

CROCO-NEXT

R. Benshila, S. Le Gac, L. Debreu, F. Lemarié, N. Ducousso, M. Brémond
+ M. Caillaud, S. Masson, S. Valat, M. Schreiber

02-07-2024

Plan

- en bref
- pourquoi ?
- bilan prospectif
- bilan pratique
- et la suite ?

En bref

CROCO-NEXT

KEZAKO ?

- Juillet 2023 : proposition d'une réécriture **raisonnée** de CROCO
- Groupe de travail : Rachid, Solène, Laurent D., Florian, Nicolas D., Maurice
- Intervenants extérieurs : S. Masson, M. Schreiber, S. Valat
- Réunions mensuelles (bimestrielles), compte-rendus sur myboxinria
- Une semaine de codage en Mai 2024

=> définir, explorer pour proposer

Pourquoi ?

Motivations

Mais, mais, mais pourquoi ?

- Éléments discutables du code actuel
 - allocation statique
 - clés cpp : maintenance, couverture de code
 - pas de norme (homogénéité, pratique)
 - pas de doc en ligne
 - **pas modulaire** (interfaces, tests unitaires)
 - duplication de code
 - prise en main (du code lui-même)
 - développeurs (scientifiques, releases)
 - *compilation*
 - *IO*
 - Préparer le futur :
 - évolution algorithmique
 - tests (unitaires)
 - modules extérieurs
 - nouvelles architectures
 - nouvelles fonctionnalités des langages (ex Co-array Fortran)
 - hybridation
- + motivation des développeurs, faisabilité de la maintenance

Wouai?



```
(base) polumekhanos:0CEAN rblod$ grep t *h *F *90 lwc  
83361  
(base) polumekhanos:0CEAN rblod$
```



Initiatives similaires

HPC, « separation of concern », IA (interface+différentiabilité)

- MITgcm (Frontend Julia)

- Oceanigans (Julia, MIT JPL)

<= nouveau coeur numérique

<https://github.com/CliMA/Oceananigans.jl>

- Veros (Jax, Hamburg U.)

<https://veros.readthedocs.io/en/latest/>

- ❖ Tolosa (Fortran 2008, Shom)

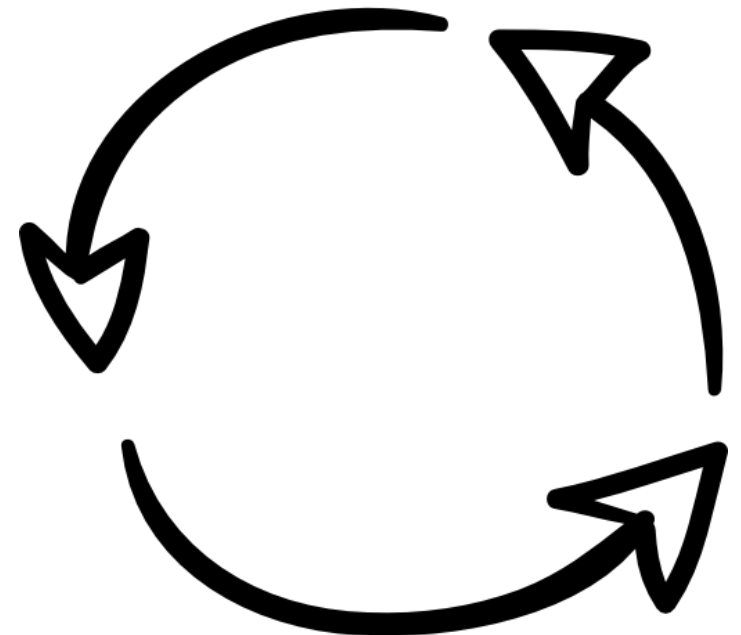
<= nouveau coeur numérique

- ♣ Discussions au niveau français (R. Fablet Pytorch)

- ✓ DSL : Psyclone, POSEIDON (HPC INRIA M. Schreiber)

Réécriture ?

