

Equivalent Latitude Computation Using Regions of Interest (ROI)

Juan A. Añel^{1,2*}, Douglas R. Allen³, Guadalupe Sáenz^{1,4}, Luis Gimeno², Laura de la Torre²

1 Smith School of Enterprise and the Environment, University of Oxford, Oxford, United Kingdom, **2** EPhyLab, Faculdade de Ciencias, Universidade de Vigo, Ourense, Spain, **3** Naval Research Laboratory, Washington, District of Columbia, United States of America, **4** Departamento de Física, Facultad de Ciencias, Universidad de Extremadura, Badajoz, Spain

Abstract

This paper introduces a novel algorithm to compute equivalent latitude by applying regions of interest (ROI). The technique is illustrated using code written in Interactive Data Language (IDL). The ROI method is compared with the “piecewise-constant” method, the approach commonly used in atmospheric sciences, using global fields of atmospheric potential vorticity. The ROI method is considerably more accurate and computationally faster than the piecewise-constant method, and it also works well with irregular grids. Both the ROI and piecewise-constant IDL codes for equivalent latitude are included as a useful reference for the research community.

Citation: Añel JA, Allen DR, Sáenz G, Gimeno L, de la Torre L (2013) Equivalent Latitude Computation Using Regions of Interest (ROI). PLoS ONE 8(9): e72970. doi:10.1371/journal.pone.0072970

Editor: Matteo Convertino, University of Florida, United States of America

Received: March 15, 2013; **Accepted:** July 16, 2013; **Published:** September 25, 2013

This is an open-access article, free of all copyright, and may be freely reproduced, distributed, transmitted, modified, built upon, or otherwise used by anyone for any lawful purpose. The work is made available under the Creative Commons CC0 public domain dedication.

Funding: This work was partially supported by the TRODIM (CGL2007-65891-C05-01) and ExCIRs projects (CGL2011-24826) funded by the Spanish Ministry of Science and Innovation, Spanish Ministry of Economy and Competitiveness and by the Xunta de Galicia under “Programa de Consolidación e Estruturación de Unidades de Investigación (Grupos de Referencia Competitiva)” funded by European Regional Development Fund. JAA research is partially supported by the University of Oxford and the Engineering and Physical Sciences Research Council of the United Kingdom. Work at the Naval Research Lab is sponsored by the United States Office of Naval Research. Part of the simulations performed were carried out in the Centro de Supercomputación de Galicia through a “Reto Computacional 2009” project. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: JAA is Academic Editor of PLOS ONE. This does not alter the authors’ adherence to all the PLOS ONE policies on sharing data and materials.

* E-mail: juan.anel@smithschool.ox.ac.uk

Introduction

In atmospheric sciences Equivalent Latitude (ϕ_e) is a Lagrangian coordinate defined as the geographical latitude enclosing the same area as the isoline of a given atmospheric field on a 2D (longitude by latitude) surface [1,2]. It is computed as:

$$\phi_e = \arcsin[1 - (A/(2\pi R^2))]$$

R : radius of the sphere (e.g. Earth’s radius).

Therefore computing ϕ_e is simply the problem of computing the area A enclosed by the isoline of the studied property, which is essentially the determination of a function from a subset of data points.

One common technique to compute ϕ_e works by assuming the value of the atmospheric field to be constant within each grid cell (hereafter the ‘piecewise-constant’ method). To determine ϕ_e for a given threshold value of the field, you simply sum the areas for all grid cells with field values less than the threshold value.

$$A(q_0) = \sum_{i=0}^n A_n(q < q_0)$$

q : value of the field. q_0 : value to be evaluated. n : grid cell.

The piecewise-constant approach is only first-order accurate, not allowing the atmospheric field to vary across the cell. Increased accuracy could be attained by higher-order approximations for the field variation (e.g., linear, parabolic, cubic interpolation). In this

paper, we develop a higher-order solution to the problem of calculating ϕ_e by using contour mapping techniques. This approach, known also as regions of interest (ROI), is a well known concept in medical imaging and is also used for geographical information systems. The general approach involves defining a region from a field (up to 4D), which is selected for a posteriori analysis. That is, given a field or sample, a subsample is selected which meets certain properties and that is the region in which we are interested for further analysis. Here we limit the field variation to the surface of a sphere (2D). The use of ROI based on interpolated contours is more accurate than the piecewise-constant method, since it is a better approximation to the real function (exact area). This is illustrated in figure 1, which maps the areas enclosed by a threshold value of potential vorticity (PV) using both the piecewise-constant and ROI methods. This is an advantage when considering, for example, small isolated and closed contours, since the size of the grid becomes more important in order to take them into account or not. For example, the ROI method is more accurate to assess the real area of a closed contour when it is slightly smaller or bigger than a cell or subset of cells of the grid.

