

and its behaviour is controlled by the following entries in the parameter file:

```
BEGIN GRID
  GEO_XMIN = -180.;
  GEO_NX   =  361 ;
  GEO_DX   =    1.;
  GEO_YMIN =    0.;
  GEO_NY   =   91 ;
  GEO_DY   =    1.;
  CLON     =  -65.;
  CLAT     =   45.;
  CROT     =    0.;
END GRID
```

The first six parameters define the geographical grid of the input files. More specifically the left grid boundary is at 180° W (GEO_XMIN), the number of grid points in the zonal (west-to-east) direction is 361 (GEO_NX) and the grid resolution in zonal direction is 1° longitude (GEO_DX). Analogously, the parameters of the grid in meridional (south-to-north) direction are given: Lowest latitude 0° N (GEO_YMIN), number of grid points 91

(GEO_NY) and grid resolution 1° latitude (GEO_DY). The rotation itself is described by the last three entries (CLON, CLAT and CROT), which have the exactly same meaning as given in section 4.4.2

Note again, that the transformation of a scalar quantity is considerably easier than the transformation of a vectorial quantity. In the latter case, the specific transformation of the vectorial components must be correctly handled. The transformation algorithm for a scalar quantity is based upon the transformation rules given in Appendix 9.1 (Fortran subroutines *lmtolms* and *phtophs*). Essentially, the steps are the reverse of the ones presented in section 4.4.2. The rotated (quasi-cartesian) latitude/longitude corresponding to a grid point in the geographical co-ordinate system is obtained in the following way. Firstly, the geographical latitude/longitude $\phi_{geo}, \lambda_{geo}$ is transformed into

$$\begin{aligned} \lambda'_{rot} &= lmtolms(\phi_{geo}, \lambda_{geo}, 90 - \phi_{cen}, \lambda_{cen} - 180) \\ \phi'_{rot} &= phtophs(\phi_{geo}, \lambda_{geo}, 90 - \phi_{cen}, \lambda_{cen} - 180)) \end{aligned}$$

where the centre of the PV anomaly (the centre of the quasi-cartesian co-ordinate system in geographical latitude/longitude co-ordinates) is given by ϕ_{cen} and λ_{cen} (CLAT and CLON in the parameter file, see above). These two values are then, in a second rotation, transformed into rotated latitude and longitude:

$$\begin{aligned} \lambda_{rot} &= 90 + lmtolms(\phi'_{rot}, \lambda'_{rot} - 90, 90 + \alpha, -180) \\ \phi_{rot} &= phtophs(\phi'_{rot}, \lambda'_{rot} - 90, 90 + \alpha, -180) \end{aligned}$$

where α is the rotation angle given in the parameter file (CROT, see below). With the correspondence $(\lambda_{geo}, \phi_{geo}) \leftrightarrow (\lambda_{rot}, \phi_{rot})$ it is straightforward to perform the transformation of a scalar field. In the program package this is done by means of linear interpolation.

The fields which are written to GEO_20060116_18 are given in the following table:

U	Zonal velocity (m/s)	GEO_20060116_18
V	Meridional velocity (m/s)	GEO_20060116_18
T	Temperature $(°C)$	GEO_20060116_18
P	Pressure (hPa)	GEO_20060116_18

4.6.2 Transformation from height to hybrid ECMWF co-ordinates [post3]

The vertical co-ordinate in the back-rotated field is still geometrical height, whereas the input fields are given on the hybrid ECMWF grid. In this step the transformation to the hybrid vertical co-ordinate is performed. The call is:

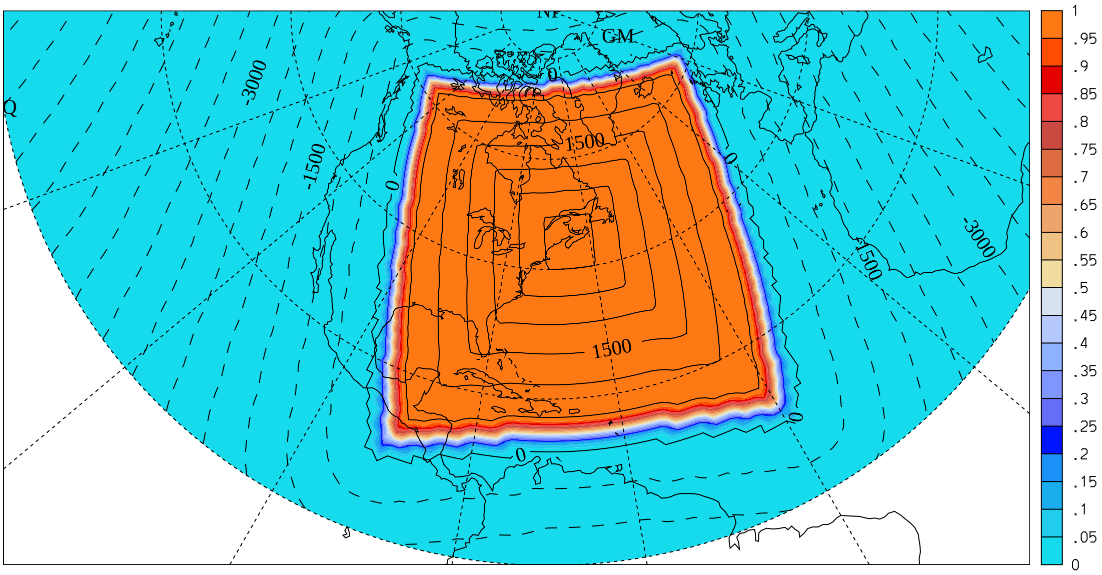


Figure 32: *Weighting function (in color) for the insertion of the modified fields into the original P file. Additionally, the distance from the boundary of insertion region is plotted as contour lines (positive inside the region, negative outside region). The zero contour line of the distance corresponds to the boundary of the insertion domain.*

```
inversion.sh post3
```

The transformation is based upon a cubic spline interpolation. The fields which are overwritten in the P20060116_18 file are:

U	Velocity component in zonal direction (m/s)	P20060115_18
V	Velocity in meridional direction (m/s)	P20060115_18
T	Temperature $(°C)$	P20060115_18
PS	Surface pressure (hPa)	P20060115_18

An impression of the modified flow field is given in Fig. 33.

4.6.3 Calculating secondary fields [post4]

In this last step, Ertel's PV and potential temperature are calculated on the ECMWF grid. This allows to directly check whether the PV anomaly was removed and to investigate how the corresponding potential temperature is changed in the absence of the PV anomaly. The call to this step is:

```
inversion.sh post4
```

The program creates the S file and writes the following two fields onto it:

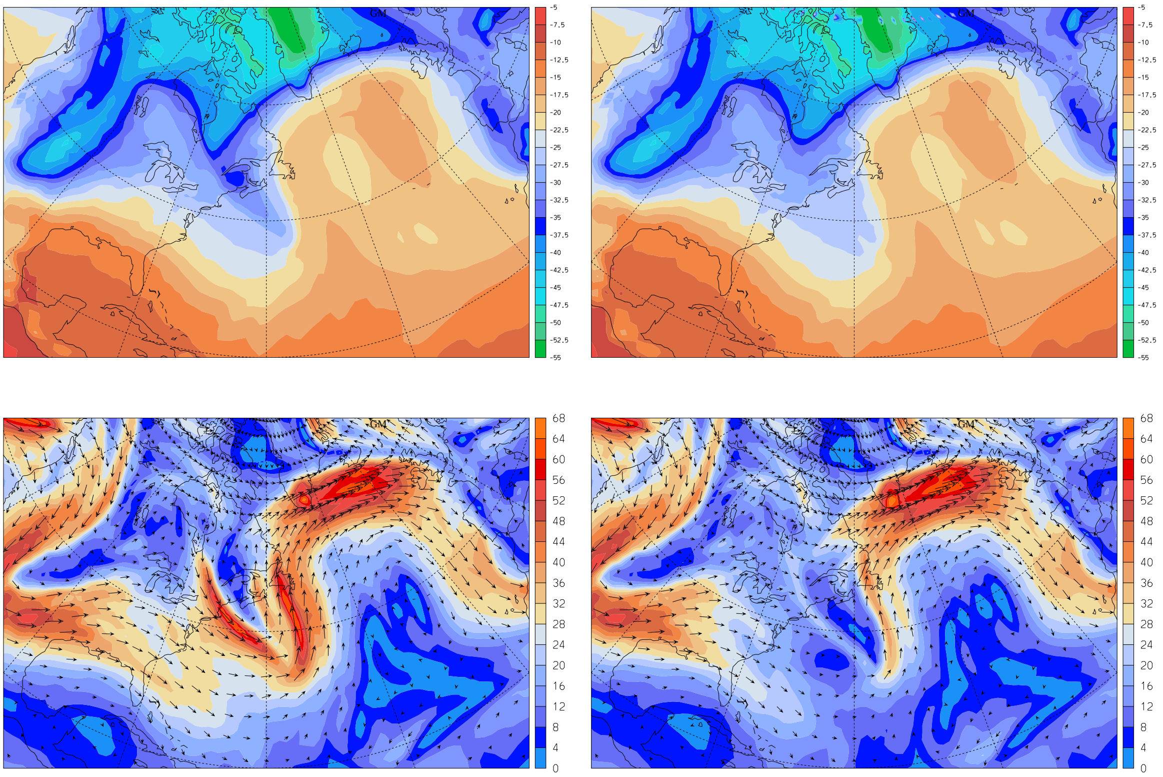


Figure 33: *Comparison of original (with PV streamer, left) and modified (without PV streamer, right) temperature (top panels) and velocity/wind vectors bottom panels) at model level 38 of the ECMWF grid.*

TH	Potential temperature (K)	S20060116_18
PV	Ertel's PV (pvu)	S20060116_18

Figure 34 shows a direct comparison of the input and of the output Ertel-PV and potential temperature. It can clearly be seen that the stratospheric PV streamer is essentially removed from the output file. Moreover, the potential temperature is considerably changed. Whereas in the presence of the PV streamer the isolines of potential temperature are pulled upwards, they are much more horizontally aligned in its absence. This "vacuum cleaner" effect of upper-level PV structures is well documented and predicted by theoretical considerations (see also Fig. 2 in section 1).

