

Offline momentum and Kinetic Energy budgets for NEMO-based simulations with CDFTOOLS

QUENTIN JAMET*

Univ. Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble, France

WILLIAM K. DEWAR

Univ. Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble, France, and Dept. of EOAS, Florida State University, Tallahassee, FL, USA

THIERRY PENDUFF, JULIEN LE SOMMER, JEAN-MARC MOLINES

Univ. Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble, France

STEPHANIE LEROUX

Ocean Next, Grenoble, France

ABSTRACT

In this note, we provide details on *offline* diagnostics we are currently developing in order to recompute the Kinetic Energy (KE) budget of the eNATL60 simulation recently produced. This diagnostics are developed as CDFTOOLS, i.e. offline FORTRAN codes based on the NEMO General Circulation Model. We first provide the numerical details of the momentum equation as implemented in eNATL60 on which the KE equation builds upon. We then provide details of their implementation in the form of CDFTOOLS, and test the accuracy of this *offline* computation against model outputs. First tested at model time steps, we achieve a relatively good reproducibility of the terms associated with the momentum equation. The error associated with the *offline* computation, at the model time step, are of the order of 10^{-3} - 10^{-4} for time rate of change, advection, Coriolis and metric term and pressure gradients computated at the baroclinic time step. The surface pressure correction associated with the time-splitting scheme has proven difficult to implement offline, due to 1/ sub-domain boundary instabilities in the computation of the barotropic mode, and 2/ replication of the interpolation scheme used in NEMO for atmospheric forcing fields (atmospheric surface pressure, evaporation, precipitations, runoff). Sensitivity to the time discretization of the third order upstream biased part of UBS advecive scheme (i.e. forward in time) reveal that not accounting for this numerical detail degrades the accuracy of the recomputation by about one order of magnitude in sharp front regions, but is otherwise relatively weak when spatially integrated. Finally, the use of one hour model outputs leads to significant reduction of the accuracy, i.e. by about one order of magnitude. This appears to be the larger source of errors for these *offline* recomputations.

1. Introduction

In this note, we present recently developed tools which allows to recompute, from the model outputs, the momentum and Kinetic Energy (KE) budgets of the eNATL60 simulation. This tool is developed as part of the CDFTOOLS diagnostic package for the analysis of NEMO model output (<https://github.com/meom-group/CDFTOOLS.git>). The code, written in FORTRAN 90, follows the numerical implementation of the NEMO General Circulation Model (GCM; Madec

et al. 2017). Details for the momentum budget equation, on which the KE builds upon, are first given on Section 2. Implementation of the different terms contributing to this budget in the form of CDFTOOLS is detailed in Section 3, along with a model validation. This validation is made using model time steps model outputs of the MEDWEST60 simulation, which as been designed with the exact same numerical details than eNATL60, but focusing on the western Mediterranean Sea (Fig. 1).

2. The momentum budget in eNATL60/MEDWEST60

The momentum equation of the Primitive Equations under Boussinesq and hydrostatic approximations can be

*Corresponding author address: Univ. Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble, France
E-mail: quentin.jamet@univ-grenoble-alpes.fr

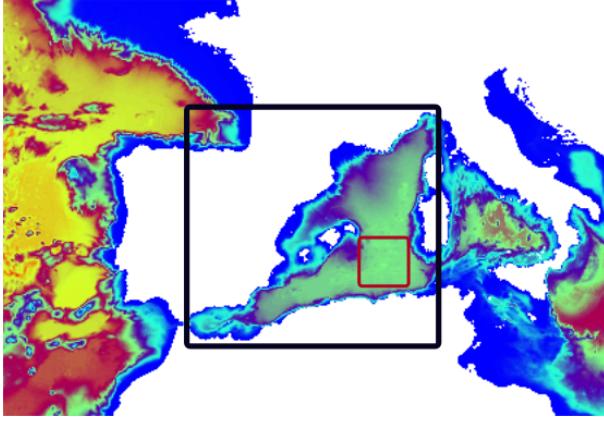


FIG. 1. **MEDWEST60 domain** The black rectangle represents the domain of the MEDWEST60 simulation, and the red rectangle the sub-domain used for validation in Section 3.

written in a flux form as (cf Eq. 2.14 in Madec et al. (2017)):

$$\partial_t u = -\nabla \cdot \mathbf{u}u + fv - \frac{1}{\rho_0} \partial_x p + \mathbf{F}_u \quad (1)$$

$$\partial_t v = -\nabla \cdot \mathbf{u}v - fu - \frac{1}{\rho_0} \partial_y p + \mathbf{F}_v \quad (2)$$

with $\mathbf{u} = (u, v, w)$ the tri-dimensional velocity field, $f = 2\Omega \sin(\phi)$ the Coriolis frequency, p is the pressure field (hydrostatic and surface), and $\mathbf{F}_{u,v}$ the forcing and dissipative terms. We summarize in the upcoming subsections the numerical implementation of these different terms as done in eNATL60/MEDWEST60. For completeness, the compiled version of the NEMO code for this configuration (without explicit tides) can be found at <https://github.com/meom-configurations/eNATL60-BLB002>.

a. Vertical grid

We first describe the vertical grid used in the eNATL60/MEDWEST60 simulations. This choice has implications for the computation of the momentum budget terms. The eNATL60/MEDWEST60 simulations are run with a z-coordinate vertical grid with partial steps (*zgr_zps*). It is computed by the NEMO routine *domzgr.F90*.

An additional scale factor is added to follow the non-linear free surface evolution, such that the vertical grid spacing becomes a variable of the model, i.e. it includes time and space variations (cf Section 3a for further details).

b. Time rate of change (*dynnxxt.F90* and *dynzdf.imp.F90*)

The model is integrated forward in time following a Leap-Frog scheme, which can be represented as:

$$x^{t+\Delta t} = x^{t-\Delta t} + 2\Delta t \text{RHS}^{t-\Delta t, t, t+\Delta t} \quad (3)$$

where x stands for u , v , T or S ; RHS is the Right-Hand-Side of the corresponding time evolution equation; Δt is the time step (40 sec in eNATL60 and 80 sec in MEDWEST60); and the superscripts indicate the time at which a quantity is evaluated. Each term of the RHS is evaluated at a specific time step depending on the associated physics. In the case of implicit vertical diffusion and non-linear free surface computation (cf Section 2h and Section 2f), a time-splitting option is used.

The Leap-Frog time stepping allows the coexistence of a numerical and a physical mode, thus the divergence of odd and even time steps may occur (cf Section 3.2 of Madec et al. (2017)). To prevent it, a Robert-Asselin time filter (Robert 1966; Asselin et al. 1972) is applied, such that:

$$x_F^t = x^t + \gamma [x_F^{t-\Delta t} - 2x^t + x^{t+\Delta t}] \quad (4)$$

where the subscript F denotes filtered values, and $\gamma = 0.1$ is the Asselin coefficient. Note that the modification of the filtering proposed by Leclair and Madec (2009) has not been accounted for in eNATL60/MEDWEST60, explaining the larger value of the Asselin coefficient as compared to the default value of $\gamma = 10^{-3}$ (Section 3.2 of Madec et al. 2017). This leads to a non-conservative scheme for tracer since the time filtering also applies on the forcing. The Asselin filtering momentum tendency is shown on Fig. 2 (right panel) and is compared to the total momentum tendency, i.e. before filtering (left panel).

Note the formulation of the trends in Section 6 of Madec et al. (2017) is given, in flux form, as:

$$NXT = COR + ADV + HPG + SPG + LDF + ZDF \quad (5)$$

where NXT refers to the time rate of change of momentum $\partial_t u$ (before application of the Asselin filter!). Other shorthands refer to the different terms of the budget, i.e. COR stands for Coriolis, ADV for advection and so on. However, when using the UBS scheme, trends due to horizontal advection are outputted in KEG , and trends due to vertical advection are outputted in ZAD . This is somewhat misleading with the documentation where KEG and ZAD refers to the vector invariant formulation. The actual budget, in flux form and using the UBS advective scheme, thus reads:

$$NXT = PVO + RVO + KEG + ZAD + HPG + SPG + ZDF \quad (6)$$

where PVO and RVO refers to the Coriolis and the metric term contributions, respectively. Note also that with

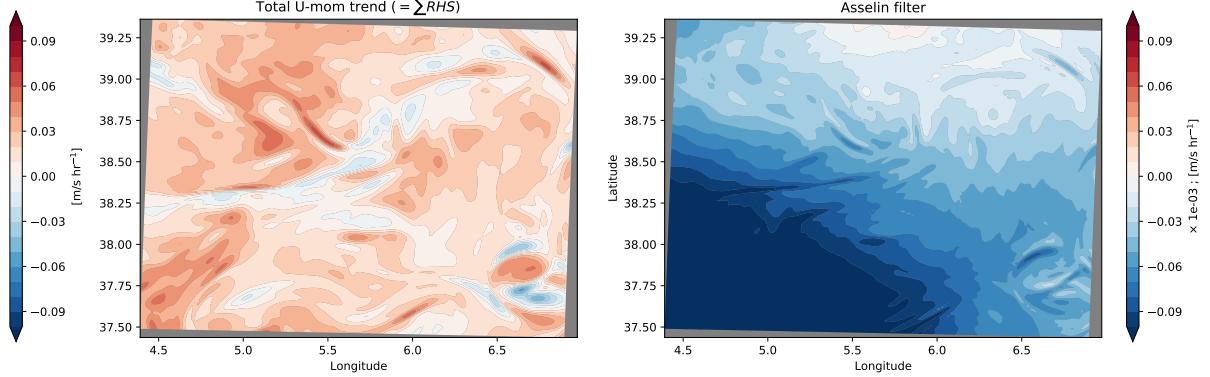


FIG. 2. **Asselin filter** Total surface layer U-momentum trend before Asselin time filtering ($\Sigma R H S$, left), and the correction associated with the Asselin filter (right). Note the 10^{-3} scale factor used for the Asselin filter on the right colorbar.

the UBS scheme, no lateral diffusion ($L D F$ in (5)) are required since this scheme is hyper-diffusive (cf Section 2c). Adding these terms together and comparing the RHS of (6) with time rate of change leads to a momentum budget at model time step closed with a machine precision (cf Fig. 3)

The incremental time stepping (3) is performed by *dynzdf_imp.F90* in the implicit vertical diffusion case, while the Asselin filtering is performed by *dynnxt.F90*.

c. Advection (*dynadv_ubs.F90*)

The advection term in eNATL60 is expressed in a flux form with the third order upstream biased scheme based on an upstream-biased parabolic interpolation (UBS, Shchepetkin and McWilliams 2005) (cf Section 6.3.2 in Madec et al. (2017)). This scheme has two terms, a (2^{nd} or 4^{th}) centred part (first term in Eq. (6.17) of Madec et al. (2017)) and a upstream biased third order part (second term in Eq. (6.17)). The former is evaluated with the *now* velocities, while the later is evaluated with the *before* velocities. This scheme is hyper-diffusive due to its upstream biased third order part. No explicit lateral diffusion on momentum is thus required.

Note that the centered part of the scheme is 4^{th} order as in ROMS. This is hard coded in *dynadv_ubs.F90* with the parameter $gamma2 = \frac{1}{32}$. In the documentation, it is stated about this 4^{th} order option that: "This is an error and should be suppressed soon.". It is nonetheless used in eNATL60/MEDWEST60.

The UBS scheme is only applied on the horizontal, the vertical advection term is computed following the 2^{nd} order centered scheme instead. The associated time-stepping is performed using a Leap-Frog scheme in conjunction with an Asselin time-filter, so u and v are the *now* velocities.

d. Coriolis and metric term (*dynvor.F90*)

In flux form, the vorticity term reduces to a Coriolis term in which the Coriolis parameter has been modified to account for the "metric" term. This *metric* term is meant to account for the curvilinear nature of the coordinate system, which emerges when the advection term of momentum is expressed in the so-called *flux form*. This altered Coriolis parameter is discretised at vorticity (f-) points as:

$$f + \zeta_{metric} = f + (v\partial_x\Delta y - u\partial_y\Delta x)^1 \quad (7)$$

This term is evaluated with an energy and enstrophy conserving scheme (*vor_een*), and using the *now* velocities (i.e. Leap-Frog scheme).

e. Hydrostatic pressure gradient (*dynhpg.F90*)

The hydrostatic pressure gradient trend is computed following the standard jacobian (s-coordinate) formulation (*hpg_sco*) with an explicit Leap-Frog time stepping, i.e. the *now* density. It follows Eq. (6.18) and (6.19) in Madec et al. (2017). The hydrostatic pressure can be obtained by integrating the hydrostatic equation vertically from the surface. However, the pressure is large at great depth while its horizontal gradient is several orders of magnitude smaller. This may lead to large truncation errors in the pressure gradient terms. To avoid such errors, it is favorable to instead vertically integrate the horizontal pressure gradient. Hydrostatic pressure gradient is then computed as the vertically integrated horizontal gradient of density anomaly:

$$\frac{1}{\rho_0} \nabla_h p(z) = \int_{\eta}^z \frac{1}{2} g \nabla_h \frac{\rho(z') - \rho_0}{\rho_0} dz' \quad (8)$$

with η the free surface elevation, $\rho_0 = 1026 \text{ Kg m}^{-3}$ the reference density and ρ the in-situ density computed with

¹There is a typo in Eq. (6.14) of Madec et al. (2017), it should read ' $[e_{2v}]'$ instead

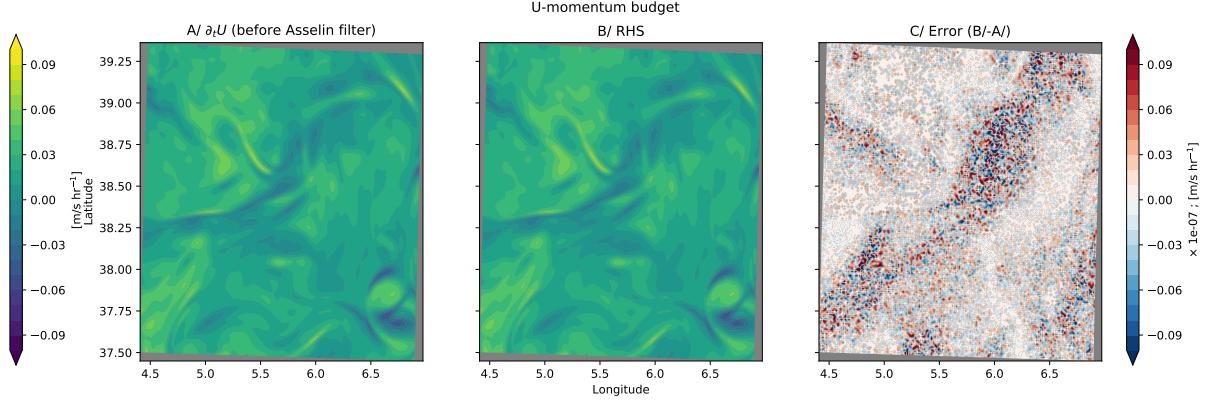


FIG. 3. U-momentum budget in NEMO Upper layre time rate of change $\partial_t U$ before the application of the Asselin filter (left), the associated RHS of (6) including the contribution from all terms in the budget (see next Sections for details ; center), and the absolute error in the budget (right). Note the $\times 10^{-7}$ scale factor in the errors.

modified polynomial TEOS-10 equation of state of Roquet et al. (2015).

Hydrostatic pressure gradient is a horizontal gradient, i.e. computed along geopotential surfaces. In the case of variable volume level (*lk_vvl*) associated with the non-linear free surface formulation, a correction is made on the pressure gradient between two adjacent model grid cells. Note that in this case (i.e. variable volume level), the surface pressure gradient is included in the hydrostatic pressure through the space and time variations of the vertical grid spacing. This provides a 'first guess' estimates of the surface pressure gradient term at the baroclinic time step, which is then 'corrected' through a more robust computation of the external mode in the case of the split-explicit formulation, detailed in Section f.

Additionally, with partial bottom cells, tracers in horizontally adjacent cells generally live at different depths. Before taking horizontal gradients between these tracer points, a linear interpolation is used to approximate the deeper tracer as if it actually lived at the depth of the shallower tracer point.

f. Surface pressure gradient (dynspg_ts.F90)

We recall that in the case of variable volume level (*lk_vvl*), a 'first guess' estimate of surface pressure gradient is computed at the baroclinic time step in the hydrostatic pressure subroutine *dynhpg.F90* through the space and time variations of the vertical grid spacing.

In the split-explicit formulation of Shchepetkin and McWilliams (2005) implemented in eNATL60/MEDWEST60, this 'first guess' estimate is then corrected through a more accurate computation of the external gravity wave allowed in the equations with the (non-)linear free surface formulation. The general idea is to solve the free surface equation and the associated barotropic velocity equations with a smaller time step

than the actual model time step (Δt), and add it to the general momentum trend. The size of the small time step used in eNATL60/MEDWEST60 is $\Delta t_e = \frac{\Delta t}{2*nn_baro}$, with *nn_baro* = 30. This computation is centered in time, starting from *before* iteration (*ln_bt_fw=FALSE*), proving a barotropic momentum trend centered at *now* time step.

The barotropic mode solves the following equations:

$$\partial_t \bar{\mathbf{u}}_h = -f \mathbf{k} \times \bar{\mathbf{u}}_h - g \nabla_h \eta - \frac{c_b^u}{H + \eta} \bar{\mathbf{u}}_h + \bar{\mathbf{G}} \quad (9)$$

$$\partial_t \eta = -\nabla \cdot [(H + \eta) \bar{\mathbf{u}}_h] + P - E \quad (10)$$

where $\bar{\mathbf{G}}$ is a forcing term held constant, containing coupling term between modes, surface atmospheric forcing (atmospheric pressure loading, wind stress), bottom friction, as well as slowly varying barotropic terms not explicitly computed to gain efficiency. The third term on the right hand side of (9) represents the bottom stress, explicitly accounted for at each barotropic iteration. The barotropic equations is integrated starting from *before* time step (*ln_bt_fw=false*), and a time filtering is applied on barotropic quantities (*ln_bt_av=true*) to avoid aliasing of fast barotropic motions into three dimensional equations. The baroclinic to barotropic forcing term is thus given at *now* time step, such that they are centred in the middle of the integration window. This removes part of splitting errors between modes and increases the overall numerical robustness. Since external mode equations written at baroclinic time steps finally follow a forward time stepping scheme, asselin filtering is not applied to barotropic quantities.

The atmospheric pressure loading is computed as an inverse barometer sea surface height as:

$$\eta_{ib} = \frac{1}{g\rho_0} (P_{atm} - P_0), \quad (11)$$

where P_{atm} is the atmospheric pressure and $P_0 = 101000.0 \text{ N m}^{-2}$ a reference atmospheric pressure.

g. Lateral diffusion

The advective scheme used (UBS) being hyperdiffusive, there is no lateral diffusion term computed for the specific stepup of eNATL60/MEDWEST60.

h. Vertical diffusion (*dynzdf.imp.F90*)

The vertical diffusion of momentum (note that it similarly applies on tracer) is computed with a backward (implicit) time differencing scheme (*ln_zdfexp=false*). This contribution is the last term evaluated. In the case of the *time splitting* option for the computation of surface pressure gradient (cf Section f), it is evaluated after the tracer equations, i.e. all the other terms for momentum trends are first computed, then the trends for tracer is computed, and then the trends of momentum due to vertical physics are evaluated and the model is step forward (cf *step.F90*).

The vertical physics subroutines solves an equation for

$$\mathbf{D}_{\mathbf{u}_h}^{vm} = \partial_z (\mathbf{A}_{\mathbf{u}_h}^{vm} \partial_z \mathbf{u}_h) \quad (12)$$

with $\mathbf{A}_{\mathbf{u}_h}^{vm}$ the vertical eddy viscosity coefficients. The way these coefficients are evaluated depends on the vertical physics used (cf below). Wind stress and bottom friction enter in the equation as surface and bottom boundary conditions for Eq. (12), respectively, such that:

$$(\mathbf{A}_{\mathbf{u}_h}^{vm} \partial_z \mathbf{u}_h) |_{z=\eta} = \frac{1}{\rho_0} \tau \quad (13)$$

and

$$(\mathbf{A}_{\mathbf{u}_h}^{vm} \partial_z \mathbf{u}_h) |_{z=-H} = \mathbf{F}_u^b \quad (14)$$

The surface stress τ_u^{surf} is computed following the CORE bulk flux formulation (Large and Yeager 2004).

The vertical diffusive flux at the bottom is computed as a quadratic bottom friction (cf Section 10.4.2 in Madec et al. (2017)), such that:

$$\mathbf{F}_u^b = c_b^{\mathbf{u}_h} \mathbf{u}_h^b = C_D \sqrt{u_b^2 + v_b^2 + e_b} \mathbf{u}_h^b \quad (15)$$

where C_D is a drag coefficient, and $e_b = 2.5 \cdot 10^{-3} \text{ m}^2 \text{s}^{-1}$ a bottom turbulent kinetic energy due to tides, internal waves breaking and other short time scale currents. The drag coefficient follows a log-layer formulation

$$C_D = \left(\frac{\kappa}{\log(0.5e3t/l_{bfr})} \right)^2 \quad (16)$$

where $\kappa = 0.4$ is the von-Karman constant, $e3t$ is the last wet layer grid point, and $l_{bfr} = 3 \cdot 10^{-3} \text{ m}$ is a roughness length. For stability, the drag coefficient is bounded following:

$$2.5 \cdot 10^{-3} < C_D < 1 \cdot 10^{-1} \quad (17)$$

The bottom friction is imposed in the code by adding the trend due to the bottom friction to the general momentum trend. For this purpose it is convenient to compute and store coefficients which can be simply combined with bottom velocities and geometric values to provide the momentum trend due to bottom friction. The coefficients $c_b^{(u,v)}$ in Eq. (15) are thus computed in *zdfbfr.F90*, and used in *dynzdf.imp.F90* as a bottom boundary condition for the computation of momentum trend. Bottom stress is thus computed with an implicit formulation, following what is done for the parameterization of the vertical physics. The coefficients $c_b^{(u,v)}$ computed by *dynzdf.imp.F90* are used for both baroclinic and barotropic modes, such that the action of bottom friction for the barotropic modes is linearized, i.e. it does not evolve with the smaller barotropic time step. Note that due to the use of the split-explicit time splitting time stepping, care must be taken to avoid the double counting of the bottom friction in the 2-D barotropic momentum equations (cf Section 10.4.6 of Madec et al. (2017)).

Away from surface and bottom boundaries, the parameterization used in eNATL60/MEDWEST60 to represent vertical physics and compute the vertical eddy viscosity coefficients $\mathbf{A}_{\mathbf{u}_h}^{vm}$ is the Turbulent Kinetic Energy (TKE) turbulent closure model (cf Section 10.1.3 of Madec et al. (2017)). It is based on a prognostic equation for the turbulent kinetic energy \bar{e} and a closure assumption for the turbulent length scales (e.g. Redelsperger and Sommeiria 1981; Mellor and Yamada 1982; Gaspar et al. 1990; Blanke and Delecluse 1993). The prognostic equation for $\bar{e} = \frac{1}{2}(u'^2 + v'^2 + w'^2)$ where (u', v', w') are the subgrid scale non-hydrostatic velocities and $\bar{\bullet}$ represents an averaging on the model grid cell, is approximated as:

$$\partial_t \bar{e} = K_m \left[(\partial_z u)^2 + (\partial_z v)^2 \right] - K_p N^2 + \partial_z [A^{vm} \partial_z \bar{e}] - c_e \frac{\bar{e}^{1/2}}{l_e} \quad (18)$$

where $K_m = C_k l_k \sqrt{\bar{e}}$, with $C_k = 0.1$ and l_k the mixing length scale; $K_p = \frac{A^{vm}}{P_r}$, with P_r the Prandtl number, set as a function of the local Richardson number R_i , and N^2 is Brunt-Vaisälä frequency; $C_e = \sqrt{2}/2 \approx 0.7$, and l_e is the dissipation length scale. [NEED TO INCLUDE THE COMPUTATION OF THE LENGTH SCALES l_k and l_e]. From left to right, these terms represent the production of \bar{e} through vertical shear, its destruction through stratification, its vertical diffusion, and its dissipation of Kolmogorov type (Kolmogorov 1942). At the surface, the value of \bar{e} is prescribed from the wind stress field as $\bar{e}_0 = e_{bb} \frac{|\tau|}{\rho_0}$, with e_{bb} a prescribed constant. To account for the surface wave breaking energy input, this constant is set to $e_{bb} = 67.83$, corresponding to a "wave age" of 100 (cf Eq. (10.10) of Madec et al. (2017) for further details). To avoid numerical instabilities associated with too weak vertical diffusion, a cut-off is applied on K_m ($10^{-4} \text{ m}^2 \text{s}^{-1}$) and K_p ($10^{-5} \text{ m}^2 \text{s}^{-1}$). And to avoid negative values in the

time integration of the \bar{e} equation, a second cut-off is applied on the minimum value of \bar{e} ($10^{-6} \text{ m}^2 \text{ s}^{-2}$), which is larger for surface values ($10^{-4} \text{ m}^2 \text{ s}^{-2}$).

The former TKE parameterization has been improved with two additional components to account for energy inputs due to Langmuir cells, and mixing just below the mixed layer. Their main action is to improve the formulation of the prognostic equation for \bar{e} near the surface. These parameterization have been used in eNATL60/MEDWEST60. Details can be found in Section 10.1.3 of Madec et al. (2017).

In NEMOv3.6, the momentum trends outputed by *dynzdf.F90* are not correct in the case of implicit formulation. In this case, the outputs produced by *dynzdf.imp.F90* are actual *after* velocities ua and va , while those quantities refer to trends in (almost) all the other routines. A correction is thus need to account for time stepping at the end of *dynzdf.F90* when momentum trends are computed. It reads:

$$ztrdu = \frac{ua - ub}{2\Delta t} - ztrdu \quad (19)$$

$$ztrdv = \frac{va - vb}{2\Delta t} - ztrdv \quad (20)$$

where $ztrdu$ and $ztrdv$ in the right hand side are the trends due to other terms (advection, pressure gradients, vorticity). This correction has been made by Robin Waldman in November 2019 (<http://forge.ipsl.jussieu.fr/nemo/attachment/ticket/1584/dynzdf.2.F90>), and should now be included in NEMOv4.

Note that since in this setup (eNATL60/MEDWEST60), lateral diffusion is handled by the UBS advection scheme, and surface and bottom stress are included as a boundary condition for solving the vertical diffusion problem, the diffusive term $\mathbf{D}_{\mathbf{u}_h}^{vm}$ in Eq. (12) completely describes the forcing and dissipative term $\mathbf{F}_{u,v}$ of Eq. (1) and (2).

i. The Kinetic Energy equation (*trdkn.F90*)

An equation for the Kinetic Energy (KE) is derived by multiplying the u- and v- momentum equations (1) and (2) by $\rho_0 u$ and $\rho_0 v$, respectively, where $\rho_0 = 1026 \text{ kg m}^{-3}$ is the reference density, and summing, leading to:

$$\partial_t K = -\nabla \cdot \mathbf{u}K - (u\partial_x p + v\partial_y p) + \rho_0(u\mathbf{D}_u^{vm} + v\mathbf{D}_v^{vm}) \quad (21)$$

where $K = \frac{\rho_0}{2}(u^2 + v^2)$ and \mathbf{u} the three dimensional velocity field.

In order to achieve machine precision in KE budget, care must be taken in the way the u- and v- momentum equations are interpolated on the same grid points. NEMO uses an Arakawa C-type grid with zonal (meridional) velocity points located to the east (north) of tracer points. The KE equation is thus computed at tracer (t-) points with

the following averaging:

$$K = \rho_0 \left(\overline{u * u_{trd} * e1e2e3_u^i} + \overline{v * v_{trd} * e1e2e3_v^j} \right) / e1e2e3_t \quad (22)$$

where $(u, v)_{trd}$ denotes trends associated with the different terms of the momentum budget, $e1e2e3_{u,v,t}$ the volume grid cell at u-, v- and t-points, respectively, and $\overline{\bullet}^i$ and $\overline{\bullet}^j$ the average of the respective zonal and meridional momentum neighbors of a given tracer point.

The KE budget for NEMO model outputs appears on Fig. 4. We note that this budget is not (yet) closed to machine precision due to the current lack of *online* estimates of time rate of change of kinetic energy $\partial_t K$. This term of the KE budget was not coded in NEMOv3.6 used for the production of eNATL60/MEDWEST60 simulations. It is thus recomputed *offline* following the procedure (22), with $u_{trd} = \partial_t u$ and $v_{trd} = \partial_t v$.

3. Offline computation of momentum trends

We have presented in the previous section the details of the momentum and Kinetic Energy equations as set up for the configurations eNATL60 and MEDWEST60. We now detail the different steps and approximations taken to adapt these *online* computations into an *offline* version, allowing the recomputation of the full momentum and Kinetic Energy budget of these simulations. As a general rule, stable numerical schemes used for time discretization are required to perform *online* simulations and avoid the potential growth of numerical instabilities. In an *offline* context however, such details are not mandatory since numerical instabilities are not able to grow. Errors made in the *offline* computation are however sensitive to these numerical details, such that we provide some test on the associated errors.

Along with the description of the *offline* computation of momentum trend terms, we provide a validation against model outputs. We use for this the MEDWEST60 configuration, which is an exact replication of the eNATL60 simulation, but run only on the western Mediterranean sea (Fig. 1, <https://github.com/stephanieлерoux/MEDWEST60>). The validation is first performed at the model time step. Errors associated with time averaged model outputs is further evaluated in Section 4.

a. Vertical grid (CDFTOOLS)

In the case of variable volume level (*vvl*), the vertical mesh is a three dimensional space and time variable. For storage limitation purposes, this variable has not been saved during the production of eNATL60. It thus need to be recomputed *offline* based on the free surface elevation η (Sea Surface Height, SSH) and the total depth of the ocean basin. From the model state *at rest* (i.e. $e3t_0$ and

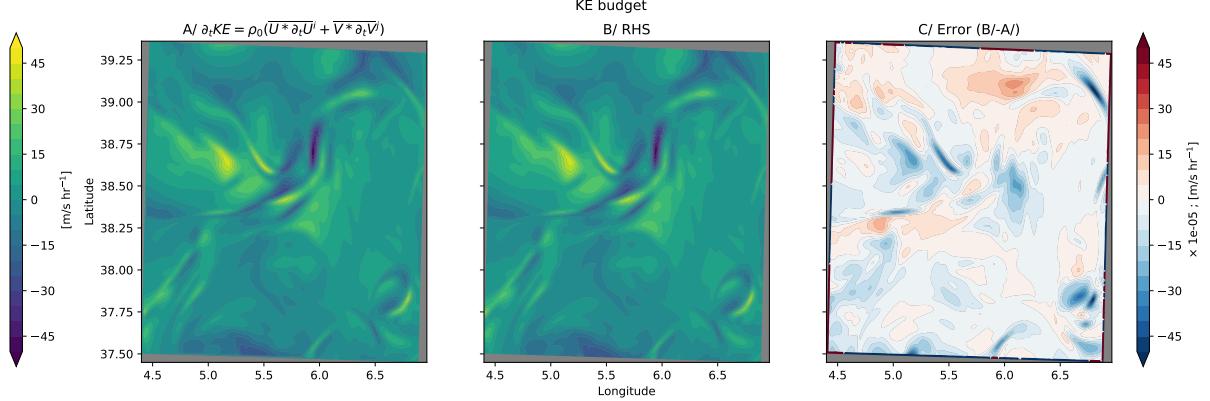


FIG. 4. **KE budget in NEMO** Upper layre time rate of change of kinetic energy $\partial_t K$ before the application of the Asselin filter (left), the associated RHS of (22) including the contribution from all terms in the budget, and the absolute error in the budget (right). Note the $\times 10^{-5}$ scale factor in the errors.

$\eta(x,y) = 0$), the vertical mesh $e3t(x,y,z,t)$ is recomputed as:

$$e3t = e3t_0 \left(1 + \frac{\eta}{H}\right) * tmask \quad (23)$$

where H is the depth of the ocean *at rest*. Vertical mesh at u-, v-, w- and f- points are interpolated following *domvvl.F90*. When using a z-coordinate with partial steps, the model reprocesses the bathymetry file provided as input to account for partial stepping near the bathymetry. If this '*model*' bathymetry is not available from model outputs, it can be recomputed as:

$$H(x,h) = \sum_{k=0}^N e3t_0(i,j,k) * tmask(i,j,k) \quad (24)$$

where k is the model level index and $N = 300$ the number of vertical levels, $e3t_0$ is the thickness of the model grid *at rest*, i.e. when $\eta(x,y) = 0$, and $tmask$ is the land mask on t-points.

b. Time rate of change (CDFTOOLS)

Time rate of change for momentum is computed by the CDFTOOLS *cdf_dyndt.f90*. To follow the flux form of the equations solved by NEMO in the eNATL60/MEDWEST60 configurations, and due to time varying vertical model grid, the trend is computed as follow:

$$ztrdu = \frac{u_a * e3u_a - u_b * e3u_b}{2 * \Delta t * e3u_n} \quad (25)$$

where subscripts b , n and a refers to quantities evaluated at *before*, *now* and *after* time step. This formulation has little effects on the absolute error (less than 5%) at model time step.

Performances of the offline computation are evaluated against model outputs (i.e. the *real* model trend) on Fig. 5. Although errors associated with the recomputation of the

time rate of change are about one order of magnitude larger than the correction due to the Asselin filter (cf Fig. 2), their pattern closely match, suggesting a strong link between the two.

c. Advection (CDFTOOLS)

Trends due to horizontal and vertical advection are computed by the CDFTOOLS *cdf_dynadv_ubs.f90*. Note that caution is required in the definition of vorticity points mask *fmask*. In eNATL60/MEDWEST60, a no-slip boundary conditions (*rn_shlat*=2) is applied at the lateral boundaries. This is done by setting the value of *fmask* to 2 along the coastline, providing a direct implementation of such a condition for the vorticity field (cf Section 8.1 of Madec et al. (2017) and *dommsk.F90* for implementation).

Performances of the offline computation are evaluated against model outputs (i.e. the *real* model trend) on Fig. 6. The offline computation performs relatively well, capturing the trends associated with horizontal and vertical advection with an accuracy of about 10^{-4} (Fig. 6). Errors are surface intensified (not shown). Large errors arise near the lateral boundary of the subdomain, which are associated with the evaluation of horizontal gradients and interpolations. They are irreducible and contaminate 3 grid points on each side.

As described in Section 2c, the UBS advective scheme has an upstream biased third order part which is evaluated using the *before* velocities (forward in time). In an *offline* version of the code, this numerical details are not required, and the use of time averaged data (see Section 4) would make such implementation hill posed. For latter use, these numerical details will not be used, inducing an additional level of errors in the *offline* estimates, as illustrated on Fig. 7. Not including this numerical detail induces a significant increase in the error made in the recomputation, going from 10^{-4} to 10^{-3} locally in sharp front regions.

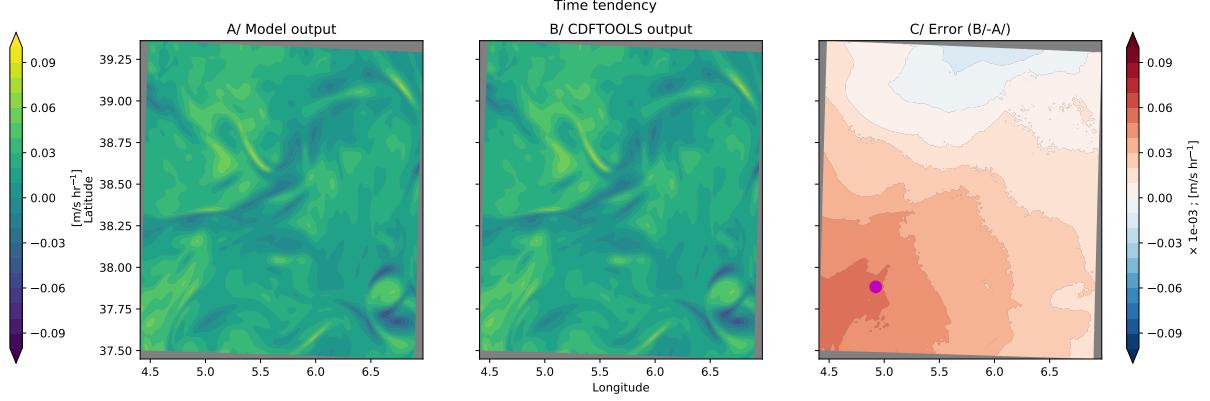


FIG. 5. Time rate of change Surface layer U-momentum trend $\partial_t u$ [$\text{m s}^{-1} \text{ hr}^{-1}$] for the model output as the sum of total momentum trend plus the Asselin filtering (*left*), its recomputed version with the CDFTOOL *cdf_dyndt.f90* following (25) (*center*), and the associated absolute error (*right*). Note the 10^{-3} scale factor used for errors.

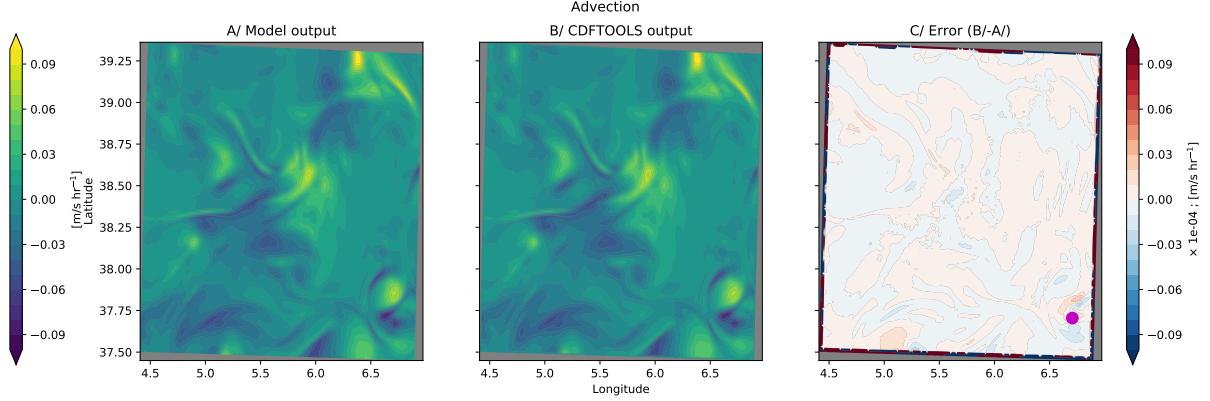


FIG. 6. Advection Surface layer U-momentum trend [$\text{m s}^{-1} \text{ hr}^{-1}$] due to three dimensional advection of momentum for the model output (*left*), its recomputed version with the CDFTOOL *cdf_dynaadv_ubs.f90* (*center*), and the associated absolute error (*right*). Note the 10^{-3} scale factor used for errors.

These errors tend to compositante when integrated horizontally (not shown).

d. Coriolis and metric term (CDFTOOLS)

Terms due to Coriolis and metric term are computed by the CDFTOOLS *cdf_dynvor.f90*. Comparison with model outputs appear on Fig. 8. The overall accuracy of this *offline* computation is of about 10^{-5} , providing a relatively good accuracy. Although small, the errors exhibit a strong barotropic signature (not shown).

e. Hydrostatic pressure gradient (CDFTOOLS)

Terms due to hydrostatic pressure gradients are computed by the CDFTOOLS *cdf_dynhpg.f90*. This includes the recomputation of the density from the TEOS-10 equation of state (Roquet et al. 2015). Comparisons with model outputs are shown on Fig. 9. The overall accuracy of this

offline computation is of about 10^{-4} , providing a relatively good accuracy.

f. Surface pressure gradient correction (CDFTOOLS)

Trends due to surface pressure gradient are computed by the CDFTOOLS *cdf_dynspg_ts.f90*.

In the eNATL60/MEDWEST60 NEMO configurations, a 'first guess' of surface pressure gradient is computed at the baroclinic time step through the use of the variable volume vertical grid spacing (cf Section f). In the split-explicit formulation used in these configurations, this 'first guess' surface pressure gradient is then corrected through a more robust computation of external modes made at a smaller time-step. This correction is outputted in the so-called 'spg' term, which we are considering here.

A first complexity in transposing this computation *offline* regards the sequence of updating momentum trends in NEMO. From now velocities, trends associated with

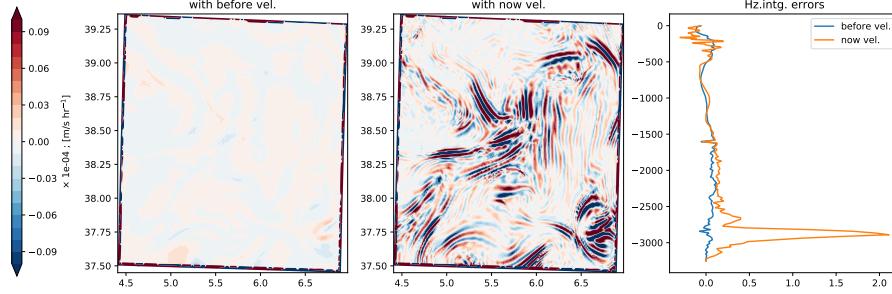


FIG. 7. Time discretization Errors on the *offline* estimates of advection u-momentum trends computed with *before* (left) and *now* (center) velocities, and the associated horizontally integrated errors (right).

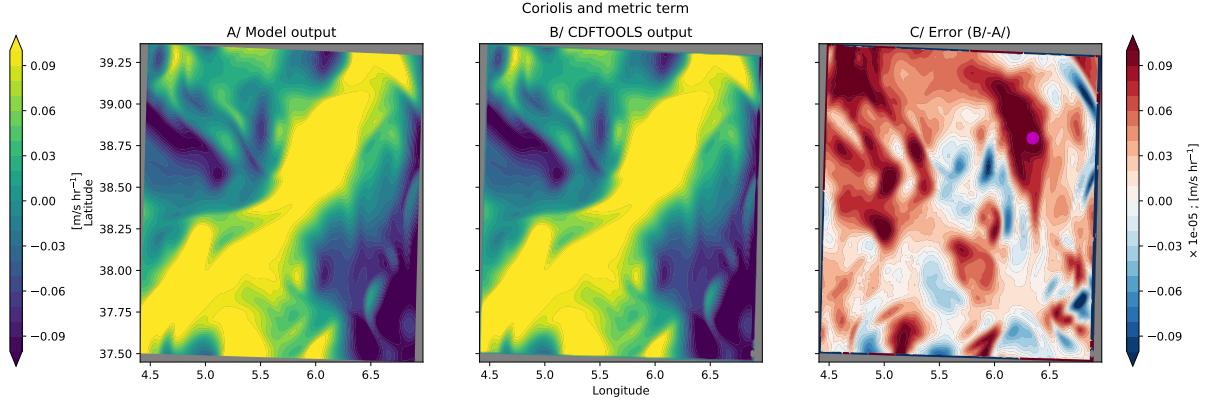


FIG. 8. Coriolis and metric term Surface layer U-momentum trend [$\text{m s}^{-1} \text{ hr}^{-1}$] due to Coriolis and metric term for the model output (left), its recomputed version with the CDFTOOL *cdf_dynvor,f90* (center), and the associated absolute error (right). Note the 10^{-5} scale factor used for errors.

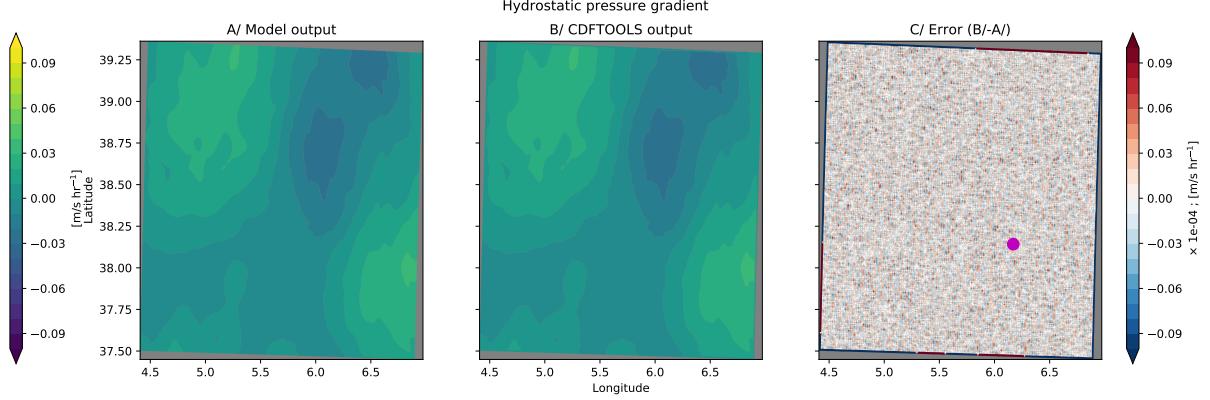


FIG. 9. Hydrostatic pressure gradient U-momentum trend [$\text{m s}^{-1} \text{ hr}^{-1}$] at $z = -1100$ m depth due to hydrostatic pressure gradient for the model output (left), its recomputed version with the CDFTOOL *cdf_dynhpg,f90* (center), and the associated absolute error (right). Note the 10^{-4} scale factor used for errors.

advection, vorticity and hydrostatic pressure gradient are first computed and sum all together. The split-explicit formulation of the surface pressure gradient then provide a correction of this general momentum trend with a more robust computation of external modes (which insure conser-

vation of tracers). This requires a communication between CDFTOOLS, which is not implemented (yet). Instead, the correction is made on the *total* momentum trend at *now*

time step, i.e.

$$ztrdu_n = \frac{ua - ub}{2\Delta t}. \quad (26)$$

Additionally, the external barotropic modes are computed with a reduced time step due to their fast dynamics. In eNATL60/MEDWEST, the sub-time step is $\frac{\Delta t}{30}$ (*nn_baro*=30 in namelist). This requires the implementation of the barotropic equations (9) and (10), thus allowing instabilities to grow, as illustrated on Figure 10.

Finally, this routine also deals with atmospheric pressure loading and freshwater fluxes associated with evaporation, precipitation and runoff. It appears that the interpolation algorithm in NEMOv3.6 suffers from inconsistency for these very high resolution simulations. As shown on Figure 11 (left panel), it seems that this interpolation is contaminated by discontinuities, likely associated with the sub-domain division used for multiprocessing. Such inconsistency are not reproducible offline, and are likely to strongly limit our ability to recompute the momentum trends associated with this term in a robust way.

Note that the contribution of this term, by its barotropic nature, has stronger impacts at depth where trends associated with other dynamical terms are small, but account for about 10% of the general momentum trend in the upper layers.

g. Vertical diffusion (CDFTOOLS)

Trends due to vertical viscous effects are computed by the CDFTOOLS *cdf_dynzdf.imp.f90*.

Computation of vertical momentum dissipation requires momentum diffusivity coefficients ($A_{u_h}^{vm}$ in (12)). This coefficients are available in model restarts, and need to be read in. This is not implemented yet, and the computation of these coefficients start from the background value of $10^{-4} \text{ m}^2 \text{ s}^{-1}$. Fig. 12 show the surface U-momentum trend associated at the end of the 2 hours available with the model time step model outputs to let the computation of viscous coefficient to adjust. The errors made in recomputing this term are much larger (10^{-1}) than those obtained for the other terms (except for the surface pressure gradient correction, cf Section f). These errors exhibits a strong 'barotropic' signature, where the *online* estimates of viscous effects have a constant value of 10^{-4} m s^{-2} , while its *offline* version show a zero background value. This need to be fixed.

Aside from this background value issue, errors may well be associated with the way this term is computed in NEMO, which is not exactly replicated in the *offline* version. Indeed, in NEMO, trends due to vertical viscous effects, when computed implicitly, are performed on the update model velocities, i.e. on the *now* velocities updated

by the trends computed by the other subroutines (advection, Coriolis, pressure terms). This would require communication between the cdftools, and is not implemented yet.

h. Kinetic energy (CDFTOOLS)

The computation of the kinetic energy associated with the momentum trends discussed in the previous section is performed within each CDFTOOLS following the procedure (22). The estimates of KE trends naturally inherit of difficulties encountered in the computation of the momentum trends, but no issues specific to the computation of KE trends have been found (so far ...).

4. Time averaged model outputs

We now turn our attention on the impacts of using time averaged quantities to recompute momentum and KE budgets. First performed at model time step for validation, we now extend our comparison between model outputs and *offline* computation of the different terms of the budgets based on time averaged quantities. In the case of eNATL60/MEDWEST60, three dimensional model outputs are available as hourly fields. We thus evaluate the errors associated with the recomputation of these terms based on hourly fields (U, V, W, T, S, SSH) against *true* estimates from model outputs. Results appear on Fig. 13 for the advective term. Note that in by performing *offline* computation based on time averaged model outputs, errors arises due to the time variations of the vertical grid size which follows the free surface evolution. These errors are however weak ($< 1\%$), except in shallow water region near the coast.

Acknowledgments. This work has been founded by the 'Make Our Planet Great Again' CONTaCTS project led by William K. Dewar. The code is available at the CDFTOOLS github repository <https://github.com/quentinjamet/CDFTOOLS>.

References

- Asselin, R., and Coauthors, 1972: Frequency filter for time integrations. *Mon. Wea. Rev.*, **100** (6), 487–490.
- Blanke, B., and P. Delecluse, 1993: Variability of the tropical Atlantic Ocean simulated by a general circulation model with two different mixed-layer physics. *J. Phys. Oceanogr.*, **23** (7), 1363–1388.
- Commission, I. O., and Coauthors, 2015: The International thermodynamic equation of seawater–2010: calculation and use of thermodynamic properties.[includes corrections up to 31st October 2015].
- Gaspar, P., Y. Grégoris, and J.-M. Lefevre, 1990: A simple eddy kinetic energy model for simulations of the oceanic vertical mixing: Tests at station papa and long-term upper ocean study site. *Journal of Geophysical Research: Oceans*, **95** (C9), 16 179–16 193.

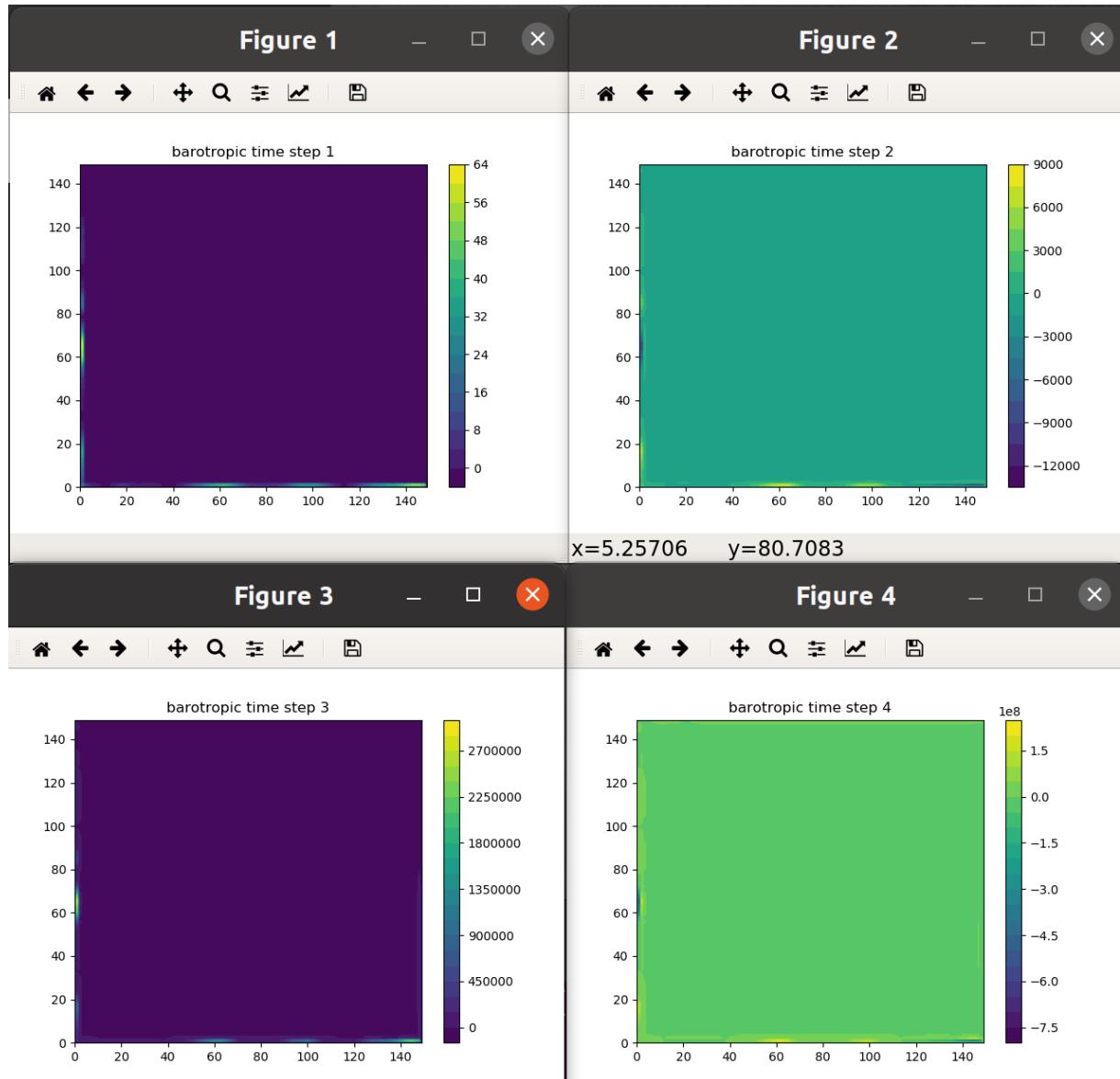


FIG. 10. Growth of instabilities in the time-splitting scheme, where each panel display the SSH for the 4 first barotropic time steps.

Kolmogorov, A., 1942: The equations of turbulent motion in an incompressible fluid (orig. in russian). *Izvestia Akad. Sci., USSR; Phys.*, **6**, 56–58.

Large, W. G., and S. G. Yeager, 2004: Diurnal to decadal global forcing for ocean and sea-ice models: the data sets and flux climatologies.

Leclair, M., and G. Madec, 2009: A conservative leapfrog time stepping method. *Ocean Modelling*, **30** (2-3), 88–94.

Madec, G., and Coauthors, 2017: Nemo ocean engine.

Mellor, G. L., and T. Yamada, 1982: Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics*, **20** (4), 851–875.

Redelsperger, J.-L., and G. Sommeria, 1981: Méthode de représentation de la turbulence d'échelle inférieure à la maille pour un modèle tri-dimensionnel de convection nuageuse. *Boundary-Layer Meteorology*, **21** (4), 509–530.

Robert, A. J., 1966: The integration of a low order spectral form of the primitive meteorological equations. *Journal of the Meteorological Society of Japan. Ser. II*, **44** (5), 237–245.

Roquet, F., G. Madec, T. J. McDougall, and P. M. Barker, 2015: Accurate polynomial expressions for the density and specific volume of seawater using the TEOS-10 standard. *Ocean Modelling*, **90**, 29–43.

Shchepetkin, A. F., and J. C. McWilliams, 2005: The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-

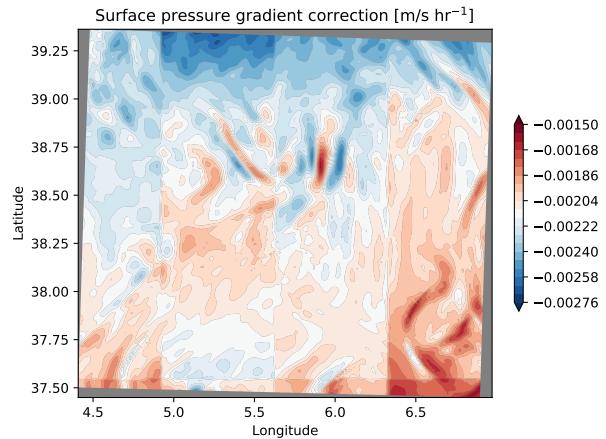


FIG. 11. Surface pressure correction term form the MEDWEST60 model simulation, i.e. the *online* estimates.

following-coordinate oceanic model. *Ocean modelling*, **9** (4), 347–404.

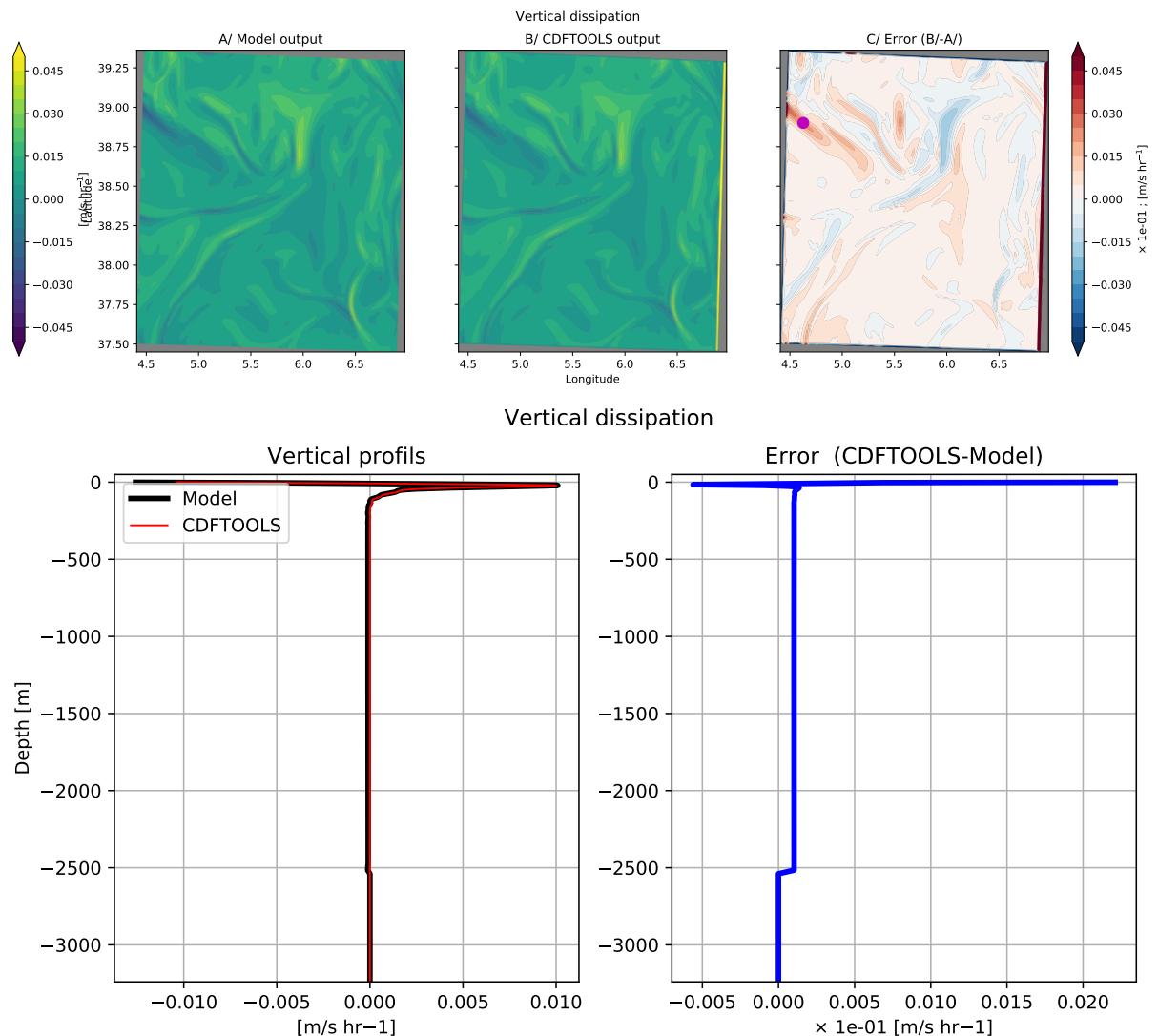


FIG. 12. **Vertical dissipation** Surface U-momentum trend [$\text{m s}^{-1} \text{ hr}^{-1}$] after 90 model time steps due to vertical viscous effects for the model output (*left*), its recomputed version with the CDFTOOL *cdf_dynamzdf_imp.f90* (*center*), and the associated absolute error (*right*). Note the 10^{-1} scale factor used for errors.

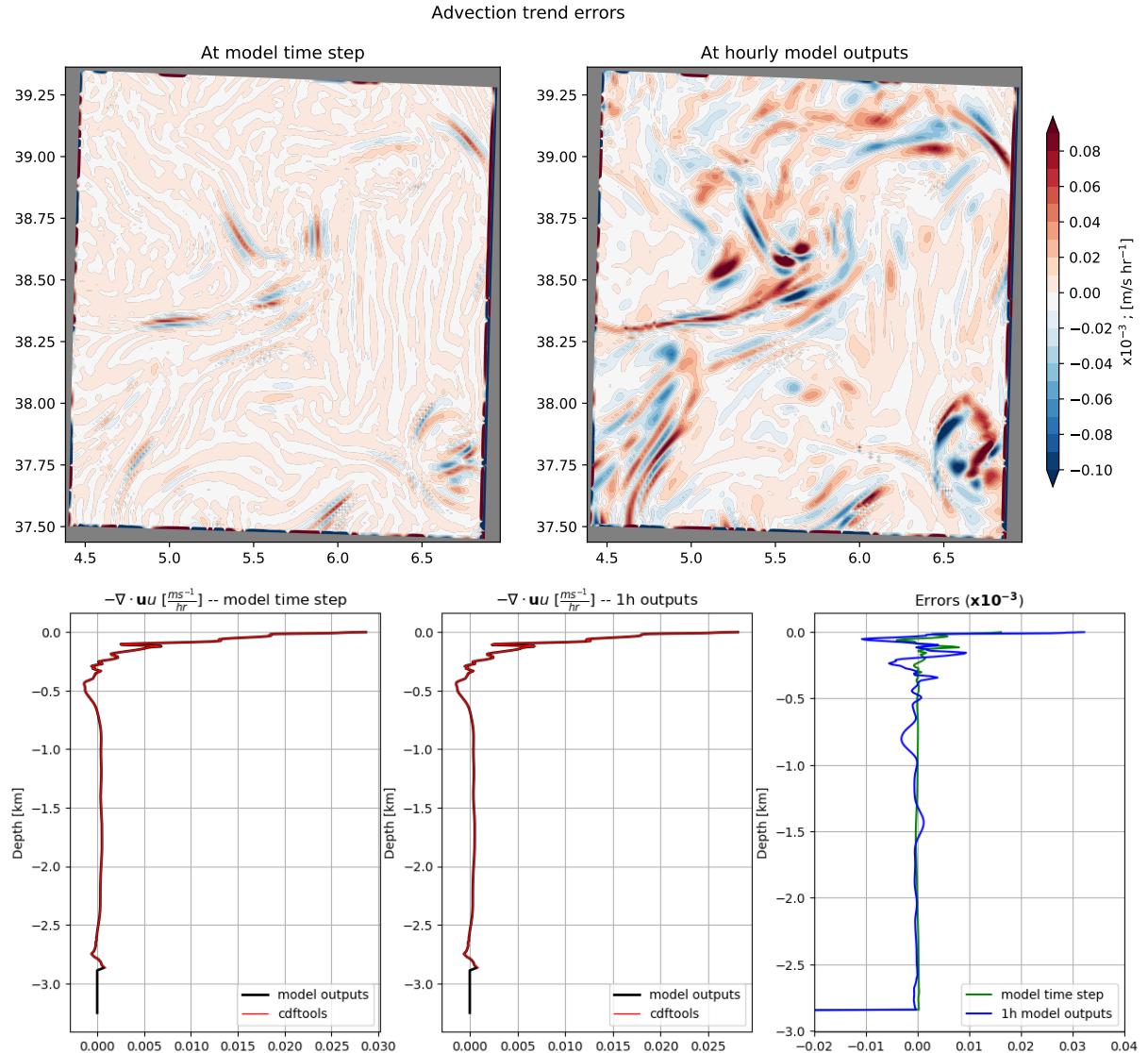


FIG. 13. Time averaging – Advection (Top) Surface u -momentum trend errors [$\times 10^{-3}$; $m s^{-1} hr^{-1}$] on advection based on (A/) model time step outputs, an (B/) hourly outputs. The errors are computed as in Fig. 6, i.e. when the upstream biased third order part of the UBS scheme is evaluated with *now* velocities. (Bottom) Associated vertical profils in a region of large error. Note the 10^{-3} scale factor used for errors.