In this paper we explore the ROI technique for the computation of ϕ_e and compare the performance and accuracy with that obtained using the piecewise-constant method. For the purposes of this paper the ROI will be a 2D field of atmospheric PV [3], usually measured in PV units (PVU), where $1PVU = 10^{-6} \frac{K \cdot m^2}{kg \cdot s}$. The reason for this choice is that computation of ϕ_e from PV is

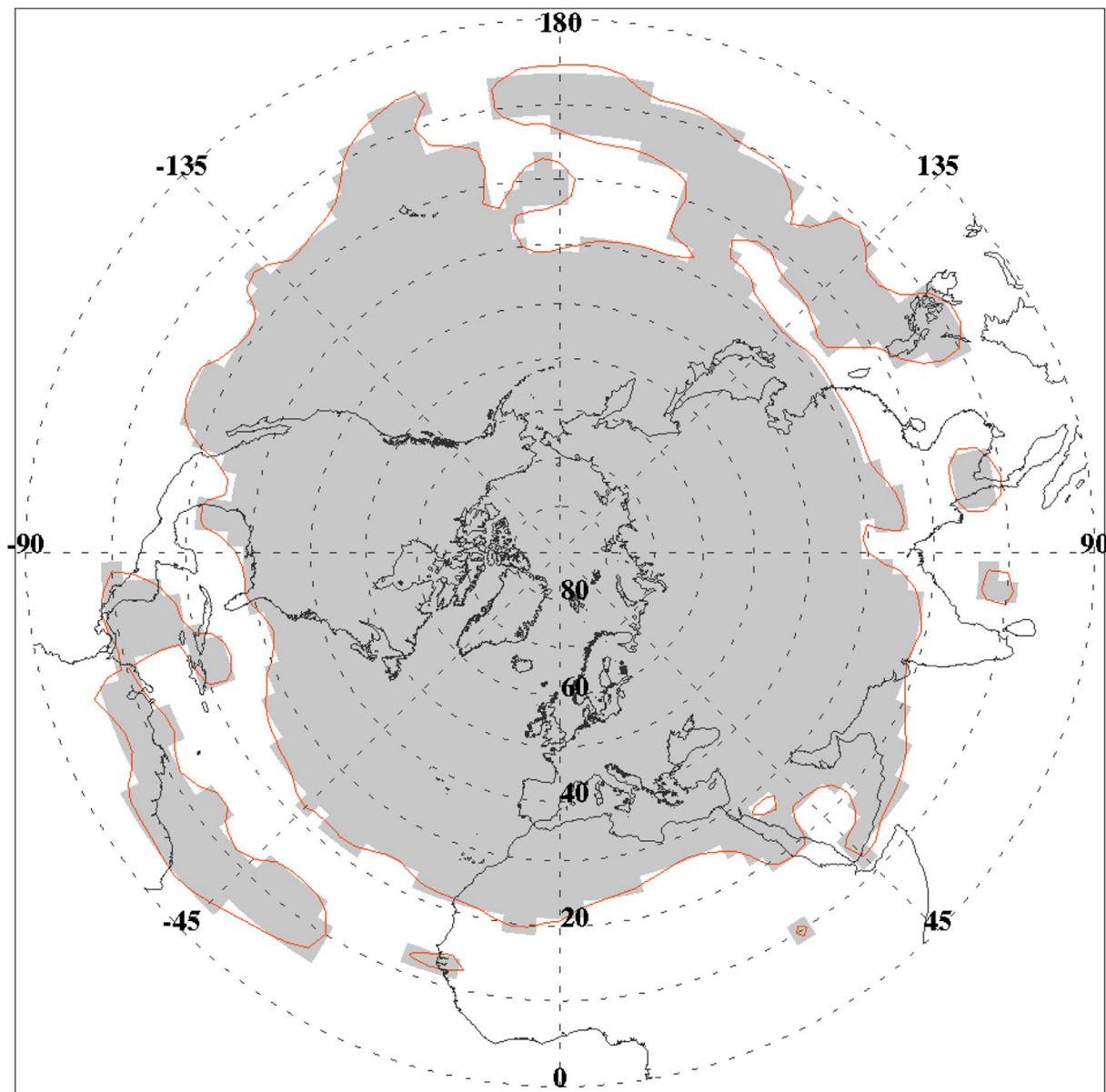


Figure 1. Comparison between the area used for the computation of ϕ_p by the piecewise-constant method (gray surface) and the ROI technique (surface inside red contours) for a grid size of $2.5^\circ \times 2.5^\circ$. The case corresponds to 1 January 1990 at 00 UTC from NCEP1 for 2 PVU on an isentropic surface of 380 K.
doi:10.1371/journal.pone.0072970.g001

commonly used in atmospheric sciences, applied for example to the study of the stratospheric polar vortex [4,5], and the goal for which we originally developed this approach. The differences between both techniques and, explained in a basic way, the steps that we follow are:

- piecewise-constant technique (traditional):
 1. PV is assumed to be constant within each grid cell;
 2. compute the area of each cell which value is less than the PV threshold that we are interested in;
 3. sum the area of the cells from step 2
- ROI technique:

1. PV is 'not' constant within each grid cell;
2. make the contour mapping function 'draw' (interpolate) the isolines of PV;
3. compute the area enclosed by the isoline corresponding to the PV threshold value that we are interested in;

We developed the ROI code using the Interactive Data Language (IDL), a programming language extensively used for research in atmospheric sciences [6], and compare with piecewise-constant code also written in IDL. In this way we put the focus on the code and the technique, and do not take into account any dependence on programming language, corresponding libraries, or dependencies on hardware.

