On the conflict between implicit stresses and barotropic (ice) sub-stepping in NEMO

Jérôme Chanut, Mercator Océan

Outline

- Bottom friction in NEMO
  - Implicit vs explicit bottom friction
  - Interaction with barotropic subcycling
- Proposed solution
- Illustration in a regional tidal model (AMM12)
- Extension to ice-ocean drag ?

Bottom friction in NEMO

- NEMO can handle bottom stresses explicitly (ln_bfrimp=F) :

$$\tau_b = -rd \frac{{u_{kbot}}^{n-1}}{h^n}$$

or implicitly (ln_bfrimp=T) :

$$\tau_b = -rd \frac{{u_{kbot}}^{n+1}}{h^n}$$

Where $rd = rn\_bfri1$ (linear)

$= rn\_bfri2 \left\| {u^n}_{kbot} \right\|$ (quadratic)

$= \left\{ \frac{\kappa}{ln(0.5\ dz_{kbot}/rn\_bfrz0)} \right\}^2 \left\| {u^n}_{kbot} \right\|$ (quadratic, resolved log layer)

- Assuming explicit bottom friction fits well into the modular code structure: it does not interfere with other model components that need the knowledge of bottom stress in advance (e.g. free surface schemes). Historically, this was implemented to ensure the compatibility with (implicit) filtered free surface.

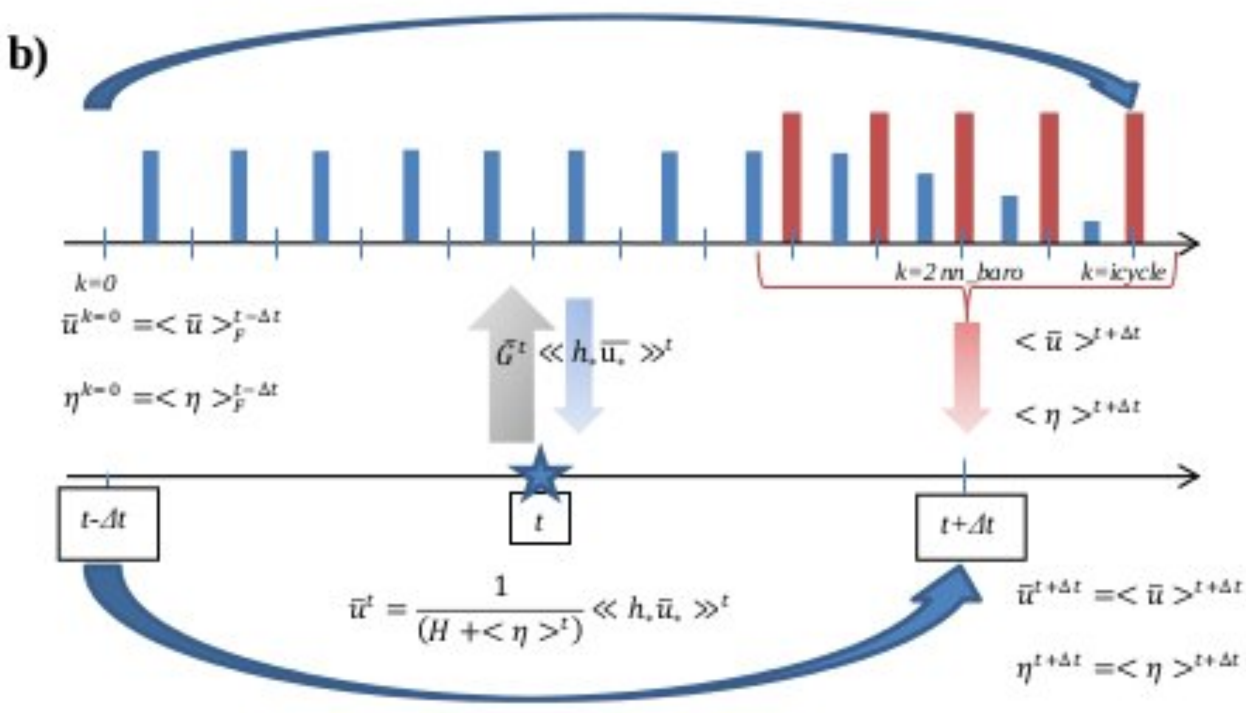
- However, in the explicit case, stability constraint impose to bound the bottom friction coefficient:

$$rd < dz_{k=kbot} / 2\Delta t$$

Which can be a severe and unphysical constraint on the shelf, with tides and small bottom thicknesses (this inevitably occurs with partial bottom cells).

Bottom friction in NEMO

We assume in the following NEMO time splitting adapted to leapfrog (eg, ln_bt_fw=F)



n-1 (« before ») n (« now ») n+1 (« after »)

Bottom friction in NEMO: conflict with barotropic sub-stepping

- Barotropic substepping needs bottom stress but in turn updates barotropic tendencies hence « after » velocities.

- How to reconciliate implicit bottom stress and split explicit free surface is a bit unclear in many models. This problem was raised by A. Shchepetkin in 2012 in ROMS. http://people.atmos.ucla.edu/alex/ROMS/SaltLakeCity2012Talk.pdf

- ROMS has explicit vertical mixing (with CFL limited bottom friction) to garanty almost identical identical bottom stresses between 2d and 3d modes. However, clipping can have a strong impact.

- In the implicit case, each model seems to have a different strategy. Most of the time (e. g. in POM, Symphonie, BOM), vertical implicit mixing (incl. bottom stresses) is performed at the end of the full 3d time stepping, leading to a mismatch between what comes out the barotropic loop and the bottom stress effectively felt by 3d velocities. A barotropic correction is then added as a necessary evil but it breaks bottom boundary conditions (see illustration later).