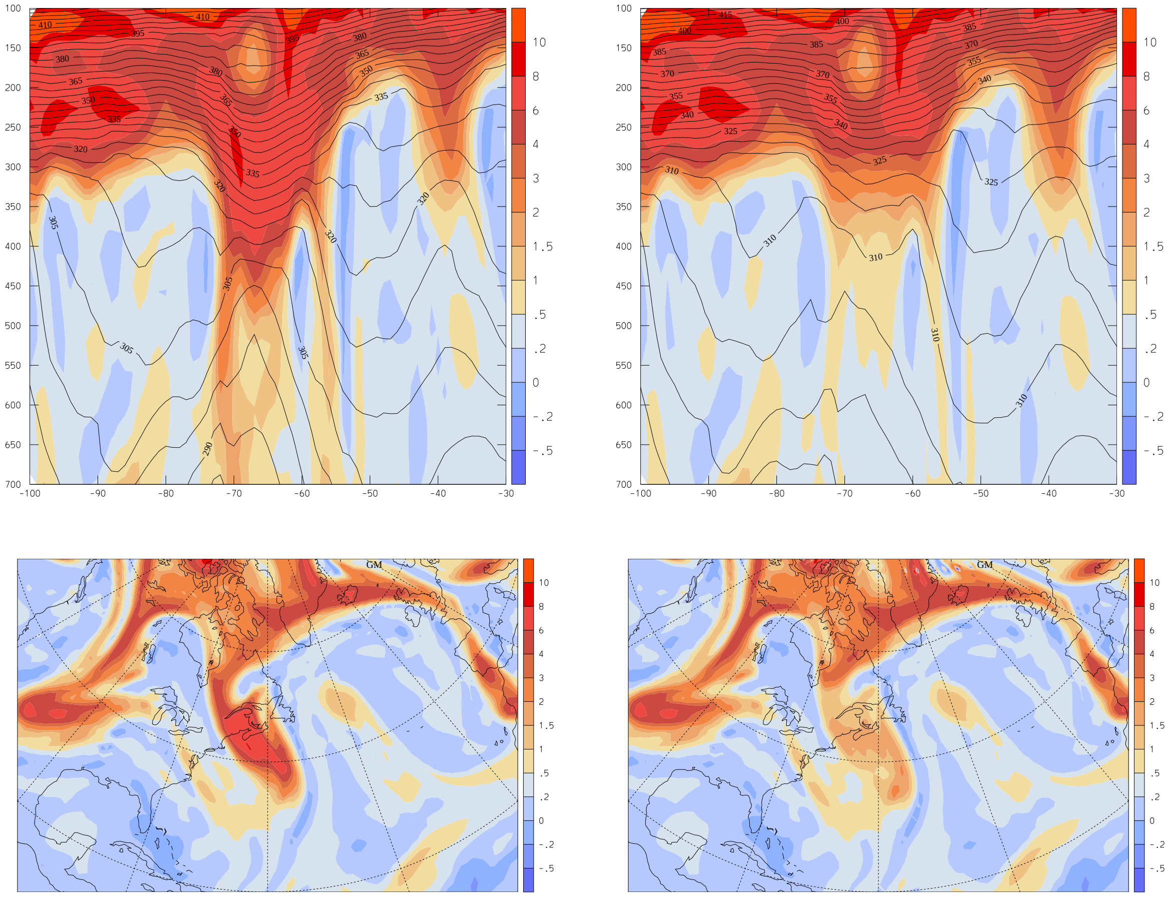


Figure 34: *Comparison of original (with PV streamer, left) and modified (without PV streamer, right) potential temperature and PV in a west/east cross section along the* 45° *N latitude circle (top panels) and PV at 350 hPa (bottom panels)*

5 Diagnostic Tools

5.1 A consistency check for the boundary condition [diag1]

The problem posed by the inversion equations constitutes a von Neumann boundary value problem. It has only a solution if some additional conditions are fulfilled between the interior PV distribution and the values of potential temperature on the upper and lower boundaries and the values of horizontal wind components on the lateral sides of the inversion domain. What follows is a detailed description of these additional requirements. The discussion very closely follows the one given in Fehlmann (1997).

To facilitate the derivation, let's define a new vertical co-ordinate such theta the quasi-geostrophic PV q can be expressed as the divergence of a vector field $\vec{E}$. Let

$$\eta(z) = -\frac{p_0(z)}{gf^2} \qquad \text{and} \qquad \Delta_h = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

where $p_o(z)$ is the reference profile of pressure, f is the Coriolis parameter and g is the

Earth's gravity. Note that η is a scaled pressure co-ordinate. If we use the hydrostatic assumption $\partial\eta/\partial z = \rho_0/f^2$, the following expression for the quasi-geostrophic PV q results:

$$q = \Delta_h \psi + \frac{\partial}{\partial\eta}(\frac{\rho_0^2}{f^2 N_0^2} \cdot \frac{\partial\psi}{\partial\eta})$$

Here, $\rho_o(z)$ and $N_o^2(z)$ are the reference profiles of density and squared Brunt-Vaisälä frequency, respectively, and ψ is the streamfunction. Hence, q is now expressed as the divergence of a vector field $\vec{E}$, and Gauss' theorem can be applied:

$$\int_G q(x,y,\eta)dxdyd\eta = \int_{\partial G} \vec{E}(x,y,\eta)d\vec{\sigma} \qquad \text{with} \qquad \vec{E} = (\frac{\partial\psi}{\partial x}, \frac{\partial\psi}{\partial y}, \frac{\rho_0^2}{f^2 N_0^2} \cdot \frac{\partial\psi}{\partial\eta})$$

Note that this integration has to be carried out in the co-ordinate system (x,y,η). If we now specify the inversion domain B as a two-dimensional domain in the (x,y)-plane and $[z_a, z_b]$ be the vertical domain, then $G = B \times [z_a, z_b]$ in the Cartesian space. This compatibility condition can be transformed back into the Cartesian co-ordinate system and the integration be split over the boundary of the domain G into an integration over the surface, the lid and the lateral boundaries of the domain. It then follows that

$$\begin{aligned}
\int_G \rho_0(z)q(x,y,z)dxdydz &= \int_{z_a}^{z_b}(\int_{\partial B}\vec{v}(x,y,z)\cdot\vec{dr})\rho_0(z)dz \\
&- \int_B \frac{\rho_0(z_a)gf\theta^*(x,y,z_a)}{\theta_0(z_a)N_0^2(z_a)}dxdy \\
&+ \int_B \frac{\rho_0(z_b)gf\theta^*(x,y,z_b)}{\theta_0(z_b)N_0^2(z_b)}dxdy
\end{aligned}$$

Only if this condition is fulfilled, does there exist a unique solution (up to a constant). This means that not every formulation of the inversion problem needs to have a solution.

The inversion program has to check whether the just discussed integral constraint is fulfilled to satisfactory accuracy. In order to estimate, how big these inconsistencies really are, we ask for a constant perturbation potential temperature θ^* which is needed in order to fulfill the integral constraint. If readily follows from the above discussion that this value is:

$$\begin{aligned}
\theta^* &= (\int_G \rho_0(z)q(x,y,z)dxdydz - \int_{z_a}^{z_b}(\int_{\partial B}\vec{v}(x,y,z)\cdot\vec{dr})\rho_0(z)dz) \\
&\quad (-\int_B \frac{\rho_0(z_a)gf}{\theta_0(z_a)N_0^2(z_a)}dxdy + \int_B \frac{\rho_0(z_b)gf}{\theta_0(z_b)N_0^2(z_b)}dxdy)^{-1}
\end{aligned}$$

If this quantity is calculated for the present example, θ^* =-2.8 K results. This value seems sufficiently small that problem can be solved despite the inconsistency. But note that the problems due to the ill-posedness of the problem are not well understood (see references in Fehlmann, 1997).