PIECEWISE-CONSTANT

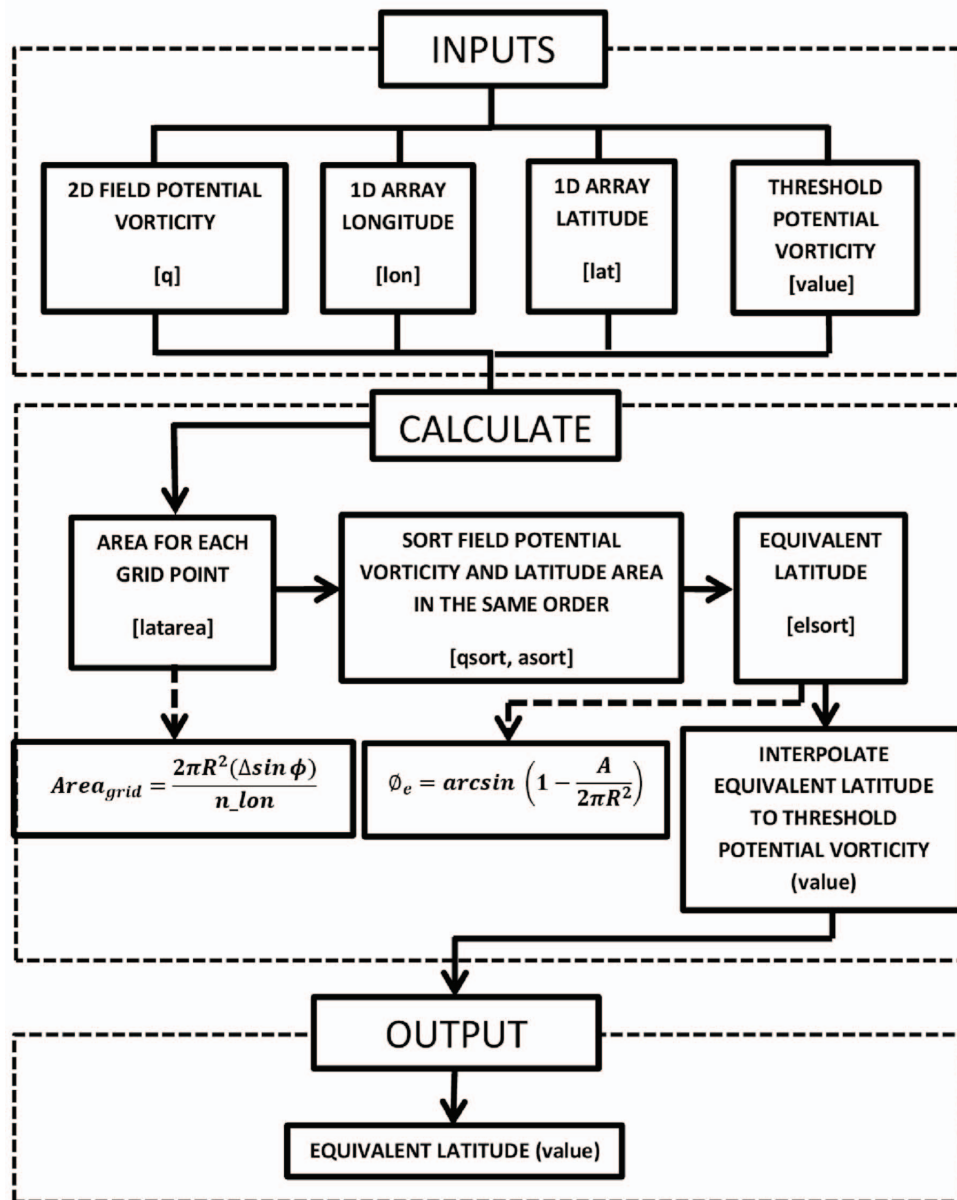


Figure 2. Flow diagram corresponding to the piecewise-constant code.
doi:10.1371/journal.pone.0072970.g002

The following section describes the data used followed by the design and implementation of the ROI routine, addressing the ways in which we have solved the shortcomings of the computer language and subroutines affecting our implementation. This is followed by a results section that details the accuracy and performance of the ROI method relative to the piecewise-constant method. To conclude, we briefly discuss several ways of improving the solution here proposed, including using another programming languages and language interpreters.