Bottom friction in NEMO: current flowchart

NEMO's strategy (POM/BOM like):

1. Freeze bottom stress due to baroclinic velocities at previous time step estimate in 2d sub-stepping: $\tau'_b = -rd\left(u_{kbot}{}^{n-1} - \bar{u}^{n-1}\right)/h^n$
2. Update at each barotropic sub step the bottom stress.
3. Finalize 2d sub-stepping to get $\bar{u}^{n+1} and \;\; \eta^{n+1}$
4. Perform implicit vertical mixing incl. bottom stress at step n+1: finalize u^{n+1}
5. Ensure: $\int_{-H}^{\eta^{n+1}} u^{n+1} dz = (H + \eta^{n+1})\bar{u}^{n+1}$

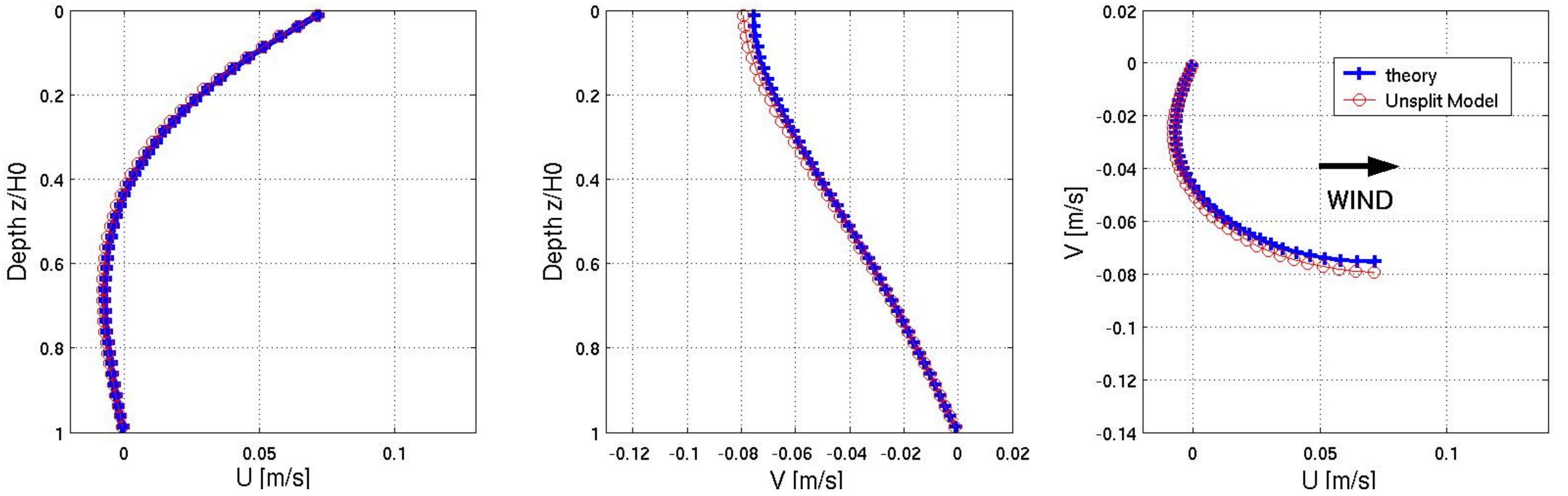
```
          CALL dyn_ldf ( kstp )   ! lateral mixing
          CALL dyn_hpg ( kstp )   ! horizontal gradient of Hydrostatic pressure
1,2,3     CALL dyn_spg ( kstp )   ! surface pressure gradient

          IF( ln_dynspg_ts ) THEN
                   CALL div_hor ( kstp )
                   IF(.NOT.ln_linssh) CALL dom_vvl_sf_nxt( kstp, kcall=2 )
                   CALL wzv   ( kstp )   ! now cross-level velocity
          ENDIF

          CALL dyn_bfr ( kstp )   ! bottom friction (if explicit)
4         CALL dyn_zdf ( kstp )   ! vertical diffusion
5         CALL dyn_nxt ( kstp )   ! finalize (bcs) velocities at next time step and swap
```

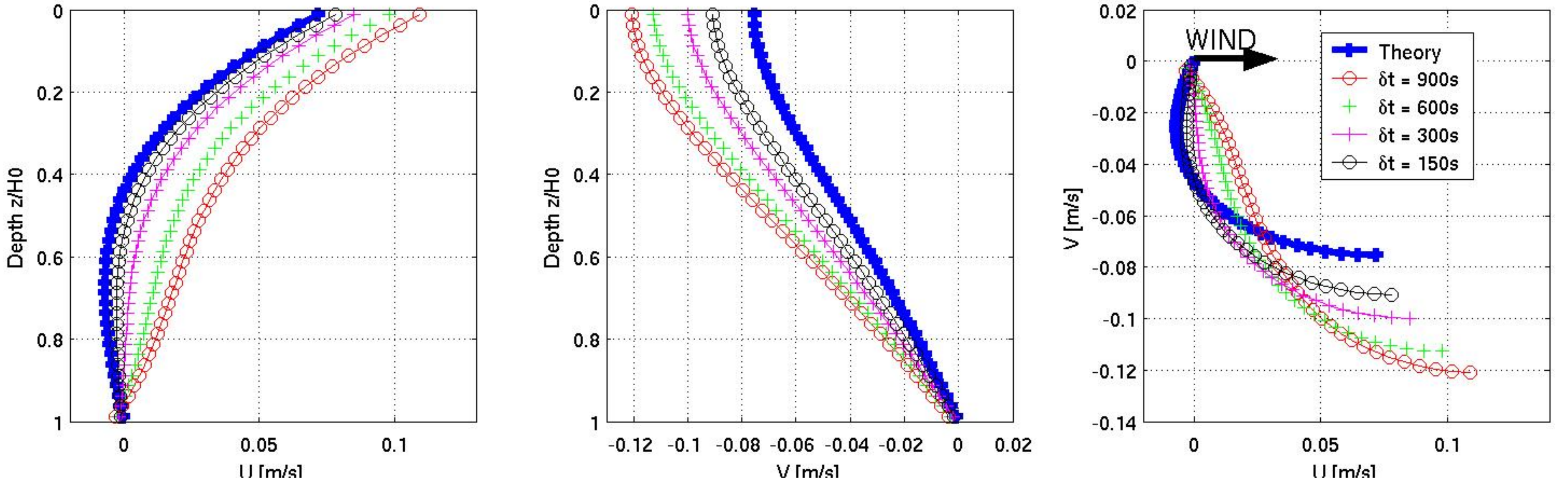
Illustration: Shchepetkin (2012) test case

- 1d wind forced ekman problem, shallow water (H=10m, jpk=41, dz=25cm)
- Constant Eastward Wind = 0.04 N m^{-2}
- Constant viscosity avmv0=13 $cm^2 s^{-1}$
- Strong linear bottom friction to emulate no slip bottom: rn_bfri1=1 $cm.s^{-1}$
- Latitude=45°N
- Vary time step, keep barotropic time step constant=15s , hence vary splitting ratio



- Unsplit model is ok

Illustration: Shchepetkin (2012) test case



- ➢ Stationary solution depends on baroclinic time step.
- ➢ Same conclusion drawn by SH12 (solution converges however differently).