The program handling the consistency check related to the boundary values is:

```
inversion.sh diag1
```

The output from this program is given in the following table (the labels A,B,C and D are entered for easier reference in the text):

```
A   integ                    =    1.7558686E+12
B   denombot                 =    5.6644076E+11
C   denomtop                 =    5.9614413E+09
D   denom                    =   -5.6047934E+11
    theta adjustment         =    -3.132798
    theta shift @ top        =     0.000000        613.2715
    theta shift @ bot        =     0.000000        271.9355
```

The first line is the integral which includes the quasi-geostrophic PV in the interior, the velocity integrals on the laterals boundaries and the potential temperature integrals at the lower and upper boundary:

$$
\begin{aligned}
A &= \int_G \rho_0(z) q(x,y,z) dxdydz - \int_{z_a}^{z_b} (\int_{\partial B} \vec{v}(x,y,z) \cdot \vec{dr}) \rho_0(z) dz \\
&+ \int_B \frac{\rho_0(z_a) g f \theta^*(x,y,z_a)}{\theta_0(z_a) N_0^2(z_a)} dxdy - \int_B \frac{\rho_0(z_b) g f \theta^*(x,y,z_b)}{\theta_0(z_b) N_0^2(z_b)} dxdy
\end{aligned}
$$

In exact fulfillment of the integral constraint, this term A would vanish, i.e. $A = 0$. The second and third line give the integrals over the upper and lower boundary only:

$$
B = \int_B \frac{\rho_0(z_a) g f \theta^*(x,y,z_a)}{\theta_0(z_a) N_0^2(z_a)} dxdy
$$

$$
C = \int_B \frac{\rho_0(z_b) g f \theta^*(x,y,z_b)}{\theta_0(z_b) N_0^2(z_b)} dxdy
$$

The difference between these two lines is the contents of the next line:

$$D \;=\; \int_B \frac{\rho_0(z_a) g f \theta^*(x,y,z_a)}{\theta_0(z_a) N_0^2(z_a)} dxdy - \int_B \frac{\rho_0(z_b) g f \theta^*(x,y,z_b)}{\theta_0(z_b) N_0^2(z_b)} dxdy$$

Note that this term D corresponds to the last two terms in expression A. With these expressions an estimate can be made for the correction which would eliminate the inconsistency of the boundary conditions with respect to the interior distribution of quasi-geostrophic PV. This adjustment of potential temperature is the next line

$$\theta^* \;=\; A/D$$

and corresponds with the aforementioned expression for θ^*. Finally, in the last two lines the shift in potential temperature at the upper and at the lower boundary is given, together with the mean potential temperatures at these two boundaries. At the moment, no shift (value 0 in the output lines) is performed, i.e. the inconsistency in the boundary conditions is not "solved".

5.2 Convergence of the quasi-geostrophic inversion [diag2]

The key numerical algorithm is the inversion of the quasi-geostrophic PV equation (see section 4.5.3). It relates the PV to the streamfunction by

$$q_a = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z} \left(\frac{\rho_0}{N_0^2} \cdot \frac{\partial \psi}{\partial z} \right.$$

with the boundary conditions of potential temperature at the lower and upper lid, and of horizontal wind components at the lateral sides of the inversion domain:

$$g \cdot \frac{\theta^*}{\theta_0} = f \cdot \frac{\partial \psi}{\partial z} \qquad u = -\frac{\partial \psi}{\partial y} \qquad v = \frac{\partial \psi}{\partial x}$$

The numerical solution by means of the successive over-relaxation (SOR) technique then completely determines the perturbations of potential temperature and of horizontal wind. These meteorological fields can be expressed in the following way by the streamfunction (see also section 2):

$$\begin{aligned} \theta^* &= \frac{f}{g} \cdot \theta_0 \cdot \frac{\partial \psi}{\partial z} \\ u &= -\frac{\partial \psi}{\partial y} \\ v &= \frac{\partial \psi}{\partial x} \end{aligned}$$

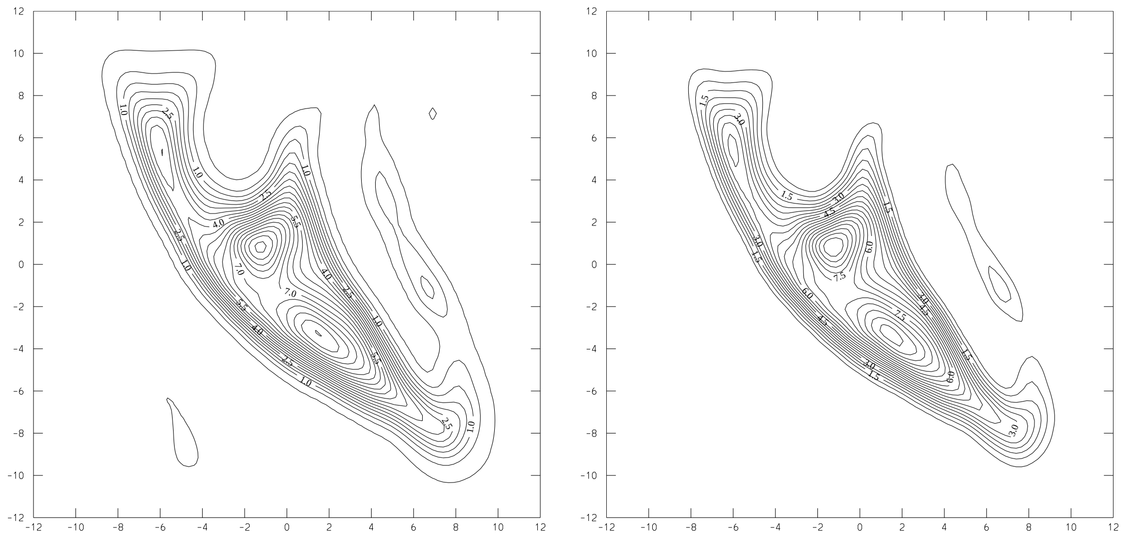


Figure 35: *Comparison of calculated (left) and pre-described (right) quasi-geostrophic PV.*

This allows a simple test whether the inversion of the quasi-geostrophic equation was successful. Indeed, the quasi-geostrophic PV can be calculated from the perturbation fields:

$$q_c = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{f^2}{\rho_0} \cdot \frac{\partial}{\partial z}(\frac{\rho_0}{N_0^2} \cdot \frac{\partial \psi}{\partial z})$$

If the inversion was successful, this calculated quasi-geostrophic PV (q_c) must coincide with the pre-described PV (q_a). For the present case study, the validation is shown in Fig. 35 for the first step of the iteration. The two panels show the calculated and the pre-described PV, respectively, and it becomes immediately clear that a reasonable degree of convergence was reached.

The calculation of the quasi-geostrophic PV is accomplished with the call

```
        inversion.sh diag2 ANO
```

where the second argument (ANO) specifies that the ANO_20060116_18 file is taken for the calculation. It is implictely assumed that the reference file is available on the corresponding REF file. The output of the calculation is then written to the same ANO_20060116_18 file, with the name QGPV_DIAG. Hence the table of modified fields and files is:

QGPV_DIAG	Calculated quasi-geostrophic PV ($1/s$)	ANO_20060116_18

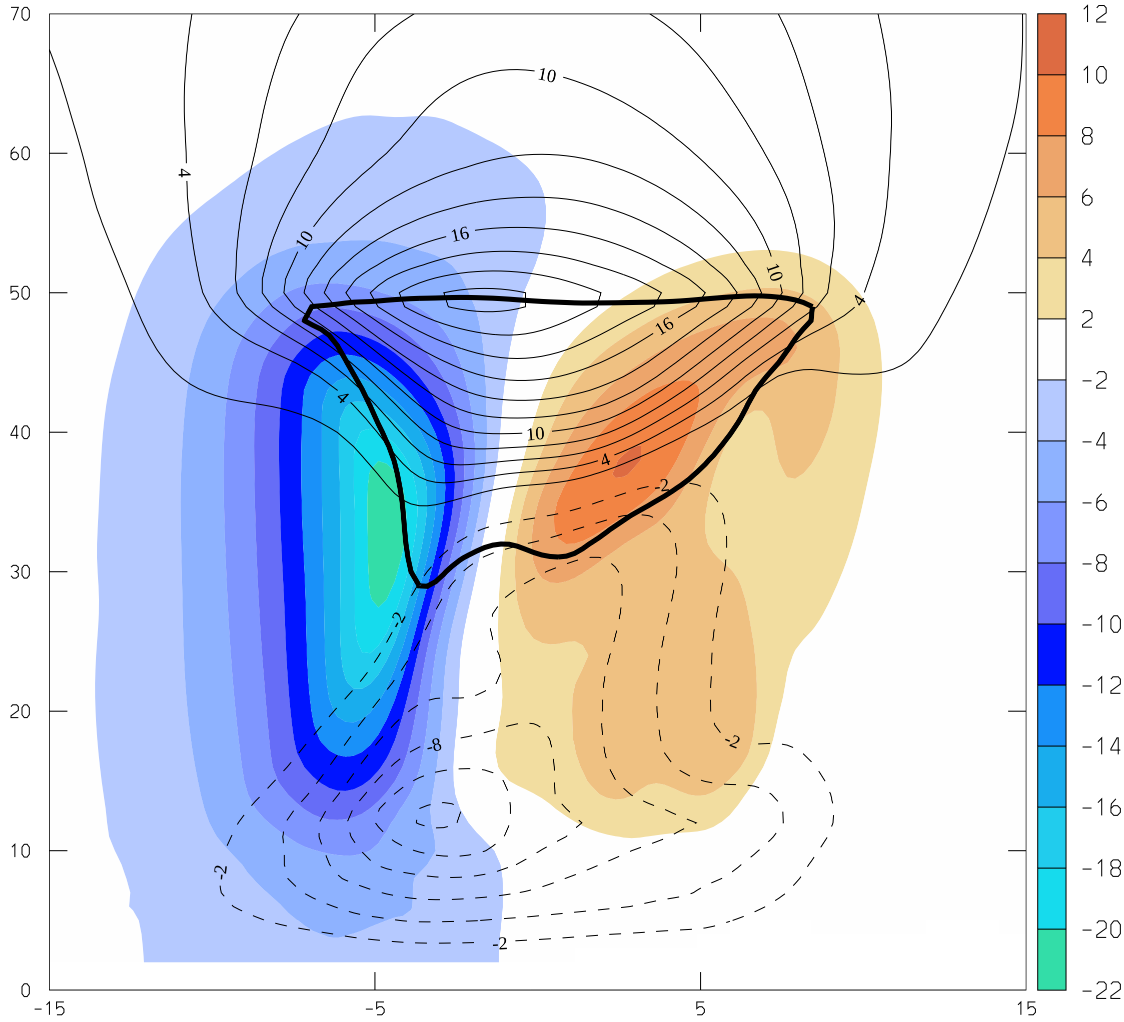


Figure 36: *Difference between the ORG and the MOD file in a vertical cross section. In color, the difference of the meridional wind velocity is shown. The difference of potential temperature is given by the thin lines (black:positive, blue:negative). The bold black line is the 2.5 isoline of the difference of Ertel's PV.*

5.3 Difference between two files [diag3]

The impact of a PV anomaly is most easily determined if the meteorological fields (temperature, velocity, pressure,...) in the presence of the anomaly is subtracted from the corresponding fields in the absence of the anomaly. The change in the meteorological fields can then be attributed to the PV anomaly. In order to facilitate this with/without comparison, a diagnostic tool is provide which calculates the difference. For instance, the call to

```
inversion.sh diag3 ORG MOD
```

takes the ORG and MOD file in the run directory and calculates the difference of all fields which are available on both input files. The difference fields are then written to a new file with prefix DIA. An example is shown in Fig. 36.

6 Final Remarks and Outlook

Several additional extensions are possible for the PV inversion tool introduced in this study. Most of them will attract a lot of interest in the research and education community. Some extensions which are and will be undertaken recently in our institute are:

a The PV inversion is not only of interest in real-case studies, as was described in this work. Considerable insight can also be gained by looking at highly idealised experiments. Such experiments enforce the existence of a tool which allows to set set up them. In the course of this diploma thesis, such a tool was developed. However, it was considered to be preferable to limit the thesis's documentation itself to the real-case experiments.

b Removing and adding, or more generally modifying the initial PV distribution, is particularly interesting if the modified state of the atmosphere is used as the input -initial and boundary values- for a regional weather prediction model. This kind of experiments allows to assess in a quantitative way how the PV field influences the future weather development. In parallel to this thesis a program package has been developed at the Institute for Atmospheric and Climate Science at ETH Zurich (IACETH) which allows to transform the (modified) P files of the PV inversion directly into the input and boundary files for the Climate HRM model. It is the aim that some case studies will be undertaken in this way by a diploma student at our institute.

c From the quasi-geostrophic omega equation it is well known that PV structures are able to enforce vertical motions. These are particularly interesting because vertical motion leads to condensation or convective instability, if the vertical temperature and humidity profile are suitable. Therefore, it would be of great interest to see how the modified PV distribution is associated with different vertical motions. Technically this means that an additional equation, the aforementioned quasi-geostrophic omega equation, has to be solved. In the coming summer semester, a student with focus on scientific programming will attack this problem.

d Teaching atmospheric dynamics at the graduate level involves several challenges. One particular challenge is associated with the abstract concept of potential vorticity. An easy-to-use PV inversion tool might be of great relieve to the teacher since he can easily illustrate and apply the concepts to real and idealised cases. It is quite certain that such a visualisation help would be much appreciated by graduate students in atmospheric dynamics. It is one aim of the dynamic's group at *IAC*ETHto improve teaching in this respect, while keeping the mathematical and physical level of the taught contents.

7 Acknowledgment

There are many persons which need to be mentioned in this acknowledgment. Perhaps, most crucially is a heartily thanks to Rene Fehlmann who initially developed the inversion tool. I had the joy to start my atmospheric research under Rene's supervision, and his enthusiasm for mathematics, numerics and atmosphere always was and still is very inspiring for me.

Recently I read that success in science cannot be attributed solely to one's own intelligence an "superiority". And with my long experience in science, I fully agree with that statement. Success needs luck, and last but not least it needs excellent teachers, supervisors and colleagues. I had the great pleasure to work together and get help from excellent scientists and teachers: Christoph Schär, Heini Wernli and Huw Davies. I appreciated very much their scientific qualities, but probably still more their warmth and friendliness as colleagues.

Many research colleagues were indirectly important in the development of this PV inversion tool. Conny Schwierz, Olivia Martius, Mischa Croci-Maspoli, Sarah Kew. Certainly, thanks should also go to Linda Schlemmer who accepted a diploma thesis based on this tool, and is now testing all aspects of it.

Finally, this work was done in the frame of the postgraduate course in Computer Science at the Fernfachhochschule Schweiz. Thanks goes to all teachers, and in particular to Lars Kruse who accepted to supervise this diploma thesis and discussed with me the different steps. Although we had some different views what should be included into this thesis ("Why do you want to send this part to the appendix? This is a diploma thesis,...") I enjoyed working with him, particularly because of his interest in the atmospheric sciences.

8 References

[**1**] Acheson, D. J., 1990: Elementary Fluid Dynamics. *Oxford Applied Mathematics and Computing Science Series,*, Oxford UP, 395pp.

[**2**] Appenzeller, C., H. C. Davies, and W. A. Norton, 1996: Fragmentation of stratospheric intrusions. *J. Geophys. Res.,* **101,** 1435-1456.

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

[**4**] Bluestein, H. B., 1993: Synoptic-Dynamic Meteorology in Midlatitudes. Volume II: Observations and Theory of Weather Systems, Oxford University Press, 594 pp.

[**5**] Charney, J. B., 1993: On the scale of atmospheric motions. Geofysisk. Publ.,**17**, No. 2.

[**6**] Davis, C. A., 1992: Piecewise potential vorticity inversion. J. Atmos. Sci., **49**, 1397-1411.

[**7**] Davis, C. A., and K. A. Emanuel, 1992: Potential vorticity diagnostic of cyclogenesis. Mon. Wea. Rev., **119**, 1929-1953.

[**8**] Ertel, H., 1942: Ein neuer hydrodynamischer Wirbelsatz. Meteor. Z., **59**, 277-281.

[**9**] Fehlmann, R., 1997: Dynamics of seminal PV elements. PhD thesis No. 12229, ETH Zürich, 143 pp.

[**10**] Fehlmann, R. and H. C. Davies, 1997: Misforecasts of synoptic systems: Diagnosis via PV retrodiction. Mon. Wea. Rev., **125**, 2247-2264.

[**11**] Holton, J. R., 1992: An Introduction to Dynamic Meteorology. Third Edition. Academic Press, San Diego, California, 511pp.

[**12**] Kleinschmidt, E., 1950: Über Aufbau und Entstehung von Zyklonen. Teil 1. Meteor. Rundsch., **3**, 1-6.

[**13**] Martius, O., E. Zenklusen, C. Schwierz, and H. C. Davies 2006: Episodes of alpine heavy precipitation with an overlying elongated stratospheric intrusion: a climatology. Intl. J. Climatology., **Vol. 96, Issue 9**, 1149-1164.

[**14**] McIntyre, M., 1997: GEFD (Geophysical and Environmental Fluid Dynamics) Summer School, DAMPTP University of Cambridge.

[**15**] Lynch, P., 2006: The Emergence of Numerical Weather Prediction. Richardson's Dream. Cambridge UP, Cambridge, UK. 279pp.

[**16**] Nebeker, F., 1995: Calculating the Weather. Meteorology in the 20th Century.. Academic Press, San Diego, California, 255pp.

[**17**] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1992: Numerical Recipes in FORTRAN: The Art of Scientific Computing. Cambridge University Press, Cambridge, UK, 1447pp.

[**18**] Sprenger, M. and H. Wernli, 2003: A northern hemispheric climatology of cross-tropopause exchange for the ERA15 time period (1979-1993). *J. Geophys. Res.*, **108**(D12), 8521, doi:10.1029/2002JD002636.

