Croco - Online Analysis Interface

B. Lemieux-Dudon, F. Auclair, C. Nguyen, et al.

2020

Summary

1. Introduction

2. Fonction test pour verifier l'analyse en ondelette de Morlet
   - Verification de l'analyse avec double calcul python
   - Verification orthogonalité numérique des ondelettes de Morlet S2+S4

3. Conclusion

Introduction

This report participate to the objective of implementing an efficient interface between the Croco model and the Online Analysis module. The Online Analysis module enables to perform online Wavelet and spectral analysis targetting specific ocean model fields. A preliminary version of this interface was set up in Spring/Summer 2020. This version was pushed to the GitLab INRIA repository (branch dev_2020_online_analysis). A part of the Online Analysis (OA) are now operational online but not the full OA options have been yet tested. This report presents the tests that were conducted to validate the developments :
A preliminary version has been set up with two test cases involving the Morlet wavelet :

- a test case a uniform field varying as the linear combination of two cosine functions with pulsations ω_{S_2} and ω_{S_4}, respectively. In this test case, we set the two harmonics to match the two pseudo tidal waves S2 (12h) and S4 (6h) and we set the amplitude to 5 and 3, respectively :

$$u(i,j,k,t) = 5 * \cos \omega_{S_2}\, t + 3.\cos \omega_{S_4}\, t$$

- the academic IGW test case which is a 2Dxz academic configuration with a fidal forcing (S2) applied at the Western boundary and the continental slope at the Eastern boundary.

2. Fonction test pour verifier l’analyse en ondelette de Morlet
   - Verification de l’analyse avec double calcul python
   - Verification orthogonalité numérique des ondelettes de Morlet S2+S4

Test fonction $u = 5\cos(\omega_{S2} t) + 3\, cos(\omega_{S4} t)$ - S2 Wavelet analysis

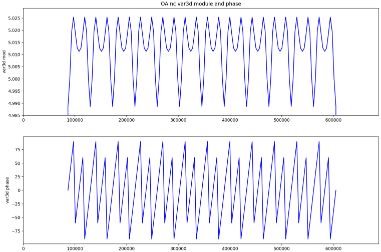

Figure: S2 Wavelet analysis every 3600s.

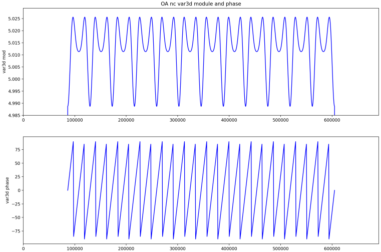

Figure: S2 Wavelet analysis every 600s.

Test fonction $u = 5\cos(\omega_{S2}t) + 3\,cos(\omega_{S4}t)$ - S2 Wavelet analysis

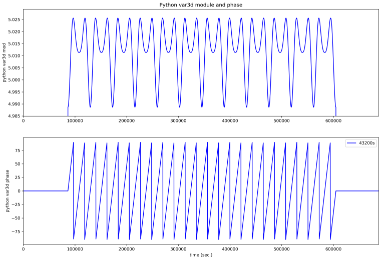

Figure: PYTHON S2 Wavelet analysis every 150s.

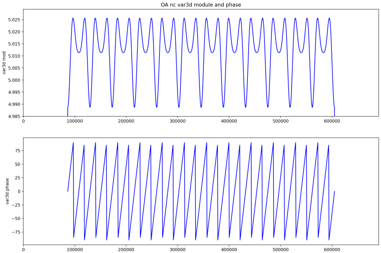

Figure: OA S2 Wavelet analysis every 600s.

Test fonction $u = 5\ \cos(\omega_{S2}t) + 3\,cos(\omega_{S4}t)$ - S4 Wavelet analysis

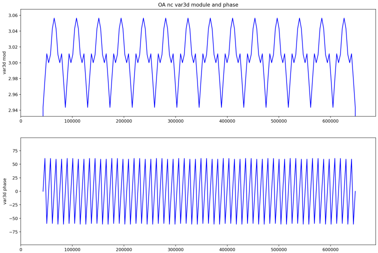

Figure: S4 Wavelet analysis every 3600s.

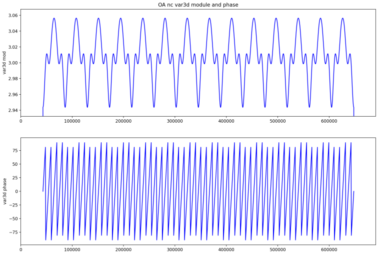

Figure: S4 Wavelet analysis every 600s.

Test fonction $u = 5\cos(\omega_{S2} t) + 3\, cos(\omega_{S4} t)$ - S4 Wavelet analysis

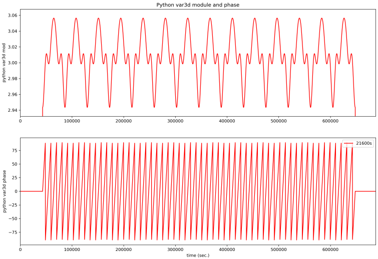

Figure: PYTHON S4 Wavelet analysis every 150s.