Proposed solution

- One can fix this by considering the tridiagonal matrix solved during implicit vertical mixing. Inspiration comes from Hallberg and Adcroft (2009)*.

- The « after » barotropic velocity increment corresponding to a given barotropic tendency could be extracted from the matrix inversion.

- implicit bottom friction should then be properly implemented in the barotropic sub-stepping if one can extract an *effective* (depending on viscosity profiles) bottom friction drag.

*Hallberg, R. and A. Adcroft, 2009: Reconciling estimates of the free surface height in Lagrangian vertical coordinate ocean models with mode-split time stepping. Ocean modelling, 29, 15-26.

Proposed solution: practically speaking

1d velocity diffusion (backward, unconditionnaly stable):

$$\frac{u^{n+1} - u^{n-1}}{2\tau} = T(u^n, u^{n-1}) - \partial_z \mu \partial_z u^{n+1}$$

Which is solved by inverting a tridiagonal matrix depending on viscosities, thicknesses and bottom drag:

$$U^{n+1} = (I - A)^{-1}(2\tau T + U^{n-1})$$

Nothing new until now. Just consider a possibly unknown barotropic *tendency* increment $\delta T(\overline{u})$, this translates into a baroclinic velocity increment:

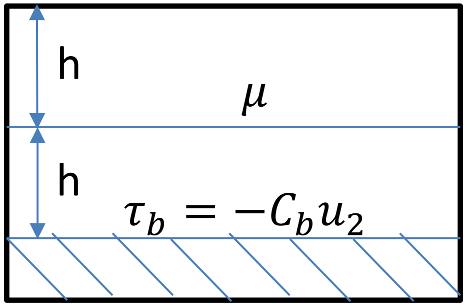
$$\Delta U = 2\tau\ \delta T(\bar{u})\ \ (I - A)^{-1} \begin{pmatrix} 1 \\ 1 \\ \dots \\ 1 \\ 1 \end{pmatrix} = 2\tau\ \delta T(\bar{u})\ \mathrm{B}$$

Where B is a (jpk,1) matrix which gives the projection of a barotropic tendency into a baroclinic current. If implicit bottom stress is null, then B=1.

Taking the vertical weighted sum, gives the barotropic increment. One can define a linear « effective » drag as seen by the barotropic mode:

$$\Delta\bar{u} = 2\tau\ \delta T(\bar{u})\bar{B} = 2\tau\ \delta T(\bar{u})\ /(1 + \frac{2\tau\ C_{beff}}{H})$$

2 layer case



1- Matrix inversion:

$$I - A = \begin{pmatrix} 1+\alpha & -\alpha \\ -\alpha & 1+\alpha+\beta \end{pmatrix}$$

$$(I-A)^{-1} = \frac{1}{1+2\alpha+\beta(1+\alpha)} \begin{pmatrix} 1+\alpha+\beta & \alpha \\ \alpha & 1+\alpha \end{pmatrix}$$

2- Projection of barotropic tendency into baroclinic velocities:

$$\mathrm{B} = (I-A)^{-1}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix} - \frac{\beta}{1+2\alpha+\beta(1+\alpha)}\begin{pmatrix}\alpha\\1+\alpha\end{pmatrix}$$

$$\alpha = \frac{2\tau\mu}{h^2}$$

$$\beta = \frac{2\tau C_b}{h}$$

3- Projection of barotropic tendency into barotropic velocity:

$$\bar{B} = \frac{1}{2h} B \begin{pmatrix}h\\h\end{pmatrix} = 1 - \frac{\beta(1+2\alpha)}{2\big(1+2\alpha+\beta(1+\alpha)\big)}$$

Which can be rewriten as an « effective » linear barotropic drag:

$$C_{beff} = \frac{2h}{2\tau}\frac{1-\bar{B}}{\bar{B}} = C_b \frac{(1+2\alpha)}{\big((1+2\alpha)+\beta/2\big)} < C_b$$

$$\mu \to \infty \; then \; C_{beff} \to C_b$$

$$\tau \to 0 \; then \; C_{beff} \to C_b$$

Proposed solution:
Ad hoc bottom stress formulation in the barotropic mode

$$u^{k+1} = \frac{u^k + dt\,T^{k+0.5}}{1 + C_b dt/h}$$ k=0, nn_baro-1

$$u^{k+1} \approx \left(u^k + dt\,T^{k+0.5}\right)(1 - C_b dt/h)$$
assuming $C_b dt/h$ is << 1

$$u^{k+1} \approx u^k + dt\,(1 - C_b dt/h)\,T^{k+0.5} - C_b dt/h\,u^0$$
linearizing around initial barotropic state.

These modifications enable to recover a time integrated stress (over the barotropic integration window) that matches what is felt by the 3D dynamics.

Proposed solution: global flowchart

1- Guess « after » baroclinic velocities including vertical diffusion: $u'^{\,n+1}\ \ B = (I - A)^{-1}\begin{pmatrix}1\\1\\\dots\\1\\1\end{pmatrix}$
Compute and store matrices B (3d) and $\bar{B}$ (2d)

2- Compute barotropic forcing and remove its contribution from 3d velocities:

$$F^n = \frac{\bar{u}'^{n+1} - \bar{u}^{n-1}}{2\tau\bar{B}} \qquad u^{*\,n+1} = u'^{\,n+1} - 2\tau F^n \mathrm{B}$$

3- Remove explicitly computed forcing during barotropic iterations

$$F'^n = F^n - F$$

4-Substep barotropic equations from baroclinic steps n-1 to n+1:

```
DO k=1, nn_baro
```

$$\bar{u}^{k+1} = \bar{u}^k + dt\,\{F'^n + F^k\}\,\bar{B}$$

```
END DO
```

5-Correct 3d velocities:

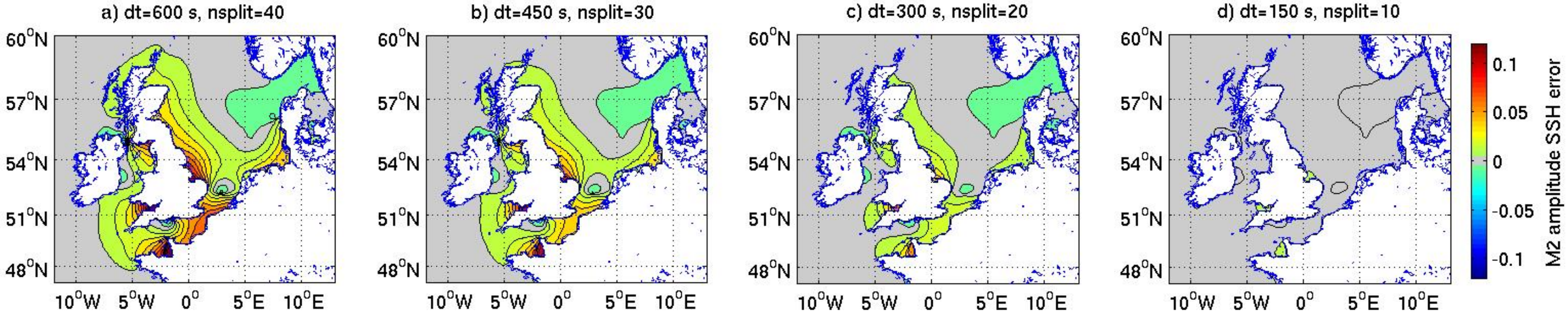
$$u^{n+1} = u^{*n+1} + (\bar{u}^{n+1} - \bar{u}^{n-1})\,\frac{B}{\bar{B}}$$

One can easily show that $\int_0^h u^{*n+1}\,dz = \bar{u}^{n-1}$ hence $\int_0^h u^{n+1}\,dz = \bar{u}^{n+1}$

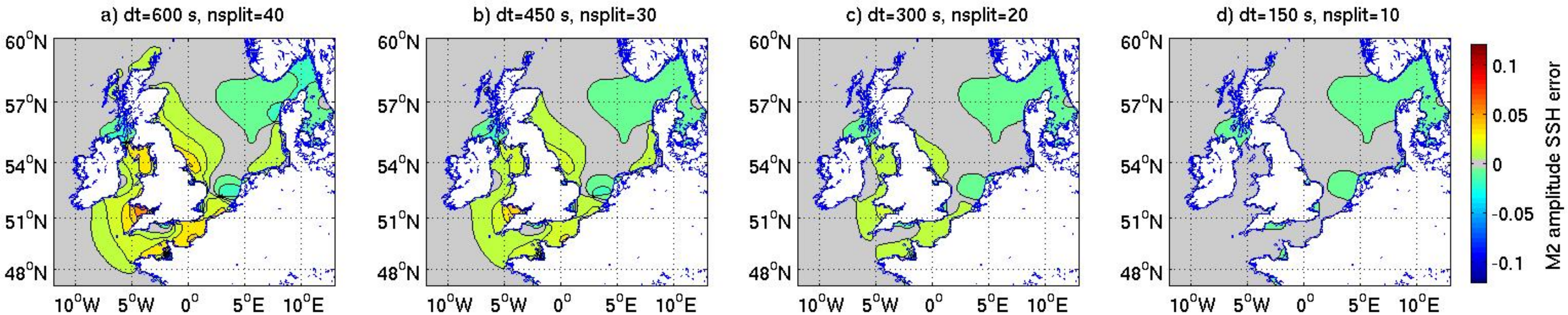
Illustration: AMM12 (North East Atlantic)

- Tides only, no stratification, k-epsilon vertical mixing
- M2 amplitude vs unsplit solution
- Test convergence at different baroclinic time step
- Investigate other parameters Asselin filter, barotropic mode filtering

M2 SSH amplitude error vs unsplit solution



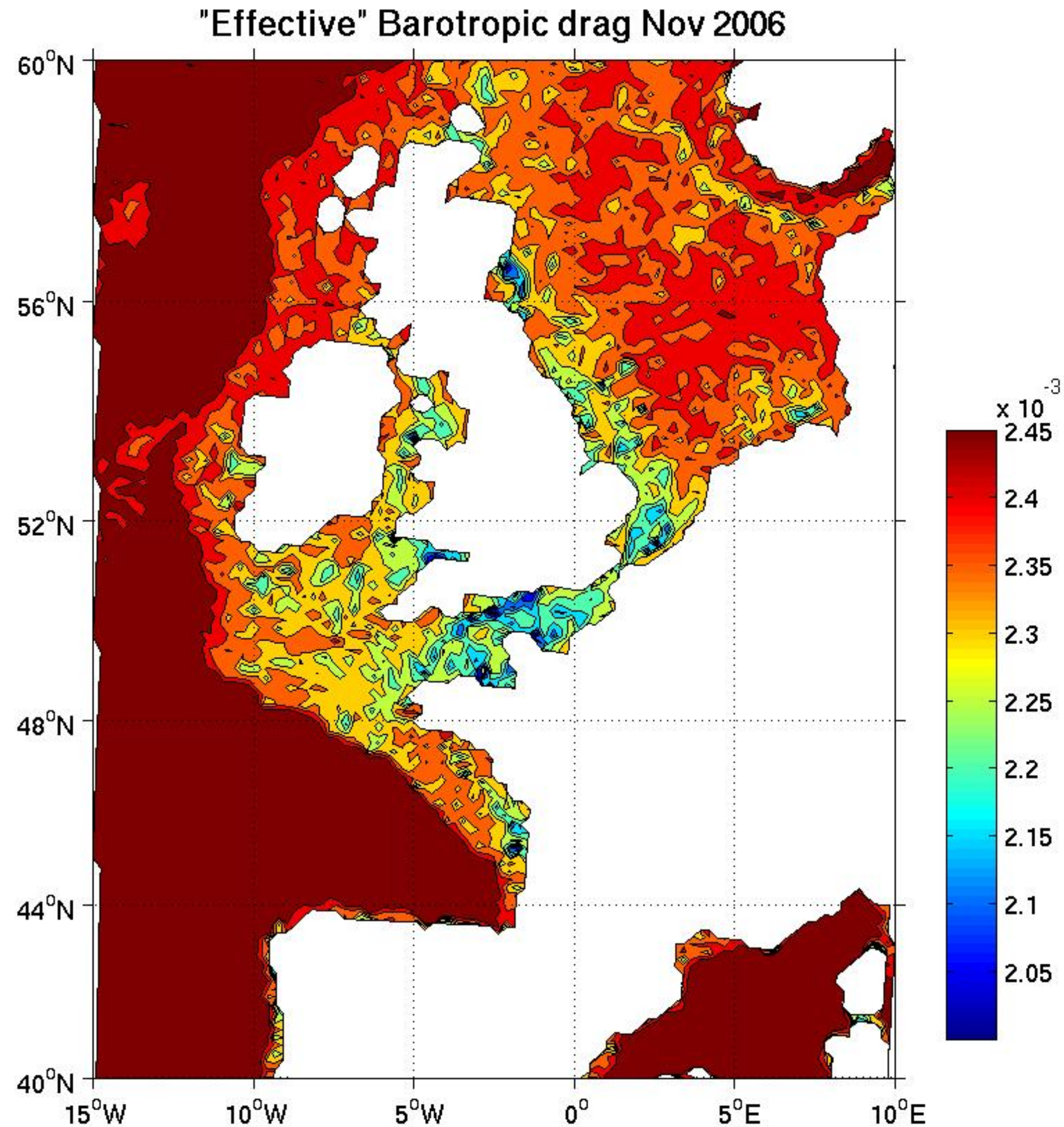
1) Correct conflict between time splitting and implicit bottom friction



600 s

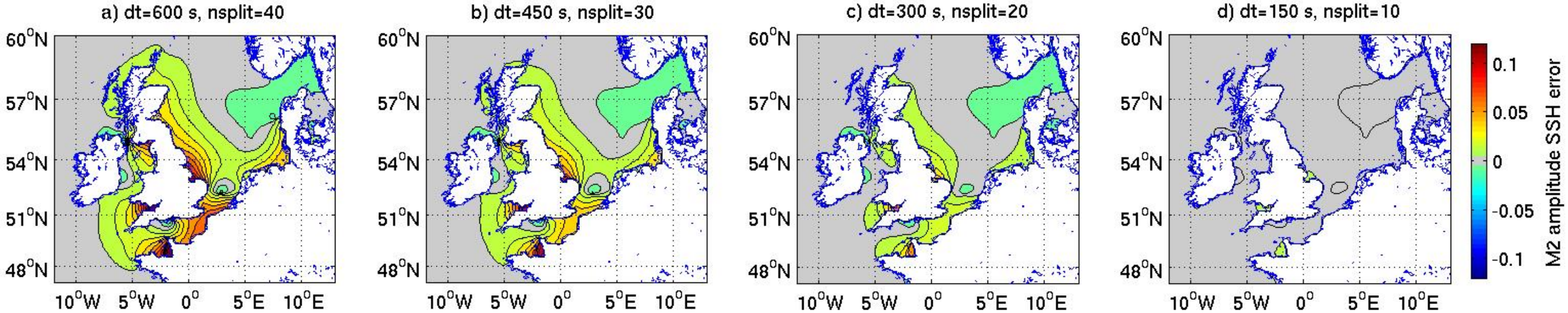
Baroclinic time step

150 s

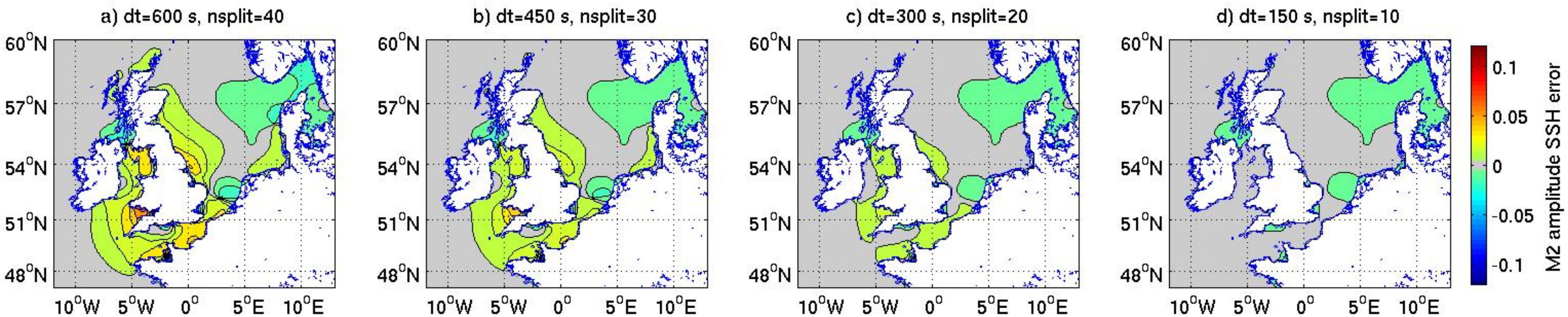


User defined bottom drag = 2.5 10^{-3}

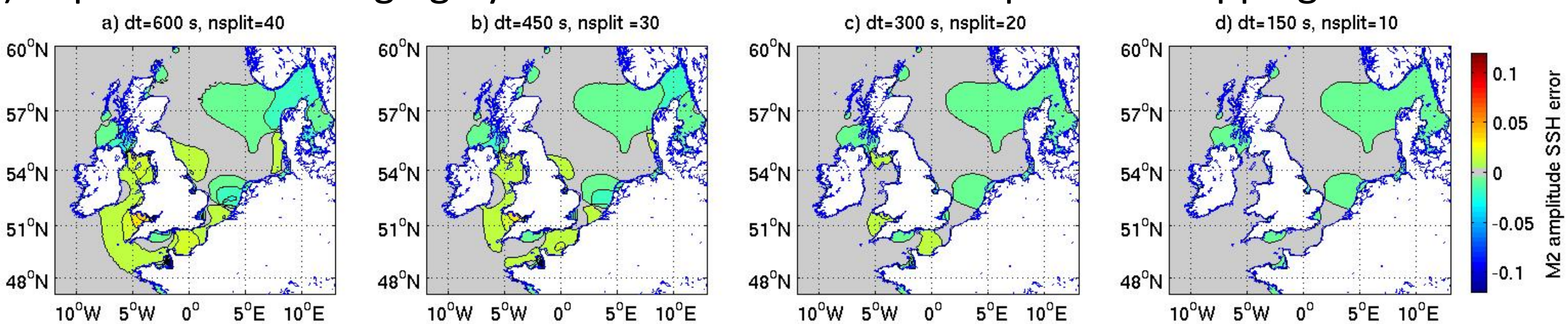
M2 SSH amplitude error vs unsplit solution



