Geosci. Model Dev. Discuss., 7, C3467–C3468, 2015
www.geosci-model-dev-discuss.net/7/C3467/2015/

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on “NEMOTAM: tangent and adjoint models for the ocean modelling platform NEMO” *by* A. Vidard et al.

A. Vidard et al.

vidard@imag.fr

Received and published: 3 March 2015

Dear Jemma Shipton,

Thank you for your valuable comments, we tried to address them all in the revised version of the paper. Some specific answers are listed below:

- *“In particular, the readability of the manuscript would be improved by checking for language errors, some of which I have indicated below, and by using fewer acronyms.”:* Agreed, I tried to limit the number of acronyms to the strict minimum and corrected a significant number of language errors.

- “However, more information on how to obtain and run the code would be valuable.”: we added a section about code availability at the end of the paper.

- *“"Automatic tools are now mature enough [. . .]" I thought this was a slightly strange way to start the discussion of the relative merits of automatic and hand coded TAM given that you go on to say that automatic tools not yet good enough.”:* well, actually autodiff tools ARE mature enough, they are just not adapted to the particular needs that originally motivated the nemotam development (multi-resolution incremental 4D-Var), I tried to rephrase this paragraph for a clearer message.

- *“You give an example of forward and backward singular vectors in figure 3. I understand they’re an illustration of the capability of the TAM code rather than to answer any particular question about the flow or the model, but could you give a brief comment on what this example shows, in addition to your general comments on the usefulness of such vectors?”* Indeed, this section was rather insubstantial as also pointed out by reviewer 2. Without going too much into detail, we added some comments about what the figure shows and another figure with the amplification factors related to resolution and time window length.

The other comments, although relevant do not require a detailed answer. We accounted for all of them, apart “Pg 6715, line 7: is this really a strict equals sign?” since there is no equation in line 7. Anyway, equations in this part of the paper were not quite right (unfortunate cut and paste, I guess), so maybe this question is no longer valid (maybe it still is).

Interactive comment on Geosci. Model Dev. Discuss., 7, 6705, 2014.

Geosci. Model Dev. Discuss., 7, C3469–C3472, 2015
www.geosci-model-dev-discuss.net/7/C3469/2015/

Interactive comment on "NEMOTAM: tangent and adjoint models for the ocean modelling platform NEMO" *by* A. Vidard et al.

A. Vidard et al.

vidard@imag.fr

Received and published: 3 March 2015

Dear Reviewer,

Thank you for your numerous valuable comments, we tried to answer properly to all of them. Some specific answers are listed below.

- *"The development of an adjoint for a complex and nonlinear model, such as NEMO, is notoriously difficult. Algorithmic differentiation (AD) tools exist that can greatly facilitate the derivation. However, since NEMO has not been developed in compatibility with an AD tool":* There is a misunderstanding in the reason why we did not go through the automatic way. This is not due to the way NEMO is coded, we actually did a first attempt using tapenade and obtained a satisfactory

version of the tangent and adjoint models in a reasonable amount of time. The main reason is due to the original motivation of building a multi-resolution incremental 4D-Var, as mentioned in our answer to the first reviewer. We modified the relevant paragraph to make this message clearer.

- *"Furthermore, I feel that much of the mathematical description is - although relevant - well-known and could be replaced with suitable references."* We kind of disagree, the paper is short enough that we can leave this description in place, it makes understanding easier for readers not familiar with the topic without disturbing readability

- *"Section 4 would be greatly improved by placing emphasis in each subsection on computational aspects specific to NEMOTAM":* Good point, we added some comment about computational aspect in each subsection.

- *"Although detailed discussion of the dynamics implied by the diverse applications is not required here, confirmation that NEMOTAM output is sensible is certainly suitable, but is missing from section 4.2.":* indeed, this paragraph was lacking details, we added some more comments about what the figure shows and an additional figure about the evolution of the amplifying factors respect to model resolution and time window length.

- *"Since NEMOTAM is hand-coded it would be helpful if the authors would offer some concluding remark on the flexibility of the current release. Although it is hinted that "more flexibility" would be "very beneficial" in the conclusions section, I feel it is appropriate to give a more explicit indication of the potential for NEMOTAM to be applied (to sensitivity, assimilation, stability investigations etc.) under different experimental configurations. I feel that confirming some degree of existing flexibility is important in ensuring that the substantial effort invested in NEMOTAM translates to a useful scientific contribution.":* indeed, this is a drawback of hand-coded TAM, but it offers some flexibility as it is, we added comment on that in the conclusion.

- *"Is the oscillatory advection term retained in the full nonlinear model but neglected by the gradient computations (i.e in both the TLM and the adjoint)? I am confused by the justification of the choice here, since the exactness of the adjoint with respect to the full nonlinear model is relevant":* This is actually a tricky bit. There is no "right way" to differentiate a non-oscillatory scheme since it is highly non differentiable. Doing it as an algorithmic differentiation tool would (retaining the non linear branching) is clearly not the right way to do as the result proves highly unstable. Both other approaches (ignoring the limiter or adding it in the tam) induce some errors and it is not clear if one is always better than the other. However the latter make the adjoint and tangent tests to fail since it induces some non linearities. My wild guess is that for shorter period of time the former is better and for longer you'd better add the limiter. But so far, my experience is that, even with few month of seabass1/12° configuration integration the absence of the limiter in TAM is not causing trouble. During the revision of this paper we added the limiter (not available in the distributed version) and no significant impact were seen on the computed sensitivities.

- *" Please elaborate a little on what is meant by "well-chosen" (i.e choice is determined by resources? nonlinearities in the code? etc.). "* see below.

- *"I may have misunderstood this part of the discussion, but linearly interpolating between checkpoints to relieve storage demands seems drastic. Please included additional citations if this approach is taken in other model frameworks."* It is indeed a bit drastic, even though it does not seem to a have had a significant impact so far in our applications. My guess is that it is a common practice to all TAM codes that are used with incremental 4D-Var (ROMS, IFS, ARPEGE, . . .).

  ' Although this simplification does not appear to significantly impact the TAM for

  the test cases referenced in table 2, it will surely become important in other experimental configurations; for example where checkpoints are spaced further apart during longer integrations. " Maybe our use of the word "checkpoint' is a bit misleading, even if we store at each checkpoint, there will be some recomputation involved. Only a limited subsample of direct variables are stored (the pronostic ones and one or two diagnostic that are expensive to compute). For a global configuration that amounts to about 12 3D variables. They are stored on disk, so this is indeed a limitation, depending on the disk space available, but it is not that problematic. The real drawback is the IO overhead implied and it can be significant. However this will be alleviated by the presence of a IO server available now in NEMO (but not yet used for NEMOTAM trajectory).

  "Could the authors not employ a higher level checkpointing scheme here?" No, it would require to be able to rerun the full non linear model, which is what we wanted to avoid and that prevented us to use algorithmic differentiation in the first place. We modified this paragraph

- *"I found section 3.3 confusing. I think the aim is to define an error measure for approximating the full model physics in the generation of the TLM but must admit that I can't see how this is provided by E in Eq 9. Are the I components of " in Eq 7 related to different approximations made to the nonlinear model M ? And then is it necessary to assume linearity to obtain Eq 8?"* Sorry, the equations here were not quite right, I hope this is clearer now.

The other comments, although relevant do not require a detailed answer. We accounted for all of them.

Interactive comment on Geosci. Model Dev. Discuss., 7, 6705, 2014.

Manuscript prepared for Geosci. Model Dev.
with version 2014/07/29 7.12 Copernicus papers of the LaTeX class copernicus.cls.
Date: 6 March 2015

NEMOTAM: tangent and adjoint models for the ocean modelling platform NEMO

Arthur Vidard[1], **Pierre-Antoine Bouttier**[2], **and Franck Vigilant**[1]

[1]Inria, Univ. Grenoble Alpes, CNRS, LJK, F-38000 Grenoble, France
[2]Univ. Grenoble Alpes, CNRS, LGGE, F-38000 Grenoble, France

Correspondence to: A. Vidard
(vidard@imag.fr)

Abstract. The tangent linear and adjoint model (TAM) are efficient tools to analyse and to control dynamical systems such as NEMO. They can be involved in a large range of applications such as sensivity analysis, parameter estimation or the computation of characteristics vectors. TAM is also required by the 4D-VAR algorithm which is one of the major ~~method~~ methods in Data Assimilation. This paper describes the development and the validation of the Tangent linear and Adjoint Model for the NEMO ocean modelling platform (NEMOTAM). The diagnostic tools that are available alongside NEMOTAM are detailed and discussed and several applications are also presented.

1 Introduction and history

Tangent linear and adjoint models (TAM in the following) are powerful modelling ~~tool~~tools. The tangent linear model (TLM) provides the directional derivatives with respect to a trajectory of the corresponding ~~non linear~~ non-linear system. The adjoint of the TLM gives information about the response of the system to variations of its input. TAM are therefore widely used for variational assimilation applications, but also for the analysis of physical processes, since they can be used for sensitivity analysis, parameter identification and for the computation of characteristic vectors (singular vectors, Liapunov vectors, etc. see Moore et al. (2004) for an extended review).