Materials and Methods

Previous code and data

Routines to compute ϕ_e are available through internet, e.g. the ones using the NCAR Command Language (http://www.ncl.ucar.edu/Applications/equiv_lat.shtml) or pascal (<http://www.bodekscientific.com/how-do-i/calculate-equivalent-latitude>). To check our new methodology against the more traditional one, we use the code (not previously published) supplied as (file S1) that applies the piecewise-constant method. This code takes as input a global gridded field with supplied latitude and longitude grids

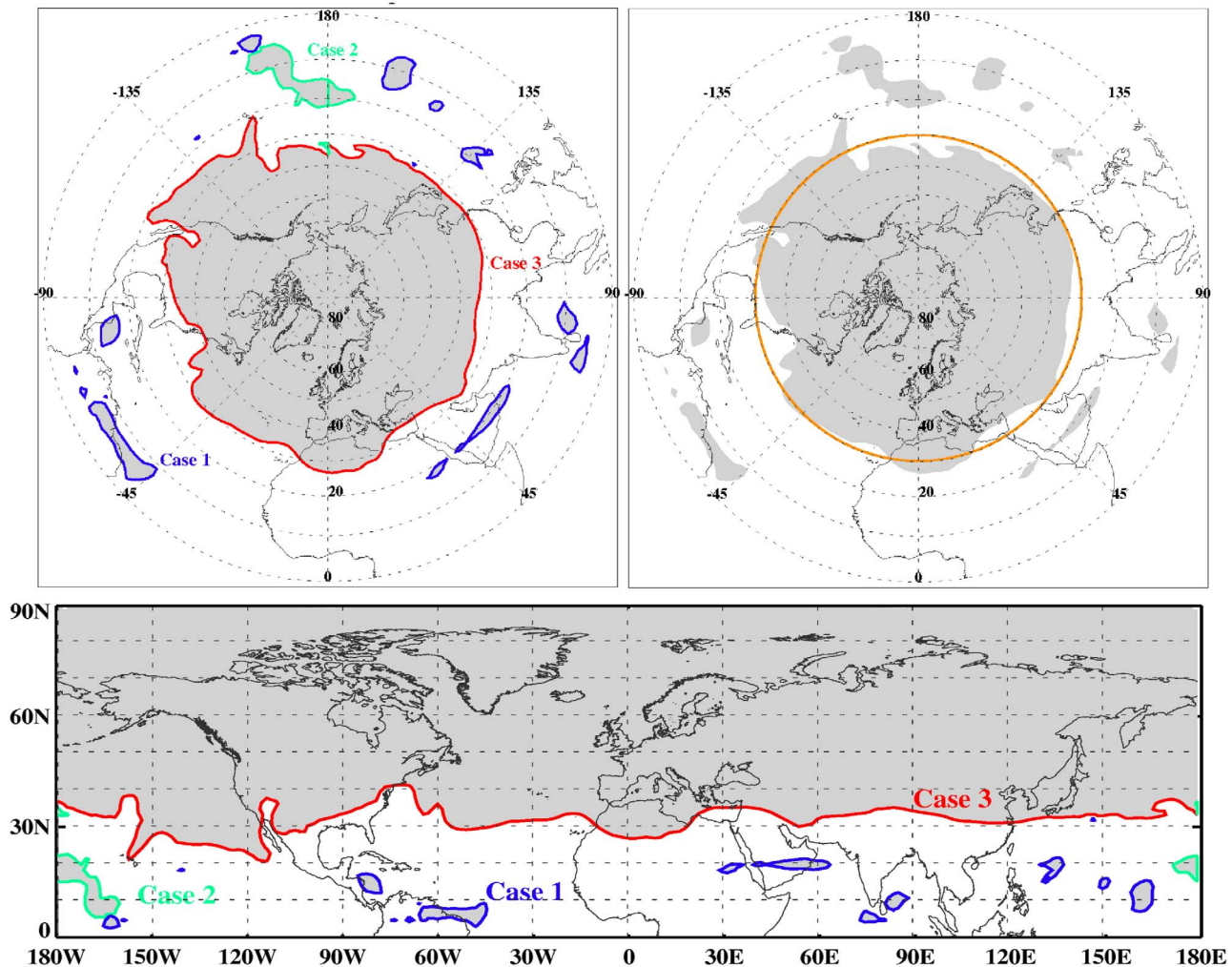


Figure 3. Upper left: global isosurface of 4 PVU on 1 January 1990 from ERA-Interim reanalysis data. Upper right: ϕ_e corresponding to the area covered by the 4 PVU isosurface (in this case 30.41°). Lower panel: cylindrical projection corresponding to the upper left panel. doi:10.1371/journal.pone.0072970.g003

along with a desired threshold value of the field at which ϕ_e is calculated. The grid cells are sorted by the value of the field in each cell, and ϕ_e is calculated for every grid point on the globe. ϕ_e is then interpolated to the desired threshold value and is output to the user. A flow diagram is provided in figure 2.

We have tested our new ROI routine using fields of PV computed for five sets of reanalyzed data (NCEP/NCAR (hereafter NCEP1) [7], NCEP/DOE (hereafter NCEP2) [8], JRA-25 [9], ERA-Interim [10], MERRA [11] and one chemistry-climate model (WACCM3 [12]). Additional information about the computation of the fields of PV, the datasets and results obtained with this routine can be obtained from Añel et al. [13].

