Momentum, Kinetic Energy and Barotropic Vorticity online Diagnostics for CROCO

Jonathan Gula

June 21, 2019

Contents

The physical meaning and numerical implementation of the cpp key DIAGNOSTICS_UV, DIAGNOSTICS_EK and DIAGNOSTICS_VRT in CROCO are described below.

- DIAGNOSTICS_UV = outputs 3d terms from the momentum equation in a separate file.
- DIAGNOSTICS_EK = outputs vertically integrated terms from the kinetic energy equation in a separate file. The DIAGNOSTICS_EK_FULL is optional. It impacts the computation of the terms but do not impact the choice or the size of the outputed fields (see below). The key DIAGNOSTICS_EK_MLD can be added to also output terms averaged over the depth of the surface mixed-layer.
- DIAGNOSTICS_VRT = outputs 2d terms from the barotropic vorticity equation in a separate file.

The 3 options needs to be activated in the cppdefs.h file:

```
# define DIAGNOSTICS_VRT
# define  DIAGNOSTICS_EK
# ifdef DIAGNOSTICS_EK
#  define DIAGNOSTICS_EK_FULL
#  define DIAGNOSTICS_EK_MLD
# endif
```

and require the following addition to the namelist (roms.in):

```

diagnosticsM:   ldefdiaM   nwrtdiaM    nrpfdiaM /filename
                   F          0          0
                                 sarga_diaM.nc

diagM_avg: ldefdiaM_avg  ntsdiaM_avg  nwrtdiaM_avg  nprfdiaM_avg /filename
               F          1           0            0
                                 sarga_diaM_avg.nc

diagM_history_fields: diag_momentum(1:2)
                            T T

diagM_average_fields: diag_momentum_avg(1:2)
                            T T

diags_ek:   ldefdiags_ek, nwrtdiags_ek, nrpfdiags_ek /filename
                   T          72        100
                                 roms_diags_ek.nc

diags_ek_avg: ldefdiags_ek_avg  ntsdiags_ek_avg  nwrtdiags_ek_avg  nprfdiags_ek_avg /filename
               T          1           10          100
                                 roms_diags_ek_avg.nc

diags_ek_history_fields: diags_ek
                            T

diags_ek_average_fields: diags_ek_avg
                            T

diags_vrt:   ldefdiags_vrt, nwrtdiags_vrt, nrpfdiags_vrt /filename
                   T          72        100
                                 roms_diags_vrt.nc

diags_vrt_avg: ldefdiags_vrt_avg  ntsdiags_vrt_avg  nwrtdiags_vrt_avg  nprfdiags_vrt_avg /filename
               T          1          72          100
                                 roms_diags_vrt_avg.nc

diags_vrt_history_fields: diags_vrt
```

```
                          T
diags_vrt_average_fields: diags_vrt_avg
                          T
```

1 Momentum equation

1.1 Continuous equation

The horizontal momentum equations in the Boussinesq approximation are:

$$\frac{\partial u}{\partial t} = -u_j \frac{\partial u}{\partial x_j} - w\frac{\partial u}{\partial z} + fv - \frac{P_x}{\rho_0} + \mathcal{V}_u + \mathcal{D}_u + \mathcal{S}_u + \mathcal{B}_u \quad (1)$$

$$\underbrace{\frac{\partial v}{\partial t}}_{rate} = -\underbrace{u_j \frac{\partial v}{\partial x_j}}_{hadv} - \underbrace{w\frac{\partial v}{\partial z}}_{vadv} - \underbrace{fu}_{cor} - \underbrace{\frac{P_y}{\rho_0}}_{Prsgrd} + \underbrace{\mathcal{V}_v}_{vmix} + \underbrace{\mathcal{D}_v}_{hmix} + \underbrace{\mathcal{S}_v}_{nudg} + \underbrace{\mathcal{B}_v}_{baro} \quad (2)$$

where Cartesian tensor notation with summation convention has been used for $j = 1, 2$; $\vec{u} = (u, v)$ is the horizontal velocity vector, w is the vertical velocity, f is the Coriolis parameter, P is the pressure anomaly, $\vec{\mathcal{V}} = (\mathcal{V}_u, \mathcal{V}_v) = \frac{\partial}{\partial z}\left(K_{Mv}\frac{\partial \vec{u}}{\partial z}\right)$ is the vertical mixing, $\vec{\mathcal{D}} = (\mathcal{D}_u, \mathcal{D}_v)$ the horizontal diffusion, $\vec{\mathcal{B}} = (\mathcal{B}_u, \mathcal{B}_v)$ the term due to barotropic/baroclinic coupling, and $\vec{\mathcal{S}} = (\mathcal{S}_u, \mathcal{S}_v)$ other sources and sinks (due to restoring, nudging, boundary conditions, etc.).