This is particularly true for geophysical applications such as meteorology and oceanography where many data assimilation systems rely on the availability of such models. However, only few ocean general circulation models are routinely provided with their TAM. Among them, one can cite MITgcm (MIT general circulation model) and ROMS (Regional Ocean Modelling System). They are quite symbolic of the two possible routes for deriving tangent and adjoint model, either using automatic differentiation ~~tool~~ tools (MITgcm uses automatic differentiation (Marotzke et al., 1999)) or hand-coded (ROMS' TAM was developed by a pool of researchers (Moore et al., 2004)).

Automatic ~~tools are nowmature enough~~differentiation tools have been available for some time now, and they show some significant advantages over the hand-coded route. Differentiation of a numerical code is a tedious and error-prone task, the use of an automatic tool could alleviate this difficulty. Moreover when an updated version of the ~~non linear~~ non-linear model is provided the corresponding TAM can in general be obtained effortlessly. Additionally automatic tools offer an important flexibility when one wants to change the variable to differentiate around. Indeed, typically, for data assimilation purpose, the TAM is differentiated around the initial condition, while for ~~parameters~~ parameter estimation it is differentiated around the sought parameter set. Most of the time the obtained TAM are very similar, but in some cases (e.g. for grid related parameter) going from one to another can lead to a significant amount of code changes.

On the other hand, automatic differentiation ~~suffer~~ suffers from some limitations compared to hand-coding. First some newer (or archaic) language features may not be supported at first by the automatic tools, second the numerical performance of ~~automatic~~ automatically derived TAM is still relatively poor compared to that of the hand-coded one, in particular, the handling of the parallelization is still an open issue. ~~Third non differentiable part~~ Moreover that are not fully automatic since non-differentiable parts of the original code ~~still require~~ are still required to be dealt with by specialists in the field. All these issues can however be overcame by mixing automatic and manual coding.

The NEMO ocean engine (Madec, 2008) was previously known as the OPA model (Madec et al., 1998). It used to have a TAM (called OPATAM), fully hand-coded. OPATAM was initially developed for a Pacific ocean configuration, and targeted at variational data assimilation applications in the framework of OPAVAR (Weaver et al., 2003, 2005). OPATAM/OPAVAR were extended to other regional basins (Mediterranean sea (Rémy, 1999), North Atlantic 1/3° (Forget et al., 2008), South Atlantic 1°), to the global ocean (ORCA 2° (Daget et al., 2009)), and were used for methodological studies such as control of the 3D model error (Vidard, 2001), control of the surface forcing and open boundary conditions (Deltel, 2002; Vossepoel et al., 2003). OPATAM was also used for sensitivity studies (Sévellec et al., 2008) ~~,~~ and singular vectors (Moore et al., 2003; Sévellec et al., 2009)~~, etc~~.

For several reasons, mainly because of lack of workforce, OPATAM, OPAVAR and related developments were not included in the standard release of OPA. As a consequence, synchronisation of OPATAM with OPA's releases could not be achieved on a regular basis, and all developments were on individual branches, without feedback to the OPATAM/OPAVAR system. The pool of potential users was therefore reduced significantly. The complete rewriting of the model during the transition to NEMO rendered OPATAM and OPAVAR obsolete. As part of the NEMOVAR initiative (a variational data assimilation with NEMO, Mogensen et al. (2009)) a first prototype of NEMOTAM was obtained by Tber et al. (2007) using the TAPENADE automatic differentiation tool (Hascoët and Pascual, 2004) for a fixed and somewhat simplified configuration (ORCA 2° with all non-differentiable options switched off (Tber et al., 2007)). ~~Even~~ This initiative was successful in a reasonable amount of time, however, even for this simplified configuration ~~, however,~~ substantial human intervention and additional work was required to obtain ~~a useable~~ an efficient product from the raw generated code. Three main drawbacks were identified for this application. First, the memory management and CPU performance of the raw code were rather poor. Second, the version of TAPENADE at that time generated single-processor code only and could not handle directives from the C-PreProcessor (CPP keys), which are widespread in NEMO. ~~Third, the technique of binomial checkpointing that is used to handle nonlinearities (see) is not compatible, at least in its present implementation, with the incremental algorithm of NEMOVAR, which employs separate executables and (possibly) different resolutions for the outer and inner loops.~~ Improved memory management and extensions to support MPP and CPP keys are planned in future versions of TAPENADE ~~so the~~ and even as it is now, it can be overcome manually with a reasonable additional effort, so these first two deficiencies are not fundamental. ~~The third deficiency, however, is more problematic and the trajectory management for nonlinearities in NEMOTAM is done differently from TAPENADE, along the lines of the simpler strategy implemented in OPAVAR.~~ There is a third problem that is an incompatibility between the way automatic differentiation tools handle non linearities and the so-called incremental 4D-Var assimilation algorithm used in NEMOVAR that was the original motivation for developing NEMOTAM. Indeed when a non linearity occurs in the direct model, the value of the corresponding variable needs to be made available to the tangent and adjoint model to differentiate around. To that purpose, on the one hand automatic differentiation tools run the direct model alongside the tangent and store one way or another the relevant value for the backward integration of the adjoint (*e.g.* binomial checkpointing as in Tber et al. (2007)). On the other hand, incremental 4D-Var would perform a minimisation using only tangent and adjoint integrations, meaning it would run several instances of the tangent model for one instance of the direct (and possibly at a different resolution), so direct and tangent models cannot be run alongside.

The modifications required to make any ~~AD tool ,~~ automatic differentiation tool compatible with the multi-incremental approach are really substantial and cannot be done in a short or medium term. Moreover the numerical performances of the automatically generated TAM do not ~~allow yet their~~ yet allow its use for 'big' configurations and for operational applications. The writing of an adjoint code is not a simple technical task, this explains why ~~Autodiff~~ automatic differentiation tools can seldom be used as a black-box. Numerous potential problems can arise and in general they should be dealt with on a case by case basis by someone who masters both the model and differentiation techniques.

From that experience it has been decided to go toward the hand-coding approach, the use of ~~AD~~ automatic differentiation tool being left aside for the time being. We may reconsider this in a medium or long term though. An optimal mix of both approaches is likely to be the preferred choice ~~in the short term future~~ and lead to a semi-automatic way of generating NEMOTAM.

This paper will first discuss the methodology used for the NEMOTAM development and explain some of the particular choices that have been made. Then the validation tools that are available alongside NEMOTAM will be detailed. Finally some application examples will be presented.

2 Methodology and choices for NEMOTAM

The approach used in developing NEMOTAM is based on the well known differentiation rules as described in Giering and Kaminski (1998). It relies in particular on the definition of active and passive variables. The former depends on the control variables (variables one differentiates around) variations, while the latter are independent of the control (typically model or grid parameters). Active variables have tangent and adjoint ~~counterpart~~ counterparts while passive variables do not.

In the current version (3.4.1), only the general circulation component of NEMO is supported by NEMOTAM and a few key components are still missing, namely the variable volume~~(VVL)~~, open boundary conditions ~~(OBC) and the nesting capabilities (AGRIF~~and the grid nesting capabilities). There is no fundamental reason for ~~non~~ not supporting these options (support for ~~OBC is actually on its way~~open boundary conditions is actually in progress), but it was not flagged as priority. In particular ~~VVL~~ "variable volume" option would require a tremendous amount of coding since it makes the surface grid cells size depending on the flow (~~they become then~~ thus becoming active variables). This is a typical case where a semi-automatic route would be beneficial.

When coding tangent and adjoint models, either automatically or manually, one has to face several difficulties. The main ones are listed in the ~~remaining~~ remainder of this section and discussed in the particular case of NEMOTAM.

2.1 Non-differentiabilities: Problem and solutions

In realistic models like those included in NEMO, non-differentiable physics are quite frequent. They can be non-differentiable by essence (*e.g.*, step processes), due to the way they have been programmed or the chosen numerical methods (*e.g.* non-oscillatory schemes). Coding-wise they can be represented by an IF statement with a condition on an active variable (or a max, abs, *etc.* statement but it is equivalent). In order to manage these non-differentiable parts several options are available.

- Regularisation: either mathematical regularisation (thanks to the introduction of a differentiable connection), or a physical regularisation whenever it is possible (by rewriting the processes in a differentiable manner).
- Approximation: The direct model physics is approximated in TAM in order to transform the non-differentiable part. One should obviously be careful that the approximation is not too strong.
- ~~Non linear~~ Non-linear branching: the non-differentiable part is kept in TAM but the same branching (same side of the IF) ~~than~~ as in the direct model is used (*i.e.* one left- and right-differentiates separately). This is in general the preferred choice, but in some cases it can lead to pathological behaviour and should be done very carefully.

The choice of treatment heavily relies on the type of the non-differentiability and it is more a matter of an educated choice. Indeed it requires an important knowledge of both the direct model and differentiation techniques. A typical example of this in NEMO is the vertical mixing schemes included in OPA~~(TKE in particular, but it is true for the other options as well)~~. The current version is strongly non-differentiable. In NEMOTAM ~~this has been sorted out by simplification~~ differentiation is achieve by first simplifying of the physics, *i.e.* some active variables are computed by the direct model and treated as passive variable in the TAM. Adopting this strategy has the advantage ~~to keep~~ of keeping the full physics in the ~~non linear~~ non-linear model, but is at the expense of an approximation in the TAM. However, most of the physical process it models being quite regular, it may be possible to rewrite the direct version to make it differentiable[1].