Design and Implementation

The method here proposed is based on a simple idea: given the field that we want to study (PV), we just need to plot it using a global contour map and then to make use of the IDL tools to detect what parts of the field meet a criterion (in our case that the value is below a given PVU threshold). Then the area that meets the criterion is automatically computed by the ROI detection tools available in IDL and we just need to apply the corresponding equation to get ϕ_e .

The first problem to be solved for the implementation of the computation of the area using ROI is to determine the configuration of a given isoline of a global field projected on a sphere (the planet). Depicting the field with a cylindrical projection, an isosurface of PV can show three different configurations, directly related to the shape of the isolines over this projection (see figure 3):

- case 1: contours closed for any region of the planet, the easiest problem as it requires just applying the computation of the area using the available routines for ROI;
- case 2: contours going through the meridian 180 and not enclosing the pole are not closed in a cylindrical projection, so they need to be split in two different contours (one in each hemisphere) and the area has to be computed as the sum of the two different areas;
- case 3: the contours around the poles are not closed in a cylindrical projection. They go from the the lowest to the highest value in longitude. They are closed using a large number of points (resolution of 0.1°) at the latitude near the pole (North or South) and then the enclosing area is calculated using routines for ROI.

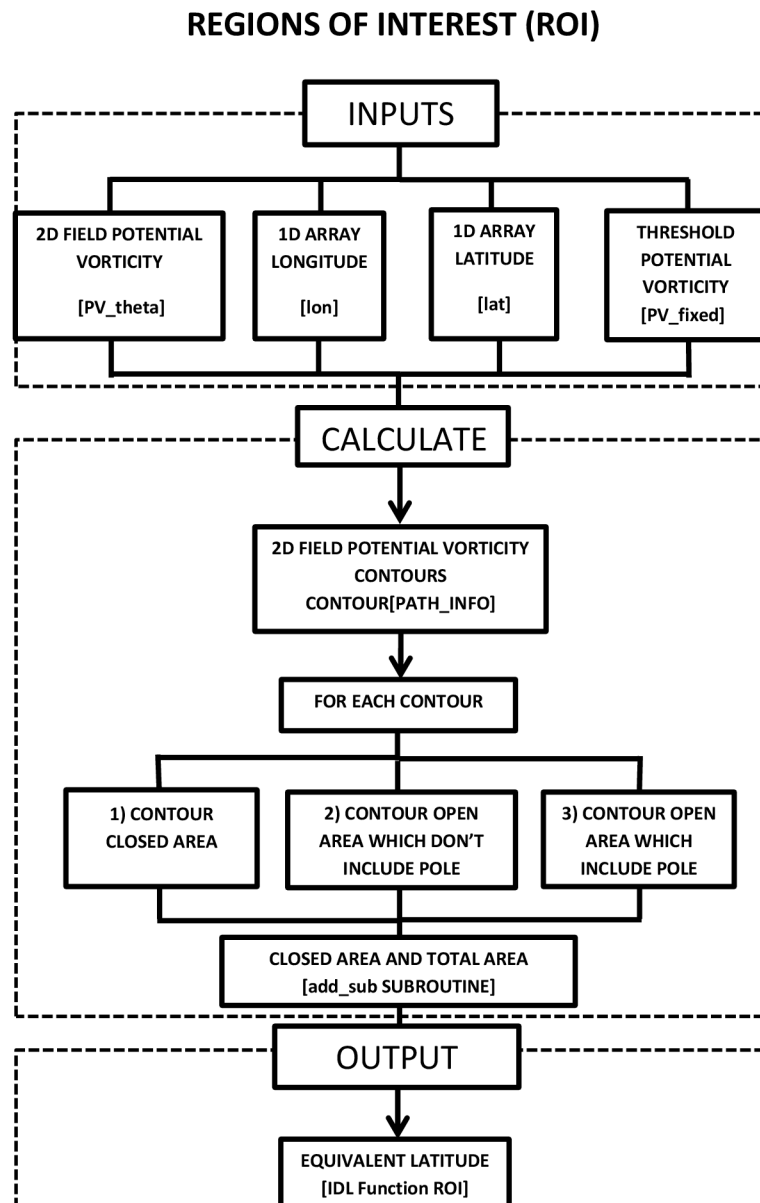


Figure 4. Flow diagram corresponding to the ROI code.
doi:10.1371/journal.pone.0072970.g004

Another problem addressed during the design of this code was that the computation of ROI in IDL gives erroneous negative values if one line of the isolines of PV drawn by the function 'contour' crosses itself. This happens mainly for latitudes near to the poles in rectangular grids where cell size and isolines get narrow and the IDL ROI implementation fails for unknown reasons. We solve this problem by increasing the latitudinal resolution in these regions, studying the ROI with a resolution of 0.1° to avoid the problem of crossing isolines.

It was also necessary to create new matrix of data in order to close the contours not closed in the cylindrical projection, just because for example they cut the meridian zero as we mentioned before. An additional problem was discovered for the computation of cosines for 90° and -90° . IDL presents a bug and the results are negative values instead of zero. To avoid this problem we close the contours in a latitude of 89.99° instead of 90° .

