How to calculate equivalent latitude

Greg Bodeker
2 November 2005

The purpose of this document is to tell you how to calculate equivalent latitude from a potential vorticity (PV) field on a prescribed isentropic level. For a more general discussion on the uses of equivalent latitude and extending it to define the vortex edge, read Nash, E.R., P.A. Newman, J.E. Rosenfield, and M.R. Schoeberl, An objective determination of the polar vortex using Ertel's potential vorticity, *Journal of Geophysical Research, 101 (D5)*, 9471-9478, 1996. But for now, this document tells you exactly how to do it.

All code in this document is pascal code (more specifically Borland Delphi). If you don't use Delphi, you should. If you want to stick with weird old languages like fortran, you should still be able to understand the code below.

Step 1

The calculation of equivalent latitude is done separately for each hemisphere. The first step is to find the maximum and minimum PV value in the hemisphere of interest. I extract the PV data at 2.5°×2.5° resolution but this isn't an absolute requirement. Here is my code that does that:

```
if AHemis=south then StartInd:=1 else StartInd:=37;
if AHemis=south then EndInd:=37 else EndInd:=73;
MaxPV:=-1E25;
MinPV:=1E25;
for I:=1 to 144 do
begin
  Long:=(I-1)*2.5;
  for J:=StartInd to EndInd do
  begin
    Lat:=-90+(J-1)*2.5;
    CurrentPV:=ExtractPVFromGrid(Long,Lat);
    if ((CurrentPV<NullValue) and (CurrentPV>MaxPV)) then MaxPV:=CurrentPV;
    if ((CurrentPV<NullValue) and (CurrentPV<MinPV)) then MinPV:=CurrentPV;
  end;
end;
```

Step 2

Calculate the geographic area enclosed by each of 100 PV isolines. Consider 100 PV values, from the minimum value to the maximum value, and for each PV value calculate the area enclosed by that PV isoline. This is easier than it sounds. First, make an array of 1..100 and then go through all of the PV values in the hemisphere of interest. For each PV value calculate the area of the cell that PV value represents. Find out which element in your array of 1..100 this PV value corresponds to and add this area to that element and to all lower order (1..N-1) elements (see code below for how this is done). Here is my code to do step 2:

```
PVStep:=(MaxPV-MinPV)/99.0;
for I:=1 to 100 do AreaTotal[I]:=0.0;
for I:=1 to 144 do
begin
  Long:=(I-1)*2.5;
  for J:=StartInd to EndInd do
  begin
    Lat:=-90+(J-1)*2.5;
```

Now, if I am right at the North pole, right at the South pole, or right at the equator, I have to do something different because at the poles the cell for which I extract the PV value has no area, and at the equator it's not quite in the hemisphere of interest. Recall that I extract the PV data in 2.5^{o} latitude steps. So near the poles the next closest value is extracted at 87.5^{o} latitude, representative of the cell whose latitude borders are 88.75^{o} and 86.25^{o}. So instead of extracting a PV value at the pole, I extract a PV value half way between 88.75^{o} and 90^{o} (i.e. at 89.375^{o}) and say that it is representative of the cell bordered by 88.75^{o} and 90.0^{o}. I then do something similar at the equator. Here is my code that does that:

```
      if ((AHemis=south) and (J=StartInd)) then Lat:=-89.375;
      if ((AHemis=south) and (J=EndInd)) then Lat:=-0.625;
      if ((AHemis=north) and (J=StartInd)) then Lat:=0.625;
      if ((AHemis=north) and (J=EndInd)) then Lat:=89.375;
      CellWidth:=1.25;
      if ((J=StartInd) or (J=EndInd)) then CellWidth:=0.625;
      CurrentPV:=ExtractPVFromGrid(Long,Lat);
```

The CellWidth value gets used below so I calculate it here. Now it might be that the PV field you are using doesn't span the range of latitudes for which you need to extract PV data e.g. it might not go all the way to the pole or all the way to the equator. To extract PV poleward of the highest latitude of available data, I just take the value at the highest possible latitude. To extract data equatorward of the lowest latitude of available data I do the following:

```
        if AHemis=north then CurrentPV:=MinPV;
        if AHemis=south then CurrentPV:=MaxPV;
```

Note that because PV is highest (most positive) at the North Pole and lowest (most negative) at the South Pole, in the southern hemisphere the maximum PV value is close to the equator while in the northern hemisphere the minimum PV value is closest to the equator.