Another important source of non differentiability is the non-oscillatory part of one of the most popular tracer ~~’s advection term in NEMO(TVD). A non linear branching of such scheme~~ advection schemes in NEMO. This kind of scheme is highly non differentiable and a classical non-linear branching (as would do an automatic differentiation tool) can lead to an very unstable TAM. Following Thuburn (2001) two viable approximations are at hand: either one ~~differentiate~~ differentiates at continuous level and then ~~apply~~ applies the same non oscillatory discretisation scheme as for the direct model or the non oscillatory correcting term is removed altogether in the tangent and adjoint schemes. ~~The~~ Both ways present some approximations, the former is generally a better approximation for the tangent model in the long run, but it introduces some non linearities and degrades the exactness of the adjoint model with respect to the tangent model~~(see next section)~~. For this reason, in NEMOTAM the second solution has been adopted.

2.2 Checkpointing

One of the issue one has to tackle when dealing with TAM is the storage and/or re-computation of the non-linear trajectory that is required for the differentiation of the non-linear terms. This is particularly important for the adjoint model which ~~need~~ needs the trajectory in reverse order as it is produced by the ~~non linear~~ non-linear model. A common practice is the so-called checkpointing where ~~well-chosen~~ snapshots of the trajectory (the checkpoints) are stored and intermediate variables are re-computed between checkpoints. ~~This is the strategy~~ A similar strategy is used in NEMOTAM, a subset of direct variables are stored on disk at the end of every time step and only required intermediate variables are recomputed. The number of non-linearities in NEMO being relatively small, the need for storage and re-computation is limited. However, for long ~~period of time~~ integrations the amount of required storage may become too ~~important~~severe; in that case a possible approximation is to subsample the output of the trajectory (say one per day) and NEMOTAM will interpolate linearly between checkpoints, additionally it can be stored in single precision. The validity of these approximations is discussed in section 3. Automatic differentiation tools generally allow for the use of more efficient multi-level

[1] This was the strategy adopted in ROMS for the ~~KPP~~ vertical mixing scheme

checkpointing, but it is not possible here due to the fact that NEMO and NEMOTAM are not run together.

2.3 Numerical issues

The numerical characteristics of the direct model are not always conserved in the adjoint model. In particular the most common difficulty is related to the convergence speed of an iterative algorithm that may be different for the direct and its tangent/adjoint counterparts.

In that case, a specific solution should be provided, it can go as far as replacing the problematic scheme by a more TAM-friendly one. Once again, in OPATAM this kind of problem arose, and one had to replace the conjugate gradient solver used for the computation of the surface pressure gradient (while self-adjoint, in theory) by a red-black SOR solver. However, this scheme being on the verge of depreciation in NEMO, NEMOTAM will soon adopt the time-splitting-based direct solver.

Stability is not the only numerical issue: the TAM are, by essence, more expensive numerically than the direct model because of the additional computations they require. Moreover, and it is especially true for the adjoint model, direct code optimisations may not be so optimal for the TAM. Therefore it is important to specifically optimise the TAM code, and this can only be done through a careful performance analysis. As of today, such an optimisation may not be achieved automatically. The tendency toward application at high and very high resolution make the computing cost aspect a crucial issue. NEMOTAM adopted the same domain decomposition strategy as for NEMO and is therefore fully parallel, and some specific targeted ~~optimisation~~ optimisations have been performed. Additionally some irrelevant (as far as TAM are concerned) on-line diagnostics have been removed. Thanks to all these ~~effort~~ efforts the additional computing burden has been contained to around twice the non-linear integration cost (more or less, depending on the application).

3 Validation

Due to the numerous caveats mentioned in section 2 validation of TAM is a crucial aspect of the model development. Fortunately there are several numerical tests that can be performed to ensure this validity, moreover these tests can be applied both to the whole model and to individual modules separately.

3.1 Numerical validation

In NEMOTAM, modules include both the tangent and adjoint parts, as well as the *validation interface*, for numerical verification of the TAM source codes. The *adjoint test* would always be present, while the *tangent test* could be optional and reserved ~~to~~ for specific and problematic routines.

In the following $\mathcal{M}(\mathbf{x})$ stands for the full non-linear NEMO model with initial state vector $\mathbf{x}$, $\mathbf{L}(\mathbf{x}) \equiv (\partial\mathcal{M}/\partial\mathbf{x})|_{\mathbf{x}=\mathbf{x}}$ its tangent linear model (possibly simplified), and $\mathbf{L}^*(\mathbf{x})$ the adjoint of $\mathbf{L}(\mathbf{x})$.

3.1.1 Adjoint validation

The adjoint part is actually relatively easy to check, indeed, by definition of the adjoint one gets:

$$(\mathbf{L}(\mathbf{x})\delta\mathbf{x}, \delta\mathbf{y}) = \langle \delta\mathbf{x}, \mathbf{L}^*(\mathbf{x})\delta\mathbf{y} \rangle \tag{1}$$

where $\langle .,. \rangle$ and $(.,.)$ denote the appropriate dot product. Equation 1 being exact, the relative error between the two computed scalar products must be close to zero barring rounding errors. In NEMOTAM, the actual test performed is

$$(\mathbf{L}(\mathbf{x})\delta\mathbf{x})^*\mathbf{W}\delta\mathbf{y} = \delta\mathbf{x}^*\mathbf{L}^*(\mathbf{x})\mathbf{W}\delta\mathbf{y} \tag{2}$$

where $\delta\mathbf{x}$ is a random vector, $\mathbf{W}$ is a diagonal matrix of scale factors and $\delta\mathbf{y} = \mathbf{L}(\mathbf{x})\delta\mathbf{x}$. This test only ~~ensure~~ ensures that the adjoint is indeed the adjoint of the tangent linear, which in turn has to be validated.

3.1.2 Tangent validation

Validation of tangent modules are more tricky since there is generally no affordable exact tests available. A classical method of testing a numerical tangent linear model $\mathbf{L}$ is to compare the evolution of a perturbation by $\mathbf{L}$ with the difference of two evolutions, with and without the perturbation, by the full ~~nonlinear~~ non-linear model $\mathcal{M}$.
Considering a fixed small perturbation vector ~~$\gamma\delta\mathbf{x}$, where~~ $\delta\mathbf{x}_0$, and γ ~~is~~ a scale parameter, the Taylor expansion of $\mathcal{M}$ reads:

$$\mathcal{M}(\mathbf{x}_0 + \gamma\,\delta\mathbf{x}_0, t) = \mathcal{M}(\mathbf{x}_0, t) + \gamma\,\mathbf{L}(\mathbf{x}, t)\,\delta\mathbf{x}_0 + O(\gamma^2) \tag{3}$$

If ~~$\mathcal{N}(\mathbf{x}_0, \gamma\delta\mathbf{x}_0, t_0, t)$ denotes the non linear~~ $\mathcal{N}(\gamma\delta\mathbf{x}_0, t)$ denotes the non-linear evolution of a perturbation~~and we use the simplified notation $\mathcal{N}(\gamma\delta\mathbf{x}_0, t)$~~,

$$\mathcal{N}(\gamma\delta\mathbf{x}_0, t) = \mathcal{M}(\mathbf{x}_0 + \gamma\delta\mathbf{x}_0, t) - \mathcal{M}(\mathbf{x}_0, t) \tag{4}$$

~~The linearization~~

The linearisation error $\mathcal{E}(\gamma\delta\mathbf{x}_0, t)$ is defined by:

$$\mathcal{E}(\gamma\delta\mathbf{x}_0, t) \quad = \quad \mathcal{N}(\gamma\delta\mathbf{x}_0, t) - \gamma\mathbf{L}(\mathbf{x}, t)\delta\mathbf{x}_0 \tag{5}$$

From (3), $\mathcal{E}(\gamma\delta\mathbf{x}_0, t)$ behaves ~~in~~ like $O(\gamma^2)$.

~~And the~~ The first order accuracy index ϵ_γ is given by:

$$\epsilon_\gamma = \frac{\|\mathcal{N}(\gamma\delta\mathbf{x}_0, t)\|}{\|\mathbf{L}(\mathbf{x}, t)\gamma\delta\mathbf{x}_0\|} \tag{6}$$

ϵ_γ tends to ~~one when~~ 1 as γ tends to ~~zero~~0, validates $\mathbf{L}(\mathbf{x}, t)$, moreover when γ is small enough, $\mathcal{N}$ enters a linear regime and ϵ_γ converges toward 1 with a rate γ.

Table 11 shows an example of such tests on a single routine. It illustrates nicely the expected behaviour of ϵ_γ which gains one digit in precision when γ is divided by ten. This ~~diagnostics~~ diagnostic gives information of both first order (tends to 1) and second order (at rate γ).

[Table 1 about here.]