- **Pas** une traduction (.F => .F90, ex ROMS-RUTGERS)
mais pas de changement du coeur de calcul
- Review des codes existants
- Spécifications
- Prototype mini-app
- Présentation, discussion (extérieurs inclus)
- Code complet



Bilan prospectif

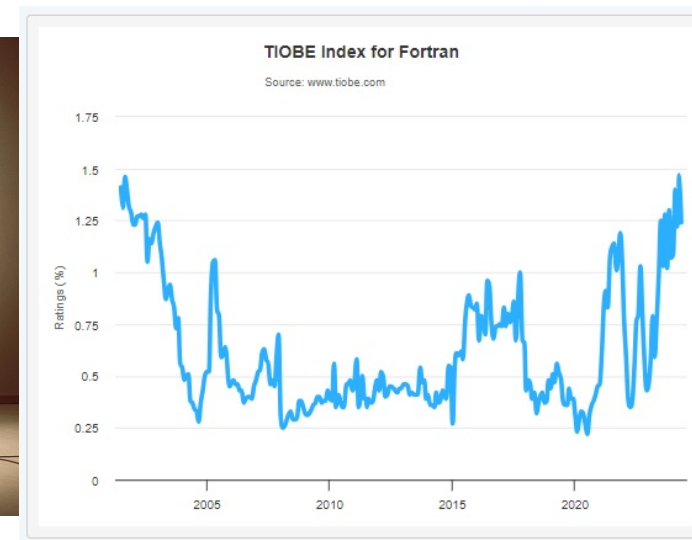
Spécifications

- Première partie de l'année :
- Review code extérieurs
- Définitions :
 - Fonctionnel : environnement, interfaçage, IO, tests etc
 - Coeur de calcul lui-même: langage, organisation

Fonctionnel

- tests : unitaires, **physiques, perfs**, efficacité énergétique
- timing simple : **timing NEMO**
- documentation incore : **Ford**, Doxygen
- interfaçage IA
- différentiabilité : **Tapenade**
- Agrif
- Psyclone

Coeur de calcul



- langage : **Fortran 90** (vs Julia, Tensorflow etc) : savoir-faire vs âge des développeurs, perfs, compatibilité, Agrif, born-again Fortran (indice TIOBE)
- parallélisation : MPI (3), GPU, OpenMP (?), co-array(?)
- allocation statique ou **dynamique**
- clés cpp : intérêt ? => namelist
- évolutivité algorithmique: **schéma en temps**, coordonnée verticale
- **modularité**

Point de départ

Mini-app

- CROCO light => BASIN => **BASIN2D** :
 - reproduire les résultats
 - sans perte de perf
 - focus sur la modularité
- code 2DV Nicolas

https://gitlab.inria.fr/croco-next/croco_mini

https://gitlab.inria.fr/croco-next/croco_nico

Bilan pratique


(4 jours de dev)

Fonctionnel

- squelette (boucle temporelle + IO simples)
- namelist
- IO simples
- organisation
- compilation simple
- tests CI
- doc
- timing
- Licence...

CROCO-MINI Documentation Source Files Modules Procedures Derived Types Namelists Programs

Search



CROCO
Coastal and Regional Ocean COmmunity model

CROCO-MINI model, from.F to .F90

CROCO-MINI














This is CROCO-MINI project, because we must start somewhere...










Source Files	Modules	Procedures	Derived Types
<ul style="list-style-type: none">ad.F90adv2d.F90agrif.F90bdy2d.F90bottom_drag.F90constants.F90constants_gen_fb.F90constants_phys.F90core.F90core_wrap.F90All source files...	<ul style="list-style-type: none">adadv2dagrifbdy2dbottom_dragconstantsconstants_gen_fbconstants_physcorecore_wrapAll modules...	<ul style="list-style-type: none">ad_set_parametersad_stepadd_xxxadv2d_flux_c2adv2d_flux_up3adv2d_read_paramAgrif_CFixedAgrif_FixedAgrif_Rootallocate_croco_varAll procedures...	<ul style="list-style-type: none">croco_varcroco_vartimertimerAll derived types...

CROCO-MINI © 2024

Documentation generated by [FORD](#)