[**19**] Stohl, A., P. Bonasoni, P. Cristofanelli, W. Collins, J. Feichter, A. Frank, C. Forster, E. Gerasopoulos, H. Güggeler, P. James, T. Kentarchos, H. Kromp-Kolb, B. Krüger, C. Land, J. Meloen, A. Papayannis, A. Priller, P. Seibert, M. Sprenger, G. J. Roelofs, H. E. Scheel, C. Schnabel, P. Siegmund, L. Tobler, T. Trickl, H. Wernli, V. Wirth, P. Zanis, and C. Zerefos: 2003. Stratosphere-troposphere exchange: A review, and what we have learned from STACCATO. J. Geophys. Res.,**Vol. 108**, No. D12, 8516, doi:10.1029/2002JD002490.

[**20**] Wallace, J. M. and P. V. Hobbs, R., 1977: Atmospheric Science. An Introductory Survey. Academic Press, San Diego, California, 467pp.

9 Appendix

9.1 Scalar co-ordinate transformation

Four real function are needed for the transformation between the geographical latitude/-longitude grid used by ECMWF and the quasi-cartesian grid used by the PV inversion. Physically the new quasi-cartesian co-ordinate system is obtained by introducing a new latitude/longitude grid, but now the two co-ordinates being defined relative to a rotated north pole. Therefore, the new longitude and latitude are referred to as rotated co-ordinates. The position (in geographical latitude and longitude) of the new north pole is given as parameters POLPHI and POLLAM. Then, the functions

```
LAM = REAL FUNCTION LMSTOLM (PHIS, LAMS, POLPHI, POLLAM)
PHI = REAL FUNCTION PHSTOPH (PHIS, LAMS, POLPHI, POLLAM)
```

relate the rotated latitude (PHIS) and longitude (LAMS) to corresponding latitude (PHI) and longitude (LAM) in the geographical system. Correspondingly, the following two functions give the reverse transformation, i.e. to every position in geographical latitude/longitude a rotated pair of co-ordinates is attributed. Again, the pole position (in the geographical grid) must be passed as parameters (POLPHI and POLLAT).

```
LAMS = REAL FUNCTION LMTOLMS (PHI, LAM, POLPHI, POLLAM)
PHIS = REAL FUNCTION PHTOPHS (PHI, LAM, POLPHI, POLLAM)
```

Here, the geographical co-ordinates (PHI, LAM) are transformed into rotated co-ordinates (PHIS, LAMS). If the corresponding pair of coordinates are given, any scalar field can easily be transformed. The following Fortran functions are taken from the source code from the numerical weather prediction model of the German/Swiss weather service.

```
C     ------------------------------------------------------------------------
      REAL FUNCTION LMSTOLM (PHIS, LAMS, POLPHI, POLLAM)
C     ------------------------------------------------------------------------
C
C**** LMSTOLM  -   FC:BERECHNUNG DER WAHREN GEOGRAPHISCHEN LAENGE FUER
C****                 EINEN PUNKT MIT DEN KOORDINATEN (PHIS, LAMS)
C****                 IM ROTIERTEN SYSTEM. DER NORDPOL DES SYSTEMS HAT
C****                 DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   AUFRUF   :   LAM = LMSTOLM (PHIS, LAMS, POLPHI, POLLAM)
C**   ENTRIES  :   KEINE
C**   ZWECK    :   BERECHNUNG DER WAHREN GEOGRAPHISCHEN LAENGE FUER
C**               EINEN PUNKT MIT DEN KOORDINATEN (PHIS, LAMS)
C**               IM ROTIERTEN SYSTEM. DER NORDPOL DIESES SYSTEMS HAT
C**               DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   VERSIONS-
C**   DATUM    :   03.05.90
```

```
C**
C**   EXTERNALS:   KEINE
C**   EINGABE-
C**   PARAMETER:   PHIS     REAL   GEOGR. BREITE DES PUNKTES IM ROT.SYS.
C**                LAMS     REAL   GEOGR. LAENGE DES PUNKTES IM ROT.SYS.
C**                POLPHI   REAL   WAHRE GEOGR. BREITE DES NORDPOLS
C**                POLLAM   REAL   WAHRE GEOGR. LAENGE DES NORDPOLS
C**   AUSGABE-
C**   PARAMETER:   WAHRE GEOGRAPHISCHE LAENGE ALS WERT DER FUNKTION
C**                ALLE WINKEL IN GRAD (NORDEN>0, OSTEN>0)
C**
C**   COMMON-
C**   BLOECKE  :   KEINE
C**
C**   FEHLERBE-
C**   HANDLUNG :   KEINE
C**   VERFASSER:   D.MAJEWSKI

      REAL        LAMS,PHIS,POLPHI,POLLAM

      DATA        ZRPI18 , ZPIR18  / 57.2957795 , 0.0174532925 /

      ZSINPOL = SIN(ZPIR18*POLPHI)
      ZCOSPOL = COS(ZPIR18*POLPHI)
      ZLAMPOL = ZPIR18*POLLAM
      ZPHIS   = ZPIR18*PHIS
      ZLAMS   = LAMS
      IF(ZLAMS.GT.180.0) ZLAMS = ZLAMS - 360.0
      ZLAMS   = ZPIR18*ZLAMS

      ZARG1   = SIN(ZLAMPOL)*(- ZSINPOL*COS(ZLAMS)*COS(ZPHIS)  +
     1                          ZCOSPOL*           SIN(ZPHIS)) -
     2          COS(ZLAMPOL)*           SIN(ZLAMS)*COS(ZPHIS)
      ZARG2   = COS(ZLAMPOL)*(- ZSINPOL*COS(ZLAMS)*COS(ZPHIS)  +
     1                          ZCOSPOL*           SIN(ZPHIS)) +
     2          SIN(ZLAMPOL)*           SIN(ZLAMS)*COS(ZPHIS)
      IF (ABS(ZARG2).LT.1.E-30) THEN
        IF (ABS(ZARG1).LT.1.E-30) THEN
          LMSTOLM =   0.0
        ELSEIF (ZARG1.GT.0.) THEN
              LMSTOLAM =  90.0
            ELSE
              LMSTOLAM = -90.0
            ENDIF
      ELSE
```

```
        LMSTOLM = ZRPI18*ATAN2(ZARG1,ZARG2)
      ENDIF

      RETURN
      END

C     -------------------------------------------------------------------------
      REAL FUNCTION PHSTOPH (PHIS, LAMS, POLPHI, POLLAM)
C     -------------------------------------------------------------------------
C
C**** PHSTOPH  -   FC:BERECHNUNG DER WAHREN GEOGRAPHISCHEN BREITE FUER
C****                 EINEN PUNKT MIT DEN KOORDINATEN (PHIS, LAMS) IM
C****                 ROTIERTEN SYSTEM. DER NORDPOL DIESES SYSTEMS HAT
C****                 DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   AUFRUF   :   PHI = PHSTOPH (PHIS, LAMS, POLPHI, POLLAM)
C**   ENTRIES  :   KEINE
C**   ZWECK    :   BERECHNUNG DER WAHREN GEOGRAPHISCHEN BREITE FUER
C**               EINEN PUNKT MIT DEN KOORDINATEN (PHIS, LAMS) IM
C**               ROTIERTEN SYSTEM. DER NORDPOL DIESES SYSTEMS HAT
C**               DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   VERSIONS-
C**   DATUM    :   03.05.90
C**
C**   EXTERNALS:   KEINE
C**   EINGABE-
C**   PARAMETER:   PHIS     REAL   GEOGR. BREITE DES PUNKTES IM ROT.SYS.
C**               LAMS     REAL   GEOGR. LAENGE DES PUNKTES IM ROT.SYS.
C**               POLPHI   REAL   WAHRE GEOGR. BREITE DES NORDPOLS
C**               POLLAM   REAL   WAHRE GEOGR. LAENGE DES NORDPOLS
C**   AUSGABE-
C**   PARAMETER:   WAHRE GEOGRAPHISCHE BREITE ALS WERT DER FUNKTION
C**               ALLE WINKEL IN GRAD (NORDEN>0, OSTEN>0)
C**
C**   COMMON-
C**   BLOECKE  :   KEINE
C**
C**   FEHLERBE-
C**   HANDLUNG :   KEINE
C**   VERFASSER:   D.MAJEWSKI

      REAL        LAMS,PHIS,POLPHI,POLLAM

      DATA        ZRPI18 , ZPIR18  / 57.2957795 , 0.0174532925 /

      SINPOL = SIN(ZPIR18*POLPHI)
      COSPOL = COS(ZPIR18*POLPHI)
      ZPHIS  = ZPIR18*PHIS
      ZLAMS  = LAMS
      IF(ZLAMS.GT.180.0) ZLAMS = ZLAMS - 360.0
      ZLAMS  = ZPIR18*ZLAMS
      ARG     = COSPOL*COS(ZPHIS)*COS(ZLAMS) + SINPOL*SIN(ZPHIS)

      PHSTOPH = ZRPI18*ASIN(ARG)

      RETURN
      END

C     ------------------------------------------------------------------------
      REAL FUNCTION LMTOLMS (PHI, LAM, POLPHI, POLLAM)
C     ------------------------------------------------------------------------
C
C%Z% Modul %M%, V%I% vom %G%, extrahiert am %H%
C
C**** LMTOLMS  -   FC:UMRECHNUNG DER WAHREN GEOGRAPHISCHEN LAENGE LAM
C****                 AUF EINEM PUNKT MIT DEN KOORDINATEN (PHIS, LAMS)
C****                 IM ROTIERTEN SYSTEM. DER NORDPOL DES SYSTEMS HAT
C****                 DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   AUFRUF   :   LAM = LMTOLMS (PHI, LAM, POLPHI, POLLAM)
C**   ENTRIES  :   KEINE
C**   ZWECK    :   UMRECHNUNG DER WAHREN GEOGRAPHISCHEN LAENGE LAM AUF
C**               EINEM PUNKT MIT DEN KOORDINATEN (PHIS, LAMS) IM
C**               ROTIERTEN SYSTEM. DER NORDPOL DIESES SYSTEMS HAT
C**               DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   VERSIONS-
C**   DATUM    :   03.05.90
C**
C**   EXTERNALS:   KEINE
C**   EINGABE-
C**   PARAMETER:   PHI    REAL BREITE DES PUNKTES IM GEOGR. SYSTEM
C**               LAM    REAL LAENGE DES PUNKTES IM GEOGR. SYSTEM
C**               POLPHI REAL GEOGR.BREITE DES N-POLS DES ROT. SYSTEMS
C**               POLLAM REAL GEOGR.LAENGE DES N-POLS DES ROT. SYSTEMS
C**   AUSGABE-
C**   PARAMETER:   WAHRE GEOGRAPHISCHE LAENGE ALS WERT DER FUNKTION
C**               ALLE WINKEL IN GRAD (NORDEN>0, OSTEN>0)
C**
C**   COMMON-
C**   BLOECKE  :   KEINE
C**
C**   FEHLERBE-
```

```
C**   HANDLUNG :   KEINE
C**   VERFASSER:   G. DE MORSIER

      REAL        LAM,PHI,POLPHI,POLLAM

      DATA        ZRPI18 , ZPIR18  / 57.2957795 , 0.0174532925 /

      ZSINPOL = SIN(ZPIR18*POLPHI)
      ZCOSPOL = COS(ZPIR18*POLPHI)
      ZLAMPOL =     ZPIR18*POLLAM
      ZPHI    =     ZPIR18*PHI
      ZLAM    = LAM
      IF(ZLAM.GT.180.0) ZLAM = ZLAM - 360.0
      ZLAM    = ZPIR18*ZLAM

      ZARG1   = - SIN(ZLAM-ZLAMPOL)*COS(ZPHI)
      ZARG2   = - ZSINPOL*COS(ZPHI)*COS(ZLAM-ZLAMPOL)+ZCOSPOL*SIN(ZPHI)
      IF (ABS(ZARG2).LT.1.E-30) THEN
        IF (ABS(ZARG1).LT.1.E-30) THEN
          LMTOLMS =   0.0
        ELSEIF (ZARG1.GT.0.) THEN
              LMTOLMS =  90.0
            ELSE
              LMTOLMS = -90.0
            ENDIF
      ELSE
        LMTOLMS = ZRPI18*ATAN2(ZARG1,ZARG2)
      ENDIF

      RETURN
      END

C     -------------------------------------------------------------------------
      REAL FUNCTION PHTOPHS (PHI, LAM, POLPHI, POLLAM)
C     -------------------------------------------------------------------------
C
C%Z% Modul %M%, V%I% vom %G%, extrahiert am %H%
C
C**** PHTOPHS  -   FC:UMRECHNUNG DER WAHREN GEOGRAPHISCHEN BREITE PHI
C****                 AUF EINEM PUNKT MIT DEN KOORDINATEN (PHIS, LAMS)
C****                 IM ROTIERTEN SYSTEM. DER NORDPOL DES SYSTEMS HAT
C****                 DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   AUFRUF   :   PHI = PHTOPHS (PHI, LAM, POLPHI, POLLAM)
C**   ENTRIES  :   KEINE
C**   ZWECK    :   UMRECHNUNG DER WAHREN GEOGRAPHISCHEN BREITE PHI AUF
```

```
C**                EINEM PUNKT MIT DEN KOORDINATEN (PHIS, LAMS) IM
C**                ROTIERTEN SYSTEM. DER NORDPOL DIESES SYSTEMS HAT
C**                DIE WAHREN KOORDINATEN (POLPHI, POLLAM)
C**   VERSIONS-
C**   DATUM    :   03.05.90
C**
C**   EXTERNALS:   KEINE
C**   EINGABE-
C**   PARAMETER:   PHI    REAL BREITE DES PUNKTES IM GEOGR. SYSTEM
C**                LAM    REAL LAENGE DES PUNKTES IM GEOGR. SYSTEM
C**                POLPHI REAL GEOGR.BREITE DES N-POLS DES ROT. SYSTEMS
C**                POLLAM REAL GEOGR.LAENGE DES N-POLS DES ROT. SYSTEMS
C**   AUSGABE-
C**   PARAMETER:   ROTIERTE BREITE PHIS ALS WERT DER FUNKTION
C**                ALLE WINKEL IN GRAD (NORDEN>0, OSTEN>0)
C**
C**   COMMON-
C**   BLOECKE  :   KEINE
C**
C**   FEHLERBE-
C**   HANDLUNG :   KEINE
C**   VERFASSER:   G. DE MORSIER

      REAL        LAM,PHI,POLPHI,POLLAM

      DATA        ZRPI18 , ZPIR18  / 57.2957795 , 0.0174532925 /

      ZSINPOL = SIN(ZPIR18*POLPHI)
      ZCOSPOL = COS(ZPIR18*POLPHI)
      ZLAMPOL = ZPIR18*POLLAM
      ZPHI    = ZPIR18*PHI
      ZLAM    = LAM
      IF(ZLAM.GT.180.0) ZLAM = ZLAM - 360.0
      ZLAM    = ZPIR18*ZLAM
      ZARG    = ZCOSPOL*COS(ZPHI)*COS(ZLAM-ZLAMPOL) + ZSINPOL*SIN(ZPHI)

      PHTOPHS = ZRPI18*ASIN(ZARG)

      RETURN
      END
```

