Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-9-RC1, 2020

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "Semi-Lagrangian advection in the NEMO ocean model" *by* Christopher Subich et al.

Florian Lemarie (Referee)

florian.lemarie@inria.fr

Received and published: 4 May 2020

This paper describes the implementation of a semi-Lagrangian (SL) advection scheme in the NEMO oceanic model. Sec. 2 discusses the fundamental difference between the Eulerian and SL formulation of advection. It is also shown how a SL advection fits within the existing NEMO framework. Such advection algorithm is made of two specific ingredients: (1) a backward integration to determine the departure points of trajectories (2) an interpolation from the on-grid values to the off-grid departure points found in step (1). The interpolation is described in Sec. 3 while the back trajectory computation is discussed in Sec. 4. Sec. 3 also shows a comparison between the proposed SL algorithm and a standard Eulerian scheme for a simple two-dimensional advection problem with prescribed velocity to check numerically the proper implementation of the

Printer-friendly version

Discussion paper

interpolation step. Finally in Sec. 5, two additional numerical tests are presented. A first semi-idealized test of a flow past an Island is implemented to test the adequacy of the proposed algorithm to find departure points for the trajectories. Then, results from realistic global simulations at $1/4$-degree resolution are shown with a systematic comparison between the SL approach and the Eulerian one. In the conclusion, the authors provide the remaining steps toward a more robust and efficient implementation of their SL algorithm in the NEMO model.

The paper is generally well written and the underlying motivations for the study are clearly stated. The authors try to tackle a very challenging task for numerous reasons: (1) the NEMO code has been fundamentally thought and built to work with Eulerian advection schemes (2) in the oceanic community SL approaches are not very popular because they do not conserve the domain integral of a tracer, they are thought to be very dissipative, and also because there is a general belief that the time-step of oceanic models is mostly constrained by internal gravity waves propagation (3) besides the numerical issues described in the manuscript, the actual code implementation of a SL algorithm in the context of parallel computing is a arduous task to guarantee both an acceptable time-to-solution and scalability. It is always very instructive to explore alternatives to challenge approaches well established for many years. Even if the paper does not conclude to an indisputable success of the proposed method in the context of global realistic applications, **I recommend publication of the manuscript provided the revisions mentioned below. The paper clearly fills a gap in the oceanic modelling community literature and fits perfectly in the GMD NEMO special issue. Moreover, I particularly liked the honesty of the authors because they do not try to oversell their method despite the significant amount of work needed to get it accomplished (the authors deserve the greatest respect for bringing their work to completion). The manuscript follows a scientifically sound reasoning with systematic testing of the different blocks of the proposed method**. I think a bit of work is necessary in the introduction and in Sec. 2 to put the reader in the best possible conditions for the rest of the paper. Moreover, I would expect more discussion about

the performance issues in Sec. 5 and 6 and compelling arguments for not accounting for the Robert-Asselin filter in your analysis.

Please see below my comments and suggestions.

Major remarks and suggestions :

- I initially had a negative feeling about the paper during my first read because I was asking myself several questions (only partially answered in the conclusion) which diverted my attention. For example, the fact that the proposed method is not conservative and is implemented only in the linear free-surface case (i.e. when the vertical grid does not move with time) should be mentioned from the introduction. Another question not tackled in the paper is about the interference between the Robert-Asselin filtering and the SL algorithm for nonlinear simulations. I don't think it is so obvious that the time filtering does not impact the general rationale developed in Sec. 2 (see my comment below)

- **Title**: since the paper does not describe a completely finalized work applicable whatever the numerical options available in NEMO, I would suggest a title like: "*Semi-Lagrangian advection in the NEMO ocean model: implementation and first experiments*". From the conclusions of the present study we can anticipate a subsequent publication with more mature results.

- **Introduction and Sec. 2:** some work is needed to be more pedagogue considering that a typical reader would be either familiar with the SL methods but not with oceanic modeling (e.g. a typical reader from the atmospheric community) or the other way around (i.e. a typical reader from the oceanic community). In

my sense, what makes your study challenging is that you have to deal with a problem under constraints. You already explain the operational constraints but I would also explain more clearly two other levels of constraints to unambiguously justify that you have to deal with problems not already tackled by the atmospheric community

- **Specificity of the oceanic context with consequences on the design of SL schemes:**
  - → *Geometric problems due to the coastlines* ⇒ impact on trajectory calculation
  - → *Stiffness of the problem*. In particular, you take for granted that the time-step of global configurations is constrained by the advective CFL, without supporting reference. From my own experience, it has been difficult to convince ocean modelers that the time-step of global simulations could be mostly limited by advection rather than internal gravity waves (IGW) propagation (I usually do not like to suggest citing my own work but you may use Lemarié et al. (2015) and at a lesser extent Shchepetkin (2015) to further motivate the fact that the time-step is indeed constrained by the advective CFL at $1/4$-degree resolution and higher). After all, your results shown in Sec. 5 and some developments done recently in NEMO (see below) confirm this is indeed the case for ORCA025 configurations.
  - → *Mode-splitting for the treatment of external gravity waves* (it would be the occasion to make clear that the computational grid is considered fixed in your study). There is no such thing as mode-splitting between a 2D barotropic and a 3D baroclinic mode in the atmospheric context. The overwhelming majority of NEMO applications are done with $z^\star$ coordinate (i.e. the grid follows the barotropic motions) and ALE

coordinates will be increasingly used in the future.

- **Specificity of the NEMO context with consequences on the design of SL schemes:**
  - → *Structured ORCA grid* ⇒ small-cell problem. You seem to consider that the small-cell problem by itself justifies that the time-step is constrained by advection. However small cells also penalize the stability limit associated with IGWs.
  - → *Time stepping algorithm* ⇒ Leapfrog (with Robert-Asselin filtering in nonlinear settings)
  - → *Grid-staggering* ⇒ C-grid less natural and efficient than A-grid for SL methods

- You could also make it clear in the introduction that your approach considers a finite-difference interpretation of model variables (in NEMO, tracers are often interpreted in a finite-volume sense).

- In subsection 1.2, you don't give any detail on the existing work regarding the interpolation step. The conservation issue could be mentioned in this subsection and maybe you could motivate why you do not consider conservative SL approaches like the cell-integrated semi-Lagrangian (CISL) schemes or a Lin & Rood type methodology (e.g. Lauritzen, 2007).

- In Sec. 2, the notations could be improved at least to make it more explicit which quantities are on-grid and off-grid and which time-level should be considered. For example in eq. (1), $f^{t_0+\Delta t}(\boldsymbol{x})$ is computed knowing $f^{t_0-\Delta t}(\boldsymbol{x})$ at the same grid point such that $\boldsymbol{x}$ is constant with time and corresponds to the standard NEMO fixed computational grid. I would suggest to give a

specific notation for this fixed reference grid (e.g. $\boldsymbol{x}_{\mathsf{ref}}$). If I understand well, in Eq. (3) $\boldsymbol{x}(t_0+\Delta t)$ exactly corresponds to $\boldsymbol{x}_{\mathsf{ref}}$ and what is noted $\boldsymbol{x}(t)$ (it should probably be $\boldsymbol{x}(t_0)$?) corresponds to the departure point $\boldsymbol{x}_d(t_0)$ with $\boldsymbol{x}_d(t_0) \neq \boldsymbol{x}_{\mathsf{ref}}$. In this case it would make very clear that $\mathsf{RHSL}_f^{t_0}(\boldsymbol{x}_d)$ has to be evaluated off-grid in the SISL approach. Some RHSX terms have a superscript for time and some others don't which complicates the proper understanding. Personally I think that something around the following lines to reformulate equations (1) to (6) would be much easier to understand for a reader:

$$
\begin{aligned}
(1) \Rightarrow \quad f^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) &= f^{t_0-\Delta t}(\mathbf{x}_{\mathsf{ref}}) + 2\Delta t \left\{ \mathsf{ADV}_f^{t_0}(\mathbf{x}_{\mathsf{ref}}) + \mathsf{RHS}^{t_0\pm\Delta t;t_0}(\mathbf{x}_{\mathsf{ref}}) \right\} \\
(3) \Rightarrow \quad f^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) &= f^{t_0}(\mathbf{x}_d^{t_0}) + \frac{\Delta t}{2} \left\{ \mathsf{RHS}^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) + \mathsf{RHS}^{t_0}(\mathbf{x}_d^{t_0}) \right\} \\
(4) \Rightarrow \quad f^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) &= f^{t_0-\Delta t}(\mathbf{x}_d^{t_0-\Delta t}) \\
(5) \Rightarrow \quad f^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) &= f^{t_0-\Delta t}(\mathbf{x}_{\mathsf{ref}}) + 2\Delta t\ \mathsf{ADV}_f^{t_0}(\mathbf{x}_{\mathsf{ref}}) = f^{t_0-\Delta t}(\mathbf{x}_d^{t_0-\Delta t}) \\
(6) \Rightarrow \quad \mathsf{ADV}_f^{t_0}(\mathbf{x}_{\mathsf{ref}}) &= \frac{1}{2\Delta t} \left(f^{t_0-\Delta t}(\mathbf{x}_d^{t_0-\Delta t}) - f^{t_0-\Delta t}(\mathbf{x}_{\mathsf{ref}}) \right)
\end{aligned}
$$

At this point, I reiterate my question regarding the impact of the Robert-Asselin (RA) filtering. For example, looking at equation (A.5) in Shchepetkin and McWilliams (2005) it can be found that your equation (1) with RA filter and $\mathsf{RHS}^{t_0\pm\Delta t;t_0}(\mathbf{x}_{\mathsf{ref}}) = 0$ would be

$$
f^{t_0+\Delta t}(\mathbf{x}_{\mathsf{ref}}) = (1-2\nu) f^{t_0-\Delta t}(\mathbf{x}_{\mathsf{ref}}) + 2\nu f^{t_0}(\mathbf{x}_{\mathsf{ref}}) + 2\Delta t \left\{ \mathsf{ADV}_f^{t_0}(\mathbf{x}_{\mathsf{ref}}) + \nu \mathsf{ADV}_f^{t_0-\Delta t}(\mathbf{x}_{\mathsf{ref}}) \right\}
$$

with ν the RA filter parameter ($\nu = \mathsf{atfp} = 0.1$ in your simulations). Please explain how you can reconcile it with your equation (6) ?

- **Stability constraints for Eulerian schemes:** throughout the paper the exact form of the stability constraint for Eulerian schemes is fuzzy. In Sec. 3.3 (l. 327) the exact constraint for the Leapfrog (LF) with C2 discretized on a C-grid should be:

$$\Delta t \left(\frac{\max(u_{i+1/2,j},0) - \min(u_{i-1/2,j},0)}{\Delta x_{i,j}} + \frac{\max(w_{i,j+1/2},0) - \min(w_{i,j-1/2},0)}{\Delta z_{i,j}} \right) \leq 1$$

  Note that in 1D the LF-C2 scheme is exact for a Courant number equals to 1. For this reason it could be interesting to add a third case with a maximum Courant number around 0.95 to see how the two approaches are robust to changes in time-step value.

  In Sec. 5.1 you consider the QUICKEST scheme which is a coupled space-time approach implemented via a directional splitting[1], meaning that the associated stability constraint is

$$\Delta t \max \left\{ \frac{\max(u_{i+1/2,j},0) - \min(u_{i-1/2,j},0)}{\Delta x_{i,j}}, \frac{\max(w_{i,j+1/2},0) - \min(w_{i,j-1/2},0)}{\Delta z_{i,j}} \right\} \leq 1$$

  which is less restrictive than the one of LF-C2 (in the absence of directional splitting, the stability constraint for QUICKEST in 2D is much more difficult to predict). You should also define how you compute the Courant number in a discrete sense in the SL case on a C-grid. Moreover, is your scheme subject to a Lipschitz criterion ?

- **Courant vs CFL number:** throughout the paper there is a confusion between the Courant number and the CFL number. Please correct it. For example in the

[1] I have been aware of early implementations of the QUICKEST scheme in NEMO without directional splitting. This is just wrong because the resulting multi-dimensional scheme would be unconditionally unstable in the absence of any external source of dissipation. Hopefully you consider a properly implemented version of it.

caption of Fig. 3 you should not talk about CFL number here but about maximum Courant number (the CFL number for LF-C2 is 1 whatever the time-step value). Same thing in Fig. 5 where you should not talk about CFL but maximum Courant number (again the CFL is constant and is equal to 1 for the Control run).

- **Sec. 5.2 and 6:**
  - It is striking to see that the time-step is $\Delta t = 10$ min for your control run with NEMO-CICE whereas with NEMO-LIM3 the standard time-step for ORCA025 is $\Delta t = 20$ min. It seems that there is clearly room for improvement in your ocean/sea-ice coupling.
  - Interestingly, the methodology described in Shchepetkin (2015) to remove the CFL constraint from vertical advection has been recently implemented in NEMO and allowed to further increase the time-step of ORCA025 up to $\Delta t = 30$ min for a marginal increase of the computational cost per time-step. This result is consistent with your remark l. 583-584.
  - The fact that you can increase from $\Delta t = 10$ min to $\Delta t = 15$ min with your SL approach suggests that the advective CFL for ORCA025 with Eulerian advection should lead to $\Delta t = 10$ min (independently from the sea-ice coupling which apparently sets a limit at $\Delta t = 15$ min). However I don't quite understand why your time-step is limited at $\Delta t = 10$ min whereas with NEMO-LIM3 it can be increased to $\Delta t = 20$ min with basically the same advection schemes. You should be able to go at least to $\Delta t = 15$ min for your control run.
  - Given the elapsed time to solution you report in Sec. 6 we can easily find that the computational cost per time-step is 3 times larger with the SL

approach compared to the Eulerian approach (it could be said explicitly in the paper). To be competitive you thus need to be able to take a time-step three times larger. Could you tell what is the contribution of the trajectory computation vs the interpolation step in the computational overhead ? If the overhead is due to the trajectory computation (which has to be done once whatever the number of tracers) it means that your method could gain efficiency in case it is needed to integrate a large number of tracers provided that you can guarantee positivity.

– For the ORCA025 simulations discussed in Sec. 5.2, what is the maximum number of iterations for the trajectory computation (40 ?). Could you comment on the impact of truncated iterations on solution quality and computational efficiency for example if we divide by a factor of 4 the number of iterations per time-step ?

– In the atmospheric context, Ritchie et al. (1995) report only a 20% overhead of their SL version of the ECMWF model compared to the Eulerian version for a time-step multiplied by 5. Do you think it is conceivable in the future to target such a small overhead within the NEMO framework ? The main difference probably comes from the fact that the computational cost of an atmospheric model is largely dominated by physical parameterizations. In NEMO, parameterizations only account for about 10% of the cost while advection in the Eulerian case represents about 12% on average (8% for tracer advection + 4% for momentum advection). I think it makes it much more challenging in the oceanic context to deliver an SL implementation more efficient than Eulerian approaches.

– The parallel implementation of SL methods can be quite cumbersome. Is there a limitation on how far you go to find the departure points, do you

restrict the search for the departure points to the neighbouring MPI subdomains or can you go further ? Loop vectorization is important but with an SL approach you potentially also increase significantly the number of MPI exchanges especially with your iterative algorithm to find departure points ? Please also comment on your expectations in terms of scalability. My intuition is that the smaller the MPI subdomains the larger the computational overhead will be compared to an Eulerian approach.

Minor comments :

- **p. 1, l. 16-17:** What do you mean by "persistence of the initial conditions" ? Shouldn't it be "persistence of the boundary conditions" ?
- **p. 1, l. 18:** "systems" is repeated twice
- **p. 2, l. 49:** "CFL numbers" → "Courant numbers" (the CFL number is virtually infinite for a SL algorithm)
- **p. 4, Eq. (1):** The vector x is not defined
- **p. 5, Eq. (3):** Is it really $x(t)$? Shouldn't it be $x(t_0)$.
- **p. 5, Eq. (4):** f should not be in bold ?
- **p. 13, l. 312:** "CFL numbers" → "maximum Courant numbers"
- **p. 14, l. 327:** "is only stable to a CFL number of" → "is only stable for Courant numbers such that"
- **p. 14, l. 328:** "a CFL number of" → "for a maximum Courant number of"

- **l. 336,340,341,caption of Fig. 3,505,506,515,Fig. 5,552,581:** inconsistent use of CFL where it should refer to the maximum Courant number encountered in the simulation.
- **p. 22, l. 525:** This is a long-standing belief in the NEMO community but there is no such thing as a Total Variation Diminishing scheme in NEMO. The TVD acronym in the so-called TVD scheme in NEMO actually means *Tracer Variance Dissipation* and not *Total Variation Diminishing* which are completely different properties. The tracer advection scheme referred to as TVD scheme in NEMO is an FCT (Flux Corrected Transport) scheme, see (Lévy et al., 2001) for a description in the NEMO framework.
- **p. 22, footnote 9:** in the context of NEMO, the exact reference for this fix and for the consequences of not using it is Ducousso et al. (2017).
- **p. 24, l. 566:** "mangitude" → "magnitude"

Florian Lemarié

References

Ducousso, N., Sommer, J.L., Molines, J.M., Bell, M., 2017. Impact of the "symmetric instability of the computational kind" at mesoscale- and submesoscale-permitting resolutions. Ocean Modell. 120, 18 – 26.

Lauritzen, P.H., 2007. A stability analysis of finite-volume advection schemes permitting long time steps. Mon. Weather Rev. 135(7), 2658–2673.

Lemarié, F., Debreu, L., Madec, G., Demange, J., Molines, J., Honnorat, M., 2015. Stability constraints for oceanic numerical models: implications for the formulation of time and space discretizations. Ocean Modell. 92, 124 – 148.

Lévy, M., Estublier, A., Madec, G., 2001. Choice of an advection scheme for biogeochemical models. Geophys. Res. Lett. 28(19), 3725–3728.

Ritchie, H., Temperton, C., Simmons, A., Hortal, M., Davies, T., Dent, D., Hamrud, M., 1995. Implementation of the semi-lagrangian method in a high-resolution version of the ecmwf forecast model. Mon. Weather Rev. 123(2), 489–514.

Shchepetkin, A.F., 2015. An adaptive, courant-number-dependent implicit scheme for vertical advection in oceanic modeling. Ocean Modell. 91, 38 – 69.

Shchepetkin, A.F., McWilliams, J.C., 2005. The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model. Ocean Modell. 9(4), 347 – 404.