1.2 Discrete formulation

The model momentum equations computes the momentum at the time-step n+1 [*step3d_uv1.F, step3d_uv2.F*]:

$$H^{n+1}u_i^{n+1} = H^n u_i^n + \Delta t \sum_j M_{i,j}^{n+\frac{1}{2}}$$

where u_i are the horizontal components of the velocity vector ($i = 1, 2$), Δt is the baroclinic time-step of the model, H the vertical grid spacing, and $M_{i,j}^{n+\frac{1}{2}}$ are the terms from the momentum equation (with units of velocities × cell height / time) corresponding to horizontal advection, vertical advection, coriolis force, pressure gradient, vertical mixing, horizontal diffusion (implicit and/or explicit), and various sources and sinks.

The terms are divided by H^{n+1} before writing in the file such that they all have dimensions of m s^{-2}. Variables included in the roms_diags_uv.nc files are:

- u_rate, v_rate = rate of change of momentum [*step3d_uv2.F*]
- u_xadv, v_xadv = advection + implicit dissipation along xi- axis + grid curvature terms (see CURVGRID) [*rhs3d.F*]
- u_yadv, v_yadv = advection + implicit dissipation along eta- axis [*rhs3d.F*]
- u_vadv, v_vadv = vertical advection [*rhs3d.F*]
- u_cor, v_cor = Coriolis term [*rhs3d.F*]
- u_Prsgrd, v_Prsgrd = Pressure gradient [*prsgrd.F*]
- u_Baro, v_Baro = barotropic/baroclinic coupling [*step3d_uv2.F*]
- u_hmix, v_hmix = Horizontal mixing (explicit) [*uv3dmix4_GP.F, uv3dmix_GP.F, uv3dmix_spg.F, uv3dmix4_S.F, uv3dmix_S.F*]
- u_vmix, v_vmix = Vertical mixing [*step3d_uv2.F*]
- u_nudg, v_nudg = Nudgind, restoring, boundary conditions, etc. [*step3d_uv2.F*]

All variables are 3D on horizontal u- and v- grids and vertical rho-grid (N levels).

The following pointwise budgets are closed:

u_rate = u_xadv + u_yadv + u_vadv + u_Prsgrd + u_Baro + u_cor + u_vmix + u_hmix + u_nudg
v_rate = v_xadv + v_yadv + v_vadv + v_Prsgrd + v_Baro + v_cor + v_vmix + v_hmix + v_nudg

2 Kinetic energy equation

2.1 Continuous equation

The kinetic energy equation is formed by taking the inner product of the horizontal velocities with the momentum equations:

$$\frac{1}{2}\frac{\partial u_i^2}{\partial t} + u_j\frac{\partial \frac{1}{2}u_i^2}{\partial x_j} + w\frac{\partial \frac{1}{2}u_i^2}{\partial z} = -\frac{u_i}{\rho_0}\frac{\partial P}{\partial x_i} + \mathcal{V}_i u_i + \mathcal{D}_i u_i + \mathcal{S}_i u_i \tag{3}$$

where Cartesian tensor notation with summation convention has been used, $i = 1, 2$, $j = 1, 2$; u_i are the horizontal components of the velocity vector u_j; $u_3 = w$ is the vertical velocity.