https://croco-next.gitlabpages.inria.fr/croco_mini

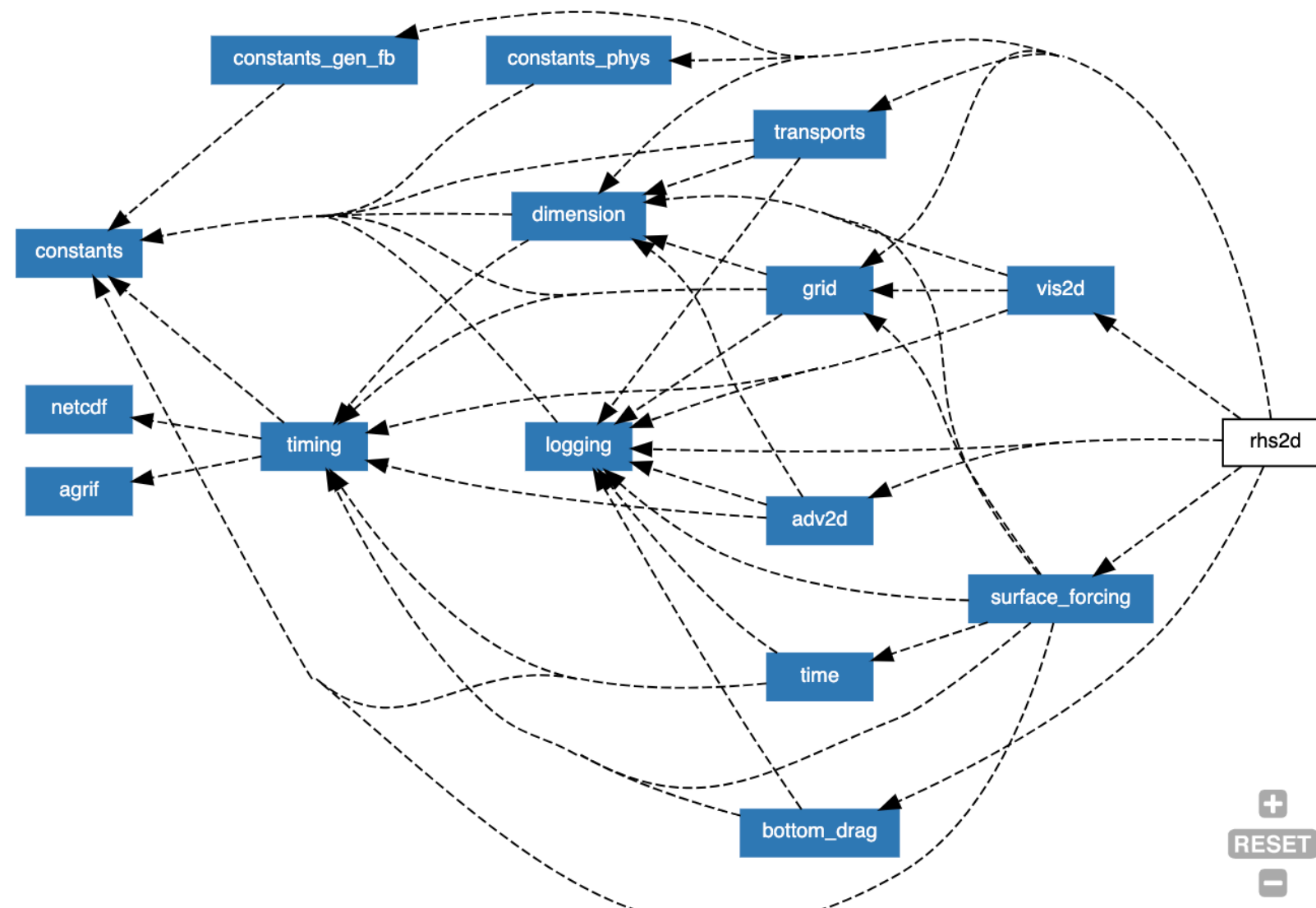
Nom	Dernière validation	Dernière mise à jour
 compilers	Update Makedefs.gfortran	il y a un mois
 doc	exclude build directory : temporary fix, build should be...	il y a 4 mois
 misc	Adding LICENCE.md AUTHORS.md and heading in file	il y a un mois
 src	Merge branch 'toF90_2D_notype' into toF90_2D_core	il y a un mois
 test	add default value in adv2d namelist	il y a un mois
 .croco_ascii.txt	To f90 compile	il y a 3 mois
 .gitignore	up	il y a un mois
 .gitlab-ci.yml	keep artefact for timing	il y a un mois
 AUTHORS.md	Adding LICENCE.md AUTHORS.md and heading in file	il y a un mois
 CHANGELOG.md	empty croco	il y a 5 mois
 LICENCE.md	Adding LICENCE.md AUTHORS.md and heading in file	il y a un mois
 Makefile	prettify	il y a un mois
 README.md	Adding LICENCE.md AUTHORS.md and heading in file	il y a un mois

Nom	Dernière validation
..	
 core	fprettify + capitalize keyword + output modify def_var with ids in module vars
 domain	fprettify
 forcing	fprettify
 init	fix error msg handling
 main	Merge branch 'toF90_2D_ND' into toF90_2D_core
 output	fprettify + capitalize keyword + output modify def_var with ids in module vars
 tools	fix error msg handling
 type	fprettify + capitalize keyword + output modify def_var with ids in module vars
 Makefile	prettify

Core 2D

Uses

[vis2d](#) [constants_gen_fb](#) [surface_forcing](#) [dimension](#) [transports](#) [logging](#) [adv2d](#) [bottom_drag](#) [grid](#) [constants_phys](#)



Timing

Flat profile:

Each sample counts as 0.01 seconds.

