Potential vorticity diagnostics in NEMO

Jamie Mathews[1], Arnaud Czaja[1] and Adam Blaker[2]
[1] Imperial College London
[2] National Oceanography Centre, Southampton.

March 30, 2021

Contents

1 Motivation

We briefly report here on the calculation of potential vorticity (PV) in NEMO with an application to the ORCA12 configuration used in Mazzorchi et al. (2015). The rationale is to include a full calculation of the PV field in this model, including contributions from the relevant horizontal components of the vorticity vector. These are known to make significant contributions to the PV in the atmospheric and oceanic literature on frontal regions (Hoskins, 1982; Korty and Schneider, 2007; Thomas et al., 2013), and as the latter are becoming increasingly well resolved in routine simulations, it is important to include their contribution accurately.

This note is adapted in large part from the MSc thesis of Jamie Mathews at Imperial College. A Python code produced by Jamie is available on Github at XX and also upon request to Arnaud Czaja (a.czaja@imperial.ac.uk).

2 Potential vorticity conserved by NEMO

Denoting in situ density by ρ and potential density by σ, the Navier-Stokes equation leads, in absence of dissipative and diabatic processes, and neglecting the non linear effects associated with the equation of state of seawater (McDougall, 1988), to material conservation of the Ertel's potential vorticity Q given by:

$$Q \equiv -\frac{\boldsymbol{\zeta}_a . \boldsymbol{\nabla}\sigma}{\rho} \tag{1}$$

in which $\boldsymbol{\zeta}_a$ is the "absolute vorticity vector", the sum of the planetary vorticity vector $2\boldsymbol{\Omega}$ (where $\boldsymbol{\Omega}$ is the angular rotation vector of the Earth) and the relative vorticity vector $\boldsymbol{\zeta}$ (Gill, 1982).

In the terminology of White et al. (2005) though, the equations solved by NEMO are the "non hydrostatic primitive equations" and as such, they do not conserve Q in (1). Instead, the conserved potential vorticity q in this class of model is:

$$\boxed{q = -\left(f\frac{\partial \sigma}{\partial z} + \boldsymbol{\zeta}'.\boldsymbol{\nabla}\sigma\right)/\rho} \tag{2}$$

where f is the Coriolis parameter:

$$f \equiv 2\Omega \sin\phi, \tag{3}$$

ϕ is latitude, and where the modified relative vorticity $\boldsymbol{\zeta}'$ has components:

$$\zeta'_x = -\frac{\partial v}{\partial z} \tag{4}$$

$$\zeta'_y = \frac{\partial u}{\partial z} \tag{5}$$

$$\zeta'_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{6}$$

in a local Cartesian system with eastward direction $\boldsymbol{i}$, coordinate x and velocity component u, northward direction $\boldsymbol{j}$, coordinate y and velocity component

v, upward direction $\boldsymbol{k}$, coordinate z and velocity component w. Note that q does not include contributions to PV arising from the vertical velocity field, which are neglected in ζ'_x and ζ'_y above, and that it does not include the contribution to PV arising from the poleward component of the planetary vorticity (a $2\Omega\cos\phi$ term in this coordinate system). Even though these terms are typically small and indeed negligible compared to the contributions in (2) when diagnosed in NEMO (at least in the ORCA12 configuration discussed below), they should not be computed as part of the PV since NEMO is not expected to conserve (1), as mentionned above.

3 Calculation of potential vorticity in NEMO

3.1 NEMO grid

Global NEMO model configurations use the ORCA mesh (Madec and Imbard, 1996), a tripolar mesh that is isotropic Mercator south of 20°N, and north of 20°N the mesh is distorted to give two poles over the land masses of Canada and Russia. This distortion of the mesh results in the grid becoming weakly anisotropic, avoids the convergence of the meridians at the North Pole that results in numerical instability. We refer the reader to the NEMO book for more in depth description and further references (NEMO manual, 2016). Briefly, it is useful to define the metric:

$$ds^2 = dr^2 + r^2 d\varphi^2 + r^2 cos^2\varphi\, d\lambda^2 \tag{7}$$

where φ and λ is the angle of latitude and longitude respectively and r is the distance to the Earth's centre. Writing the metric in terms of the new coordinates (i, j, k) which are functions of (λ, φ, z), we can relate the NEMO grid indices to the more conventional distance metric (7) from:

$$ds^2 = e_1^2 di^2 + e_2^2 dj^2 + e_3^2 dk^2 \tag{8}$$

In this expression, the scale factors e_1, e_2 and e_3 are such that:

$$e_1 = R\Big[\Big(\frac{\partial\lambda}{\partial i}\cos(\varphi)\Big)^2 + \big(\frac{\partial\varphi}{\partial i}\big)^2\Big] \tag{9a}$$

$$e_2 = R\Big[\Big(\frac{\partial\lambda}{\partial j}\cos(\varphi)\Big)^2 + \big(\frac{\partial\varphi}{\partial j}\big)^2\Big] \tag{9b}$$

$$e_3 = \frac{\partial z}{\partial k} \tag{9c}$$

In order to compute various objects such as vorticity or the gradient of the potential density which are needed in (2), it is important to know where