3.1.3 Estimation of the approximations error

As mentioned above, when differentiating realistic models, approximations have to be ~~done~~made. To estimate the effect of these approximations on the numerical tangent linear model $\mathbf{L}$, one must first ~~estimates~~ estimate the truncated part ~~, $O((\gamma\delta\mathbf{x})^2)$,~~ of the Taylor expansion ~~(see equation~~ of equation (3). In order to do this, following Lawless et al. (2003), one can write the Taylor expansion of ~~$\mathcal{E}(\gamma\delta\mathbf{x},t)$ whose each individual component~~ $\mathcal{E}(\delta\mathbf{x},t)$ whose individual components l ~~follows~~ follow:

$$\mathcal{E}_l(\gamma\delta\mathbf{x}_0,t) = \frac{\gamma^2}{2}\frac{1}{2}\partial^2\mathcal{M}_l\delta\mathbf{x}_0^2 + \frac{\gamma^3}{6}\frac{1}{6}\partial^3\mathcal{M}_l\delta\mathbf{x}_0^3 + O(\gamma^4)\dots \tag{7}$$

On ~~another~~ the other hand, from two ~~nonlinear~~ non-linear perturbations:

$$\begin{aligned}\mathcal{N}(\delta\mathbf{x}_0,t) &= \mathcal{M}(\mathbf{x}_0+\delta\mathbf{x}_0,t)+O((\delta-\mathcal{M}(\mathbf{x}_0)^2,t)\\ \mathcal{N}(\gamma\delta\mathbf{x}_0,t) &= \mathcal{M}(\mathbf{x}_0+\gamma\delta\mathbf{x}_0,t)+O((\gamma\delta-\mathcal{M}(\mathbf{x}_0)^2,t)\end{aligned} \tag{8}$$

one can compute

$$\mathrm{E}(\gamma\delta\mathbf{x}_0,t) = \frac{\mathcal{N}(\gamma\delta\mathbf{x}_0,t)-\gamma\mathcal{N}(\delta\mathbf{x}_0,t)}{\gamma^2-\gamma} \tag{9}$$

whose Taylor expansion reads (for each individual component l)

$$\mathrm{E}_l(\gamma\delta\mathbf{x}_0,t) = \frac{1}{2}\partial^2\mathcal{M}_l\delta\mathbf{x}_0^2 + \frac{1+\gamma}{6}\partial^3\mathcal{M}_l\delta\mathbf{x}_0^3 + O(\gamma^4) \tag{10}$$

For small values of γ and $\delta\mathbf{x}_0$, one can compare E and $\mathcal{E}$. That way one builds-up an estimator of the numerical tangent linear model error:

$$\hat{\mathcal{E}} = 100\left(1-\frac{\|\mathrm{E}\|}{\|\mathcal{E}\|}\right) \tag{11}$$

Moreover, in NEMO, the vast majority of the ~~nonlinearities~~ non-linearities are quadratic ones, meaning the third order and above derivatives vanish from the Taylor expansion and one gets $\mathrm{E} = \mathcal{E}$.

This diagnostic is very valuable when comparing different simplifications made to the tangent linear model. Table 12 shows $\hat{\mathcal{E}}$ for two configurations of NEMO: an academic test case that is fully differentiable (SEABASS, see appendix A) and 2°-resolution global realistic configuration (ORCA2). This allows us to measure the effect of approximations mentioned in section 2.2 and 2.1

[Table 2 about here.]

4 Application Examples

As stated by Errico (1997): "the principal application of adjoint models is sensitivity analysis, and all its other applications may be considered as derived from it". Performing Sensitivity analysis means evaluating how variations on the input of the system will affect the output. This can be of use for understanding the behaviour of the system (sensitivity analysis, propagation of ~~incertitude~~uncertainty), for optimising it (through data assimilation for instance) and for performing stability analysis. This section presents an example of these three ~~kind~~ kinds of applications.

4.1 Sensitivity analysis and Data assimilation

In the variational context, sensitivity analysis is the computation of the gradient of so-called response (or cost or objective or criterium) functions with respect to given control vectors. In other words, given the model's state $\mathbf{x} = (x_1,...,x_n)^T \in \mathcal{X} \subset \mathbb{R}^n$ and a set of control parameters $\alpha = (\alpha_1,...,\alpha_p)^T \in \mathcal{P} \subset \mathbb{R}^p$ one is interested in computing gradients with respect to α of a given response function:

$$\begin{aligned} J: \quad & \mathcal{P} \to \mathbb{R} \\ & \alpha \mapsto J(\alpha) = \int_0^T \|\phi(\mathbf{x},\alpha,t)\|^2 dt \end{aligned} \tag{12}$$

where ϕ is a (possibly non-linear) function with values in $\mathbb{R}^m$. We considered here the mono-criterium case (J with values in $\mathbb{R}$) but the following can easily be extended to multi-criteria problems.

The scope of local sensitivity analysis is to compute exactly and efficiently the sensitivities of the system's response to variations in the system's parameters, around their nominal values.

This is translated by finding

$$S_\alpha = \nabla_\alpha J(\alpha) = \left(\frac{\partial J}{\partial \alpha_1}(\alpha),...,\frac{\partial J}{\partial \alpha_p}(\alpha)\right)^T \tag{13}$$

Where S_α is the local sensitivity vector of J to variations in α, it is a local sensitivity because it depends on the current estimate of α (and of $\mathbf{x}_0$).

Rewriting (12) as $J(\alpha) = \langle\varphi(\alpha),\varphi(\alpha)\rangle$, where $\langle .,. \rangle$ defines a dot product on $\mathbb{R}^m$. The Gâteaux-derivative of J in any direction $\delta\alpha$ reads

$$\begin{aligned} dJ(\alpha)[\delta\alpha] &= \left\langle \varphi(\alpha), \frac{\partial\varphi}{\partial\alpha}.\delta\alpha \right\rangle = \left\langle \left[\frac{\partial\varphi}{\partial\alpha}\right]^{*}.\varphi(\alpha), \delta\alpha \right\rangle \\ &\equiv \langle \nabla J, \delta\alpha \rangle \end{aligned} \tag{14}$$

Hence $\left[\frac{\partial\varphi}{\partial\alpha}\right]^{*}$ the adjoint of φ allows for the exact computation of the p components of ∇J at once.

One can find an example of application of such methods with NEMOTAM on the Mercator-ocean's GLORYS 1/4°global ocean reanalysis in Vidard et al. (2011). The initial objective was to try to estimate the influences of geographical areas to reduce the forecast error using an adjoint method to compute the sensitivities (while the GLORYS assimilation system is based on optimal interpolation). We conducted a preliminary study by considering the misfit to observations as a proxy of the forecast error and sought to determine the sensitivity of this difference to changes in the initial condition and/or to forcing. That should give an indication about the important phenomena to consider to improve this system.

The most easily interpreted case in this study is to consider a sensitivity criterion coming from the difference in SST maps at the final instant of the assimilation cycle, because of its dense coverage in space. This can be translated into computing the gradient:

$$J(\mathbf{x}_0,\mathbf{q}) = \frac{1}{2}\sum_{n=1}^{N_{SST}} \| H_{SST}(\mathbf{x}_n) - SST^{obs} \|^2_{\mathbf{R}^{-1}} \tag{15}$$

with a control vector made of $\mathbf{x}_0 = (u_0, v_0, T_0, S_0, \eta_0)^T$ the initial state vector (current velocities component, temperature, salinity and sea level) and of $\mathbf{q} = (q_{sr}, q_{ns}, emp)^T$ (radiative fluxes, total heat fluxes, fresh water fluxes). One can see an example of sensitivity to initial temperature (surface and 100m) as shown in the two bottom panels of Figure 11. High sensitivity will give a signal similar to the gap in observations (top left), while low sensitivity will show a white area. In this example it is clear that the SST misfit is highly sensitive to changes in surface temperature where the initial mixed layer depth (top right) is low and insensitive elsewhere. The opposite conclusion can be drawn from the sensitivity to the initial temperature at 100m. This is obviously not a surprise, and corresponds more to the purpose of verification of the model rather than the original goal of assimilation system improvement. However it highlights the importance of having a good estimate of the vertical mixing and echoes the fact they this vertical mixing is often perturbed by data assimilation. Other components of the gradient show the important role of atmospheric forcing (which again we could have guessed) and ways to improve the system also appear to point to that direction. With the objective of improving the data assimilation system, this approach is obviously not completely satisfactory because, strictly speaking, the assimilation system should be included in the optimality system. In theory, this assimilation system being linear and made of matrix multiplication, to derive its adjoint should be easy, in practice it's a different story, manipulating an operational system is never easy.

[Figure 1 about here.]