% time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name
96.24	391.96	391.96	4216340	0.09	0.09	step2d_fb_tile_
3.40	405.82	13.86	4216340	0.00	0.00	ana_smflux_tile_
0.12	406.32	0.51	4216340	0.00	0.00	u2dbc_tile_
0.12	406.81	0.49	4216340	0.00	0.00	v2dbc_tile_
0.09	407.17	0.36	4216340	0.00	0.00	zetabc_tile_
0.01	407.21	0.04	4216340	0.00	0.00	set_vbc_
0.01	407.24	0.03				MAIN__
0.00	407.26	0.02	21081700	0.00	0.00	omp_get_thread_num_
0.00	407.28	0.02	4216340	0.00	0.09	step2d_
0.00	407.30	0.02	40	0.50	0.50	get_vbc_
0.00	407.32	0.02	20	1.00	1.00	ana_initial_
0.00	407.32	0.00	16865380	0.00	0.00	omp_get_num_threads_
0.00	407.32	0.00	4216340	0.00	0.10	step2d_thread_
0.00	407.32	0.00	4216340	0.00	0.10	step_
0.00	407.32	0.00	320	0.00	0.00	lenstr_
0.00	407.32	0.00	200	0.00	0.00	cancel_kwd_
0.00	407.32	0.00	20	0.00	0.00	ana_grid_
0.00	407.32	0.00	20	0.00	0.00	check_kwds_
0.00	407.32	0.00	20	0.00	0.00	check_srcs_
0.00	407.32	0.00	20	0.00	0.00	check_switches1_
0.00	407.32	0.00	20	0.00	0.00	check_switches2_
0.00	407.32	0.00	20	0.00	0.00	closecdf_
0.00	407.32	0.00	20	0.00	0.00	get_date_
0.00	407.32	0.00	20	0.00	0.00	init_arrays_
0.00	407.32	0.00	20	0.00	0.00	init_scalars_
0.00	407.32	0.00	20	0.00	0.00	read_inp_
0.00	407.32	0.00	20	0.00	0.00	setup_grid1_
0.00	407.32	0.00	20	0.00	0.00	setup_grid2_
0.00	407.32	0.00	20	0.00	0.00	setup_grid2_tile_
0.00	407.32	0.00	20	0.00	0.00	setup_kwds_
0.00	407.32	0.00	20	0.00	0.00	start_timers_
0.00	407.32	0.00	20	0.00	0.00	stop_timers_

Flat profile:

Each sample counts as 0.01 seconds.