1) Correct conflict between time splitting and implicit bottom friction

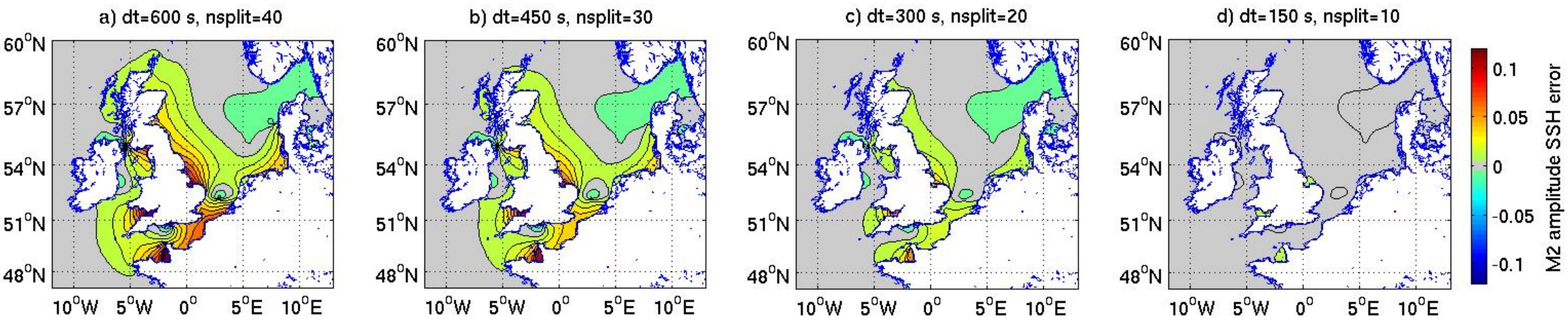


2) Replace time averaging by ad-hoc diffusion in barotropic time stepping

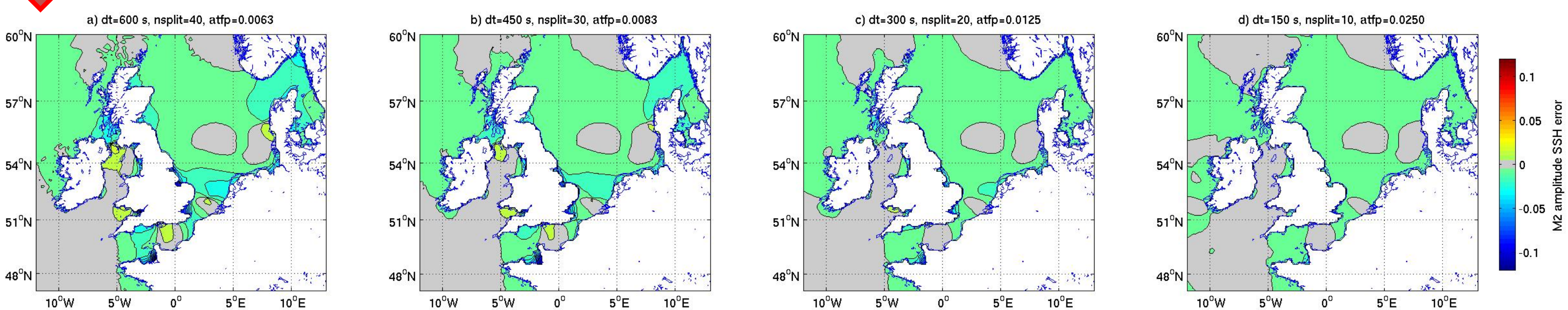


600 s Baroclinic time step 150 s

M2 SSH amplitude error vs unsplit solution

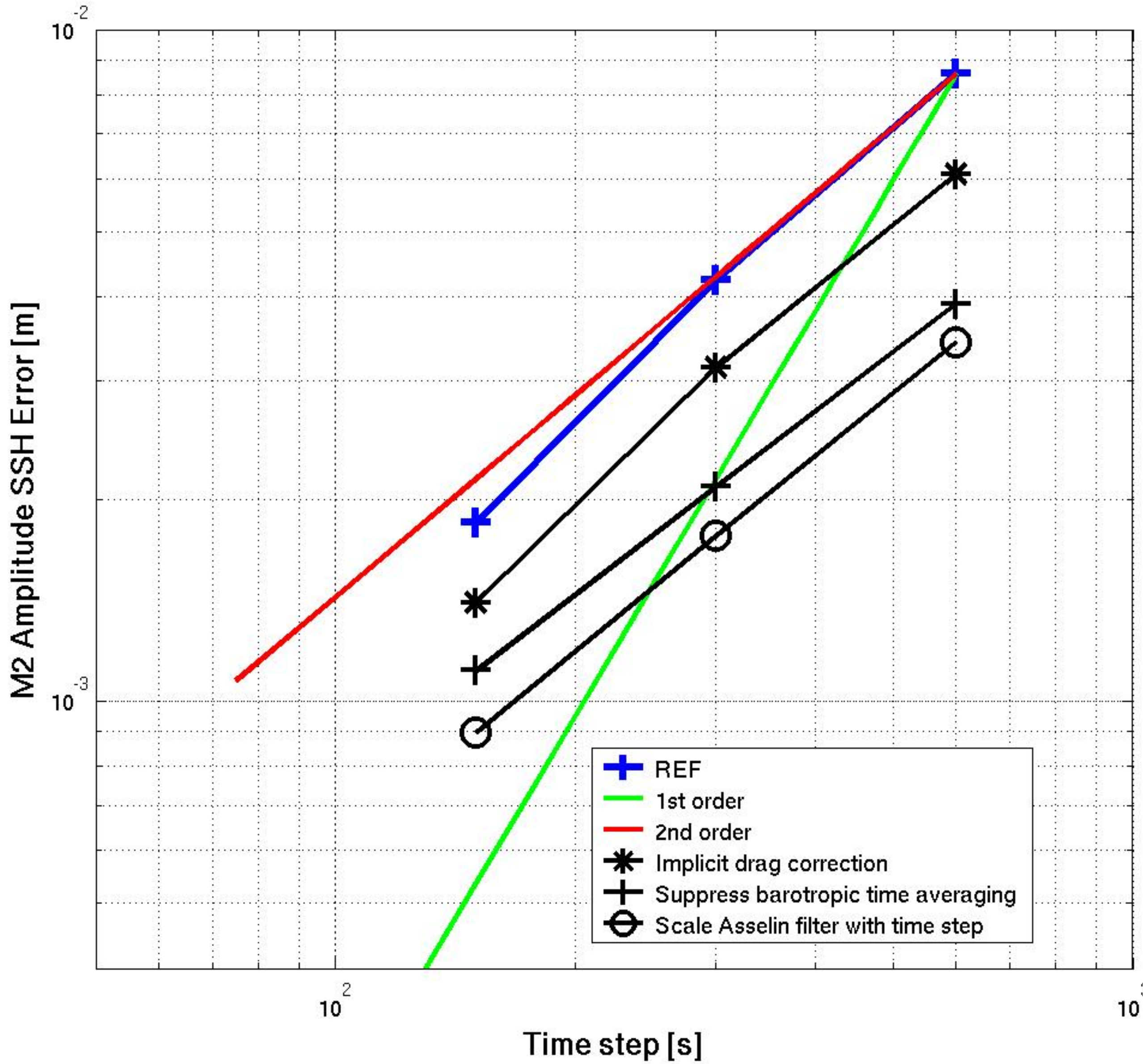


All corrections + scale Asselin coefficient so that numerical diffusion remains constant



600 s Baroclinic time step 150 s

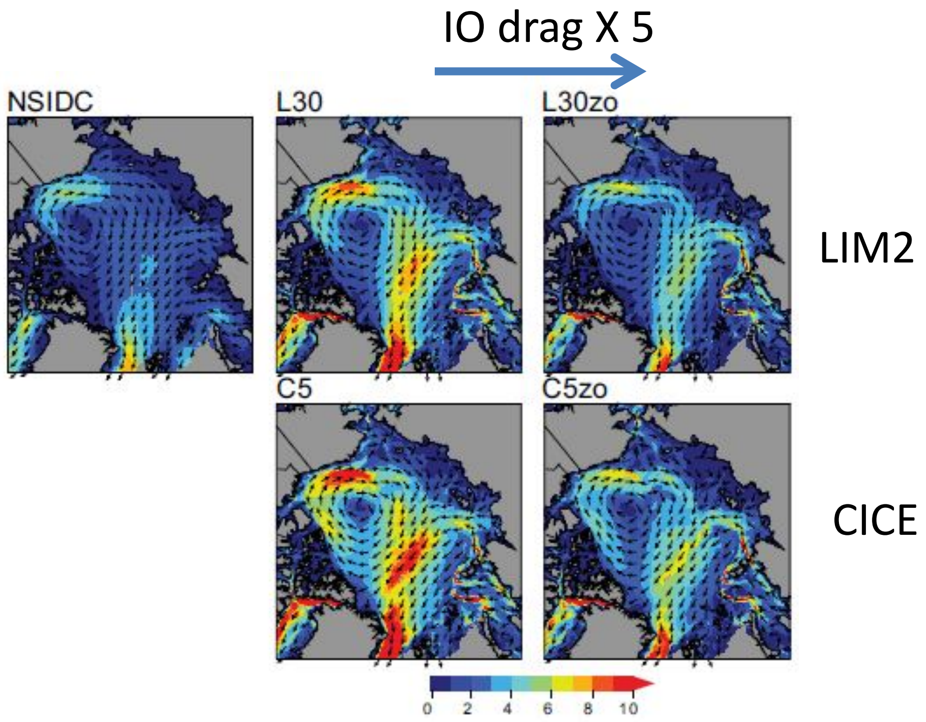
=> Solution barely depends on time step !



=> From 30 % to 50 % error reduction

What about sea-ice drag ?

- Surface vertical resolution is now around 1m (diurnal cycle resolving).
- Moving to the explicit resolution of shallow ekman layers below ice has implied to revise commonly used drag values (to basically sticks with log law).
- Explicit dynamical coupling => Drastic reduction of time steps at coarse resolution and even more with suppression of TKE input below ice (eg no wave breaking).

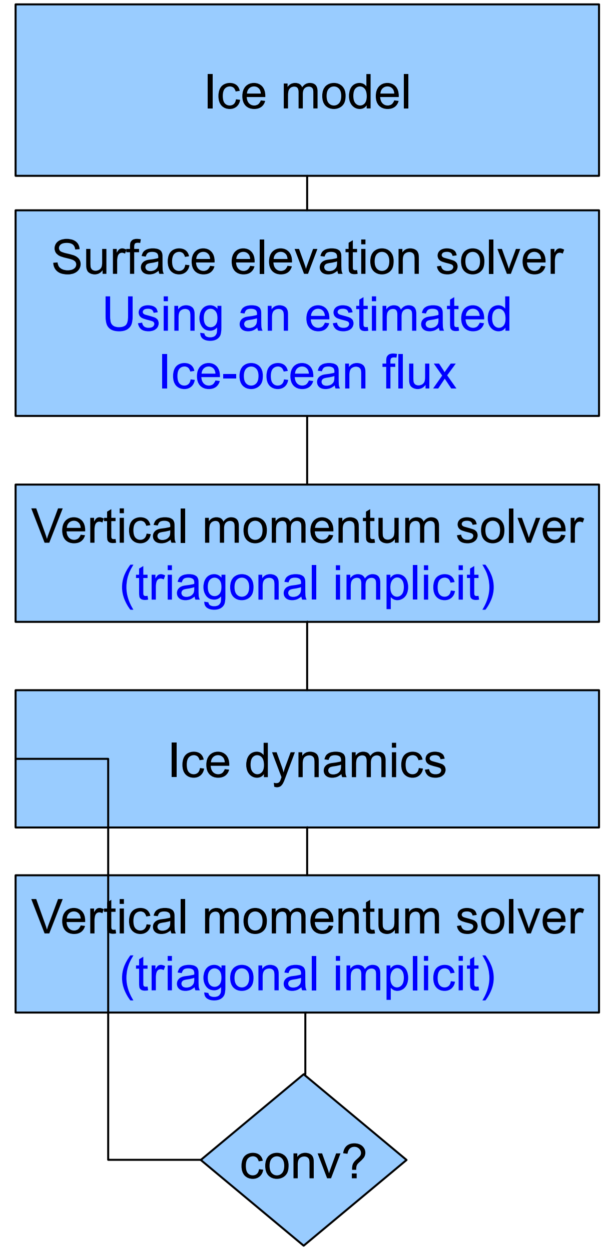


Roy et al. JGR 2015

Fred Dupont's (Env. Canada) slide a couple of years ago

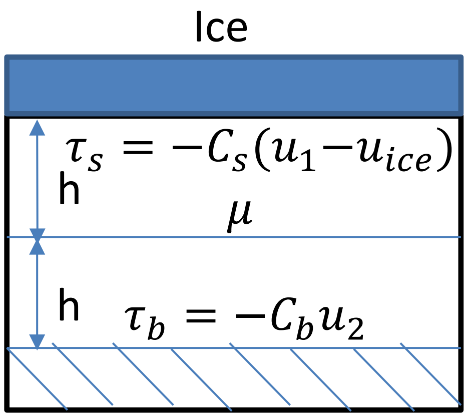
W… doing it?