So carrying on:

```
      if AHemis=north then ArrayIndex:=trunc((CurrentPV-MinPV)/PVStep)+1
      else ArrayIndex:=trunc((MaxPV-CurrentPV)/PVStep)+1;
      Area:=CalcArea(Lat-CellWidth,Lat+CellWidth);
```

The calc area function is defined as follows:

```
function CalcArea(SLat,NLat:double):double;
begin
  Result:=DegToRad(2.5)*abs(sin(DegToRad(SLat))-sin(DegToRad(NLat)));
end;
```

where DegToRad converts degrees to radians and abs calculates the absolute value. Then:

```
      if ArrayIndex=100 then Area:=0.0;
      TotalArea:=TotalArea+Area;
      for K:=ArrayIndex downto 1 do AreaTotal[K]:=AreaTotal[K]+Area;
    end; {End of loop through J.}
  end; {End of loop through I.}
```

The third last line above is where we add the cell's area to the array element and all lower order array elements. Now, because we have a line above that says `if ArrayIndex=100 then Area:=0.0` it may be that the total area for the hemisphere is not 2p. This is not a problem and we just normalize the areas as follows:

```
    if abs(TotalArea-(2*Pi))>0.0001 then
      for I:=1 to 100 do AreaTotal[I]:=AreaTotal[I]*2*Pi/TotalArea;
```

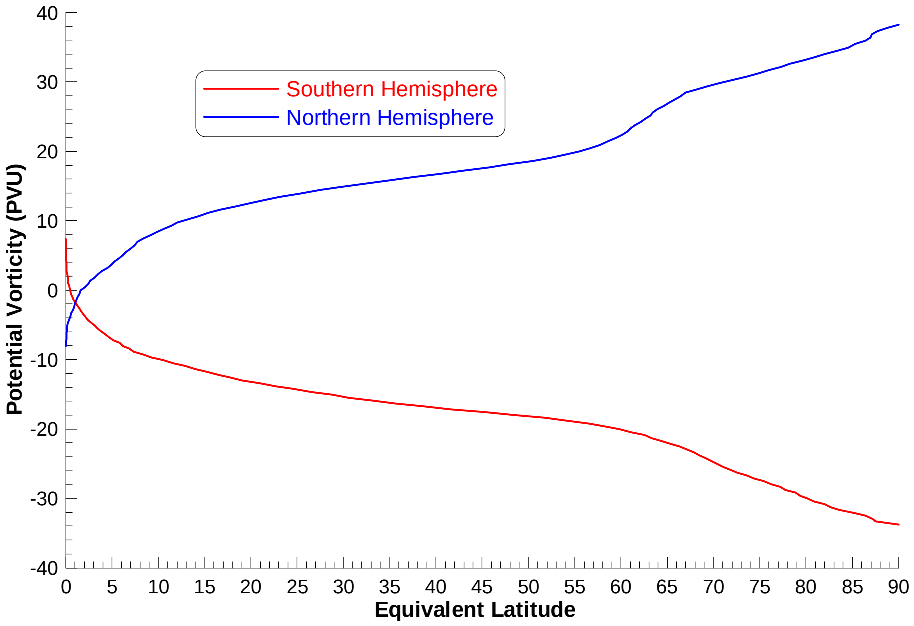

Figure 1: examples of PV vs. equivalent latitude profiles for the northern hemisphere and southern hemisphere.

So now we have our areas enclosed by 100 PV isolines. The next step is to calculate what true latitudes enclose the same areas as the areas for those 100 PV values (this is the equivalent latitude corresponding to each PV value). So:

Step 3

Here is the code I wrote to do this:

```
  for I:=1 to 100 do
  begin
    if AHemis=north then J:=I else J:=101-I;
    Eqlat[I]:=CalculateEquivLat(AreaTotal[J]);
  end;
```

where CalculateEquivLat is defined as follows:

```
function CalculateEquivLat(Area:double):double;
begin
  Result:=57.29577951*arcsin(1-(Area/6.283185307));
end;
```

So now you should have 100 PV values in your hemisphere of interest, and an equivalent latitude associated with each one. Examples of two such profiles are shown in Figure 1. So now, if for example you have an ozone measurement at a certain latitude and longitude and want to know what equivalent latitude that corresponds to, you go and get the PV value at that latitude and longitude and at the potential temperature at which your equivalent latitude calculation was done (of course this choice is somewhat arbitrary but people tend to choose 450K, 475K, or 550K), and, using the PV vs. equivalent latitude profile, determine the equivalent latitude associated with that PV value. Of course the PV vs. equivalent latitude profile is date and time specific and you need to use the one for the date and time of your measurement.

If anything I have written above is unclear, please email me at g.bodeker@niwa.co.nz and ask.