The sensitivities are of interest by themselves, but they can also be used for optimising the system. In particular this way of computing ~~gradient~~ gradients is extensively used in variational data assimilation for the minimisation of similar cost function (4D-Var). For ocean application, historically the preferred choice of data assimilation technique has been (and still is for many cases) that of Optimal Interpolation or 3D-Var ~~types~~ type schemes. These algorithms make the assumptions that the system state ~~(or the increment in their FGAT formulation)~~ is stationary over a given time window (typically 1 to 10 days) which can be a crude approximation. 4D-Var does not make this assumption and uses the adjoint model to compute the gradient of a cost function of the form:

$$J(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{t=1}^{T} \| H_t(\mathcal{M}(\mathbf{x}_0,t)) - \mathbf{y}_t^{obs} \|^2_{\mathbf{R}^{-1}} \tag{16}$$

where $\|\mathbf{z}\|^2_{\mathbf{C}} = \langle \mathbf{z}, \mathbf{C}\mathbf{z} \rangle$ and $\mathbf{B}$ (resp $\mathbf{R}$) is the background (resp observation) error covariance matrix. $\mathbf{x}^b$ its the background state, and $\mathbf{y}^{obs}$ are the observations. The gradient ∇J of this cost function can be computed using relation (14):

$$\nabla J = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \sum_{t=1}^{T} \mathbf{L}^* \mathbf{H}_t^* \mathbf{R}^{-1} (H_t(\mathcal{M}(\mathbf{x}_0,t)) - \mathbf{y}_t^{obs}) \tag{17}$$

To illustrate the application of 3D-Var and 4D-Var type schemes, one can perform single observation experiments, where only one observation at the end of the assimilation window is assimilated. In that case, after a bit of algebra and assuming $\mathcal{M}(\mathbf{x}_0 + \delta\mathbf{x}, T) = \mathcal{M}(\mathbf{x}_0, T) + \mathbf{L}.\delta\mathbf{x}$ one can write the optimal state $\mathbf{x}^a$ that minimises J as:

$$\begin{aligned} \mathbf{x}^a &= \mathbf{x}^b \\ &+ \mathbf{B}\mathbf{L}^*\mathbf{H}^* \left(\mathbf{R} + \mathbf{H}\mathbf{L}\mathbf{B}\mathbf{L}^*\mathbf{H}^*\right)^{-1} \left(H_T(\mathcal{M}(\mathbf{x}_0,T)) - \mathbf{y}_T^{obs}\right) \end{aligned}$$

For a single observation experiment, it is easy to see that $\left(\mathbf{R} + \mathbf{H}\mathbf{L}\mathbf{B}\mathbf{L}^*\mathbf{H}^*\right)^{-1} \left(H_T(\mathcal{M}(\mathbf{x}_0,T)) - \mathbf{y}_T^{obs}\right)$ is a scalar, and when multiplied by $\mathbf{H}^*$ it becomes a vector in the state space, with only one non-zero value (assuming the observation is at a grid point). In 3D-Var formulation, $\mathbf{L}^*$ is approximated by the identity operator, so the correction to the initial condition outside the observed grid point is solely driven by the prescribed background error statistics in $\mathbf{B}$, while in 4D-Var the model dynamics ~~is~~ are accounted for through the adjoint model $\mathbf{L}^*$.

An example of such differences is given in Figure 12 where a single synthetic SSH observation, close to the middle of the regional model, at the end of ~~DA~~ the data assimilation time window, is assimilated using both ~~3D-FGAT~~ 3D-Var and incremental 4D-Var algorithms from the NEMOVAR system with NEMO's SEABASS configuration (see appendix A). The observation misfit value is 0.5 m.

The 3D-Var increment (top figure) shows a perfect gaussian shape, centred around the observation location, with an maximum amplitude close to the observation value. This gaussian shape is exactly what is prescribed in the background error covariance matrix $\mathbf{B}$, and the computed increment is independent of the length of the assimilation window. On the other hand, 4D-Var increment is sensitive to the assimilation window length. Two examples are given, first with a 5 day window (bottom left) and second with a 30 days window. In the first case, the 3D-Var approximation is ~~not that bad~~acceptable, so both 3D- and 4D-Var are similar, even though the latter is slightly deformed and displaced to account for the short term dynamics. For the longer assimilation window (bottom right) however, the effect of the dynamics is more complex, in particular the ~~non linearities~~ non-linearities are more developed. As a consequence the 3D-Var approximation is no longer valid and the shape of the optimal correction is completely different.

This obviously comes at a cost, since 3D-Var would only require one direct model integration and 4D-Var would additionally require one tangent and adjoint integration per minimisation iteration. In a single observation experiment as presented above the minimisation converges in only one iteration, limiting the cost of 4D-Var to about 4 times that of 3D-Var. In a more realistic configuration one performs about 30 to 50 iterations, leading 4D-Var to be up to 200 times more cpu-expensive than 3D-Var. This is why, in many implementation the minimisation is performed at a lower resolution than the forecast.

[Figure 2 about here.]

4.2 Singular vectors

Another application of tangent and adjoint models is ~~the~~ stability analysis, that is the study of perturbations on the system. Particular tools for such analysis are the so-called singular vectors.

We classically define the growth rate of a given perturbation $\delta\mathbf{x}_0$ by

$$\rho(\delta\mathbf{x}_0) = \frac{\|\mathcal{M}(\mathbf{x}_0+\delta\mathbf{x}_0,T)-\mathcal{M}(\mathbf{x}_0,T)\|}{\|\delta\mathbf{x}_0\|} \; \frac{\|\mathcal{M}(\mathbf{x}_0+\delta\mathbf{x}_0,T)-\mathcal{M}(\mathbf{x}_0,T)\|_1}{\|\delta\mathbf{x}_0\|_2} \qquad (18)$$

where ~~$\|.\|$ is a given norm~~$\|.\|_1 = <.,\mathbf{W}_1.>$ and $\|.\|_2 = <.,\mathbf{W}_2.>$ are given norms and T is the final time of the considered window.

One can then define the optimal perturbation $\delta\mathbf{x}_0^1$ so that $\rho\left(\delta\mathbf{x}_0^1\right) = \max_{\delta\mathbf{x}_0} \rho(\delta\mathbf{x}_0)$ and then deduce a family of maximum growth vectors

$$\rho\left(\delta\mathbf{x}_0^i\right) = \max_{\delta\mathbf{x}_0 \perp \mathrm{Span}\left(\delta\mathbf{x}_0^1,\ldots,\delta\mathbf{x}_0^{i-1}\right)} \rho(\delta\mathbf{x}_0) \qquad ,i \geq 2 \qquad (19)$$

By restricting the study to the linear part of the perturbation behaviour, the growth rate becomes (denoting $\mathbf{L} = \mathbf{L}(\mathbf{x},T)$ for clarity).

$$\begin{aligned}\rho^2(\delta\mathbf{x}_0) = \frac{\|\mathbf{L}\delta\mathbf{x}_0\|_1^2}{\|\delta\mathbf{x}_0\|_2^2} &= \frac{<\mathbf{L}\delta\mathbf{x}_0,\mathbf{W}_1\mathbf{L}\delta\mathbf{x}_0>}{<\delta\mathbf{x}_0,\delta\mathbf{x}_0>} \\ &= \frac{<\delta\mathbf{x}_0,\mathbf{L}^*\mathbf{W}_1\mathbf{L}\delta\mathbf{x}_0>}{<\delta\mathbf{x}_0,\mathbf{W}_2\delta\mathbf{x}_0>}\end{aligned} \qquad (20)$$

~~$\mathbf{L}^*\mathbf{L}$~~ maximising the above equation is equivalent to solving the following generalised eigenvalue problem

$$\mathbf{L}^*\mathbf{W}_1\mathbf{L}g_i^+ = \mu_i\mathbf{W}_2 g_i^+ \qquad (21)$$

Which is equivalent, using the change of variable $g_i^+ = \mathbf{W}_2^{-1/2} f_i^+$ to

$$\mathbf{W}_2^{-1/2}\mathbf{L}^*\mathbf{W}_1\mathbf{L}\mathbf{W}_2^{-1/2} f_i^+ = \mu_i f_i^+ \qquad (22)$$

$\mathbf{W}_2^{-1/2}\mathbf{L}^*\mathbf{W}_1\mathbf{L}\mathbf{W}_2^{-1/2}$ being a symmetric positive definite matrix, its eigenvalues are positive real and its eigenvectors are (or can be chosen) ~~orthogonal~~orthonormal. The strongest growth vectors are the eigenvectors of ~~$\mathbf{L}^*\mathbf{L}$~~ $\mathbf{W}_2^{-1/2}\mathbf{L}^*\mathbf{W}_1\mathbf{L}\mathbf{W}_2^{-1/2}$ corresponding to the greater eigenvalues. They are called forward singular vectors~~(FSV)~~.

~~$\mathbf{L}^*\mathbf{L}f_i^+ = \mu_i f_i^+$~~

~~$\mathbf{L}f_i^+$ is an eigenvector of $\mathbf{L}\mathbf{L}^*$.~~

The backward singular vectors~~(BSV)~~, noted f_i^-, are noted by:

$$\mathbf{L}\mathbf{W}_1^{1/2}\mathbf{L}\mathbf{W}_2^{-1/2} f_i^+ = \sqrt{\mu_i}\, f_i^-$$

the eigenvalue corresponding to f_i^- is μ_i as well. ~~FSVs~~ Forward singular vectors represent the directions of perturbation that will grow fastest, while ~~BSVs~~ backward singular vector represent the directions of perturbation that have grown the most. $\sqrt{\mu_i}$ is the amplification factor associated to the i^{th} singular vector.

The computation of the f_i^+ and f_i^- generally requires numerous matrix-vector multiplications, *i.e.* direct integrations of the model and backward adjoint integrations. The result of these calculations depends on the norm used, the time window and the initial state if the model is ~~nonlinear~~non-linear. Examples of such vectors for a $1/12^{th}$ of degree SEABASS configuration are shown in Fig.13~~, they~~. The left panel shows the sea surface height component of the first forward singular vector that exhibits a strong signal over the dominant jet of this configuration, showing that the optimal perturbation is located in the most active region (as it is shown in Fig.15). The right panel shows the corresponding backward singular vector, *i.e.* the result of the evolution of the optimal perturbation through the linear part of the dynamics. The complex structure of the original perturbation has been transformed into several individual vortices. This is similar to what (Durbiano, 2001) presented for a shallow-water model.