The IDLanROI class is used to determine the regions enclosing specified threshold values of PV, and it is combined with the application of the 'path_info' keyword in the Contour procedure. Also we had to take into account the difference between computation of the geometric area (the real value of the mathematical calculation) and the masked region (value from the pixels in the displayed area) when using ROI in IDL. Therefore we applied the ComputeGeometry function to obtain the geometric area. ComputeGeometry relies on the Contour procedure to get the function.

Finally, the total area to be passed to the routine to compute the equivalent latitude is considered as the addition of the values for the different partial areas obtained. We use -9999.99 for the total area when (according to 'path_info') there is not a corresponding contour for the threshold PV value that we try to evaluate.

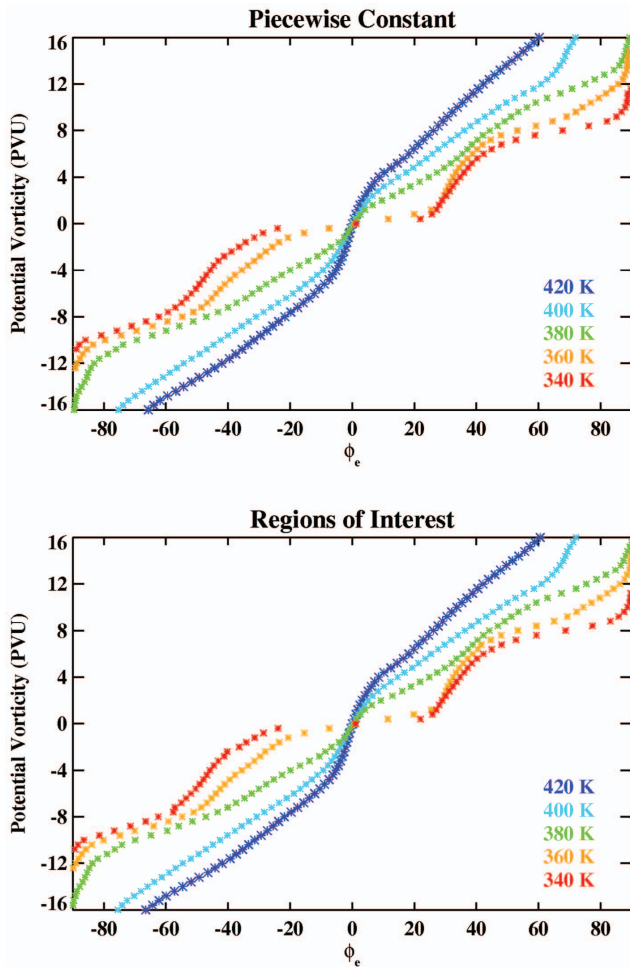


Figure 5. Comparison of PVU values for several isentropic surfaces and corresponding to different ϕ_e . Values computed using the piecewise-constant and ROI methods on 1 January 1990 from ERA-Interim data.
doi:10.1371/journal.pone.0072970.g005

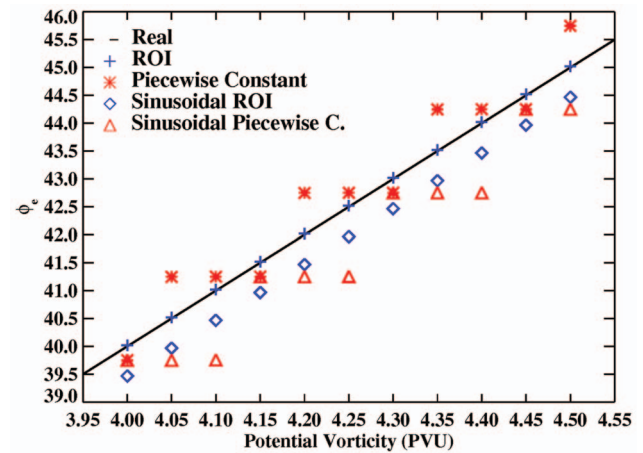


Figure 7. Accuracy of the ROI technique and the piecewise-constant method for different values of PV and latitude and for the two analytical functions.
doi:10.1371/journal.pone.0072970.g007

The process is explained in a flow diagram in the figure 4 and the ROI routine is included in file S2.

Results

Precision

First, we checked that the ROI methodology produces reasonable results for ϕ_e at all latitudes and for different isentropic surfaces. Figure 5 shows ϕ_e calculated from ERA-Interim PV data on 1 January 1990 using both the piecewise-constant and ROI methods from 340 to 420 K potential temperature. It is clear that the two methods produce very similar results, although there are some differences (e.g., near 40°S at 340 K).

Next, in order to check the accuracy of both methods we computed ϕ_e for two different cases in which the PV field is not based on real data, but rather on an analytic function ($PV = \text{latitude}/10$ and $PV = ((\text{latitude} * \sin(\text{longitude}))/10)$), so that we know the exact relationship between ϕ_e and PV. For the

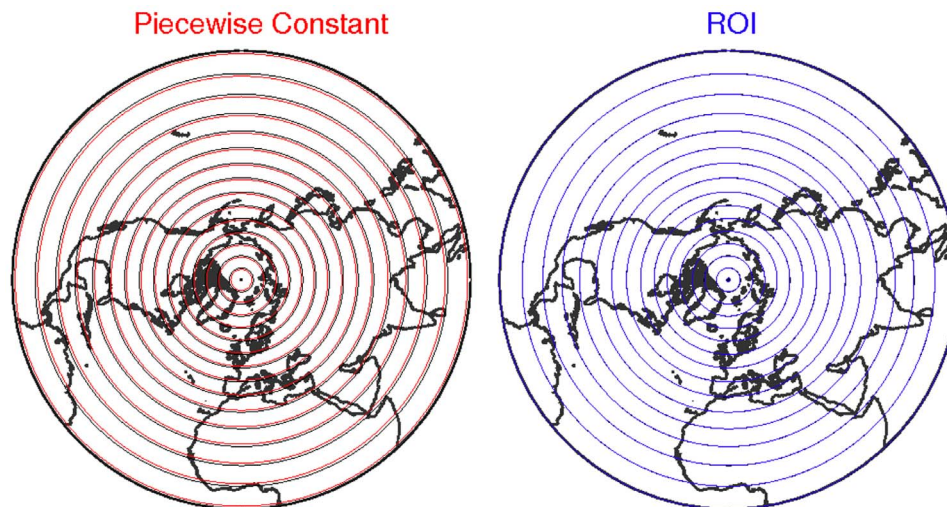


Figure 6. Red and blue circles: ϕ_e circles computed applying the piecewise-constant method and the ROI technique, respectively, from an ideal field of PV represented on a $2.5^\circ \times 2.5^\circ$ grid. Black circles: real values of ϕ_e for the PV field studied (not seen for the case of the ROI technique as the blue ones are superimposed as they match perfectly the ideal field).
doi:10.1371/journal.pone.0072970.g006

Table 1. Precision dependence with technique and grid size in degrees for the PV field given by the first analytical function.

	ROI	piecewise-constant
1,5°×1,5°	0.030±0.025	0.415±0.237
3°×3°	0.027±0.015	0.659±0.384
4,5°×4,5°	0.029±0.020	1.145±0.660
6°×6°	0.029±0.022	1.497±0.892

Mean absolute values computed as the difference between the ϕ_e given by each technique and the real one. Average and standard deviation resulting of the computation for 50 different PV isolines (25 in the Northern Hemisphere and 25 in the Southern Hemisphere).
doi:10.1371/journal.pone.0072970.t001

Table 2. Precision dependence with technique and grid size in degrees for the PV field given by the second analytical function.

	ROI	piecewise-constant
1,5°×1,5°	0.540±0.010	0.746±0.415
3°×3°	0.877±0.010	0.920±0.676
4,5°×4,5°	0.801±0.010	1.255±0.836
6°×6°	0.801±0.010	1.563±1.022

Mean absolute values computed as the difference between the ϕ_e given by each technique and the real one. Average and standard deviation computed following the same method that for table 1.
doi:10.1371/journal.pone.0072970.t002

first case, the PV isolines form perfect circles around the pole and ϕ_e is simply equal to PV*10. In the second case we simply obtain a wavy field, but the relationship between ϕ_e and PV is the same. The PV fields generated in this way are then used as input for both methods and the results are compared. Figure 6 shows the results using the piecewise-constant and ROI routines for the first method and figure 7 for both of them. Also the tables 1 and 2 quantify the variation of the error for different grid sizes. As it can be seen for a grid of 1.5°×1.5° the piecewise-constant method shows a bias respect to the corresponding ϕ_e around a half degree for the first analytic function and 0.75 for the second one. This is greater than the bias for the ROI technique. Moreover the accuracy of the piecewise-constant method depends much more on the size of the grid (see both tables and figure 7).

Performance

As we can see in tables 3 and 4 the ROI technique computed in this way is faster than the piecewise-constant method for a grid of 1.5°×1.5°, and moreover the finer the grid the greater the advantage in computational time for the ROI technique. This is something to have into account as in the future we probably will be moving to smaller grid sizes. This result is probably because the piecewise-constant method relies for the evaluation of the values of the function in each point on several FOR loops, which makes especially slow any computation in IDL. However the ROI technique simply computes the function once.

The test was performed on a computer with the following hardware: Intel Quad 2 Core with cpu frequency of 2.66 GHz and 3 MB of cache memory, 3 GB of RAM memory. The

Table 3. Computation time dependence with technique and grid size in 10⁻³ seconds for the PV field given by the first analytical function.

	ROI	piecewise-constant
1,5°×1,5°	4.09±0.16	36.97±0.25
3°×3°	1.33±0.18	9.16±0.03
4,5°×4,5°	0.83±0.18	4.18±0.02
6°×6°	0.66±0.18	2.39±0.01

Average and standard deviation computed following the same method that for table 1.
doi:10.1371/journal.pone.0072970.t003

Table 4. Computation time dependence with technique and grid size in 10⁻³ seconds for the PV field given by the second analytical function.