Variables included in the roms_diags_ek.nc files are:

- ek_rate = rate of change of depth integrated kinetic energy, $\frac{\partial}{\partial t}\int_{-h}^{\zeta}\frac{1}{2}u_i^2 dz$
- ek_hadv = $\int_{-h}^{\zeta} u_i \frac{\partial \frac{1}{2}u_i^2}{\partial x_i} dz$
- ek_vadv = $\int_{-h}^{\zeta} w \frac{\partial \frac{1}{2}u_i^2}{\partial z} dz$
- ek_Prsgrd = $\int_{-h}^{\zeta} -u_i \frac{\partial P}{\partial x_j} dz$
- ek_vmix = $\int_{-h}^{\zeta} \mathcal{V}_i u_i dz$
- ek_hmix = explicit part of $\int_{-h}^{\zeta} \mathcal{D}_i u_i dz$
- ek_hdiff = implicit part of $\int_{-h}^{\zeta} \mathcal{D}_i u_i dz$ [already included in ek_hadv]
- ek_nudg = $\int_{-h}^{\zeta} \mathcal{S}_i u_i dz$, other sources and sinks such as nudging and open boundary conditions
- ek_vol = depth integrated kinetic energy variation due to the grid breezing ($\frac{\partial \zeta}{\partial t}$)
- ek_cor = $\int_{-h}^{\zeta}(fuv - fvu)dz$, *i.e.,* zero at the continuous level, but not formally zero in the model due to the discretization of the grid
- ek_wind [already included in ek_vmix]
- ek_drag [already included in ek_vmix]

Such that the full closed budget is:

ek_rate = ek_hadv + ek_vadv + ek_Prsgrd + ek_vmix + ek_hmix + ek_hdiff + ek_nudg + ek_cor + ek_vol

Tems ek_vol and ek_cor should both be negligible and are kept only for consistency check. Note that the discretization of the Coriolis term does not unsure pointwise cancelation of ek_cor but should unsure area averages cancellation. Area average cancellation is perfect for closed boundary conditions but there might be a residual for open boundary conditions.

Examples of kinetic energy budget with ROMS using these diagnostics can be found in **?**, where the equation is further decomposed into mean and eddy parts:

$$\overline{u}_i \overline{\frac{\partial u_i}{\partial \mathrm{t}}} + \underbrace{\frac{\partial\left(\frac{1}{2}\overline{u}_j\overline{u}_i^2 + \frac{1}{\rho_0}\overline{u}_j\overline{p}\right)}{\partial x_j}}_{\text{Boundary Transport}} = \underbrace{-\overline{u}_i\frac{\partial\overline{u_j'u_i'}}{\partial x_j}}_{\text{EKE}\to\text{MKE}} + \underbrace{\overline{w}\overline{b}}_{\text{MPE}\to\text{MKE}} + \underbrace{\overline{\mathcal{V}}_i\overline{u}_i}_{\text{Vertical mixing}} + \underbrace{\overline{\mathcal{D}}_i\overline{u}_i}_{\text{Horizontal diffusion}}, \tag{4}$$

$$\overline{u_i'\frac{\partial u_i'}{\partial \mathrm{t}}} + \underbrace{\frac{\partial\left(\frac{1}{2}\overline{u_j}\,\overline{u_i'^2} + \frac{1}{2}\overline{u_j'u_i'^2} + \frac{1}{\rho_0}\overline{u_j'p'}\right)}{\partial x_j}}_{\text{Boundary Transport}} =$$
$$\underbrace{-\overline{u_j'u_i'}\frac{\partial\overline{u_i}}{\partial x_j}}_{\text{MKE}\to\text{EKE}} + \underbrace{\overline{w'b'}}_{\text{EPE}\to\text{EKE}} + \underbrace{\overline{\mathcal{V}_i'u_i'}}_{\text{Vertical mixing}} + \underbrace{\overline{\mathcal{D}_i'u_i'}}_{\text{Horizontal diffusion}}. \tag{5}$$

where the overbar denotes a time average, and the prime denotes fluctuations relative to the time average. Cartesian tensor notation with summation convention has been used, $i = 1, 2,\ j = 1, 2, 3$; u_i are the horizontal components of the velocity vector u_j; $u_3 = w$ is the vertical velocity; p is the pressure anomaly; $b = -\frac{g\rho}{\rho_0}$ is the buoyancy anomaly; $\mathcal{V}_i$ and $\mathcal{D}_i$ are the vertical mixing and horizontal diffusion terms in the horizontal momentum equations.

Note in particular the decomposition:

$$\frac{u_i}{\rho_0}\frac{\partial P}{\partial x_i} = \frac{\partial\left(\frac{1}{\rho_0}u_j p\right)}{\partial x_j} + wb$$

with summation convention $i = 1, 2,\ j = 1, 2, 3$; which shows that the contribution of the pressure gradient term can be split into the divergence of an energy flux at the boundaries of the domain volume ($\frac{\partial\left(\frac{1}{\rho_0}u_j p\right)}{\partial x_j}$) and conversion between potential and kinetic energy (wb).