%	cumulative	self		self	total	
time	seconds	seconds	calls	ms/call	ms/call	name
29.42	113.88	113.88	4216340	0.03	0.06	__rhs2d_MOD_rhs_hu2d
21.40	196.74	82.86	4216340	0.02	0.02	__adv2d_MOD_adv2d_flux_up3
16.43	260.33	63.60	4216340	0.02	0.02	__vis2d_MOD_visc2d_flux
8.10	291.69	31.35	4216340	0.01	0.09	__core_MOD_core_step2d
5.30	312.19	20.50	12649020	0.00	0.00	__rhs2d_MOD_extrap_ab3
3.38	325.26	13.07	4216360	0.00	0.00	__surface_forcing_MOD_surface_forcing
3.35	338.21	12.95	8432680	0.00	0.00	__transports_MOD_transport_v
3.28	350.91	12.70	8432680	0.00	0.00	__transports_MOD_transport_u
2.52	360.67	9.76	4216340	0.00	0.00	__rhs2d_MOD_rhs_zeta
2.42	370.03	9.36	4216340	0.00	0.00	__rhs2d_MOD_interp_am4
1.84	377.14	7.11	4216340	0.00	0.00	__transports_MOD_transport_to_vel_u
1.72	383.80	6.66	4216340	0.00	0.00	__transports_MOD_transport_to_vel_v
0.24	384.74	0.94	33731200	0.00	0.00	__timing_MOD_timing_start
0.19	385.46	0.72	4216340	0.00	0.00	__bdy2d_MOD_set_zeta_bc
0.17	386.11	0.65	33731200	0.00	0.00	__timing_MOD_timing_stop
0.14	386.66	0.55	4216340	0.00	0.00	__bdy2d_MOD_set_u2d_bc
0.09	387.00	0.34	4216340	0.00	0.00	__bdy2d_MOD_set_v2d_bc
0.02	387.08	0.08	4216340	0.00	0.09	__core_MOD_core_step
0.01	387.10	0.02	67462420	0.00	0.00	__agrif_MOD_agrif_root
0.01	387.12	0.02	20	1.00	1.00	__bottom_drag_MOD_bottom_drag_read_pa
0.00	387.13	0.01	20	0.50	0.50	__core_MOD_core_exit
0.00	387.14	0.01	20	0.50	0.50	__surface_forcing_MOD_surface_forcing
0.00	387.15	0.01	20	0.50	0.50	__vis2d_MOD_vis2d_read_param
0.00	387.16	0.01				__agrif_MOD_agrif_cfixed
0.00	387.16	0.00	600	0.00	0.00	__timing_MOD___deallocate_timing_Time
0.00	387.16	0.00	80	0.00	0.00	__logging_MOD_log_char
0.00	387.16	0.00	40	0.00	0.00	__time_MOD_dat2sec
0.00	387.16	0.00	20	0.00	0.00	__adv2d_MOD_adv2d_read_param
0.00	387.16	0.00	20	0.00	0.00	__bdy2d_MOD_bdy2d_read_param
0.00	387.16	0.00	20	0.00	0.00	__dimension_MOD_dimension_read_param
0.00	387.16	0.00	20	0.00	0.00	__grid_MOD_grid_allocate
0.00	387.16	0.00	20	0.00	0.00	__grid_MOD_grid_read_param
0.00	387.16	0.00	20	0.00	0.00	__grid_MOD_grid_setup
0.00	387.16	0.00	20	0.00	2.00	__init_MOD_init_main
0.00	387.16	0.00	20	0.00	0.00	__init_MOD_init_read_options
0.00	387.16	0.00	20	0.00	2.00	__init_MOD_init_read_param
0.00	387.16	0.00	20	0.00	0.00	__output_MOD_output_read_param
0.00	387.16	0.00	20	0.00	0.00	__surface_forcing_MOD_surface_forcing
0.00	387.16	0.00	20	0.00	0.00	__time_MOD_time_read_param
0.00	387.16	0.00	20	0.00	0.00	__timing_MOD_timing_open
0.00	387.16	0.00	20	0.00	0.00	__variables_MOD_variables_allocate
0.00	387.16	0.00	20	0.00	0.00	__variables_MOD_variables_deallocate

En test : adjoint

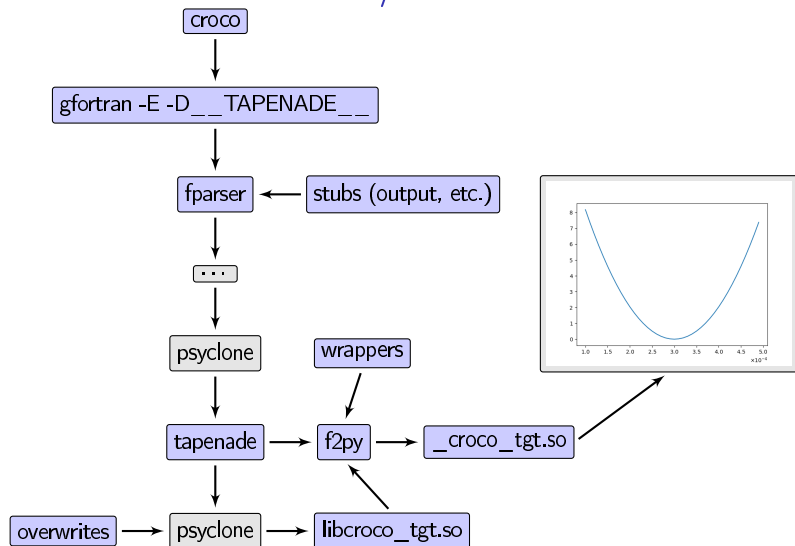
- tapenade
- fonction coût en python
- suite par Maurice