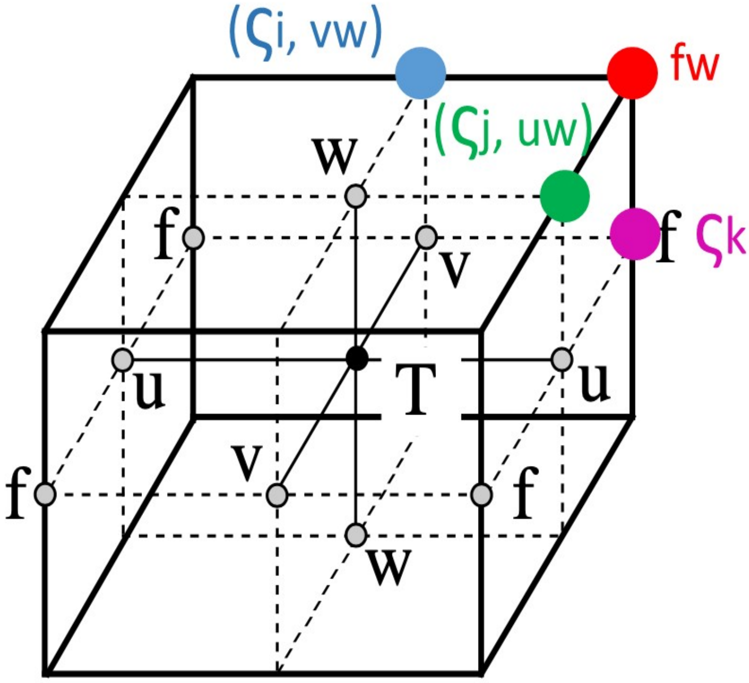

Figure 1: The NEMO C-grid. Scalars are measured at T points while at the velocities in the $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ directions are measured at the u, v and w points, respectively. At the f point, highlighted in magenta, the vertical component of the vorticity is measured, both planetary (f) and relative (ζ_k). Other important points for the calculation of q are uw, vw and fw, highlighted in green, blue and red, respectively (ζ_i is the component of $\boldsymbol{\zeta}'$ in the x-direction and ζ_j that of $\boldsymbol{\zeta}'$ in the y-direction).

these quantities are defined. The NEMO model uses the "C" grid from Arakawa's classification in three dimensions (Fig. 1). The scalar quantities such as temperature and salinity are measured at the centre of the unit cube at T points, while the velocities in the $\boldsymbol{i}$, $\boldsymbol{j}$ and $\boldsymbol{k}$ directions are measured at (u, v, w) points respectively. The vertical component of the planetary and relative vorticities are measured at f points. The indexing system used in this grid prioritises whole number indexing for scalar values and half number indexing for any other point in the cube (see Table 1).

t	i	j	k
u	i+1/2	j	k
v	i	j+1/2	k
w	i	j	k+1/2
f	i+1/2	j+1/2	k
uw	i+1/2	j	k+1/2
vw	i	j+1/2	k+1/2
fw	i+1/2	j+1/2	k+1/2

Table 1: The indexing for each point on the C grid for the NEMO model.

3.2 Calculation of q on the NEMO grid

First introduce the differencing operator:

$$\delta_i(s) = s(i+1/2, j, k) - s(i-1/2, j, k) \tag{10}$$

which can be applied to any direction $\boldsymbol{i}$, $\boldsymbol{j}$ or $\boldsymbol{k}$ and scalar s. Using this operator, the gradient operator can be rewritten, assuming s to be defined at T points, as:

$$\boldsymbol{\nabla} s = \frac{1}{e_{1u}}\delta_{i+1/2}(s) + \frac{1}{e_{2v}}\delta_{j+1/2}(s) + \frac{1}{e_{3w}}\delta_{k+1/2}(s) \tag{11}$$

Note that the the scaling term for each differencing operator is defined at the mid point between the two values.

The curl of some vector $\boldsymbol{A} = (a_1, a_2, a_3)$ can be rewritten as:

$$\begin{aligned} \boldsymbol{\nabla} \times \boldsymbol{A} = & \frac{1}{e_{2vw}e_{3vw}}\Big(\delta_{j+1/2}(e_{3w}a_3) - \delta_{k+1/2}(e_{2v}a_2)\Big)\boldsymbol{i} \\ & + \frac{1}{e_{1uw}e_{3uw}}\Big(\delta_{k+1/2}(e_{1u}a_1) - \delta_{i+1/2}(e_{3w}a_3)\Big)\boldsymbol{j} \\ & + \frac{1}{e_{1f}e_{2f}}\Big(\delta_{i+1/2}(e_{2v}a_2) - \delta_{j+1/2}(e_{1u}a_1)\Big)\boldsymbol{k} \end{aligned} \tag{12}$$