These singular vectors were computed using an energy norm to define ~~$\|.\|$~~ $\|.\|_{1,2}$ and the parpack (Lehoucq et al., 1997) external library to perform the singular value decomposition. The computational cost required for obtaining singular vectors can vary from one situation to another since it depends on the number of iterations the Arnoldi algorithm used in parpack takes to converge, which itself depends on the eigenspectrum. For instance it took from 27 to 49 iterations for computing the 10 leading singular vectors of the different cases discussed in Figure 14, each iterations requiring the integration of both tangent and adjoint models.

[Figure 3 about here.]

These vectors, thanks to the information they contain about the system behaviour, have many applications. Among them, one can cite ensemble forecast, sensitivity studies (Rivière et al. (2009) for a recent application), the order reduction in data assimilation (Blayo et al., 2003), improving the monitoring network (Qin and Mu, 2011) or allow to better select targeted observations (Mu et al., 2009). ~~Apart from the generation of ensemble, however, the potential of these vectors has yet been little exploited.~~

A by-product of the computation of singular vectors is the amplification factor $\sqrt{\mu_i}$ that represents the growth of the corresponding singular vector at T the end of the time window. Figure 14 shows the impact of the length of this time window (left) and of the model resolution (right). The impact of the latter is obvious, the higher the resolution the more active the model, hence the faster perturbations amplify. Regarding the time window length it is less obvious. For a short period of time as presented here the link is the same, perturbations get more time to grow therefore the amplification factor increases. With a longer time window (few years or decades) optimal perturbations are such that amplification factors will start to decrease. This dipping point is important in predictability studies (see Zanna et al. (2011) for more details on this topic).

[Figure 4 about here.]

5 Conclusions

The tangent and adjoint models of NEMO (NEMOTAM) is now available (with respect to the 3.4.1 version of NEMO at the time of writing this paper). It is part of the NEMO-ASSIM tools (Bouttier et al., 2012), which aim is to ease the interface between NEMO code and ~~a~~ most Data Assimilation algorithms. In the few preceding pages, these models, the technical choices made for their developments and their validation were discussed. Additionally some applications were presented as an illustration of potential use.

When developing a TAM, two main difficulties have to be addressed: the handling of the non-linearities and of the non-differentiable parts. Indeed ~~non linear~~ non-linear equations require the storage and/or ~~recompilation of the non linear~~ recomputation of the non-linear trajectory to differentiate around. In NEMOTAM this is done through the so-called checkpointing strategy, consisting in saving part of the non-linear trajectory at a given frequency (checkpoints), recomputing the missing part and, if needed, interpolating it linearly between checkpoints. On the other hand the non-differentiability issues are dealt with using three different approaches~~approaches were used, depending of~~ , depending on the discontinuity nature: numerical or physical regularisation, numerical or physical approximation and non-linear branching.

All these choices have to be validated along with the coding itself. To that end a significant effort has been ~~done~~ invested in NEMOTAM. First, adjoint tests are systematically implemented for each adjoint ~~routines and gives~~ routine and give an exact indication of the validity of the adjoint code. For the tangent linear model, there is no exact test which can strictly validate the development. However, comparing the propagation of a small perturbation by ~~TL model and and~~ the tangent-linear model and the direct model gives an idea of the validity of the ~~TL~~ tangent-linear hypothesis. Finally, an estimator of the errors due to the approximations ~~on~~ of NEMO non-differentiable parts is also provided for the corresponding routines.

The range of applications using NEMOTAM is wide. To illustrate that, three example applications were ~~exposed~~presented. First a local sensitivity analysis with a realistic NEMO configuration. Then a very simple data assimilation experiment, using a single observation is also performed, illustrating the impact of the use of an adjoint model. And finally some singular vectors were computed using NEMOTAM.

The scope of the current NEMOTAM implementation leaves ~~the~~ room for different extensions (e.g. taking in account other NEMO modules, ~~as LIM (sea ice) model, AGRIF~~such as the sea ice model, nested grids, etc.). In order to handle properly the ~~MPP~~ multi-processor aspects and optimize computing cost, the current NEMOTAM is hand-coded. ~~However,~~ Such approach can limit the range of possible input quantities to be considered when computing sensitivities. As it is it can compute, without any change of code, sensitivities to perturbations to the initial condition and/or surface boundary conditions, and with very limited modifications, sensitivities to perturbation of physical parameters (such as bottom friction for instance). However for parameters that are used throughout the model code, such as grid-related parameters, it would require a significant amount of coding. This is why, in order to provide more flexibility in the choice of variable to differentiate around and to ease the process of updating the NEMOTAM code with new features and following the evolution of the direct code, a subtle trade-off between automatic differentiation and manual intervention could be very beneficial.

6 Code availability

NEMO Tangent and Adjoint Model (NEMOTAM) is released with NEMO 3.4 STABLE version for the first time as an official NEMO component. To extract this version of NEMO, user needs to create his account on NEMO website (www.nemo-ocean.eu/user/register). After that, source code can using the relevant SVN command. Instructions to compile and to run NEMOTAM can be found at this URL (visible once logged in): http://www.nemo-ocean.eu/Using-NEMO/User-Guides/Basics/Tangent-and-Adjoint-quick-start-guide

Appendix A: SEABASS configuration

The SEABASS reference configuration for NEMO is mentioned several times in this paper. This configuration is an academic basin presenting a double-gyre circulation. The horizontal domain extends from 24°N to 44°N and over 30° in longitude. For a 1/4° horizontal resolution, the grid contains only 121 points in longitude and 81 points in latitude. The time step is 900 seconds. The ocean is sliced into 11 verticals levels, from surface to 4000 meters, described with a z-coordinate. The domain is closed and has a flat bottom. Lateral boundaries conditions are frictionless and bottom boundary condition exerts a linear friction. The circulation is only forced by a zonal wind. Lateral dissipation is performed on dynamics and tracers with a biharmonic diffusion operator. The salinity is constant over the whole domain and the initial stratification is produced using an analytical temperature profile. Details can be found in Cosme et al. (2010), for example.

Even if SEABASS is an academic configuration, it exhibits a turbulence level statistically meaningful regarding the eddy activity and the non-linearity amplitude of the actual Gulf Stream system. This SEABASS characteristics is interesting in a data assimilation context, as the oceanic turbulence is one of the major current stakes for data assimilation methods in oceanography.

[Figure 5 about here.]

Acknowledgements. The authors would like to thanks the NEMOVAR group in general and A. Weaver and K. Mogensen in particular for their participation in the early phase of the development of NEMOTAM. Another precious help came from R. Benshila who guided us through the meander of the NEMO system. This work was also financially supported by CNES and the french national research agency (ANR project VODA: ANR-08-COSI-016).

References

Blayo, E., Durbiano, S., Vidard, A., and Le Dimet, F.-X.: Reduced order strategies for variational data assimilation in oceanic models., in: Data Assimilation for Geophysical Flows., edited by Sportisse, B. and Le Dimet, F.-X., p. 18pp, Springer-Verlag, 2003.

Bouttier, P.-A., Blayo, E., Brankart, J.-M., Brasseur, P., Cosme, E., Verron, J., and Vidard, A.: Toward a data assimilation system for NEMO, Mercator Ocean Quarterly Newsletter, pp. 24–30, 2012.

Daget, N., Weaver, A. T., and Balmaseda, M. A.: Ensemble estimation of background-error variances in a three-dimensional variational data assimilation system for the global ocean., Q. J. R. Meteorol. Soc., 135, 1071–1094, 2009.

Deltel, C.: Estimation de la circulation dans l'océan Atlantique Sud par assimilation variationnelle de données in situ. Impact du contrôle optimal des forçages et de l'hydrologie aux frontières ouvertes. 185 pp., Ph.D. thesis, Univ. de Bretagne Occidentale, 2002.

Durbiano, S.: Vecteurs caractéristiques de modèles océaniques pour la réduction d'ordre en assimilation de données, Ph.D. thesis, Université Joseph Fourier, 2001.

Errico, R. M.: What Is an Adjoint Model?, Bull. Amer. Meteorol. Soc., 78, 2577–2591, 1997.

Forget, G., Ferron, B., and Mercier, H.: Combining ARGO profiles with a general circulation model in the North Atlantic. Part1: estimation of hydrographic and circulation anomalies from synthetic profiles, over a year, Ocean Modelling, doi:10.1016/j.ocemod.2007.06.001, 2008.

Giering, R. and Kaminski, T.: Recipes for adjoint code construction, ACM Trans. Math. Softw., 24, 437–474, 1998.

Hascoët, L. and Pascual, V.: TAPENADE 2.1 user's guide, Technical Report 0300, INRIA, 2004.

Lawless, A., Nichols, N., and Ballard, S.: A comparison of two methods for developing the linearization of a shallow-water model, Q.J.R. Meteorological Society, 129, 1237–1254, 2003.