2.2 Discrete formulation

The kinetic energy equation terms correspond to the momentum equation terms multiplied by the velocity at the time-step $n + \frac{1}{2}$ [*step3d_uv2.F*]:

$$\frac{1}{2}H^{n+1}(u_i^{n+1})^2 = \frac{1}{2}H^n(u_i^n)^2 + \Delta t \sum_j u^{n+\frac{1}{2}} M_{i,j}^{n+\frac{1}{2}}$$

The output terms are $u^{n+\frac{1}{2}} M_{i,j}^{n+\frac{1}{2}}$. They are vertically integrated and multiplied by the cell surface $dm\, dn = \frac{1}{pm\, pn}$, such that they correspond to volume energy tendencies. They all have dimensions of [energy × volume / time] (m^5s^{-3}). The exact formulation of the terms is:

- ek_rate $= \sum_{k=1}^{N} \left(\frac{1}{2}(u_i^{n+1})^2 - \frac{1}{2}H^n(u_i^n)^2\right)/\Delta t$
- ek_hadv $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{hadv}}^{n+1/2}$
- ek_vadv $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{vadv}}^{n+1/2}$
- ek_Prsgrd $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{Prsgrd}}^{n+1/2}$
- ek_vmix $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{vmix}}^{n+1/2}$
- ek_hmix $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{hmix}}^{n+1/2}$
- ek_hdiff $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{hdiff}}^{n+1/2}$ [already included in ek_hadv]
- ek_nudg $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{nudg}}^{n+1/2}$
- ek_vol $= \sum_{k=1}^{N} \left(\frac{H^n - H^{n+1}}{2} \left[\frac{H^n}{H^{n+1}}(u^n)^2 + 2u_n \sum_j \frac{M_j^{n+1/2}}{H^{n+1}} \right] \right)$
- ek_cor $= \sum_{k=1}^{N} u_i^{n+1/2} M_{i,\mathrm{cor}}^{n+1/2}$
- ek_wind $= \tau_i^s u_i^{n+1/2}$ [already included in ek_vmix]
- ek_drag $= \tau_i^b u_i^{n+1/2}$ [already included in ek_vmix]

Spatial discretization: Momentum terms and velocities are first computed on their native u- and v-grids, then energy terms are computed on the rho-grid:

$$\begin{aligned} \text{ek\_hadv} = &\int_{-h}^{\zeta} \frac{1}{2}\left[u(i,j)M_{u,\mathrm{hadv}}^{n+1/2}(i,j) + u(i+1,j)M_{u,\mathrm{hadv}}^{n+1/2}(i+1,j)\right] dz \\ &+ \int_{-h}^{\zeta} \frac{1}{2}\left[v(i,j)M_{v,\mathrm{hadv}}^{n+1/2}(i,j) + v(i,j+1)M_{v,\mathrm{hadv}}^{n+1/2}(i,j+1)\right] dz \end{aligned} \quad (6)$$

2.2.1 DIAGNOSTICS_EK_FULL

if DIAGNOSTICS_EK_FULL is defined, the terms from the momentum equations $M_i^{n+1/2}$ computed during time-step n, are multiplied by $\frac{u^{n+1}+u^n}{2}$. This option has the disadvantage to require several 3d arrays during the online computation to save the $M_i^{n+1/2}$ terms from the momentum equation until the end of the time-step (because the vertical integration requires u^{n+1}).

if DIAGNOSTICS_EK_FULL is NOT defined, the velocities used to multiply the momentum terms are the velocities computed after the predictor step $u^{n+1/2}$. This option has the advantage to only use 2d arrays during the online computation as the vertical integration is performed directly, but the balance will not be perfectly closed because $u^{n+1/2} \neq \frac{u^{n+1}+u^n}{2}$

2.2.2 DIAGNOSTICS_EK_MLD

This adds energy terms averaged over the mixed layer depth (currently defined as the KPP hbl).