Tangent & Adjoint

- ▶ Tapenade, 2D, 3D, MPI
- ▶ minimisation via m1qn3 et pilotes:
 - ▶ adjoint
 - ▶ tangent
 - ▶ différences finies
- ▶ fortran

croco-next

sources transformations / CMake



croco-next

python interface

```
from croco.init import init_read_options, init_read_param, \
    init_allocate
from croco.grid import grid_setup
from croco.surface_forcing import surface_forcing_update
from croco.output import output_def
from croco.time import nstart, nend, nstep
from croco.timing import timing_start, timing_stop
from croco.core import core_step

init_read_options(sys.argv[1])
[...]
timing_start('full code')
for step in range(nstart, nend):
    timing_start('core_step', nstep, nstart, nend, 1, 1000)
    core_step()
    timing_stop('core_step', nstep)
timing_stop('full code')
```

croco-next

tapenade \rightarrow *f2py*

```
[...]  
from croco.ad_diff import ad_set_parameters  
from croco.cost_diff import cost_fun_d  
[...]
```

python minimizer

```
ad_set_parameters(1, 10, 1)  
def f(x):  
    return cost_fun_d(x, 1)[0] * 1e18  
def jacf(x):  
    return cost_fun_d(x, 1)[1] * 1e18  
  
from scipy.optimize import minimize  
res = minimize(f, [1], method='BFGS',  
               jac=jacf)
```


En test : Laurent D.

- parallélisation (co-array)
- interface IO (netcdf, ADIOS)
- Fortran avancé : types structurés

Entrées / Sorties

Objectifs :

1. Avoir une gestion simple des IO au sein du code CROCO
2. Avoir une interface unique pour plusieurs logiciels IO/formats d'entrées-sorties
 - a) Fonctionnalités diverses (io, visu/diagnostics en ligne, aggrégation ...)
 - b) Diverses complexités d'utilisation (e.g., mode développeur/mode production)

Entrées / Sorties dans CROCO_NEW

Objectifs 1)

Une interface commune aux outils, indépendante de l'outil cible.

```
CALL IO_DEFINE_DIMENSION('xr',0,LMM+1,IstrR,IendR) ! Global and, optionnaly, local indices
CALL IO_DEFINE_VARIABLE('zeta',['xr','yr'])
CALL IO_WRITE_VARIABLE('zeta',zeta)
```

Objectifs 2)

CROCO_NEW a actuellement deux interfaces (implémentés comme submodules Fortran)

- NetCDF
- ADIOS2

Entrées / Sorties dans CROCO_NEW: développements potentiels et questions

1. Permettre l'utilisation de plusieurs outils au sein de la même simu. (e.g. NetCDF pour les fichiers de sortie, ADIOS2 pour les simulations interactives)
2. Ecriture automatisée du code des sorties via lecture d'un fichier de configuration (e.g. xml)
3. Implémentation d'une sous-couche XIOS3
4. Est-ce envisageable d'effectuer des tests de cette interface d'IO dans le CROCO actuel ?

Parallelisation

1. Nécessité de garder ou non le découpage en sous-domaines ("tiles") ?

Note : ceci peut avoir plus d'intérêt dans le CROCO_NEW qui est en allocation dynamique : possibilité de définir des sous-domaines (et des tableaux de travail) de tailles distinctes

Comment tester ceci dans CROCO_NEW ? Scalabilité forte d'une configuration 2D ?

2. Dans CROCO_NEW, l'indexation des tableaux est globale.

3. Des tests de l'utilisation des co-array Fortran ont été effectués. Simplifie l'implémentation mais pas de gain de performance (potentielles pertes ...)

Prévu d'implémenter des modules MPI en version "Two-sided" et "One-sided" et comparaison des performances.

Ces modules pourraient être testés dans le CROCO actuel.

FYPP Preprocessor

Le CROCO_NEW devrait privilégier la suppression d'un maximum de clef CPP (rendu en partie possible par l'allocation dynamique).

Reste que l'utilisation d'un préprocesseur plus évolué que **cpp** peut être intéressant.
(Ecriture automatique de code)

FYPP est un préprocesseur qui repose sur Python.

Exemples d'utilisation dans CROCO_NEW

- Ecriture de procédures génériques (types / précision) dans les module d'IO
- Déclaration de variables
- Ecriture des boucles de calcul (avec génération des **call check_tab** pour débogage)

Voir complémentarité / redondance éventuelles avec fParser / PSYCLONE

Et maintenant ?

?

- Comment valider
- Comment interagir avec les développeurs
- Comment décider
- Comment paralléliser
- Comment passer au 3D