9.2 Controlling Linux Shell Script

In this appendix the controlling Linux Shell script of the PV inversion is reproduced. This script offers a user-friendly interface to the inversion. It is essentially split into five different section: (1) installation and parameter settings, (2) preparatory steps, (3) iterative solution of the PV inversion problem, (4) post-processing, and (5) diagnostic tools.

```
#!/bin/csh

# Master Linux script for PV inversion
# Michael Sprenger / Winter 2006,2007

# -------------------------------------------------------------------
# Set some variables and paths
# -------------------------------------------------------------------

# Handling of input parameters
set step="help"
if ( $#argv == 1 ) then
  set step=$1
endif
if ( $#argv == 2 ) then
  set step=$1
  set file1=$2
endif
if ( $#argv == 3 ) then
  set step=$1
  set file1=$2
  set file2=$3
endif

# Set base directory for programmes
set bdir=/home/sprenger/PV_Inversion_Tool/real/

# Set name of the parameter file, the coastline file and the sample files
set parafile="${PWD}/inversion.param"
set coastfile="${bdir}/prep/coastline.dat"
set sampledir="/net/rossby/lhome/sprenger/PV_Inversion_Tool/real/inp/"

# Extract parameters from parameter file
set progex=${bdir}/inversion.perl
set    date=‘${progex} ${parafile} date           | awk ’{ print $2}’‘
set nofiter=‘${progex} ${parafile} n_of_iteration | awk ’{ print $2}’‘
set    save=‘${progex} ${parafile} save_iteration | awk ’{ print $2}’‘
set    idir=‘${progex} ${parafile} inp_dir        | awk ’{ print $2}’‘
set    rdir=‘${progex} ${parafile} run_dir        | awk ’{ print $2}’‘
set    odir=‘${progex} ${parafile} out_dir        | awk ’{ print $2}’‘

# -----------------------------------------------------------------------------
# Installation, help and sample
# -----------------------------------------------------------------------------

# Create the needed directories
if ( ${step} == "inst" ) then
  if ( ! -d ${idir} ) mkdir ${idir}
  if ( ! -d ${rdir} ) mkdir ${rdir}
  if ( ! -d ${odir} ) mkdir ${odir}
endif

# Create the needed directories
if ( ${step} == "help" ) then
  echo
  echo "Installation"
  echo "   inst:   Creates the input, run and output directory"
  echo
  echo "Sample case study"
  echo "   sample: Copy all files for a sample case study"
  echo
  echo "Preparing input files [prep]"
  echo "   prep0:  Calculate S file with PV and TH"
  echo "   prep1:  Interpolate onto height levels"
  echo "   prep2:  Rotate into local cartesian co-ordinate system"
  echo "   prep3:  Add TH,PV,NSQ and RHO to the data file"
  echo "   prep4:  Define modified and anomaly PV field and boundary values"
  echo "   prep5:  Reduce the domain size and split the input files"
  echo "   prep6:  Add the reference profile"
  echo "   prep7:  Add coastlines to REF file"
  echo "   prep8:  Move the files to the run directory"
  echo
  echo "Perform the PV inversion [pvin]"
  echo "   pvin1:  Add NSQ, TH, RHO, and PV"
  echo "   pvin2:  Change Ertel’s PV anomaly into one of quasi-geostrophic PV"
  echo "   pvin3:  Inversion of quasi-geostrophic PV anomaly"
  echo "   pvin4:  Subtract anomaly from MOD file"
  echo "   pvin5:  Keep iterative steps if save flag is set"
  echo
  echo "Postprocessing [post]"
  echo "   post1:  Copy needed files from input and run directory"
  echo "   post2:  Rotate from quasi-cartesian co-ordinate frame to lat/lon system"
  echo "   post3:  Bring modified fields back to P file"
  echo "   post4:  Calculate S file with PV and TH"
  echo "   post5:  Make clean"
  echo
  echo "Diagnostic Tools"
  echo "   diag1:  Check the consistency of the boundary conditions"
  echo "   diag2:  Calculate the quasi-geostrophic PV"
  echo "   diag3:  Get the difference between two files"’
  echo
endif


# Copy sample files (if specified)
if ( ${step} == "sample" ) then
  \cp ${sampledir}/P20060116_18 ${idir}
  \cp ${sampledir}/ml_cst       ${idir}
  \cp ${sampledir}/Z20060116_18 ${idir}
  \cp ${sampledir}/pl_cst       ${idir}
endif

# ---------------------------------------------------------------------------
# Preparatory steps
# ---------------------------------------------------------------------------

# Change to data directory
cd ${idir}

# Step 0: Calculate S file with PV and TH (not in prep mode)
if ( ${step} == "prep0" ) then
    \rm -f S${date}
    p2s P${date} TH PV
endif

# Step 1: Interpolate onto height levels
if ( ${step} == "prep1" | ${step} == "prep" ) then
  echo "P${date}"                        >! fort.10
  echo "Z${date}"                        >> fort.10
  echo "H${date}"                        >> fort.10
  ${bdir}/inversion.perl ${parafile} p2z >> fort.10
  echo "U        U       P${date}"       >> fort.10
  echo "V        V       P${date}"       >> fort.10
  echo "T        T       P${date}"       >> fort.10
  echo "Q        Q       P${date}"       >> fort.10
  ${bdir}/prep/p2z
  \mv H${date}_cst zl_cst
  changecst H${date} zl_cst
```

```
   #\rm -f fort.10
endif

# Step 2: Rotate into local cartesian co-ordinate system
if ( ${step} == "prep2" | ${step} == "prep" ) then
  echo "H${date}"                                 >! fort.10
  echo "R${date}"                                 >> fort.10
  ${bdir}/inversion.perl ${parafile} rotate_grid >> fort.10
  echo "5"                                        >> fort.10
  echo "ORO"                                      >> fort.10
  echo "U.V"                                      >> fort.10
  echo "P"                                        >> fort.10
  echo "T"                                        >> fort.10
  echo "Q"                                        >> fort.10
  ${bdir}/prep/rotate_grid
  \mv R${date}_cst ro_cst
  changecst R${date} ro_cst
  \rm -f fort.10
endif

# Step 3: Add TH,PV,NSQ and RHO to the data file
if ( ${step} == "prep3" | ${step} == "prep" ) then
  echo "TH"         >! fort.10
  echo "R${date}"   >> fort.10
  echo "R${date}"   >> fort.10
  echo "5       "   >> fort.10
  ${bdir}/prep/z2s
  echo "PV"         >! fort.10
  echo "R${date}"   >> fort.10
  echo "R${date}"   >> fort.10
  echo "5       "   >> fort.10
  ${bdir}/prep/z2s
  echo "NSQ"        >! fort.10
  echo "R${date}"   >> fort.10
  echo "R${date}"   >> fort.10
  echo "5       "   >> fort.10
  ${bdir}/prep/z2s
  echo "RHO"        >! fort.10
  echo "R${date}"   >> fort.10
  echo "R${date}"   >> fort.10
  echo "5       "   >> fort.10
  ${bdir}/prep/z2s
  \rm -f fort.10
endif
```

```
# Step 4: Set the modified PV field and boundary values
if ( ${step} == "prep4" | ${step} == "prep" ) then
  echo "R${date}"                                >! fort.10
  ${bdir}/inversion.perl ${parafile} def_anomaly >> fort.10
  ${bdir}/prep/def_anomaly
  \rm -f fort.10
endif

# Step 5: Reduce the domain size and split the input files
if ( ${step} == "prep5" | ${step} == "prep" ) then
  echo "PV        PV        R${date}    ORG_${date}" >! fort.10
  echo "U         U         R${date}    ORG_${date}" >> fort.10
  echo "V         V         R${date}    ORG_${date}" >> fort.10
  echo "TH        TH        R${date}    ORG_${date}" >> fort.10
  echo "Q         Q         R${date}    ORG_${date}" >> fort.10
  echo "P         P         R${date}    ORG_${date}" >> fort.10