3 Barotropic vorticity equation

3.1 Continuous equation

The full barotropic vorticity balance equation of the flow is obtained by integrating the momentum equations in the vertical and cross differentiating them:

$$\underbrace{\frac{\partial \Omega}{\partial t}}_{\text{rate}} = - \underbrace{\vec{\nabla}.(f\vec{\bar{u}})}_{\text{planet. vort. adv.}} + \underbrace{\frac{\mathbf{J}(P_b, h)}{\rho_0}}_{\text{bot. pres. torque}} + \underbrace{\vec{k}.\vec{\nabla} \times \frac{\vec{\tau}^{wind}}{\rho_0}}_{\text{wind curl}} - \underbrace{\vec{k}.\vec{\nabla} \times \frac{\vec{\tau}^{bot}}{\rho_0}}_{\text{bot. drag curl}}$$
$$+ \underbrace{\mathcal{D}_\Sigma}_{\text{horiz.diffusion.}} - \underbrace{A_\Sigma}_{\text{NL advection}} \tag{7}$$

where the barotropic vorticity is defined as the vorticity of the vertically integrated velocities[1]

$$\Omega = \frac{\partial \overline{v}}{\partial x} - \frac{\partial \overline{u}}{\partial y}$$

with (u, v) the (x, y) components of the horizontal flow, and the overbar denotes a vertically integrated quantity,

$$\overline{u} = \int_{-h}^{\zeta} u \, dz,$$

where $\zeta(x, y, t)$ is the free-surface height and $h(x, y) > 0$ the depth of the resting topography. $H(i, j, t) = \int_{-h}^{\zeta} dz = \zeta(i, j, t) + h(i, j)$ is the total depth of the water column. Finally, the curl of non-linear advection terms can be written as

$$A_\Sigma = \frac{\partial^2(\overline{vv} - \overline{uu})}{\partial x \partial y} + \frac{\partial^2 \overline{uv}}{\partial x \partial x} - \frac{\partial^2 \overline{uv}}{\partial y \partial y},$$

and $\mathcal{D}_\Sigma$ is the term due to the horizontal diffusion in the model implicitly part of the advective scheme, plus eventually some explicit diffusion.

Examples of barotropic vorticity budget with ROMS using these diagnostics and interpretations can be found in **?** and **?**.

3.2 Discrete formulation

Variables included in the roms_diags_vrt.nc files are:

- vrt_rate = rate of change of barotropic vorticity [*step3d_uv2.F*]
- vrt_xadv = contribution of advection + implicit dissipation along xi-axis+ grid curvature terms (see CURVGRID) [*rhs3d.F*]
- vrt_hdiff = implicit dissipation along xi- and eta- axis [*rhs3d.F*] [already included in vrt_xadv+ vrt_yadv]
- vrt_cor = planetary vorticity advection [*rhs3d.F*]

[1]Note that the barotropic vorticity is not identical to the vertically integrated vorticity. The curl and the vertical integration can be interchanged at the expense of introducing terms due to the horizontal variations of the limits of the integral. The difference $\Omega - \overline{\zeta} = \vec{u_s} \times \vec{\nabla}\zeta + \vec{u_b} \times \vec{\nabla}h$, where $\vec{u_s}$ and $\vec{u_b}$ are the horizontal velocities at the surface and bottom, respectively, can be non-negligible at places where we have both significant bottom currents and large topography slopes.

- vrt_Prsgrd = bottom Pressure torque [*prsgrd.F*]
- vrt_hmix = contribution of Horizontal diffusion (explicit) [*uv3dmix4_GP.F, uv3dmix_GP.F, uv3dmix_spg.F, uv3dmix4_S.F, uv3dmix_S.F*]
- vrt_vmix = contribution of Vertical mixing = vrt_Wind + vrt_Drag [*step3d_uv2.F*]
- vrt_nudg = contribution of Nudging, restoring, boundary conditions, etc. [*step3d_uv2.F*]
- vrt_Wind = Wind stress curl [*step3d_uv2.F*] [already included in vrt_vmix]
- vrt_Drag = Bottom drag curl [*step3d_uv2.F*] [already included in vrt_vmix]

All variables are 2D on horizontal psi-grid. The following pointwise budget is closed:

vrt_rate = vrt_xadv + vrt_yadv + vrt_Prsgrd + vrt_cor + vrt_vmix + vrt_hmix + vrt_nudg

Spatial discretization: Momentum terms are first computed and vertically averaged on their native u- and v-grids, then vorticity terms are computed on the psi-grid: