

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

3 Quentin Jamet^{*}

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

5 William K. Dewar

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

8 Thierry Penduff, Julien Le Sommer, Jean-Marc Molines

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

10 Stephanie Leroux

11 *Ocean Next, Grenoble, France*

12 *Corresponding author address: Univ. Grenoble Alpes, CNRS, IRD, Grenoble INP, IGE, Grenoble,
13 France

14 E-mail: quentin.jamet@univ-grenoble-alpes.fr

ABSTRACT

15 In this note, we provide details on *offline* diagnostics we are currently
16 developping in order to recompute the Kinetic Energy (KE) budget of the
17 eNATL60 simulation recently produced. This diagnositcs are developped as
18 CDFTOOLS, i.e. offline FORTRAN codes based on the NEMO General Cir-
19 culation Model. We first provide the numerical details of the momentum equa-
20 tion as implemented in eNATL60 on which the KE equation builds upon. We
21 then provide details of their implementation in the form of CDFTOOLS, and
22 test the accuracy of this *offline* computation against model outputs. First tested
23 at model time steps, we achieve a relatively good reproducibility of the terms
24 associated with the momentum equation. The error associated with the *of-*
25 *line* computation, at the model time step, are of the order of 10^{-3} - 10^{-4} for
26 time rate of change, advection, Coriolis and metric term and pressure gradi-
27 ents computated at the baroclinic time step. The surface pressure correction
28 associated with the time-splitting scheme has proven difficult to implement
29 offline, due to 1/ sub-domain boundary instabilities in the computation of
30 the barotropic mode, and 2/ replication of the interpolation scheme used in
31 NEMO for atmospheric forcing fields (atmospheric surface pressure, evap-
32 oration, precipitations, runoff). Sensitivity to the time discretization of the
33 third order upstream biased part of UBS advecive scheme (i.e. forward in
34 time) reveal that not accouting for this numerical detail degrades the accuracy
35 of the recomputation by about one order of magnitude in sharp front regions,
36 but is otherwise relatively weak when spatially integrated. Finally, the use of
37 one hour model outputs leads to significant reduction of the accuracy, i.e. by
38 about one order of magnitude. This appears to be the larger source of errors
39 for these *offline* recomputations.

40 **1. Introduction**

41 In this note, we present recently developed tools which allows to recompute, from the model
42 outputs, the momentum and Kinetic Energy (KE) budgets of the eNATL60 simulation. This tool
43 is developped as part of the CDFTOOLS diagnostic package for the analysis of NEMO model out-
44 put (<https://github.com/meom-group/CDFTOOLS.git>). The code, written in FORTRAN 90,
45 follows the numerical implementation of the NEMO General Circulation Model (GCM; Madec
46 et al. 2017). Details for the momentum budget equation, on which the KE builds upon, are first
47 given on Section 2. Implementation of the different terms contributing to this budget in the form
48 of CDFTOOLS is detailed in Section 3, along with a model validation. This validation is made
49 using model time steps model outputs of the MEDWEST60 simulation, which has been designed
50 with the exact same numerical details than eNATL60, but focusing on the western Mediterranean
51 Sea (Fig. 1).

52 **2. The momentum budget in eNATL60/MEDWEST60**

53 The momentum equation of the Primitive Equations under Boussinesq and hydrostatic approxi-
54 mations can be written in a flux form as (cf Eq. 2.14 in Madec et al. (2017)):

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

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

55 with $\mathbf{u} = (u, v, w)$ the tri-dimensional velocity field, $f = 2\Omega \sin(\phi)$ the Coriolis frequency, p is
56 the pressure field (hydrostatique and surface), and $\mathbf{F}_{u,v}$ the forcing and dissipative terms. We
57 summarize in the upcoming subsections the numerical implementation of these different terms
58 as done in eNATL60/MEDWEST60. For completeness, the compiled version of the NEMO

⁵⁹ code for this configuration (without explicit tides) can be found at <https://github.com/>
⁶⁰ `meom-configurations/eNATL60-BLB002.`

⁶¹ *a. Vertical grid*

⁶² We first describe the vertical grid used in the eNATL60/MEDWEST60 simulations.
⁶³ This choice has implications for the computation of the momentum budget terms. The
⁶⁴ eNATL60/MEDWEST60 simulations are run with a z-coordinate vertical grid with partial steps
⁶⁵ (`zgr_zps`). It is computed by the NEMO routine `domzgr.F90`.

⁶⁶ An additional scale factor is added to follow the non-linear free surface evolution, such that the
⁶⁷ vertical grid spacing becomes a variable of the model, i.e. it includes time and space variations (cf
⁶⁸ Section 3a for further details).

⁶⁹ *b. Time rate of change (`dynnxt.F90` and `dynzdf_imp.F90`)*

⁷⁰ The model is integrated forward in time following a Leap-Frog scheme, which can be repre-
⁷¹ sented as:

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

⁷² where x stands for u , v , T or S ; RHS is the Right-Hand-Side of the corresponding time evolution
⁷³ equation; Δt is the time step (40 sec in eNATL60 and 80 sec in MEDWEST60); and the super-
⁷⁴ scripts indicate the time at which a quantity is evaluated. Each term of the RHS is evaluated at a
⁷⁵ specific time step depending on the associated physics. In the case of implicit vertical diffusion
⁷⁶ and non-linear free surface computation (cf Section 2h and Section 2f), a time-splitting option is
⁷⁷ used.

⁷⁸ The Leap-Frog time stepping allows the coexistence of a numerical and a physical mode, thus
⁷⁹ the divergence of odd and even time steps may occur (cf Section 3.2 of Madec et al. (2017)). To

80 prevent it, a Robert-Asselin time filter (Robert 1966; Asselin et al. 1972) is applied, such that:

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

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

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

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

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

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

95 where PVO and RVO refers to the Coriolis and the metric term contributions, respectively. Note
96 also that with the UBS scheme, no lateral diffusion (LDF in (5)) are required since this scheme
97 is hyper-diffusive (cf Section 2c). Adding these terms together and comparing the RHS of (6)
98 with time rate of change leads to a momentum budget at model time step closed with a machine
99 precision (cf Fig. 3)

100 The incremental time stepping (3) is performed by *dynzdf_imp.F90* in the implicit vertical diffu-
101 sion case, while the Asselin filtering is permformed by *dynnxt.F90*.

102 *c. Advection (dynadv_ubs.F90)*

103 The advection term in eNATL60 is expressed in a flux form with the third order upstream biased
104 scheme based on an upstream-biased parabolic interpolation (UBS, Shchepetkin and McWilliams
105 2005) (cf Section 6.3.2 in Madec et al. (2017)). This scheme has two terms, a (2^{nd} or 4^{th}) centred
106 part (first term in Eq. (6.17) of Madec et al. (2017)) and a upstream biased third order part (second
107 term in Eq. (6.17)). The former is evaluated with the *now* velocities, while the later is evaluated
108 with the *before* velocities. This scheme is hyper-diffusive due to its upstream biased third order
109 part. No explicit lateral diffusion on momentum is thus required.

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

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

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

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

¹²² (f-) points as:

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

¹²³ This term is evaluated with an energy and enstrophy conserving scheme (*vor_een*), and using the
¹²⁴ *now* velocities (i.e. Leap-Frog scheme).

¹²⁵ e. *Hydrostatic pressure gradient (dynhpg.F90)*

¹²⁶ The hydrostatic pressure gradient trend is computed following the standard jacobian (s-
¹²⁷ coordinate) formulation (*hpg_sco*) with an explicit Leap-Frog time stepping, i.e. the *now* density.
¹²⁸ It follows Eq. (6.18) and (6.19) in Madec et al. (2017). The hydrostatic pressure can be obtained
¹²⁹ by integrating the hydrostatic equation vertically from the surface. However, the pressure is large
¹³⁰ at great depth while its horizontal gradient is several orders of magnitude smaller. This may lead
¹³¹ to large truncation errors in the pressure gradient terms. To avoid such errors, it is favorable to
¹³² instead vertically integrate the horizontal pressure gradient. Hydrostatic pressure gradient is then
¹³³ computed as the vertically integrated horizontal gradient of density anomaly:

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

¹³⁴ with η the free surface elevation, $\rho_0 = 1026 \text{ Kg m}^{-3}$ the reference density and ρ the in-situ density
¹³⁵ computed with modified polynomial TEOS-10 equation of state of Roquet et al. (2015).

¹³⁶ Hydrostatic pressure gradient is a horizontal gradient, i.e. computed along geopotential surfaces.
¹³⁷ In the case of variable volume level (*lk_vvl*) associated with the non-linear free surface formulation,
¹³⁸ a correction is made on the pressure gradient between two adjacent model grid cells. Note that in
¹³⁹ this case (i.e. variable volume level), the surface pressure gradient is included in the hydrostatic
¹⁴⁰ pressure through the space and time variations of the vertical grid spacing. This provides a 'first
¹⁴¹ guess' estimates of the surface pressure gradient term at the baroclinic time step, which is then

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

¹⁴² 'corrected' through a more robust computation of the external mode in the case of the split-explicit
¹⁴³ formulation, detailed in Section f.

¹⁴⁴ Additionally, with partial bottom cells, tracers in horizontally adjacent cells generally live at dif-
¹⁴⁵ ferent depths. Before taking horizontal gradients between these tracer points, a linear interpolation
¹⁴⁶ is used to approximate the deeper tracer as if it actually lived at the depth of the shallower tracer
¹⁴⁷ point.

¹⁴⁸ f. Surface pressure gradient (*dynspg_ts.F90*)

¹⁴⁹ We recall that in the case of variable volume level (*lk_vvl*), a 'first guess' estimate of surface
¹⁵⁰ pressure gradient is computed at the baroclinic time step in the hydrostatic pressure subroutine
¹⁵¹ *dynhpg.F90* through the space and time variations of the vertical grid spacing.

¹⁵² In the split-explicit formulation of Shchepetkin and McWilliams (2005) implemented in
¹⁵³ eNATL60/MEDWEST60, this 'first guess' estimate is then corrected through a more accurate
¹⁵⁴ computation of the external gravity wave allowed in the equations with the (non-)linear free surface
¹⁵⁵ formulation. The general idea is to solve the free surface equation and the associated barotropic
¹⁵⁶ velocity equations with a smaller time step than the actual model time step (Δt), and add it to
¹⁵⁷ the general momentum trend. The size of the small time step used in eNATL60/MEDWEST60
¹⁵⁸ is $\Delta t_e = \frac{\Delta t}{2*nn_baro}$, with *nn_baro* = 30. This computation is centered in time, starting from *before*
¹⁵⁹ iteration (*ln_bt_fw=.FALSE.*), proving a barotropic momentum trend centered at *now* time step.

¹⁶⁰ The barotropic mode solves the following equations:

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

¹⁶¹

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

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

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

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

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

176 g. Lateral diffusion

177 The advective scheme used (UBS) being hyper-diffusive, there is no lateral diffusion term com-
 178 puted for the specific stepup of eNATL60/MEDWEST60.

179 h. Vertical diffusion (*dynzdf_imp.F90*)

180 The vertical diffusion of momentum (note that it similarly applies on tracer) is computed with a
 181 backward (implicit) time differencing scheme (*ln_zdfexp=false*). This contribution is the last term
 182 evaluated. In the case of the *time splitting* option for the computation of surface pressure gradient

¹⁸³ (cf Section f), it is evaluated after the tracer equations, i.e. all the other terms for momentum trends
¹⁸⁴ are first computed, then the trends for tracer is computed, and then the trends of momentum due
¹⁸⁵ to vertical physics are evaluated and the model is step forward (cf *step.F90*).

¹⁸⁶ The vertical physics subroutines solves an equation for

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

¹⁸⁷ with $\mathbf{A}_{\mathbf{u}_h}^{vm}$ the vertical eddy viscosity coefficients. The way these coefficients are evaluated depends
¹⁸⁸ on the vertical physics used (cf bellow). Wind stress and bottom friction enter in the equation as
¹⁸⁹ surface and bottom boundary conditions for Eq. (12), respectively, such that:

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

¹⁹⁰ and

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

¹⁹¹ The surface stress $\tau_{\mathbf{u}}^{surf}$ is computed following the CORE bulk flux formulation (Large and Yeager
¹⁹² 2004).

¹⁹³ The vertical diffusive flux at the bottom is computed as a quadratic bottom friction (cf Section
¹⁹⁴ 10.4.2 in Madec et al. (2017)), such that:

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

¹⁹⁵ 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
¹⁹⁶ tides, internal waves breaking and other short time scale currents. The drag coefficient follows a
¹⁹⁷ log-layer formulation

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

₁₉₈ where $\kappa = 0.4$ is the von-Karman constant, $e3t$ is the last wet layer grid point, and $l_{bfr} = 3 \cdot 10^{-3}$
₁₉₉ m is a roughness length. For stability, the drag coefficient is bounded following:

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

₂₀₀ The bottom friction is imposed in the code by adding the trend due to the bottom friction to the
₂₀₁ general momentum trend. For this purpose it is convenient to compute and store coefficients which
₂₀₂ can be simply combined with bottom velocities and geometric values to provide the momentum
₂₀₃ trend due to bottom friction. The coefficients $c_b^{(u,v)}$ in Eq. (15) are thus computed in *zdfbfr.F90*,
₂₀₄ and used in *dynzdf_imp.F90* as a bottom boundary condition for the computation of momentum
₂₀₅ trend. Bottom stress is thus computed with an implicit formulation, following what is done for
₂₀₆ the parameterization of the vertical physics. The coefficients $c_b^{(u,v)}$ computed by *dynzdf_imp.F90*
₂₀₇ are used for both baroclinic and barotropic modes, such that the action of bottom friction for the
₂₀₈ barotropic modes is linearized, i.e. it does not evolve with the smaller barotropic time step. Note
₂₀₉ that due to the use of the split-explicit time splitting time stepping, care must be taken to avoid
₂₁₀ the double counting of the bottom friction in the 2-D barotropic momentum equations (cf Section
₂₁₁ 10.4.6 of Madec et al. (2017)).

₂₁₂ Away from surface and bottom boundaries, the parameterization used in
₂₁₃ eNATL60/MEDWEST60 to represent vertical physics and compute the vertical eddy vis-
₂₁₄ cosity coefficients $\mathbf{A}_{\mathbf{u}_h}^{vm}$ is the Turbulent Kinetic Energy (TKE) turbulent closure model (cf Section
₂₁₅ 10.1.3 of Madec et al. (2017)). It is based on a prognostic equation for the turbulent kinetic energy
₂₁₆ \bar{e} and a closure assumption for the turbulent length scales (e.g. Redelsperger and Sommeria
₂₁₇ 1981; Mellor and Yamada 1982; Gaspar et al. 1990; Blanke and Delecluse 1993). The prognostic
₂₁₈ equation for $\bar{e} = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2})$ where (u', v', w') are the subgrid scale non-hydrostatic

219 velocities and $\overline{\cdot}$ represents an averaging on the model grid cell, is approximated as:

$$\partial_t \bar{e} = K_m \left[(\partial_z u)^2 + (\partial_z v)^2 \right] - K_\rho N^2 + \partial_z [A^{vm} \partial_z \bar{e}] - c_\varepsilon \frac{\bar{e}^{1/2}}{l_\varepsilon} \quad (18)$$

220 where $K_m = C_k l_k \sqrt{\bar{e}}$, with $C_k = 0.1$ and l_k the mixing length scale; $K_\rho = \frac{A^{vm}}{P_{rt}}$, with P_{rt} the Prandtl
221 number, set as a function of the local Richardson number R_i , and N^2 is Brunt-Vaisälä frequency;
222 $C_\varepsilon = \sqrt{2}/2 \approx 0.7$, and l_ε is the dissipation length scale. [NEED TO INCLUDE THE COMPUTU-
223 TATION OF THE LENGTH SCALES l_k and l_ε]. From left to right, these terms represent the
224 production of \bar{e} through vertical shear, its destruction through stratification, its vertical diffusion,
225 and its dissipation of Kolmogorov type (Kolmogorov 1942). At the surface, the value of \bar{e} is pre-
226 scribed from the wind stress field as $\bar{e}_0 = e_{bb} \frac{|\tau|}{\rho_0}$, with e_{bb} a prescribed constant. To account for
227 the surface wave breaking energy input, this constant is set to $e_{bb} = 67.83$, corresponding to a
228 "wave age" of 100 (cf Eq. (10.10) of Madec et al. (2017) for further details). To avoid numerical
229 instabilities associated with too weak vertical diffusion, a cut-off is applied on K_m ($10^{-4} \text{ m}^2 \text{ s}^{-1}$)
230 and K_ρ ($10^{-5} \text{ m}^2 \text{ s}^{-1}$). And to avoid negative values in the time integration of the \bar{e} equation, a
231 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 val-
232 ues ($10^{-4} \text{ m}^2 \text{ s}^{-2}$). Note that, to be energetically consistant, parts of this equation are discretized
233 backward in time, and that the product of eddy coefficient by the shear at time t and $t - \Delta t$ must
234 be performed prior to the spatial interpolation between the w -point (where it the TKE equation is
235 solved) and the u, v, t -points where it is applied, thus the time variation of the vertical grid factors
236 need to be accounted for (cf Section 10.1.4 of Madec et al. 2017).

237 The former TKE parameterization has been improved with two additional components to ac-
238 count for energy inputs due to Langmuir cells, and mixing just below the mixed layer. Their main
239 action is to improve the formulation of the prognostic equation for \bar{e} near the surface. These pa-

240 parameterization have been used in eNATL60/MEDWEST60. Details can be found in Section 10.1.3
 241 of Madec et al. (2017).

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

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

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

247 where $ztrdu$ and $ztrdv$ in the right hand side are the trends due to other terms (advection, pres-
 248 sure gradients, vorticity). This correction has been made by Robin Waldman in November
 249 2019 (<http://forge.ipsl.jussieu.fr/nemo/attachment/ticket/1584/dynzdf.2.F90>),
 250 and should now be included in NEMOv4.

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

255 i. The Kinetic Energy equation (*trdken.F90*)

256 From the model prognostic variables, an equation for the Kinetic Energy (KE) is derived by
 257 multiplying the u- and v- momentum equations (1) and (2) by $\rho_0 u$ and $\rho_0 v$, respectively, where
 258 $\rho_0 = 1026 \text{ kg m}^{-3}$ is the reference density, and summing, leading to:

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

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

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

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

264 where $(u, v)_{trd}$ denotes trends associated with the different terms of the momentum budget,
265 $e1e2e3_{u,v,t}$ the volume grid cell at u-, v- and t-points, respectively, and $\bar{\bullet}^i$ and $\bar{\bullet}^j$ the average
266 of the respective zonal and meridional momentum neighbors of a given tracer point.

267 The KE budget for NEMO model outputs appears on Fig. 4, panel 9. The accuracy of this budget
268 as compared to the machine precision budget obtained for the momentum (Fig. 3) likely reflects
269 the contamination of the Asselin filter applied to the *now* velocities used to construct the kinetic
270 energy equation.

271 We point out that, although the Coriolis and metric term are not expected to contribute to the
272 kinetic energy, i.e. $u(f + \zeta_{metric})v - v(f + \zeta_{metric})u$ should be identically zero, this statement
273 depends on the numerical scheme used to discretize this term. eNATL60/MEDWEST60 uses
274 the kinetic energy and the potential enstrophy scheme of ?. This scheme conserves energy and
275 enstrophy in an integrated sense, but leads to significant contribution of this term at the model grid
276 point. We have verified that this contribution, once integrated over the sub-domain used for this
277 work (150x150 grid points), this contribution decreases to less than 1% of the horizontally integrated
278 KE budget.

279 **3. Offline computation of momentum trends**

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

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

295 *a. Vertical grid (CDFTOOLS)*

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

300 $\eta(x,y) = 0$), the vertical mesh $e3t(x,y,z,t)$ is recomputed as:

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

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

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

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

307 b. Time rate of change (CDFTOOLS)

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

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

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

313 Performances of the offline computation are evaluated against model outputs (i.e. the *real* model
314 trend) on Fig. 5. Although errors associated with the recomputation of the time rate of change are
315 about one order of magnitude larger than the correction due to the Asselin filter (cf Fig.2), their
316 pattern closely match, suggesting a strong link between the two.

317 *c. Advection (CDFTOOLS)*

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

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

330 As described in Section 2c, the UBS advective scheme has an upstream biased third order part
331 which is evaluated using the *before* velocities (forward in time). In an *offline* version of the code,
332 this numerical details are not required, and the use of time averaged data (see Section 4) would
333 make such implementation hill posed. For latter use, these numerical details will not be used,
334 inducing an additional level of errors in the *offline* estimates, as illustrated on Fig. 7. Not including
335 this numerical detail induces a significant increase in the error made in the recomputation, going
336 from 10^{-4} to 10^{-3} locally in sharp front regions. These errors tend to composante when integrated
337 horizontally (right panel of Fig. 7), but exhibit larger integrated errors at depth (2500-3000 m)
338 associated with errores made along the bathymetry.

339 *d. Coriolis and metric term (CDFTOOLS)*

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

344 *e. Hydrostatic pressure gradient (CDFTOOLS)*

345 Terms due to hydrostatic pressure gradients are computed by the CDFTOOLS *cdf_dynhpg.f90*.
346 This includes the recomputation of the density from the TEOS-10 equation of state (Roquet et al.
347 2015). Comparisons with model outputs are shown on Fig. 9. The overall accuracy of this *offline*
348 computation is of about 10^{-4} , providing a relatively good accuracy.

349 *f. Surface pressure gradient correction (CDFTOOLS)*

350 Trends due to surface pressure gradient are computed by the CDFTOOLS *cdf_dynspg_ts.f90*.
351 In the eNATL60/MEDWEST60 NEMO configurations, a 'first guess' of surface pressure gra-
352 dient is computed at the baroclinic time step through the use of the variable volume vertical grid
353 spacing (cf Section f). In the split-explicit formulation used in these configurations, this 'first
354 guess' surface pressure gradient is then corrected through a more robust computation of external
355 modes made at a smaller time-step. This correction is outputed in the so-called 'spg' term, which
356 we are considering here.

357 A first complexity in transposing this computation *offline* regards the sequence of updating mo-
358 mentum trends in NEMO. From now velocitites, trends associated with advection, vorticity and
359 hydrostatic pressure gradient are first computed and sum all together. The split-explicit formu-
360 lation of the surface pressure gradient then provide a correction of this general momentum trend

361 with a more robust computation of external modes (which insure conservation of tracers). This
362 requires a communication between CDFTOOLS, which is not implemented (yet). Instead, the
363 correction is made on the *total* momentum trend at *now* time step, i.e.

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

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

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

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

378 *g. Vertical diffusion (CDFTOOLS)*

379 Trends due to vertical viscous effects are computed by the CDFTOOLS *cdf_dynzdf_imp.f90*.
380 Computation of vertical momentum dissipation requires momentum diffusivity coefficients ($A_{\mathbf{u}_h}^{vm}$
381 in (12)). These coefficients are available in model restarts, and need to be read in. This is not imple-

mented 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 (exempt 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.

395 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 ...).

401 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

404 comparison between model outputs and *offline* computation of the different terms of the budgets
 405 based on time averaged quantities. In the case of eNATL60/MEDWEST60, three dimensional
 406 model outputs are available as hourly fields. We thus evaluate the errors associated with the re-
 407 computation of these terms based on hourly fields (U, V, W, T, S, SSH) against *true* estimates from
 408 model outputs. Results appear on Fig. 13 for the advective term. Note that in by performing *offline*
 409 computation based on time averaged model outputs, errors arises due to the time variations of the
 410 vertical grid size which follows the free surface evolution. These errors are however weak ($\pm 1\%$),
 411 except in shallow water region near the coast.

412 5. Conclusion

$$\rho_0 \bar{u}^t \left(\begin{array}{c} \partial_t u = -\nabla \cdot \mathbf{u}u + fv - \frac{1}{\rho_0} \partial_x p + \mathbf{D}_u \\ \end{array} \right)^t \quad (27)$$

$$+ \rho_0 \bar{v}^t \left(\begin{array}{c} \partial_t v = -\nabla \cdot \mathbf{u}v - fu - \frac{1}{\rho_0} \partial_y p + \mathbf{D}_v \\ \end{array} \right)^t \quad (28)$$

$$\nabla \cdot \bar{\mathbf{u}}^t \hat{K} = -\nabla \cdot \bar{\mathbf{u}}^t \bar{p}^t - \bar{\epsilon}^t + \nabla v \nabla \hat{K} - \bar{\mathbf{u}}_h^t \nabla \cdot \bar{\mathbf{u}} \bar{\mathbf{u}}_h^t + \bar{w}^t \bar{b}^t \quad (29)$$

$$\nabla \cdot (\bar{\mathbf{u}}^t (\hat{K} + \bar{t}) + \bar{\mathbf{u}}_h^t) = -\bar{\epsilon}^t + \nabla v \nabla \hat{K} - \bar{\mathbf{u}}_h^t \nabla \cdot \bar{\mathbf{u}} \bar{\mathbf{u}}_h^t - \bar{w}' \bar{b}' \quad (30)$$

$$NXT = (KEG + ZAD) + (HPG + SPG_{1st \ guess}) + SPG_{correction} + ZDF \quad (31)$$

413 *Acknowledgments.* This work has been founded by the the 'Make Our Planet Great Again' CON-
 414 TaCTS project led by William K. Dewar. The code is available at the CDFTOOLS github reposi-
 415 tory <https://github.com/quentinjamet/CDFTOOLS>.

416 **References**

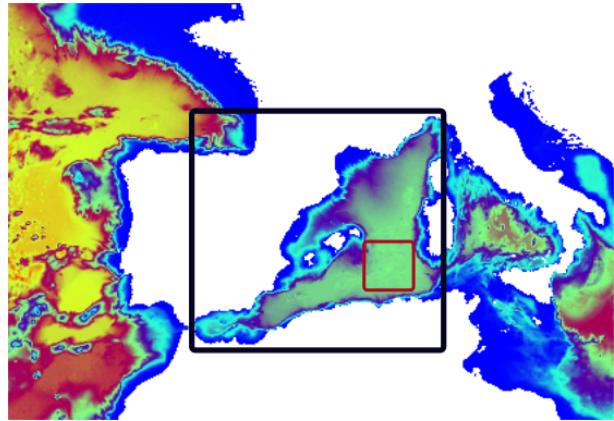
- 417 Asselin, R., and Coauthors, 1972: Frequency filter for time integrations. *Mon. Wea. Rev.*, **100** (6),
418 487–490.
- 419 Blanke, B., and P. Delecluse, 1993: Variability of the tropical Atlantic Ocean simulated by a
420 general circulation model with two different mixed-layer physics. *J. Phys. Oceanogr.*, **23** (7),
421 1363–1388.
- 422 Commission, I. O., and Coauthors, 2015: The International thermodynamic equation of seawater–
423 2010: calculation and use of thermodynamic properties.[includes corrections up to 31st October
424 2015].
- 425 Gaspar, P., Y. Grégoris, and J.-M. Lefevre, 1990: A simple eddy kinetic energy model for simula-
426 tions of the oceanic vertical mixing: Tests at station papa and long-term upper ocean study site.
427 *Journal of Geophysical Research: Oceans*, **95** (C9), 16 179–16 193.
- 428 Kolmogorov, A., 1942: The equations of turbulent motion in an incompressible fluid (orig. in
429 russian). *Izvestia Akad. Sci., USSR; Phys.*, **6**, 56–58.
- 430 Large, W. G., and S. G. Yeager, 2004: Diurnal to decadal global forcing for ocean and sea-ice
431 models: the data sets and flux climatologies.
- 432 Leclair, M., and G. Madec, 2009: A conservative leapfrog time stepping method. *Ocean Mod-
433 elling*, **30** (2-3), 88–94.
- 434 Madec, G., and Coauthors, 2017: Nemo ocean engine.
- 435 Mellor, G. L., and T. Yamada, 1982: Development of a turbulence closure model for geophysical
436 fluid problems. *Reviews of Geophysics*, **20** (4), 851–875.

- ⁴³⁷ Redelsperger, J.-L., and G. Sommeria, 1981: Méthode de représentation de la turbulence d'échelle
⁴³⁸ inférieure à la maille pour un modèle tri-dimensionnel de convection nuageuse. *Boundary-Layer*
⁴³⁹ *Meteorology*, **21** (4), 509–530.
- ⁴⁴⁰ Robert, A. J., 1966: The integration of a low order spectral form of the primitive meteorological
⁴⁴¹ equations. *Journal of the Meteorological Society of Japan. Ser. II*, **44** (5), 237–245.
- ⁴⁴² Roquet, F., G. Madec, T. J. McDougall, and P. M. Barker, 2015: Accurate polynomial expressions
⁴⁴³ for the density and specific volume of seawater using the TEOS-10 standard. *Ocean Modelling*,
⁴⁴⁴ **90**, 29–43.
- ⁴⁴⁵ Shchepetkin, A. F., and J. C. McWilliams, 2005: The regional oceanic modeling system (ROMS):
⁴⁴⁶ a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean mod-*
⁴⁴⁷ *elling*, **9** (4), 347–404.

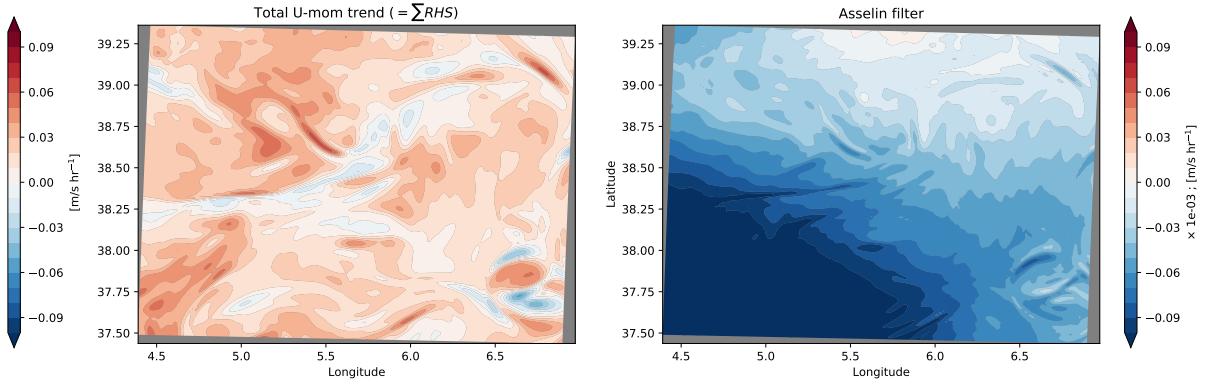
448 LIST OF FIGURES

449	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.	27
450			
451	Fig. 2. Asselin filter	Total surface layer U-momentum trend before Asselin time filtering (ΣRHS , 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.	28
452			
453			
454	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.	29
455			
456			
457			
458	Fig. 4. Decomposition of KE budget in NEMO	Contribution of each individual terms in the kinetic energy trends of the upper layer, and the balanced budget in panel 9.	30
459			
460	Fig. 5. Time rate of change	Surface layer U-momentum trend $\partial_t u$ [$m s^{-1} hr^{-1}$] for the model output as the sum of total momentum trend plus the Asselin filtering (left), its recomputed version with the CDFTOOL <code>cdf_dyndt.f90</code> following (25) (center), and the associated absolute error (right). Note the 10^{-3} scale factor used for errors.	31
461			
462			
463			
464	Fig. 6. Advection	Surface layer U-momentum trend [$m s^{-1} hr^{-1}$] due to three dimensional advection of momentum for the model output (left), its recomputed version with the CDFTOOL <code>cdf_dynadv_ubs.f90</code> (center), and the associated absolute error (right). Note the 10^{-3} scale factor used for errors.	32
465			
466			
467			
468	Fig. 7. Time discretization	Errors on the <i>offline</i> estimates of advection u-momentum trends computed with <i>before</i> (left) and <i>now</i> (center) velocities, and the associated horizontally integrated errors (right).	33
469			
470			
471	Fig. 8. Coriolis and metric term	Surface layer U-momentum trend [$m s^{-1} hr^{-1}$] due to Coriolis and metric term for the model output (left), its recomputed version with the CDFTOOL <code>cdf_dynamor.f90</code> (center), and the associated absolute error (right). Note the 10^{-5} scale factor used for errors.	34
472			
473			
474			
475	Fig. 9. Hydrostatic pressure gradient	U-momentum trend [$m s^{-1} hr^{-1}$] at $z = -1100$ m depth due to hydrostatic pressure gradient for the model output (left), its recomputed version with the CDFTOOL <code>cdf_dynhpg.f90</code> (center), and the associated absolute error (right). Note the 10^{-4} scale factor used for errors.	35
476			
477			
478			
479	Fig. 10. Growth of instabilities	in the time-splitting scheme, where each panel display the SSH for the 4 first barotropic time steps.	36
480			
481	Fig. 11. Surface pressure correction term	form the MEDWEST60 model simulation, i.e. the <i>online</i> estimates.	37
482			
483	Fig. 12. Vertical dissipation	Surface U-momentum trend [$m s^{-1} hr^{-1}$] after 90 model time steps due to vertical viscous effets for the model output (left), its recomputed version with the CDFTOOL <code>cdf_dynzdf_imp.f90</code> (center), and the associated absolute error (right). Note the 10^{-1} scale factor used for errors.	38
484			
485			
486			

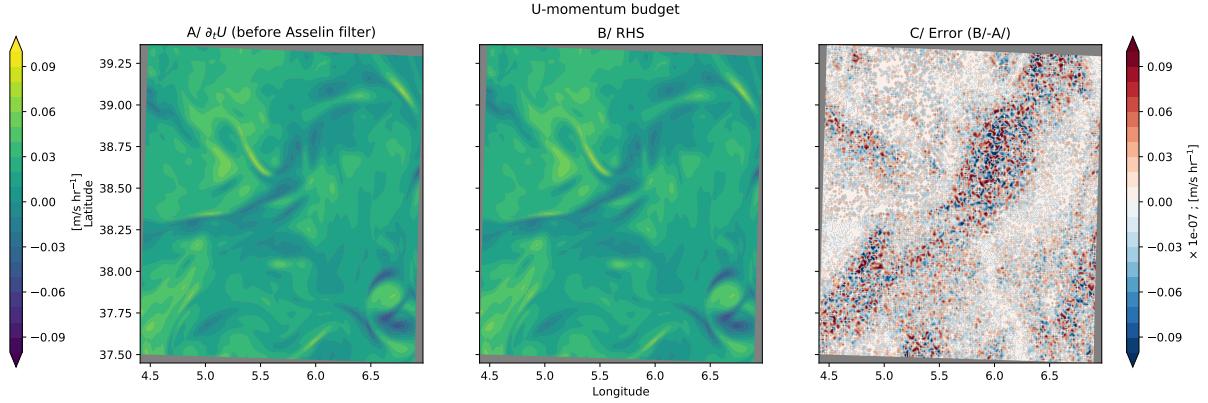
Fig. 13. Time averaging – Advection (Top) Surface u-momentum trend errors [$\times 10^{-3}$; $m s^{-1} hr^{-1}$] on advection based on (A/) model time step outputs, an (B/) hourly outputs. The errors are computed as in Fig. 6, i.e. when the upstream biased third order part of the UBS scheme is evaluated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



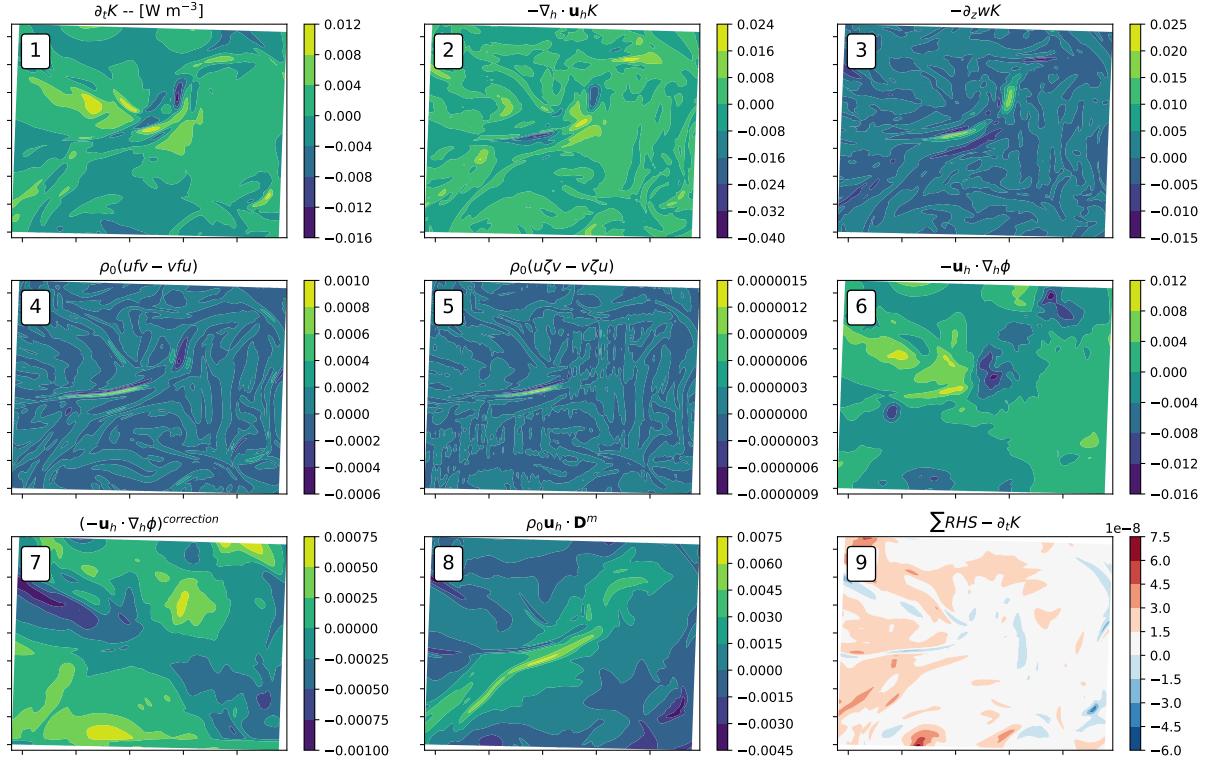
492 FIG. 1. **MEDWEST60 domain** The black rectangle represents the domain of the MEDWEST60 simulation,
493 and the red rectangle the sub-domain used for validation in Section 3.



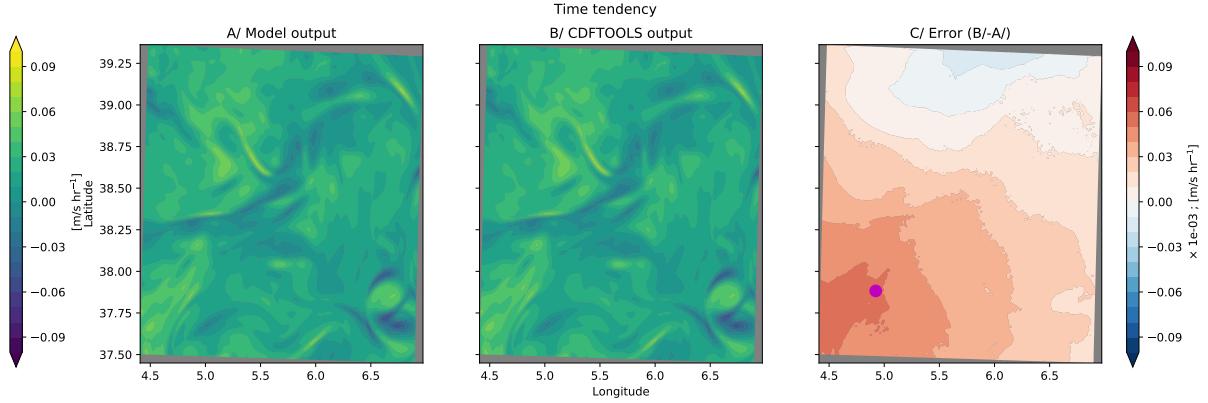
494 FIG. 2. **Asselin filter** Total surface layer U-momentum trend before Asselin time filtering (ΣRHS , left), and
 495 the correction associated with the Asselin filter (right). Note the 10^{-3} scale factor used for the Asselin filter on
 496 the right colorbar.



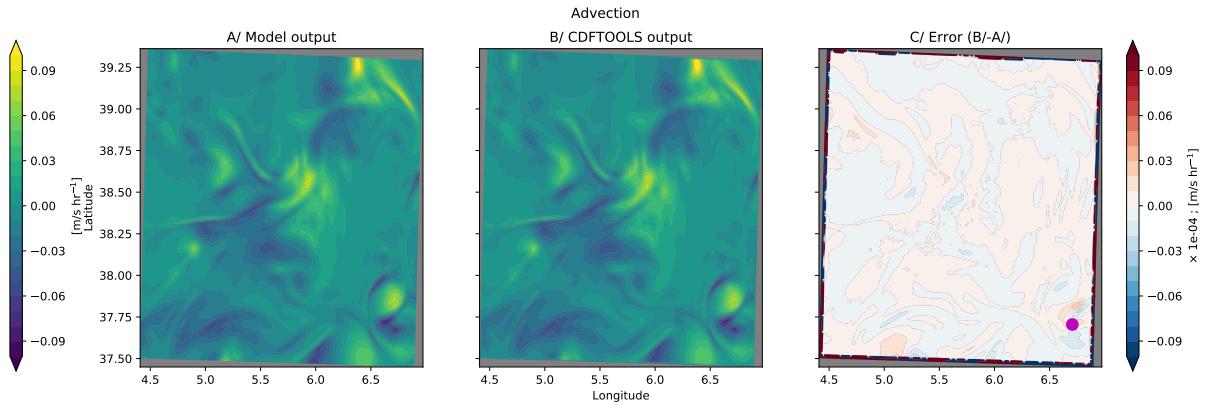
497 FIG. 3. **U-momentum budget in NEMO** Upper layre time rate of change $\partial_t U$ before the application of the
 498 Asselin filter (left), the associated RHS of (6) including the contribution from all terms in the budget (see next
 499 Sections for details ; center), and the absolute error in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



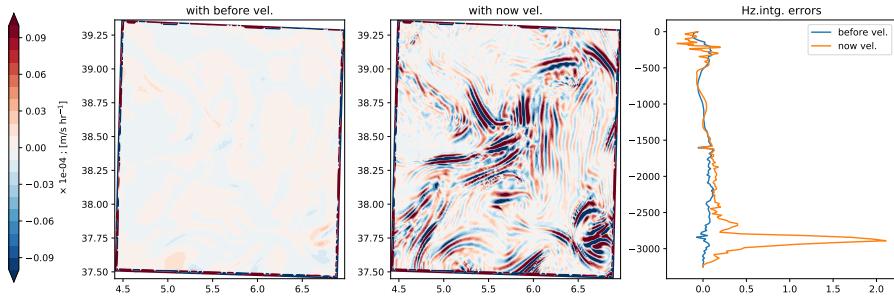
501 FIG. 4. **Decomposition of KE budget in NEMO** Contribution of each individual terms in the kinetic energy
 502 trends of the upper layer, and the balanced budget in panel 9.



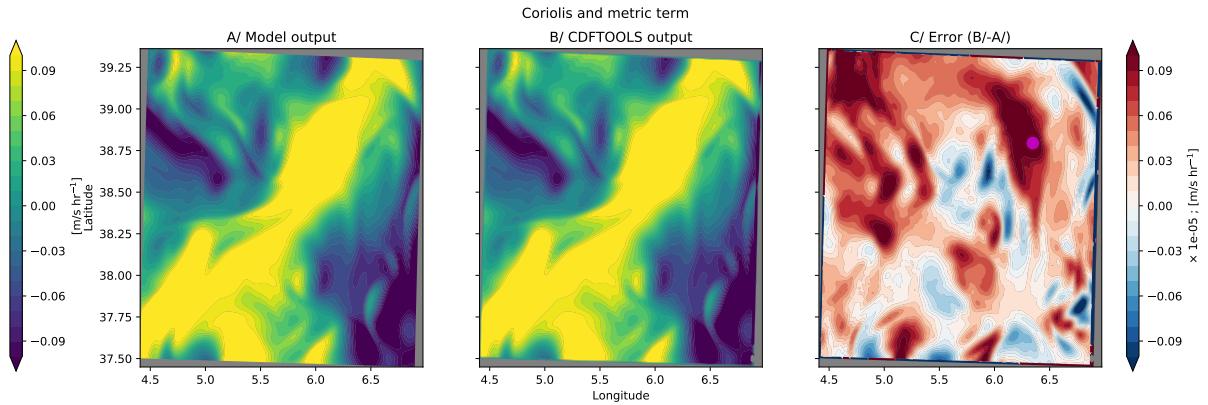
503 FIG. 5. **Time rate of change** Surface layer U-momentum trend $\partial_t u$ [$\text{m s}^{-1} \text{ hr}^{-1}$] for the model output as
 504 the sum of total momentum trend plus the Asselin filtering (*left*), its recomputed version with the CDFTOOL
 505 *cdf_dyndt.f90* following (25) (*center*),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



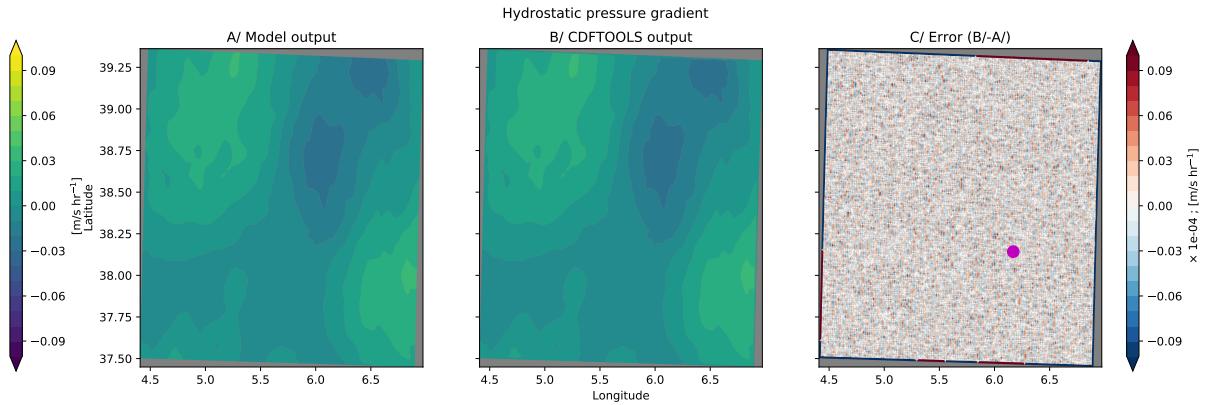
507 FIG. 6. **Advection** Surface layer U-momentum trend [$\text{m s}^{-1} \text{ hr}^{-1}$] due to three dimensional advection of
 508 momentum for the model output (*left*), its recomputed version with the CDFTOOL *cdf_dynadv_ubs.f90* (*center*),
 509 and the associated absolute error (*right*). Note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



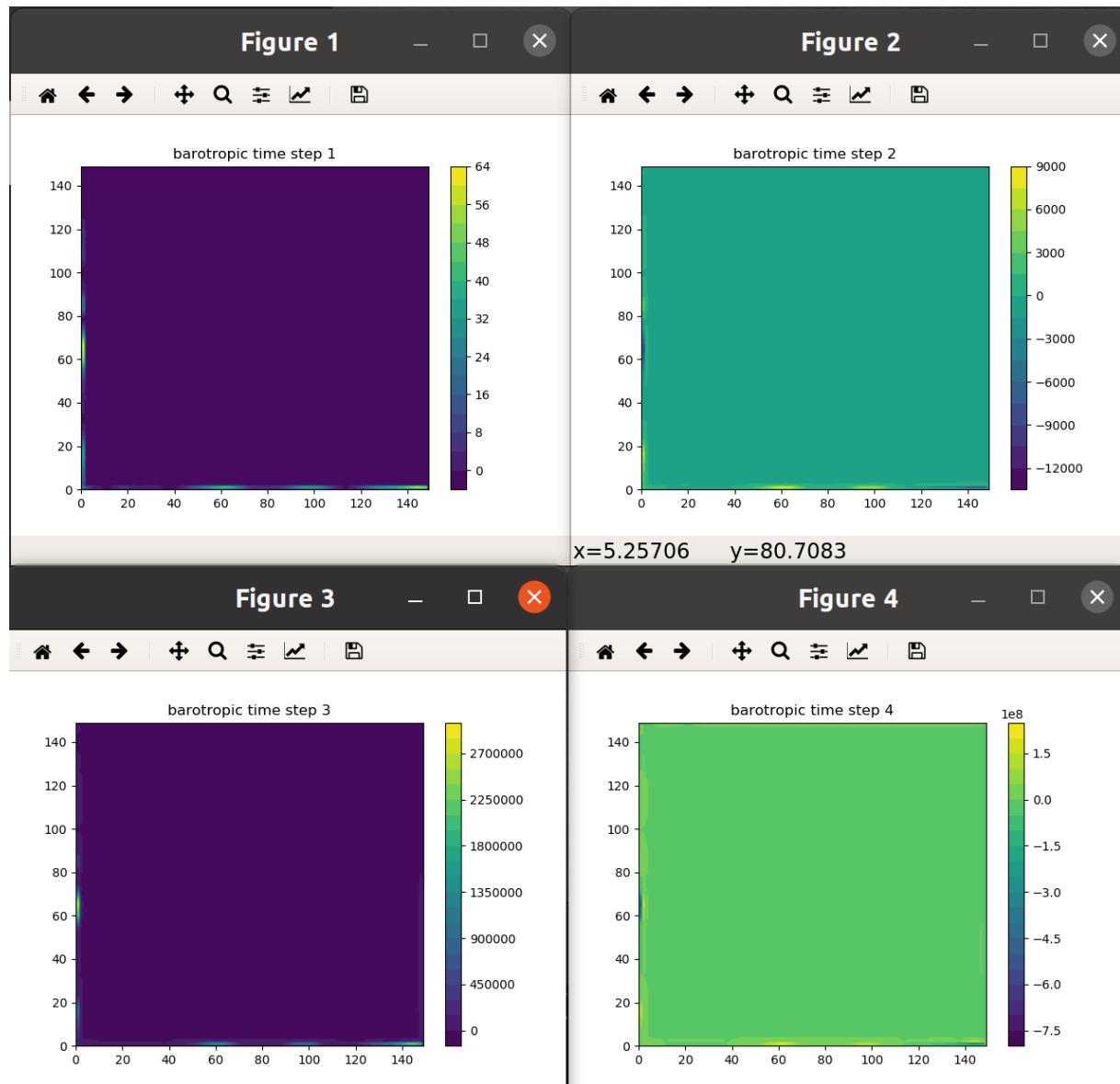
510 FIG. 7. **Time discretization** Errors on the *offline* estimates of advection u-momentum trends computed with
 511 *before* (left) and *now* (center) velocities, and the associated horizontally integrated errors (right).



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



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



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

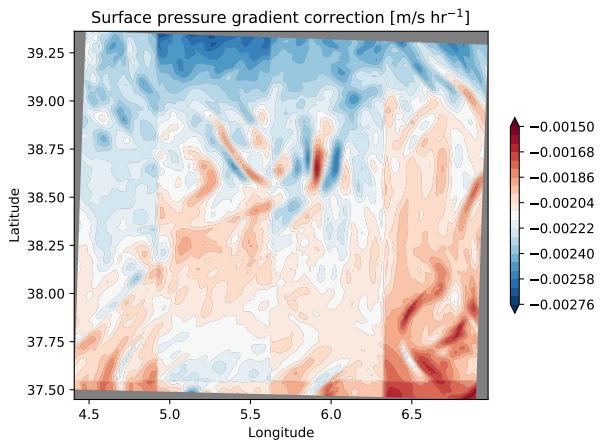
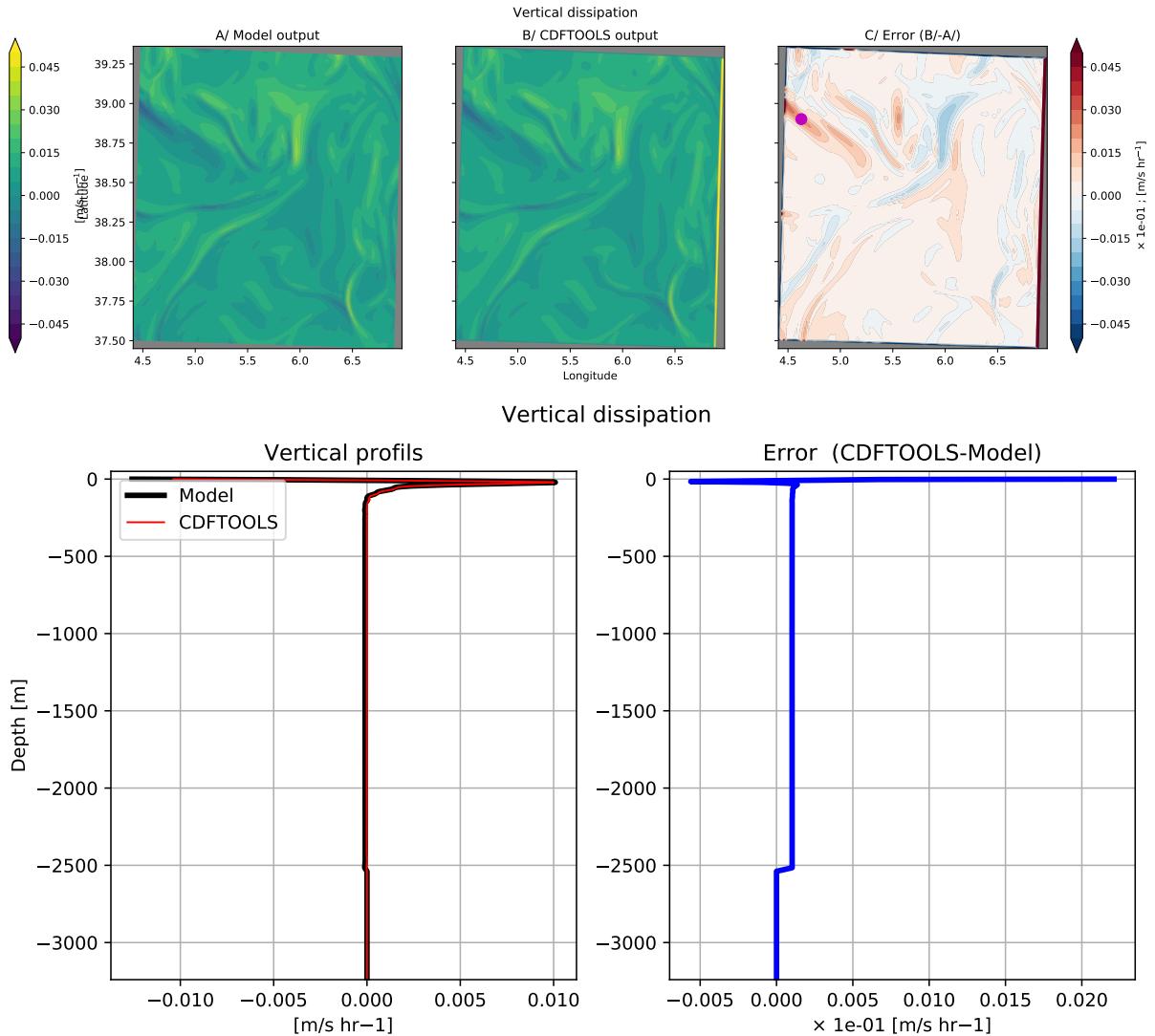
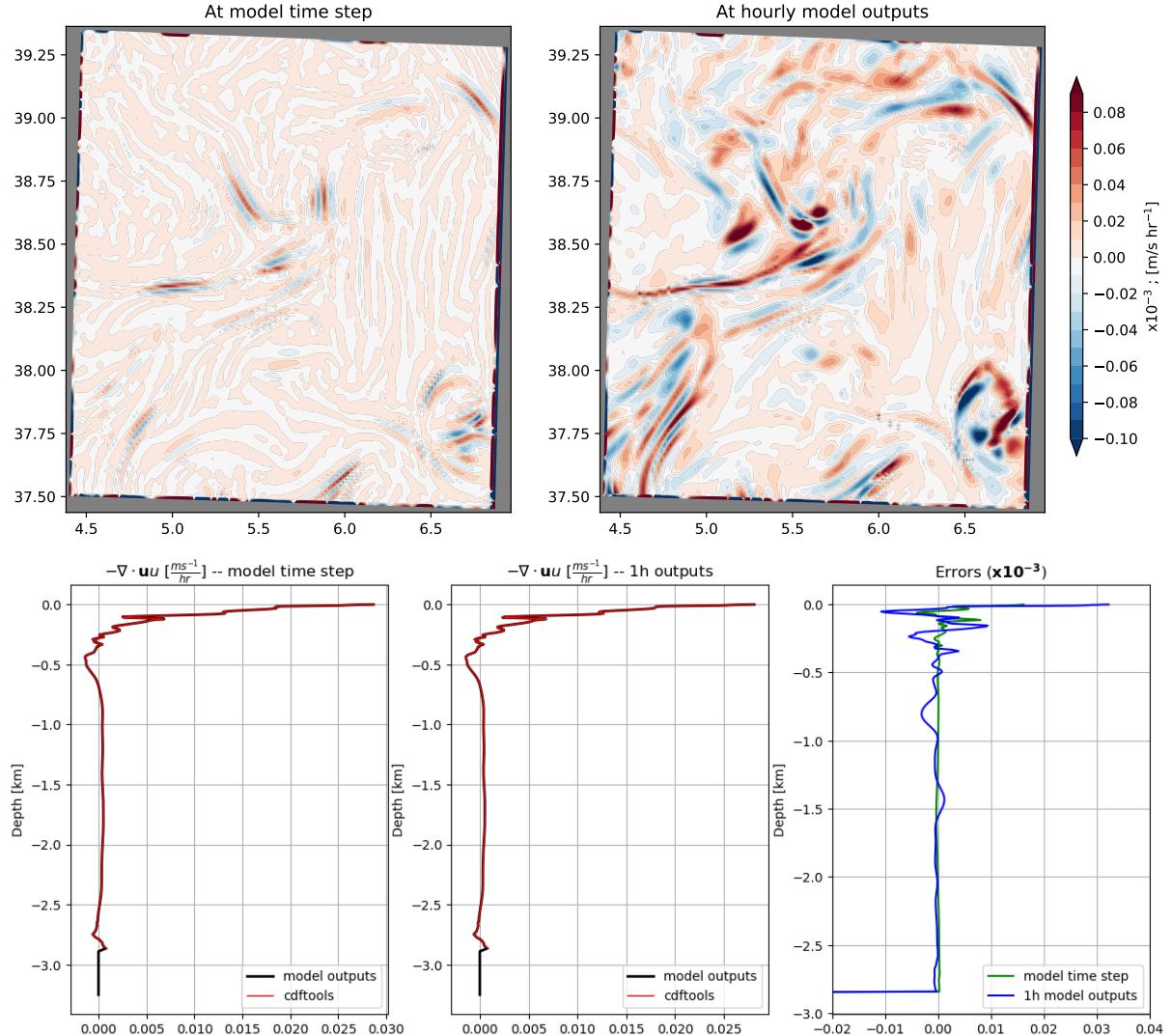


FIG. 11. Surface pressure correction term form the MEDWEST60 model simulation, i.e. the *online* estimates.



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

Advection trend errors



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