Because the horizontal grid dimension does not vary with depth, $e_{1uw} = e_{1u}$ and $e_{2vw} = e_{2v}$, so the previous equation can be rewritten as:

$$\begin{aligned} \boldsymbol{\nabla} \times \boldsymbol{A} = & \frac{1}{e_{2v}e_{3vw}}\Big(\delta_{j+1/2}(e_{3w}a_3) - \delta_{k+1/2}(e_{2v}a_2)\Big)\boldsymbol{i} \\ & + \frac{1}{e_{1u}e_{3uw}}\Big(\delta_{k+1/2}(e_{1u}a_1) - \delta_{i+1/2}(e_{3w}a_3)\Big)\boldsymbol{j} \\ & + \frac{1}{e_{1f}e_{2f}}\Big(\delta_{i+1/2}(e_{2v}a_2) - \delta_{j+1/2}(e_{1u}a_1)\Big)\boldsymbol{k} \end{aligned} \tag{13}$$

For a z-coordinate simulation we can simplify this further, noticing that in this case $e_{3uw} = e_{3w}$ and $e_{3vw} = e_{3w}$, leading to:

$$
\begin{aligned}
\boldsymbol{\nabla} \times \boldsymbol{A} = & \frac{1}{e_{2v} e_{3w}} \Big(\delta_{j+1/2}(e_{3w} a_3) - \delta_{k+1/2}(e_{2v} a_2) \Big) \boldsymbol{i} \\
& + \frac{1}{e_{1u} e_{3w}} \Big(\delta_{k+1/2}(e_{1u} a_1) - \delta_{i+1/2}(e_{3w} a_3) \Big) \boldsymbol{j} \\
& + \frac{1}{e_{1f} e_{2f}} \Big(\delta_{i+1/2}(e_{2v} a_2) - \delta_{j+1/2}(e_{1u} a_1) \Big) \boldsymbol{k}
\end{aligned}
\tag{14}
$$

or, after further simplifications:

$$
\begin{aligned}
\boldsymbol{\nabla} \times \boldsymbol{A} = & \Big(\frac{1}{e_{2vw}} \delta_{j+1/2}(a_3) - \frac{1}{e_{3vw}} \delta_{k+1/2}(a_2) \Big) \boldsymbol{i} \\
& + \Big(\frac{1}{e_{3uw}} \delta_{k+1/2}(a_1) - \frac{1}{e_{1uw}} \delta_{i+1/2}(a_3) \Big) \boldsymbol{j} \\
& + \frac{1}{e_{1f} e_{2f}} \Big(\delta_{i+1/2}(e_{2v} a_2) - \delta_{j+1/2}(e_{1u} a_1) \Big) \boldsymbol{k}
\end{aligned}
\tag{15}
$$

Equation (15) is the form used here for the calculation of potential vorticity. As shown by the denominators, the components of the curl of a vector is defined at vw, uw and f points on the cube. Therefore the components of the vorticity vector is defined at these same points. Table 2 shows where each value needed for the computation of q is defined and these are also highlighted by colors in Figure 1. It is readily seen from Fig. 1 that knowledge of σ at fw points is sufficient to estimate, by finite centred difference, the three components of $\boldsymbol{\nabla}\sigma$ at the points where the three components of the vorticity vector are estimated. *As the derivation above shows, note that (15) is a simplified form of eqs. (4.4)-(4.5)-(4.6) in the NEMO manual which does not apply if the scale factors e_3 are a function of horizontal location or if the horizontal scale factors e_1, e_2 vary with depth.*

In summary, the calculation of q is decomposed into three steps:

(i) Compute ζ_i' and $\partial\sigma/\partial i$ at vw point, and then the resulting contribution $PV_x \equiv -\zeta_x' \frac{\partial \sigma}{\partial x}$ to q

(ii) Compute ζ_j' and $\partial\sigma/\partial j$ at uw point, and then the resulting contribution $PV_y \equiv -\zeta_y' \frac{\partial \sigma}{\partial y}$ to q

(iii) Compute ζ_k', f and $\partial\sigma/\partial k$ at f point, and then the resulting contribution $PV_z \equiv -(f + \zeta_z') \frac{\partial \sigma}{\partial z}$ to q

(iv) Interpolate PV_x, PV_y, PV_z at T point and compute in situ density ρ at T point

Points	Values
t	T, S, ρ, σ, q
u	u, $\delta_{i+1/2}(s)$
v	v, $\delta_{j+1/2}(s)$
w	w, $\delta_{k+1/2}(s)$
f	f, ζ_k, $\delta_{k-1/2}(\sigma_{fw})$, PV_x
uw	ζ_j, $\delta_{j-1/2}(\sigma_{fw})$, PV_y
vw	ζ_i, $\delta_{i-1/2}(\sigma_{fw})$, PV_x
fw	σ_{fw}

Table 2: A table labeling the position of all vector components and scalars used in the potential vorticity computation.

(v) Compute q at T point as the sum $(PV_x + PV_y + PV_z)/\rho$

A Python code of these steps is given in Appendix.