  echo "T         T         R${date}    ORG_${date}" >> fort.10
  echo "PV_FILT   PV_AIM    R${date}    MOD_${date}" >> fort.10
  echo "U         U         R${date}    MOD_${date}" >> fort.10
  echo "V         V         R${date}    MOD_${date}" >> fort.10
  echo "Q         Q         R${date}    MOD_${date}" >> fort.10
  echo "P         P         R${date}    MOD_${date}" >> fort.10
  echo "T         T         R${date}    MOD_${date}" >> fort.10
  echo "NSQ       NSQ       R${date}    MOD_${date}" >> fort.10
  echo "RHO       RHO       R${date}    MOD_${date}" >> fort.10
  echo "TH        TH        R${date}    MOD_${date}" >> fort.10
  echo "PV_ANOM   PV        R${date}    ANO_${date}" >> fort.10
  echo "TH_ANOM   TH        R${date}    ANO_${date}" >> fort.10
  echo "UU_ANOM   U         R${date}    ANO_${date}" >> fort.10
  echo "VV_ANOM   V         R${date}    ANO_${date}" >> fort.10
  echo "ORO       ORO       R${date}    REF_${date}" >> fort.10
  echo "X         X         R${date}    REF_${date}" >> fort.10
  echo "Y         Y         R${date}    REF_${date}" >> fort.10
  echo "LAT       LAT       R${date}    REF_${date}" >> fort.10
  echo "LON       LON       R${date}    REF_${date}" >> fort.10
  echo "CORIOL    CORIOL    R${date}    REF_${date}" >> fort.10
  ${bdir}/prep/cutnetcdf
  \rm -f fort.10
endif

# Step 6: Add the reference profile
if ( ${step} == "prep6" | ${step} == "prep" ) then
  \rm -f fort.10
  echo "MOD_${date}"     >! fort.10
  echo "REF_${date}"     >> fort.10
  ${bdir}/prep/ref_profile
  \rm -f fort.10
endif

# Step 7: Add coastlines to REF file
if ( ${step} == "prep7" | ${step} == "prep" ) then
  \rm -f fort.10
  echo \"REF_${date}\"                         >! fort.10
  echo \"${coastfile}\"                        >> fort.10
  ${bdir}/inversion.perl ${parafile} coastline >> fort.10
  ${bdir}/prep/coastline
  \rm -f fort.10
endif

# Step 8: Move the files to the run directory
if ( ${step} == "prep8" | ${step} == "prep" ) then
  \mv MOD_${date} MOD_${date}_cst ${rdir}
  \mv ORG_${date} ORG_${date}_cst ${rdir}
  \mv ANO_${date} ANO_${date}_cst ${rdir}
  \mv REF_${date} REF_${date}_cst ${rdir}
  \rm -f fort.10
endif

# ---------------------------------------------------------------------------
# Inversion
# ---------------------------------------------------------------------------

# Change to data directory
cd ${rdir}

# Start loop
set count=0
loop:

# Step 1: Add NSQ, TH, RHO, and PV to MOD file, take grid from REF file
if ( ${step} == "pvin1" | ${step} == "pvin" ) then
  echo "NSQ"           >! fort.10
  echo "MOD_${date}"   >> fort.10
  echo "REF_${date}"   >> fort.10
  echo "5          "   >> fort.10
  ${bdir}/pvin/z2s
  echo "RHO"           >! fort.10
  echo "MOD_${date}"   >> fort.10
  echo "REF_${date}"   >> fort.10
  echo "5          "   >> fort.10
  ${bdir}/pvin/z2s
  echo "TH"            >! fort.10
  echo "MOD_${date}"   >> fort.10
  echo "REF_${date}"   >> fort.10
  echo "5          "   >> fort.10
  ${bdir}/pvin/z2s
  echo "PV"            >! fort.10
  echo "MOD_${date}"   >> fort.10
  echo "REF_${date}"   >> fort.10
  echo "5          "   >> fort.10
  ${bdir}/pvin/z2s
  \rm -f fort.10
endif

# Step 2: Change Ertel’s PV anomaly into an anomaly of quasi-geostrophic PV
if ( ${step} == "pvin2" | ${step} == "pvin" ) then
  echo "MOD_${date}"   >! fort.10
  echo "REF_${date}"   >> fort.10
  echo "ANO_${date}"   >> fort.10
  ${bdir}/pvin/pv_to_qgpv
  \rm -f fort.10
endif

# Step 3: Inversion of quasi-geostrophic PV anomaly with Neumann boundary
if ( ${step} == "pvin3" | ${step} == "pvin" ) then
  echo "ANO_${date}"   >! fort.10
  echo "REF_${date}"   >> fort.10
  ${bdir}/pvin/inv_cart
  \rm -f fort.10
endif

# Step 4: Prepare the output of the inversion for next iteration step
if ( ${step} == "pvin4" | ${step} == "pvin" ) then
  \rm -f fort.10
  echo "MOD_${date}"                                >! fort.10
  echo "ANO_${date}"                                >> fort.10
  ${bdir}/inversion.perl ${parafile} prep_iteration >> fort.10
  ${bdir}/pvin/prep_iteration
  \rm -f fort.10
endif

# Step 5: Keep iterative steps if save flag is set
if ( ${step} == "pvin5" | ${step} == "pvin" ) then
  if ( "${save}" == "yes" ) then
    set pre=’’
    if ( ${count} < 10 ) then
      set pre='0'
    endif
    \cp MOD_${date} MOD_${date}_${pre}${count}
    \cp ANO_${date} ANO_${date}_${pre}${count}
  endif
endif

# End loop for iterations
if ( ${step} == "pvin" ) then
  @ count = ${count} + 1
  if ( ${count} < ${nofiter} ) goto loop
endif

# ---------------------------------------------------------------------------
# Postprocessing
# ---------------------------------------------------------------------------

# Change to data directory
cd ${odir}

# Step 1: Copy needed files from input and run directory
if ( ${step} == "post1" | ${step} == "post" ) then
    ln -sf ${rdir}/MOD_${date} ${rdir}/MOD_${date}_cst .
    \cp ${idir}/P${date} ${idir}/ml_cst .
endif

# Step 2:  Rotate from quasi-cartesian co-ordinate frame to lat/lon system
if ( ${step} == "post2" | ${step} == "post" ) then
    echo "MOD_${date}"                                >! fort.10
    echo "GEO_${date}"                                >> fort.10
    ${bdir}/inversion.perl ${parafile} rotate_lalo    >> fort.10
    echo "3"                                          >> fort.10
    echo "T"                                          >> fort.10
    echo "U.V"                                        >> fort.10
    echo "P"                                          >> fort.10
    ${bdir}/post/rotate_lalo
    \rm -f fort.10
endif

# Step 3: Bring modified fields back to P file
if ( ${step} == "post3" | ${step} == "post" ) then
    \rm -f fort.10
    echo "P${date}"                                   >! fort.10
    echo "GEO_${date}"                                >> fort.10
    ${bdir}/inversion.perl ${parafile} add2p          >> fort.10
    ${bdir}/post/add2p
    \rm -f fort.10
endif

# Step 4: Calculate S file with PV and TH
if ( ${step} == "post4" | ${step} == "post" ) then
    \rm -f S${date}
    p2s P${date} TH PV
endif

# Step 5: Make clean
if ( ${step} == "post5" | ${step} == "post" ) then
    \rm -f MOD_${date} MOD_${date}_cst
    \rm -f GEO_${date} GEO_${date}_cst
endif

# ----------------------------------------------------------------------------
# Diagnostic Tools
# ----------------------------------------------------------------------------

# Change to run directory
cd ${rdir}

# Step 1: Check the consistency of the boundary conditions (diag1)
if ( ${step} == "diag1"  ) then
    \rm -f fort.10
    echo "ANO_${date}"  >! fort.10
    echo "REF_${date}"  >> fort.10
    ${bdir}/diag/check_boundcon
    \rm -f fort.10
endif

# Step 2: Calculate the quasi-geostrophic PV (diag2 [ORG|MOD|ANO]
if ( ${step} == "diag2"  ) then
    \rm -f fort.10
    echo "${file1}_${date}"  >! fort.10
    echo "REF_${date}"       >> fort.10
    ${bdir}/diag/calc_qgpv
    \rm -f fort.10
endif

# Step 3: Get difference between two files (diag3 [ORG|MOD|ANO] - [ORG|MOD|ANO])
if ( ${step} == "diag3"  ) then
    \rm -f fort.10
    echo "${file1}_${date}"  >! fort.10
    echo "${file2}_${date}"  >> fort.10
    echo "DIA_${date}"       >> fort.10
    ${bdir}/diag/difference
    \rm -f fort.10
endif
```

9.3 Controlling Parameter File

The inversion problem is specified in a parameter file. This file is set as a parameter in the controlling Linux Shell script (see Appendix 9.2) and is often the only file which has to be adapted.

```
BEGIN DATA
  DATE    = 20060116_18;                                ! Date for case study
  INP_DIR = /lhome/sprenger/PV_Inversion_Tool/real/inp;  ! Input directory
  RUN_DIR = /lhome/sprenger/PV_Inversion_Tool/real/run;  ! Run directory
  OUT_DIR = /lhome/sprenger/PV_Inversion_Tool/real/out;  ! Output directory
END DATA

BEGIN GRID
  GEO_XMIN = -180.;     ! Geographical grid in zonal direction
  GEO_NX   =  361 ;
  GEO_DX   =    1.;
  GEO_YMIN =    0.;     ! Geographical grid in meridional direction
  GEO_NY   =   91 ;
  GEO_DY   =    1.;
  GEO_ZMIN =    0.;     ! Vertical levels ( ZMIN and DZ in [m] )
  GEO_NZ   =  125 ;
  GEO_DZ   =  200.;
  ROT_NX   =  250 ;     ! Rotated grid ( DX and DY in [deg] )
  ROT_NY   =  250 ;
  ROT_DX   =  0.25;
  ROT_DY   =  0.25;
  CLON     =  -65.;     ! Longitude, latitude [deg] and angle of rotated grid
  CLAT     =   45.;
  CROT     =    0.;
END GRID

BEGIN ANOMALY
  BOX_XMIN = -1200.;    ! Box where to apply the digital filter
  BOX_XMAX =  1200.;
  BOX_YMIN = -1200.;
  BOX_YMAX =  1200.;
  BOX_ZMIN =  2000.;
  BOX_ZMAX = 10000.;
  NFILTER  =     5 ;    ! Number of filter iterations;
  BOUND_XY =   500.;    ! Transition zone for horizontal boundaries ( in [km] )
  BOUND_Z  =   500.;    ! Transition zone for vertical boundaries ( in [m] )
END ANOMALY

BEGIN NUMERICS
  ALPHA    =    0.5;    ! Adjustment factor after one iteration step
  NOFITER  =      6;    ! Number of "external" iterations
  SAVEITER =     no;    ! Flag whether to save iteration steps
END NUMERICS
```

9.4 List of Fortran, Perl and Linux programs

The main tasks of the PV inversion are handled by Fortran programs. The following table contains a complete list of these programs with some details. The first column gives the name of the program, the second column the section (pre-processing, inversion, post-processing, diagnostics) where this program is needed, the third column the authors (in order of their contribution), and the fourth column is a short description of the code. In the table, Fortran programs are marked with the appendix ".f", Perl programs with the appendix ".perl", and Linux Shell scripts with the appendix ".sh".

Name of program	**Section**	**Author**	**Short description**
p2z.f	prep	Michael Sprenger	Interpolation onto height levels
rotate_grid.f	prep	Michael Sprenger	Rotation to qusi-cartesian system
z2s.f	prep, pvin	Michael Sprenger	Calculation of secondary fields
def_anomaly.f	prep	Michael Sprenger	Definition of modified Ertel's PV
cutnetcdf.f	prep	Michael Sprenger Heini Wernli	Splitting of netcdf files
ref_profile.f	prep	Michael Sprenger Olivia Martius	Definition of reference profile
coastline.f	prep	Michael Sprenger	Add coastlines to reference file
pv_to_qgpv.f	pvin	Michael Sprenger	Calculate quasi-geostrophic PV
inv_cart.f	pvin	Rene Fehlmann Michael Sprenger	Inversion of quasi-geostrophic PV
prep_iteration.f	pvin	Michael Sprenger Rene Fehlmann	Prepare next iteration step
rotate_lalo.f	post	Michael Sprenger	Rotate to geographical grid
add2p.f	post	Michael Sprenger	Interpolation to hybrid grid
p2s.f	prep, post	Heini Wernli	Secondary fields on hybrid grid
check_boundcon.f	diag	Rene Fehlmann Michael Sprenger	Consistency check for boundaries
calc_qgpv.f	diag	Sebastien Dirren Michael Sprenger	Convergence check for inversion
difference.f	diag	Michael Sprenger	Difference of two netcdf files
inversion.sh		Michael Sprenger	Master Linux Shell script
inversion.perl		Michael Sprenger	Extraction of parameters