Lehoucq, R. B., Sorensen, D. C., and Yang, C.: ARPACK Users' Guide, 1997.

Madec, G.: NEMO ocean engine, IPSL, Note N° 27 du Pôle de Modélisation (LOCEAN-IPSL), 2008.

Madec, G., Delecluse, P., Imbard, M., and Lévy, C.: OPA 8.1 ocean general circulation model reference manual, IPSL, Note N° 11 du Pôle de Modélisation (LODYC), 1998.

Marotzke, J., Giering, R., Zhang, K. Q., Stammer, D., Hill, C., and Lee, T.: Construction of the adjoint MIT ocean general circulation model and application to Atlantic heat transport sensitivity, J. Geophys. Res., 104, 29 529–29 547, 1999.

Mogensen, K. S., Balmaseda, M. A., Anthony, W., Martin, M., and Vidard, A.: NEMOVAR: A variational data assimilation system for the NEMO ocean model, ECMWF newsletter, 120, 17–21, 2009.

Moore, A., Vialard, J., Weaver, A., Anderson, D., Kleeman, R., and Johnson, J.: The Role of Air-Sea Interaction in Controlling the Optimal Perturbations of Low-Frequency Tropical Coupled Ocean-Atmosphere Modes, J. Climate, 16, 951–968, 2003.

Moore, A., Arango, H., Di Lorenzo, E., Cornuelle, B., Miller, A., and Neilson, D.: A comprehensive ocean prediction and analysis system based on the tangent linear and adjoint of a regional ocean model, Ocean Modelling, 7, 227–258, 2004.

Mu, M., Zhou, F., and Wang, H.: A Method for Identifying the Sensitive Areas in Targeted Observations for Tropical Cyclone Prediction: Conditional Nonlinear Optimal Perturbation, Mon. Wea. Rev., 137, 1623–1639, 2009.

Qin, X. and Mu, M.: Influence of conditional nonlinear optimal perturbations sensitivity on typhoon track forecasts, Q.J.R. Meteorol. Soc., pp. n/a–n/a, 2011.

Rémy, E.: Assimilation variationnelle de données tomographiques simulées dans des modèles de circulation océanique, Ph.D. thesis, Univ. Paris 6, 1999.

Rivière, O., Lapeyre, G., and Talagrand, O.: A novel technique for nonlinear sensitivity analysis: application to moist predictability, Q.J.R. Meteorol. Soc., 135, 1520–1537, 2009.

Sévellec, F., Huck, T., BenJelloul, M., Grima, N., Vialard, J., and Weaver, A.: Optimal surface salinity perturbations of the meridional overturning and heat transport in a global ocean general circulation model, J. Phys. Oceanogr., 38, 2739–2754, 2008.

Sévellec, F., Huck, T., BenJelloul, M., and Vialard, J.: Non-normal multidecadal response of the thermohaline circulation induced by optimal surface salinity perturbations, J. Phys. Oceanogr., 39, 852–872, 2009.

Tber, M. H., Hascoet, L., Vidard, A., and Dauvergne, B.: Building the Tangent and Adjoint codes of the Ocean General Circulation Model OPA with the Automatic Differentiation tool TAPENADE, Research Report RR-6372, INRIA, 2007.

Thuburn, J.: Adjoints of Nonoscillatory Advection Schemes, Journal of Computational Physics, 171, 616–631, 2001.

Vidard, A.: Vers une prise en compte de l’erreur modèle en assimilation de données 4D-variationnelle, Ph.D. thesis, Université Joseph Fourier, 2001.

Vidard, A., Rémy, E., and Greiner, E.: Sensitivity analysis through adjoint method: application to the GLORYS reanalysis, Contrat n° 08/D43, Mercator Océan, 2011.

Vossepoel, F. C., Weaver, A., Vialard, J., and Delecluse, P.: Adjustment of near-equatorial wind stress with 4D-Var data assimilation in a model of the Pacific Ocean, Mon. Weather Rev., 132, 2070–2083, 2003.

Weaver, A. T., Vialard, J., and Anderson, D. L. T.: Three- and Four-Dimensional Variational Assimilation with a General Circulation Model of the Tropical Pacific Ocean. Part I: Formulation, Internal Diagnostics, and Consistency Checks., Mon. Weather Rev., 131, 1360–1378, 2003.

Weaver, A. T., Deltel, C., Machu, E., and Ricci, S.: A multivariate balance operator for variational ocean data assimilation. Appeared in 2006 in a Special Issue., Q. J. R. Meteorol. Soc., 131, 3605–3625, 2005.

Zanna, L., Heimbach, P., Moore, A. M., and Tziperman, E.: Upper-ocean singular vectors of the North Atlantic climate with implications for linear predictability and variability, Q.J.R. Meteorol. Soc., 138, 500–513, 2011.

List of Tables

[h]

Table 11. Tangent validity tests for the bn2 routine (computation of the Brünt-Väisälä frequency), Values of the first order accuracy index ϵ_γ for several values of the perturbation amplitude γ

Routine (**L**)	γ	ϵ_γ
bn2_tan	1E+00	.999961862090
bn2_tan	1E-01	.999995878199
bn2_tan	1E-02	.999999584740
bn2_tan	1E-03	.999999958442
bn2_tan	1E-04	.999999995846

[h]

Table 12. Approximation error in the tangent linear model for different configurations over 10 days

Configurations	$\hat{\mathcal{E}}$
SEABASS no simplification	negligible
SEABASS interpolated checkpoint (1/day)	negligible
SEABASS simplified TVD	3%
ORCA2 simplified vertical mixing	1%
ORCA2 idem + TVD + interp. checkpoint (1/day)	4%

List of Figures

[h]

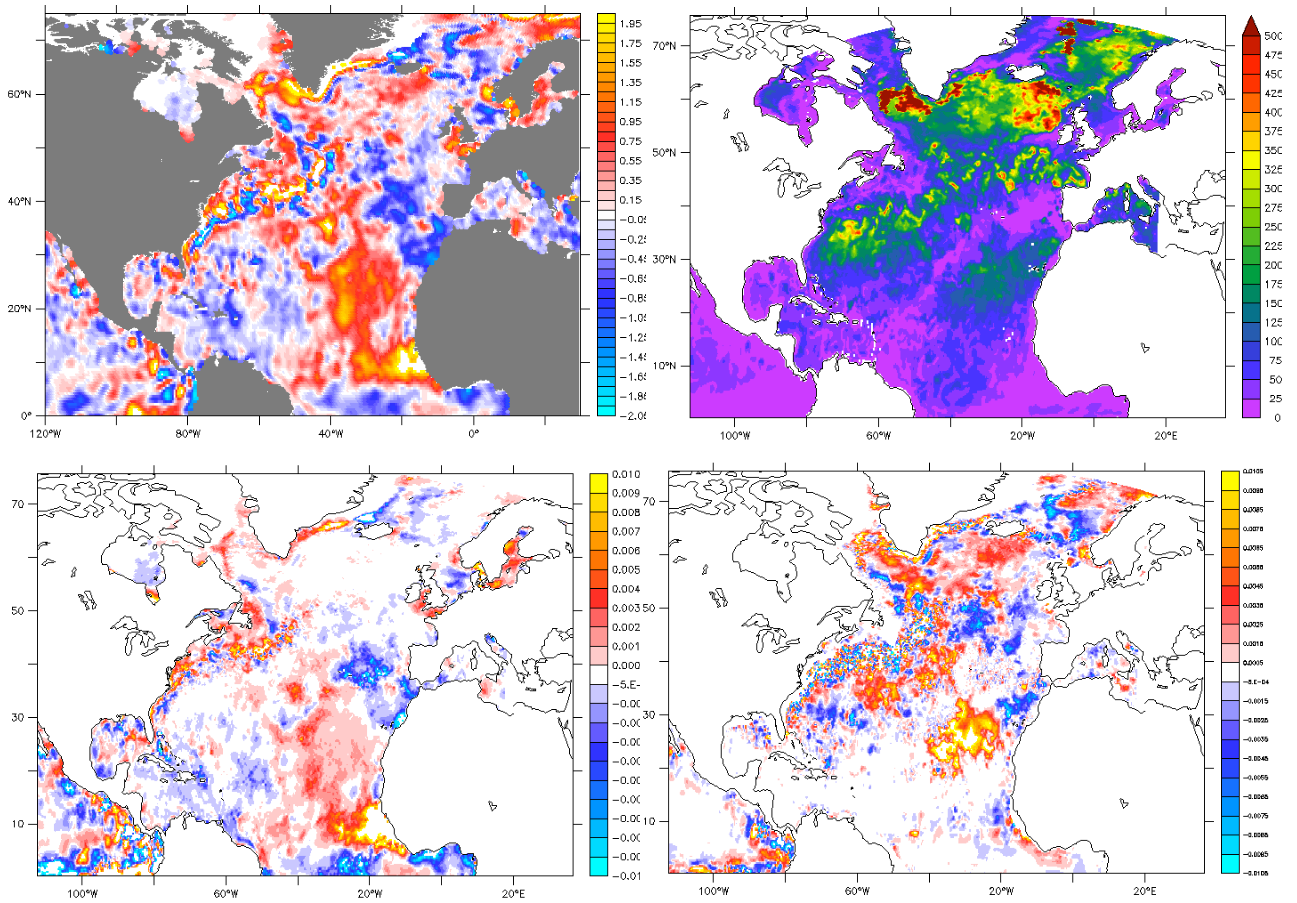

Figure 11. Top: misfit between forecast and observed SST (left) and mixed layer depth (right). Bottom: sensitivity to one week lead time SST error with respect to variations in initial surface (left) and 100m (right) temperature (courtesy E. Rémy, Mercator-Océan)