3.3 Treatment of boundary points

No specific alteration to the code was made to consider grid points adjacent to the land or bottom boundaries. The land or bottom points involved in the resulting calculations were masked so that the values of q output by the routine were only genuinely involving "valid" (or "interior") ocean gridpoints. This can lead to significant "loss" of gridpoint data for q, especially near sloping boundaries where the missing horizontal derivatives "spread" the masked values towards interior gridpoints. A special treatment of these regions should be done in future work to improve this.

4 Links to other forms of potential vorticity

The potential vorticity q in (2) can be linked explicitely to the PV computed using hydrographic sections and to that used in quasi-geostrophic theory (QG) by rewriting (2) as:

$$-\rho q = (f + \zeta'_z)\frac{\partial \sigma}{\partial z} + \zeta'_x \frac{\partial \sigma}{\partial x} + \zeta'_y \frac{\partial \sigma}{\partial y} \tag{16}$$

If the motion is close to geostrophic balance (small Rossby number), the thermal wind relation allows to relate the ζ'_x and ζ'_y components to the horizontal density gradients. Introducing the buoyancy frequency N and the

Richardson number R_i:

$$R_i \equiv N^2 / \left((\frac{\partial u}{\partial z})^2 + (\frac{\partial v}{\partial z})^2 \right), \tag{17}$$

one can then rewrite q as:

$$q \approx \frac{fN^2}{g} (1 + \zeta'_z/f - 1/R_i) \quad \text{(for small Rossby number)} \tag{18}$$

From this relation, we obtain:

- The quasi-geostrophic regime, in the limit of small Rossby number and large Richardson number. This amounts to neglecting the contributions from the horizontal vorticities in (2) –see for example Green (1970).

- The planetary geostrophic limit, where in addition to the above, the vertical component of the vorticity vector is also neglected ($\zeta'_x \ll f$) and:

$$\rho q \approx -f \frac{\partial \sigma}{\partial z} \tag{19}$$

  This leads to the familiar f/h calculation when averaged over an isopycnal layer bounded by two sigma surfaces σ_1 and σ_2 ($> \sigma_1$) located at height z_1 and z_2:

$$\frac{1}{z_1 - z_2} \int_{z_2}^{z_1} \rho q dz \approx f(\sigma_2 - \sigma_1)/h \tag{20}$$

  where we have introduced the thickness $h = z_1 - z_2$ of the layer.

5 Illustration: potential vorticity of the Gulf Stream

A zonal section of q at the approximate latitude where the Gulf Stream separates from the coast in the 1/12th ° simulation described in Marzocchi et al. (2015) is show in Fig. 2. The Gulf Stream is readily seen in the sloping isopycnals (white contours) near $73°W$ while the surface mixed layer is identified with regions where the isopycnals are nearly vertical. The PV shows large values just below the mixed layer in the western side of the section and all across around 200m depth, presumably reflecting the seasonal thermocline which hasn't yet been eroded. Particularly striking is the fact that these large values penetrate deeper along the axis of the Gulf Stream,