	ROI	piecewise-constant
1,5°×1,5°	4.13±0.17	43.13±0.79
3°×3°	1.35±0.17	10.57±0.10
4,5°×4,5°	0.85±0.17	4.82±0.04
6°×6°	0.68±0.17	3.16±0.03

Average and standard deviation computed following the same method that for table 1.
doi:10.1371/journal.pone.0072970.t004

operative system was Ubuntu GNU/Linux i686 with Linux kernel 2.6.32–37-generic. The IDL interpreter was provided by EXELIS and the version was 8.0.

Availability and Future Directions

The ROI code described in this paper is publicly available from the URL <https://github.com/eqlat/roi>. As expressed in the code of the routine the code is under the GNU General Public License (GPL) (<http://www.gnu.org/licenses>). In this way we comply with the criteria on public availability of code for research purposes [14]. To work with it, it is just necessary to get the code and to have an IDL interpreter.

We suggest as potential future directions for development the possibility of writing an equivalent ROI code in Python or Fortran, two programming languages very used in the geosciences community, but it should be had into account that very limited support for ROI computation seems to exist in both languages. To the authors knowledge just the NiPy library (<http://nipy.sourceforge.net>) for Python, currently under development, and some punctual implementations in Fortran. Also, currently is not possible to use the code based on ROI with GNUdata language (<http://gnudatalanguage.sourceforge.net>), a free IDL compiler, as computation of ROI is not implemented in it.

Supporting Information

File S1 Piecewise-constant code.

(PDF)

File S2 ROI code.

(PDF)

Acknowledgments

The authors would like to thank Andrew Gettelman and Douglas Kinnison from ACD, NCAR and Elías Berriochoa and Francisco Tugores from the Universidade de Vigo for providing useful code and comments.

References

1. Lary D, Chipperfield M, Pyle J, Norton W, Riishøjgaard L (1995) Three-dimensional tracer initialization and general diagnostics using equivalent PV latitude-potential-temperature coordinates. *Q J R Meteorol Soc* 121: 187–210.
2. Allen DR, Nakamura N (2003) Tracer equivalent latitude: A diagnostic tool for isentropic transport studies. *J Atmos Sci* 60: 287–304.
3. McIntyre ME (2003) Potential Vorticity. In: Holton JR, Curry J, Pyle J, editors, *Encyclopedia of Atmospheric Sciences*, Academic Press, volume 2, pp. 685–694.
4. Butchart N, Remsberg E (1986) The area of the stratospheric polar vortex as a diagnostic for tracer transport on an isentropic surface. *J Atmos Sci* 43: 1319–1339.
5. Schoeberl MR, Newman PA (2003) Polar Vortex. In: Holton JR, Curry J, Pyle J, editors, *Encyclopedia of Atmospheric Sciences*, Academic Press, volume 3, pp. 1321–1328.
6. Bowman KP (2006) *An Introduction to programming with IDL: Interactive Data Language*. Elsevier Academic Press, 286 pp.
7. Kalnay E, Kanamitsu M, Kistler R, Collins W, Deaven D, et al. (1996) The NCEP/NCAR 40-Year Reanalysis Project. *Bull Amer Meteorol Soc* 77: 437–471.
8. Kanamitsu M, Ebisuzaki W, Woollen J, Yang S, Hnilo J, et al. (2002) NCEP-DOE AMIP-II Reanalysis (R-2). *Bull Amer Meteorol Soc* 83: 1631–1644.
9. Onogi K, Tsutsui J, Koide H, Sakamoto M, Kobayashi S, et al. (2007) The JRA-25 Reanalysis. *J Meteor Soc Japan* 85: 369–432.
10. Uppala SM, Kållberg PW, Simmons AJ, Andrae U, da Costa Bechtold V, et al. (2005) The ERA-40 re-analysis. *Q J R Meteorol Soc* 131: 2961–3012.
11. Rienecker MM, Suarez MJ, Gelaro R, Todling R, Bacmeister J, et al. (2011) MERRA: NASA's modern-era retrospective analysis for research and applications. *J Clim* 24: 3624–3648.
12. Garcia RR, Marsh DR, Kinnison DE, Boville BA, Sassi F (2007) Simulation of secular trends in the middle atmosphere, 1950–2003. *J Geophys Res* 112: D09301.
13. Añel JA, Gettelman A, Castanheira JM Tropical widening vs tropopause rising: a PV- θ perspective of the UTLS (submitted). *PLoS ONE*.
14. Añel JA (2011) The importance of reviewing the code. *Communications of the ACM* 54: 40–41.

Author Contributions

Conceived and designed the experiments: JAA LdlT DRA. Performed the experiments: JAA LdlT DRA GS. Analyzed the data: JAA LdlT DRA GS LG. Contributed reagents/materials/analysis tools: JAA LdlT DRA GS LG. Wrote the paper: JAA LdlT DRA GS LG.