[h]

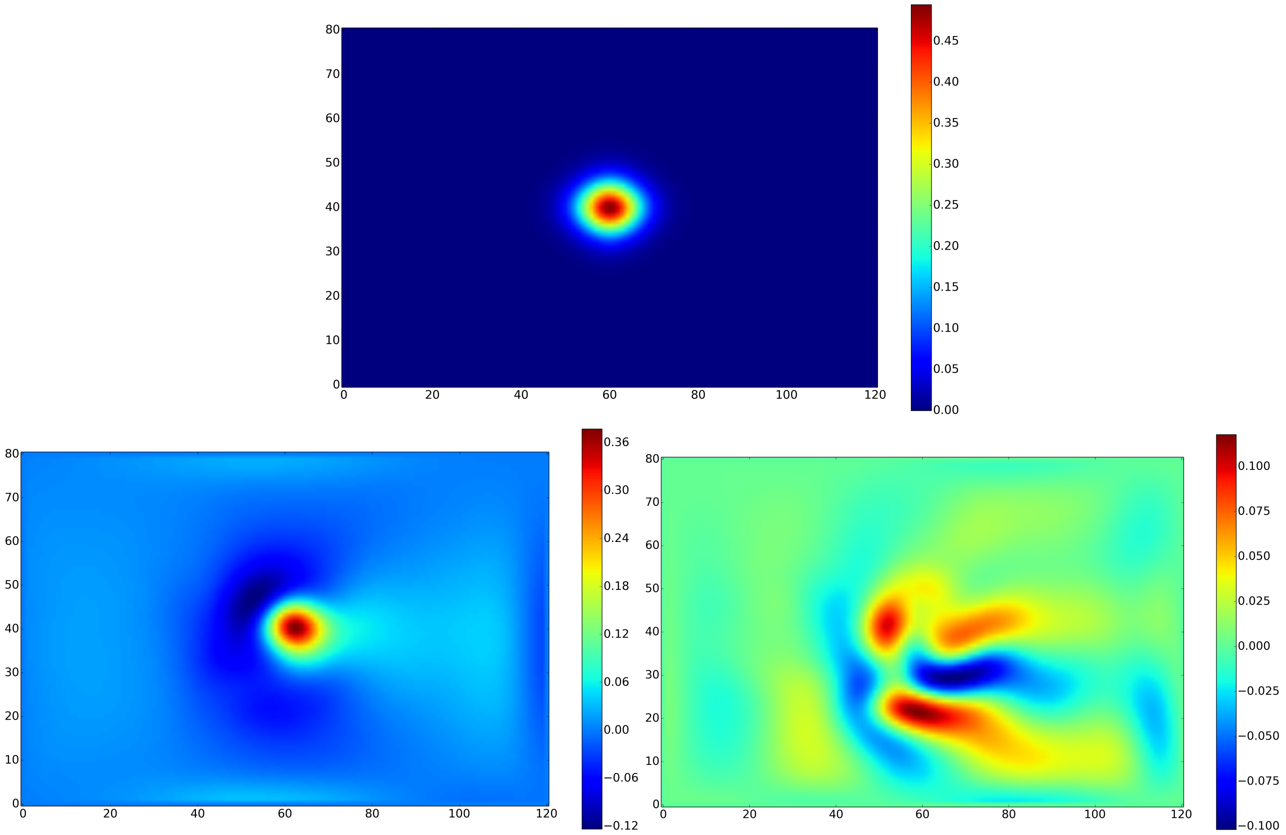

Figure 12. Assimilation increments in the SEABASS configuration corresponding to one SSH observation and a misfit of 0.5m. Coming from NEMOVAR ~~3D-FGAT~~ 3D-Var (top) and 4D-Var (bottom) formulation. Two different assimilation ~~widows length~~ window lengths are presented for the 4D-Var: 5 days (left) and 30 days (right).

[h]

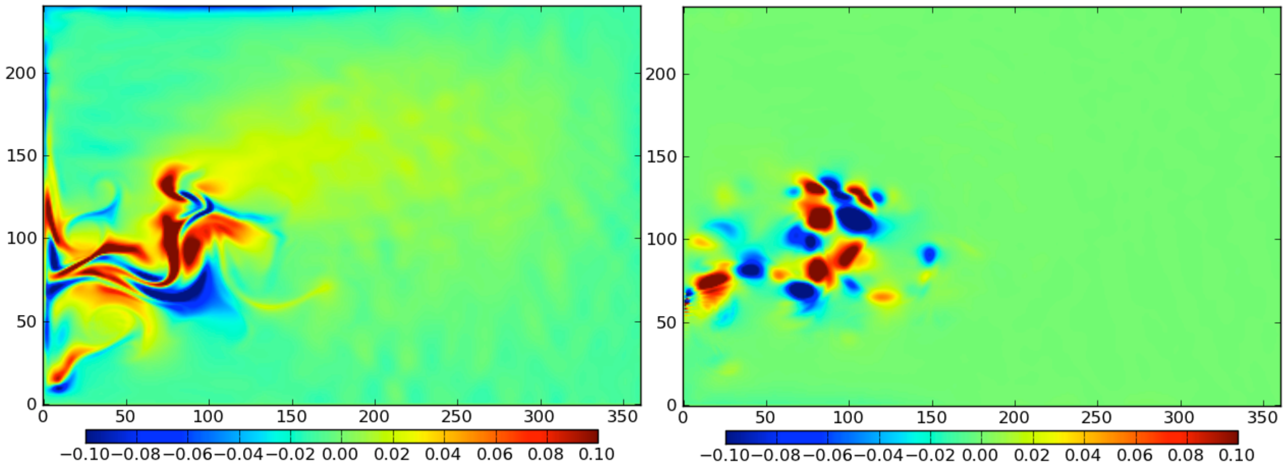

Figure 13. ~~Leading FSV~~ Sea Surface Height component of the leading forward(left) and ~~BVF~~ backward (right) singular vectors for a SEABASS configuration at 1/12° and a 10 day window

[h]

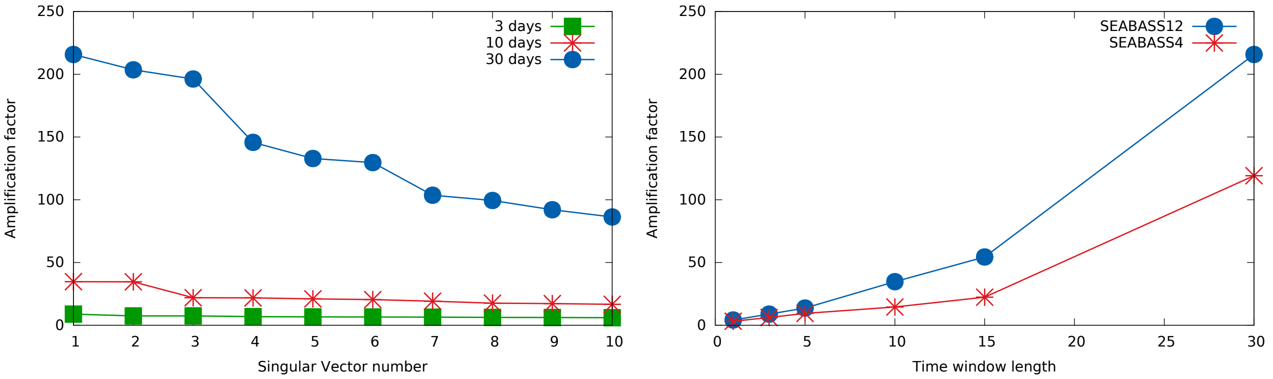

Figure 14. Amplification factor for the 10 first singular vectors of the 1/12° SEABASS configuration for several time windows (left) and comparison of the amplification factors of the first singular vector for the 1/12° and the 1/4° SEABASS configurations (right)

[h]

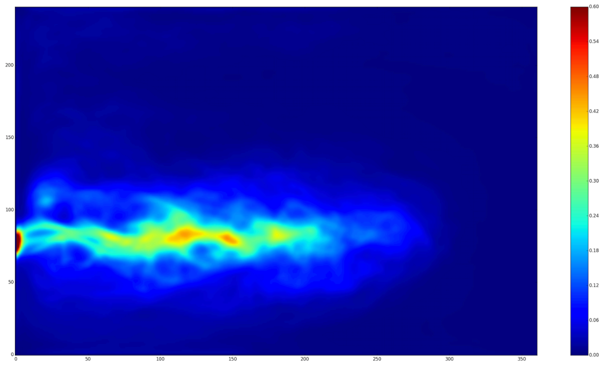

Figure 15. Surface Eddy Kinetic Energy averaged over 1 year (in $m^2.s^{-2}$), for the SEABSS configuration at a 1/12° horizontal resolution