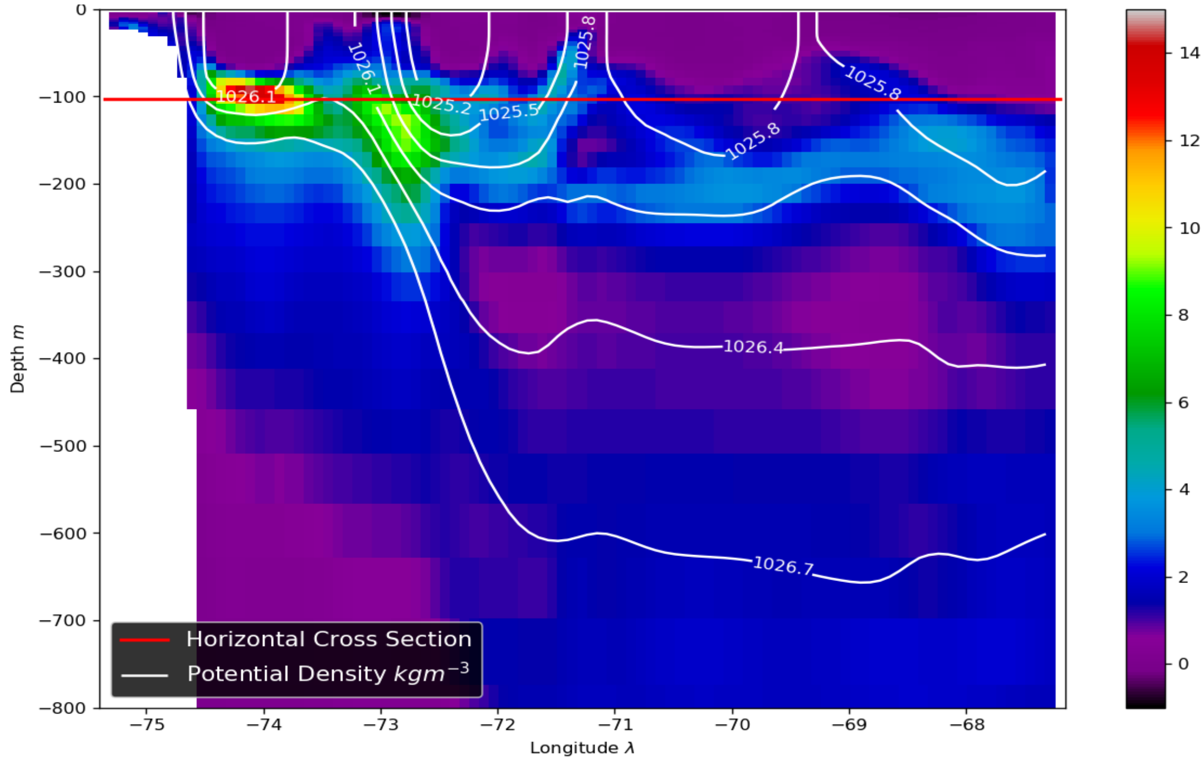

Figure 2: Zonal section of q (colors) at the approximate latitude $37°N$. White lines indicate isopycnals. In this and other plots below, q is expressed in units of $10^{-10}m^{-1}s^{-1}$ and the section is based on 5-day means data centred on 25 January 1979.

flanked to the east by a low PV water mass identified as the model version of the 18° mode water. All these features, as well as the magnitude of the PV distribution are in agreement with the observations reported in Todd et al. (2016).

The calculation allows us to further investigate the different contributions to q. The most dominant is the "planetary geostrophic" component in (19), which is shown in Fig. 3, and can be seen to capture the basic features of the PV distribution. The contribution from the vertical component of the relative vorticity is highlighted in Fig. 4 and it is seen to make a difference, as expected, on the poleward (enhancing the PV) and equatorward (diminishing the PV) flank of the Gulf Stream. Finally, the contribution from the horizontal vorticities are given in Figs. 5, 6. As expected from the discussion of the thermal wind in section 4, they make a significant contribution where the Richardson number is smaller, i.e. close to the surface (low N) and close to regions of sloping isopycnals (large shear). The effect is seen to penetrate

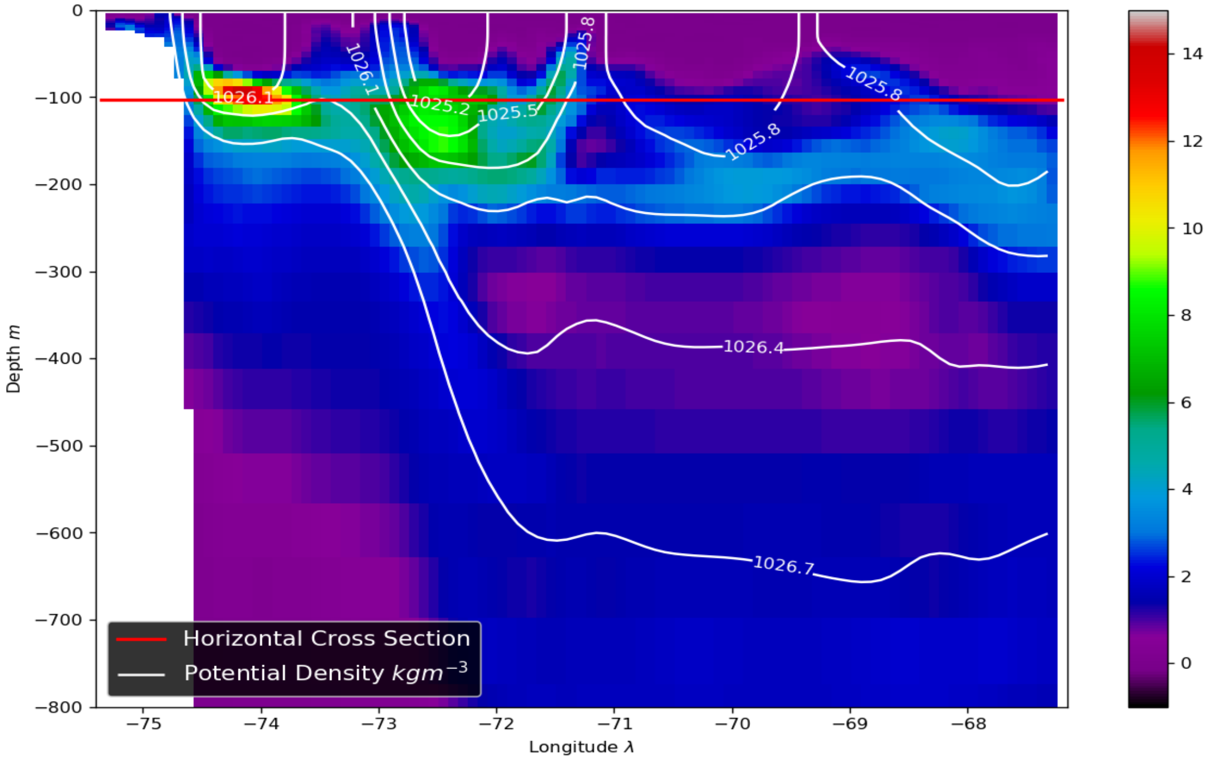

Figure 3: Same as Fig. 2 but for the contribution to q given in eq. (19).

deeply around 73°W in this section for the y-component of the vorticity (Fig. 6), reflecting the main core of the separated Gulf Stream at this longitude and at that time.