- …r case, the ice-ocean drag …oefficient was getting large enough (due to the thin surface layer and the logarithmic profile) that some numerical instabilities were resulting.
- 2 solutions:
  - reduce the time-step
  - move to an implicit solution in the ocean of the ice-ocean drag (previously explicit). But then, the resulting momentum is inconsistent with what the ice saw → need to iterate...



Implicit drag with barotropic time splitting and prognostic ice

Same philosophy as for barotropic sub-stepping except that we add one dimension to the tridiagonal system (jpk+1).



Similarly define the projection of ice dynamics tendency on baroclinic velocities:

$$\Delta U = \delta T(u_{ice}) \; (I - A')^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} = \delta T(u_{ice}) \, B_{ice}$$

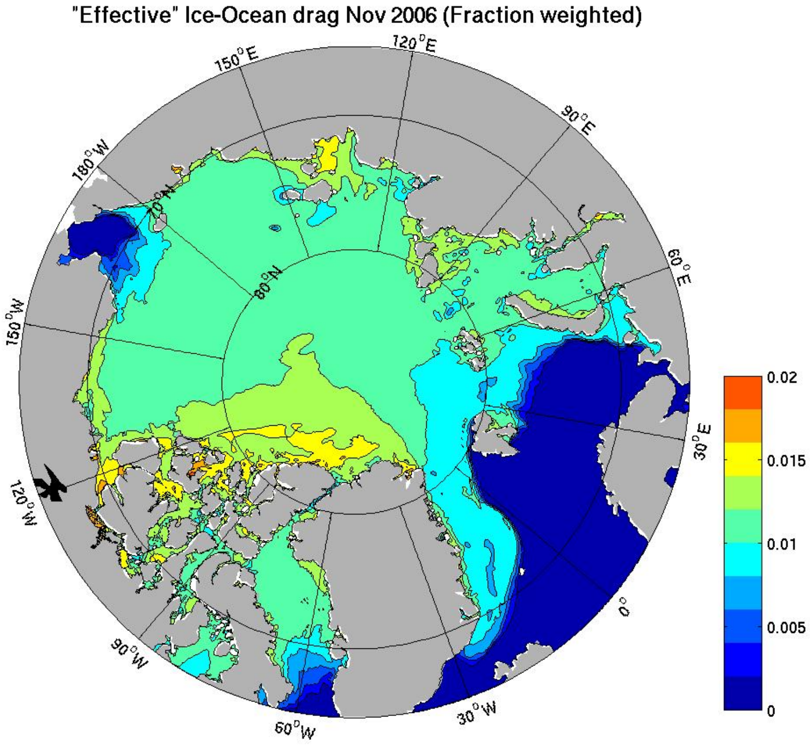
Now consider both barotropic and ice substepping at the same time

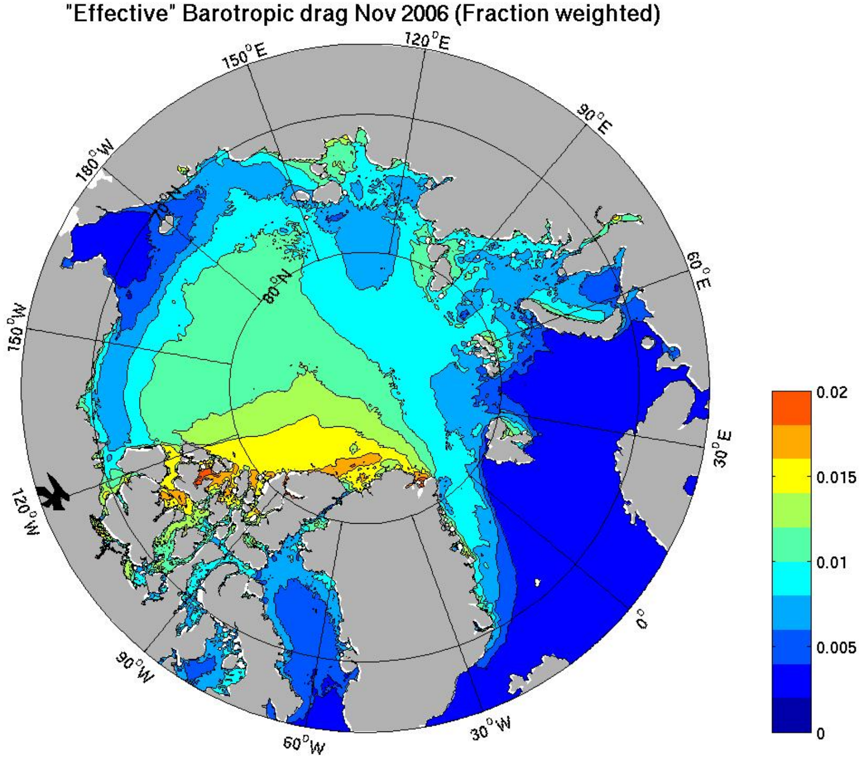
$$\Delta U = \delta T(u_{ice}) (I - A')^{-1} \begin{pmatrix} 1 \\ 0 \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix} + \delta T(\bar{u}) (I - A')^{-1} \begin{pmatrix} 0 \\ 1 \\ 1 \\ \dots \\ 1 \\ 1 \end{pmatrix} = \delta T(u_{ice}) \, B_{ice} + \delta T(\bar{u}) \, B_{bar}$$

$$\partial_t u_{ice} = \delta T(u_{ice}) B_{ice}(0) + \delta T(\bar{u}) B_{bar}(0)$$

$$\partial_t \bar{u} = \delta T(\bar{u}) \, \overline{B_{bar}} + \delta T(u_{ice}) \overline{B_{ice}}$$

« Effective » drag diagnostic from ice ocean system: CREG4+LIM3, dt=1080s, ice-ocean drag C_{io}=1.e-2





- Diagnostic of « effective » drags strongly varies in time and space depending on column viscosities, ice thickness,…
- These are significantly lower so that drag is <u>likely to be overestimated</u> if implicitly refreshed during substepping (barotropic and or ice).
- « Effective » implicit drag should be transferred between sub-components

Summary

- Proposed scheme removes unconsistency between the drag seen by 3d dynamics and sub-components (barotropic or rheology solver).

- Simple to implement (need the storage of two additionnal 3d arrays).

- We still need to figure out how to deal with non-linear free surface (Matrix depends in that case on predicted ssh).

- Implementation with sea-ice would de facto couple barotropic and rheology solvers:
  - Breaks code modular structure
  - Rheology solver certainly exhibits « unphysical » signals that should not be spread in the barotropic dynamics.