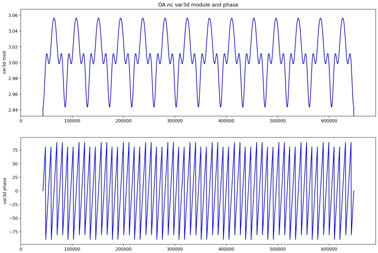

Figure: OA S4 Wavelet analysis every 600s.

2. Fonction test pour verifier l'analyse en ondelette de Morlet
   - Verification de l'analyse avec double calcul python
   - Verification orthogonalité numérique des ondelettes de Morlet S2+S4

Test fonction $u = 5\cos(\omega_{S2}t)$ - S2+S4 Wavelet analysis

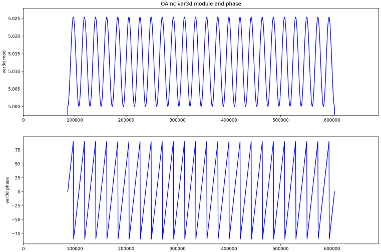

Figure: S2 Wavelet analysis every 600s.

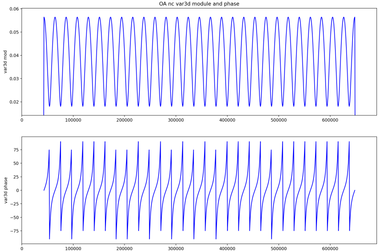

Figure: S4 Wavelet analysis every 600s.

Test fonction $u = 3\,cos(\omega_{S4} t)$ - S2+S4 Wavelet analysis

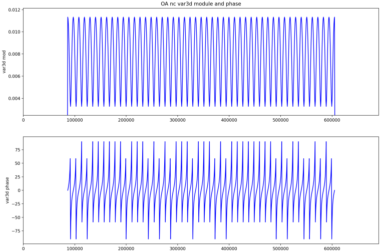

Figure: S2 Wavelet analysis every 600s.

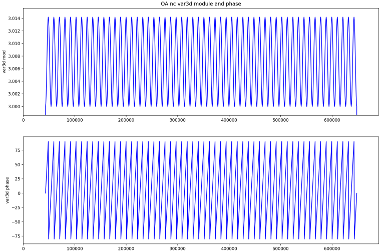

Figure: S4 Wavelet analysis every 600s.

Conclusion

This preliminary version of the Online Analysis - Croco interface provided appropriate results when applied to the two Test Cases i) ad hoc uniform field varyrying as a cosine function, ii) IGW academic test case. The Online Analysis module enables to perform online Wavelet and spectral analysis targetting specific ocean model fields. So far we have only tested the Wavelet transform with the Morlet atom. To compile the Croco code and OA module we used the intel compilers. To output the simulation fields and the Online Analysed fields we used XIOS2 executable with the properly parametrized xml files.
Currently :

- The interface is first based on calls to the subroutine croco_oa within the main.F and step.F Croco subroutines (calls are performed at the initialization step - 3D now - and at subsequent 3D time steps). The croco_oa.F90 routine includes several Croco header files (grid.h, ocean2d/3d.h,...) so that the horizontal domain and vertical grid variables and parameters (lon-lat geographical or curvilinear coordinates, ocean thickness, sub-domain indices,...) can be passed as arguments to the Stand-alone Online Analysis subroutines init_oa and main_oa.
- Within the Stand Alone module, the calculation of the correlation product between the psi_oa function (Fourier/Wavelet) and the requested ocean field value (cf namelist_oa and requested variable-configuration-analysis) is performed thanks to the var_oa external real function which returns (possibly at each time step) the value of the Croco ocean field value at a given domain location. Indeed, the var_oa external real function also includes several Croco header files (grid.h, ocean2d/3d.h,...) since most of the Croco ocean fields are not passed as arguments to the Stand-alone OA code.
- Finally, the Online Analysis data is output by the mean of the XIOS2 facility thanks to few changes implmented in the send send_xios_diags.F routine (eg, use

Perspectives

The preliminary version of the Online Analysis - Croco interface must be improved to solve several issues.

- the current Croco - Online Analysis Interface is not efficient in terms of memory and should be modofied to conform with some Croco pre-requisite (hybrid OpenMPI - OpenMP paralelization, shared header files, the way to handle precision with source pre-processing) :
- the objective is to replace croco_oa by a "loop on tile" construct specific to Croco with dedicated modules for the Croco to OA interface and OA to Croco- XIOS2 inteface. This will solve the duplication of the Croco arrays in memory with the current interface (Croco arrays are currently passed as arguments to the Online Analysis routines initi_oa and main_oa with reduced dimensions).
- A check regarding the treatment of the precision of complex numbers (intel/gnu compilation flag -r8, croco mpc.F preproc applied to F90 sources, F90 fixed format sources or not)
- the ccp keys allowing the use of mask hasn't been tested yet, neither the curvilinear coordinate.
- Tests must be conducted with GNU compilers (at this stage only only Intel compilers were tested).