6 References

-Green, J. S. A., 1970: Transfer properties of the large scale eddies and the general circulation of the atmosphere, Q.J.R. Meteorol. Soc. 96 (408): 157–185. doi:10.1002/qj.49709640802.

-Gill, A. E . J., 1982: Atmosphere-Ocean dynamics, Academic Press, Vol 30, International Geophysical Series.

-Hoskins, 1982: The mathematical theory of frontogenesis, Ann. Rev. Fl. Mech., 14, 131–151.

-Korty and Schneider, 2007: A Climatology of the Tropospheric Thermal

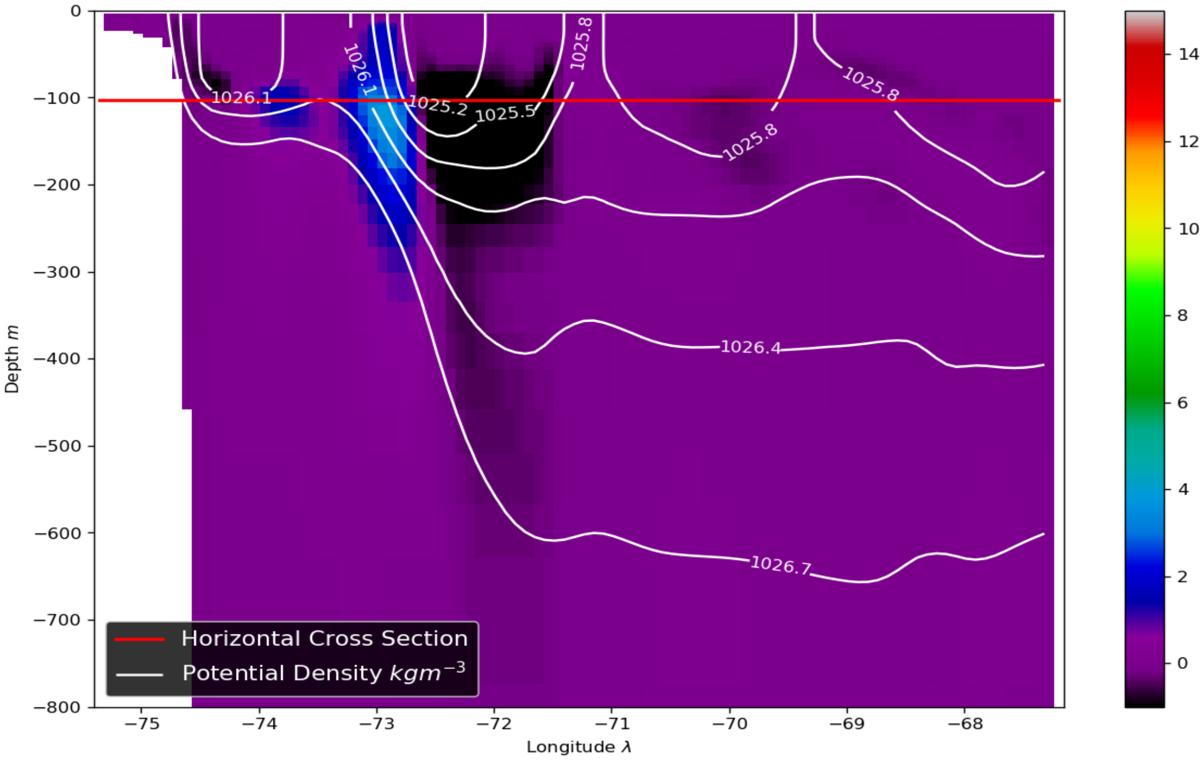

Figure 4: Same as Fig. 2 but for the contribution to q arising from the $-\zeta_z' \partial\sigma/\partial z$ term.

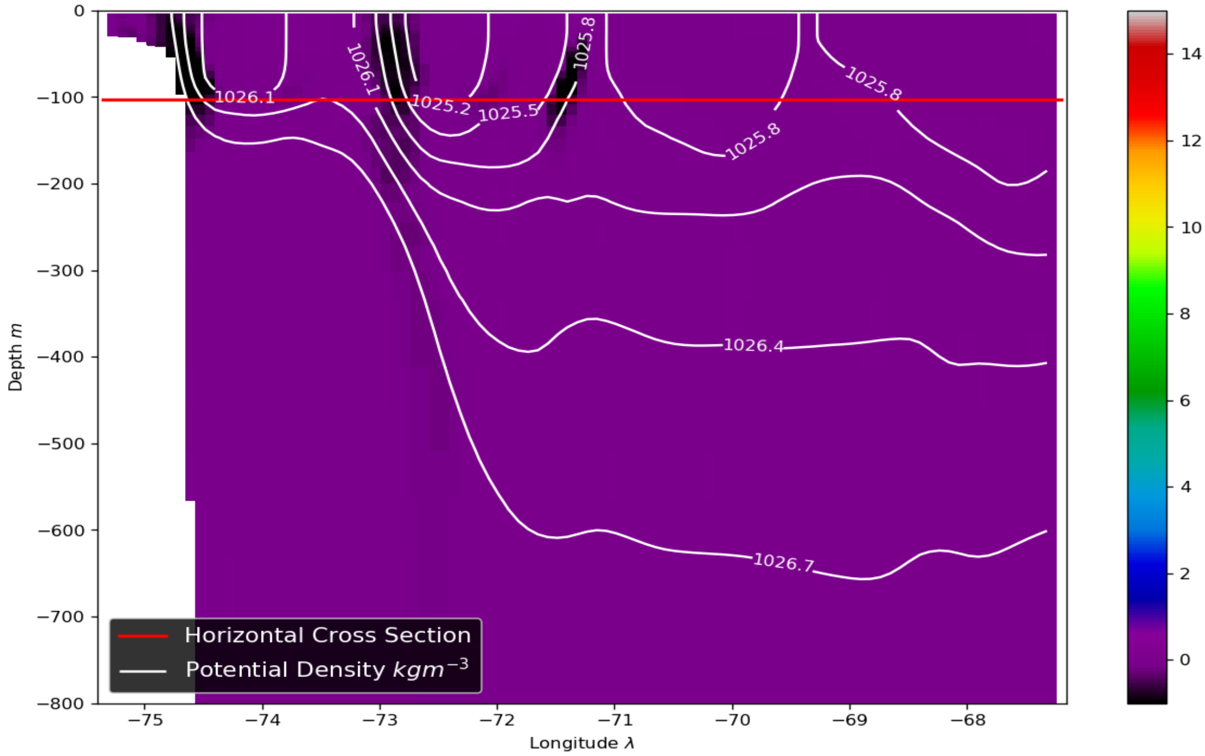

Figure 5: Same as Fig. 2 but for the contribution to q arising from the $-\zeta_x' \partial\sigma/\partial x$ term.

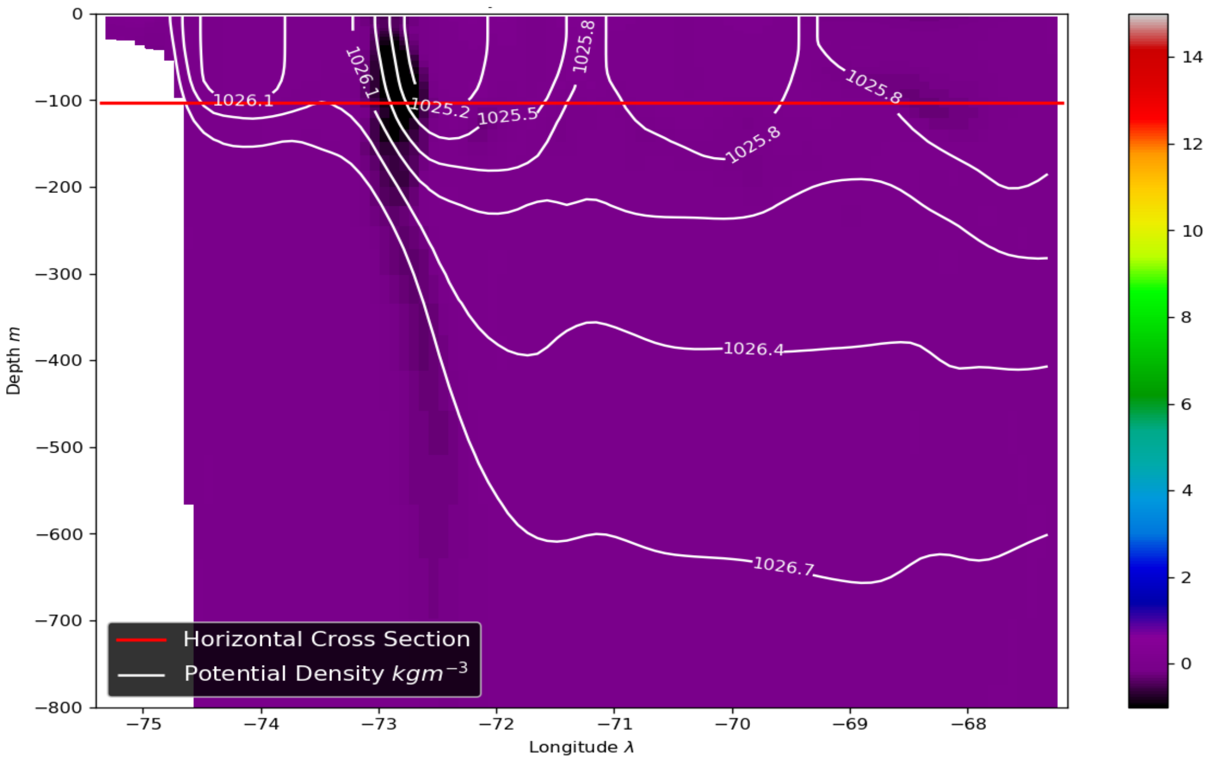

Figure 6: Same as Fig. 2 but for the contribution to q arising from the $-\zeta_y' \partial\sigma/\partial y$ term.

Stratification Using Saturation Potential Vorticity, J. Atm. Sci., 20, 5977–5991.

-Madec, G. and M. Imbard, 1996: A global ocean mesh to overcome the North Pole singularity, Climate Dynamics 12, 381–388.

-Marzocchi et al., 2015: The North Atlantic subpolar circulation in an eddy-resolving global ocean model, J. Mar. Sys., 142, 126-143.

-McDougall, T. J., 1988: Neutral surface potential vorticity, Progress in Oceanography, Volume 20, Issue 3, Pages 185–221.

-NEMO manual, 2016: The NEMO ocean engine, Institut Pierre-Simon Laplace (IPSL), Scientific Notes of Climate Modelling Center No 27.

-Thomas et al., 2013: Symmetric instability in the Gulf Stream, Deep Sea Research Part II, 91, 96–110.

-Todd et al., 2016: Potential vorticity structure in the North Atlantic western boundary current from underwater glider observations, Journal of Physical Oceanography, 46, 1, 327–348

-White et al., 2005: Consistent approximate models of the global atmosphere: shallow, deep, hydrostatic, quasi-hydrostatic and non-hydrostatic, Quarterly Journal of the Royal Meteorological Society, 131, 609, 2081–2107.

A Python codes for PV calculation

A.1 Vorticity

```
def calc_vortz(U,V,e1u,e2v,e1f,e2f):
#calculating the relative vorticity in the k direction
dA=e1f*e2f #area for the face of the cube
eU=U*e1u #term inside the derivative
eV=V*e2v
du=np.roll(eU,-1,1)-eU #difference in adjacent velocities
dv=np.roll(eV,-1,2)-eV
vort=(dv - du)/dA #defined at f points
```

```
vort=null(vort,1) #creating null points at the end of the array
return vort


def calc_vorty(U,W,e1uw,e3uw):
#calculating the relative vorticity in the j direction
du=-np.roll(U,-1,0)+U
dw=np.roll(W,-1,2)-W
vort=du/np.roll(e3uw,-1,0)-np.roll(dw/e1uw,-1,0)

vort=null(vort,1)
vort=np.roll(vort,1,0) #bring back to uw(0) point
return vort


def calc_vortx(V,W,e2vw,e3vw):
#calculating the relative vorticity in the i direction
dv=-np.roll(V,-1,0)+V
dw=np.roll(W,-1,1)-W
vort=np.roll(dw/e2vw,-1,0)-dv/np.roll(e3vw,-1,0)
#rolling e3w to correct position

vort=null(vort,1)
vort=np.roll(vort,1,0) #bring back to vw(0)
return vort
```

A.2 Potential vorticity

```
def calc_PVt(gvori,gvorj,gvork,e1vw,e2uw,e3f,RHO,SIG):

#averaging everything over to the correct positions
SIGfw=av_fw(SIG)

#calculating the gradient
dSi=-(np.roll(SIGfw,+1,2)-SIGfw)/e1vw
dSj=-(np.roll(SIGfw,+1,1)-SIGfw)/e2uw
dSk=(-np.roll(SIGfw,-1,0)+SIGfw)/e3f

#PV formula
PVi=-gvori*dSi
```

```
PVj=-gvorj*dSj
PVk=-gvork*dSk

#averaging over to T points
PVi=av(PVi,0,-1)
PVj=av(PVj,1,-1)
PVk=av(PVk,2,-1)
PVi=PVi/RHO
PVj=PVj/RHO
PVk=PVk/RHO

PV=PVk+PVj+PVi
return [PV, PVi, PVj, PVk]
```

A.3 Averaging routines

```
def av(S,d,p):
#d=0,1,2 respectively corresponds to the x,y,z plane averaging
#p=1 average +1/2, p=-1 average to -1/2
S=null(S,1)
if d==0: #move to vw
m=0
n=1
elif d==1: #move to uw
m=0
n=2
elif d==2: #move to f
m=1
n=2
S1=np.roll(S,-p,n)
S2=np.roll(S,-p,m)
S3=np.roll(np.roll(S,-p,n),-p,m)
S_av=(S+S1+S2+S3)4
S_av=np.roll(S_av,1-m,0)
return S_av

def av_fw(S):
#averaging to centre of cube
S1=np.roll(S,-1,0) #w
S2=np.roll(S,-1,1) #v
S3=np.roll(S,-1,2) #u
```

```
S4=np.roll(S1,-1,1) #vw
S5=np.roll(S1,-1,2) #uw
S6=np.roll(S2,-1,2) #f
S7=np.roll(S6,-1,0) #fw
S_av=(S+S1+S2+S3+S4+S5+S6+S7)8
S_av=null(S_av,1)
S_av=np.roll(S_av,1,0)
return S_av
```