CRIANN

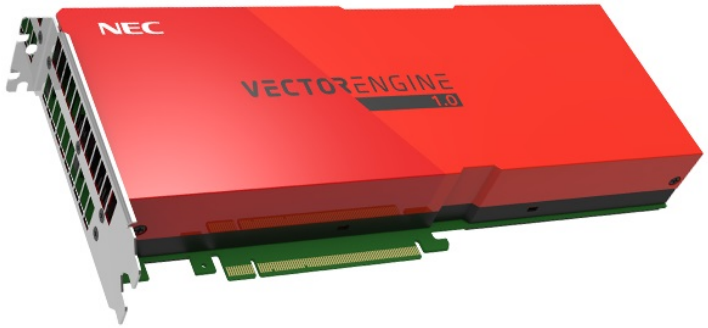
Processeurs vectoriels Aurora - Preuves de concept en cours sur prototypes et applications utilisateurs

CRIANN
19 Novembre 2020

Contexte

NEC Aurora

Architecture vectorielle



- Veille technologique CRIANN en 2020, avec le support de NEC
- Un serveur acquis (2 cartes Aurora, 2 CPU Cascade Lake)
  - Dans le cluster de test Deca
- Domaines/méthodes testés concernant la communauté d'utilisateurs du PRMN
  - HPC : CFD/LBM, FFT, chimie, dynamique moléculaire
  - IA/DL et traitement de données

Processeurs

mentionnés dans ce document

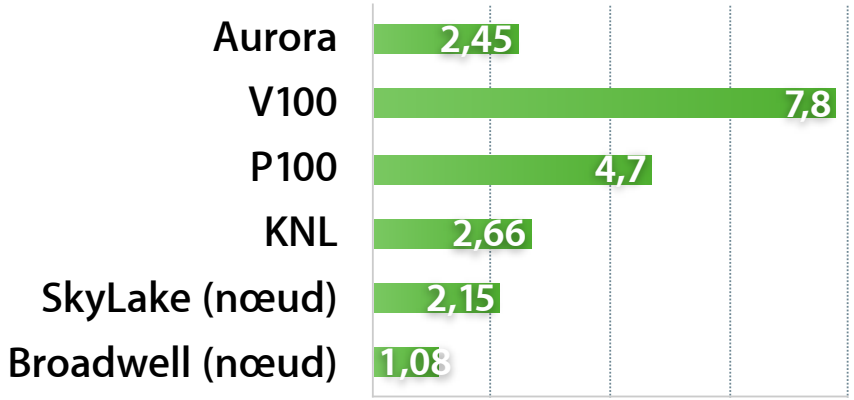
Processeur	Catégorie	Nom abrégé (pages suivantes)	High Bandwidth Memory
AMD EPYC 7642 (48 cœurs 2,3 GHz)	CPU	Rome	-
Intel Xeon E5-2680 v4 (14 cœurs 2,4 GHz)	CPU	Broadwell	-
Intel Xeon Gold 6130 (16 cœurs 2,1 GHz)	CPU	SkyLake	-
Intel Xeon Phi 7210 (64 cœurs 1,3 GHz)	CPU	KNL	16 GB MCDRAM
NVIDIA P100-PCIe	GPU	P100	12 GB HBM2
NVIDIA V100-SXM2	GPU	V100	32 GB HBM2
NEC SX-Aurora TSUBASA 10BE-P	Accélérateur vectoriel	Aurora	48 GB HBM2

Architecture

Aurora : architecture

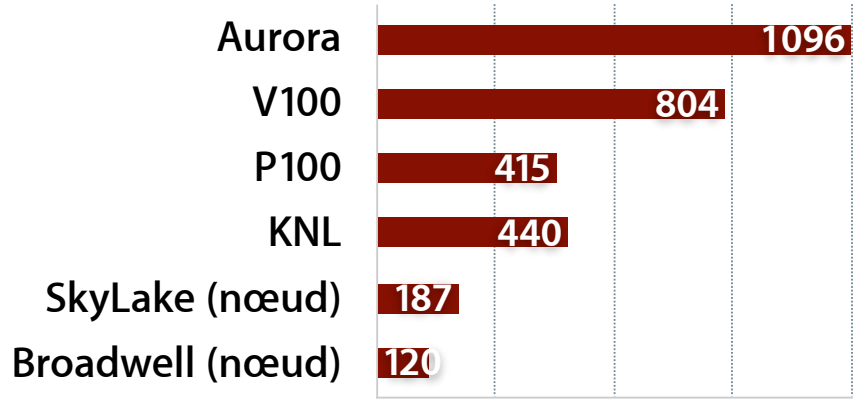
Vector Engine (VE)

Crête : TFlop/s (double précision)



*Aurora : **médian en Flop/s**, intensif en **bande passante mémoire***

Stream benchmark, triad (GB/s)



- Un VE est muni de 8 cœurs
  - Chacun de ces cœurs a un registre vectoriel de taille 256 x (64 bits)
    - 32 x (taille de registre CPU Xeon)
- Bande passante mémoire
  - > 1 TB/s mesuré par VE
- Cible d'Aurora
  - Applications *Memory Bound*
    - (imagerie, CFD, mécanique, climat, sciences de la terre, dynamique moléculaire, chimie, etc.)

Aurora : architecture

Environnement logiciel

- https://www.hpc.nec/documentation
  - Compilateurs C/C++, FORTRAN ; librairies OpenMP et MPI ; profilers processeur/MPI ; debugger
  - Numeric Library Collection (NLC), dont BLAS, LAPACK, FFT
  - Utilisation des cartes Aurora en *mode natif* ou *offload*, i.e. déport sur accélérateurs de parties de code : API *VEO (Vector Engine Offloading)*
  - NLCPy : Python accéléré (offload de fonctions NumPy)

Aurora : architecture

Programmation

- Aurora

Parallélisation

Vectorisation (*)

Rapport du compilateur de NEC

```
   61:                #pragma omp for
   62: P------>         for (iterx = sx; iterx < ex+1; iterx++) {
   63: |
   64: |+----->           for (itery = sy; itery < ey+1; itery++) {
   65: ||
   66: ||V---->               for (iterz = sz; iterz < ez+1; iterz++) {
   67: |||
   68: |||                        u_nouveau[IDX(iterx, itery, iterz)] = …
```

(*) Conditions, bonnes pratiques, directive (*ivdep*) : https://www.hpc.nec/documents/guide/pdfs/AuroraVE_TuningGuide.pdf

- vs GPU

```
61 #pragma acc kernels present(u, u_nouveau, f, coef)
62 {
63 #pragma acc loop independent
64   for (iterx = sx; iterx < ex+1; iterx++) {
65
66 #pragma acc loop independent
67     for (itery = sy; itery < ey+1; itery++) {
68
69 #pragma acc loop independent
70       for (iterz = sz; iterz < ez+1; iterz++) {
71
72         u_nouveau[IDX(iterx, itery, iterz)] = …
```

Rapport du compilateur PGI pour OpenACC

```
pgcc  -O2  -acc -Minfo=accel -ta=tesla:cc70  -c calcul.c
calcul:
 62, Generating present(f[:],u[:],u_nouveau[:],coef[:])
 64, Loop is parallelizable
 67, Loop is parallelizable
 70, Loop is parallelizable
  Generating Tesla code
   64, #pragma acc loop gang /* blockIdx.y */
   67, #pragma acc loop gang, vector(4) /* blockIdx.z threadIdx.y */
   70, #pragma acc loop gang, vector(32) /* blockIdx.x threadIdx.x */
```

Aurora : architecture

Intérêts potentiels

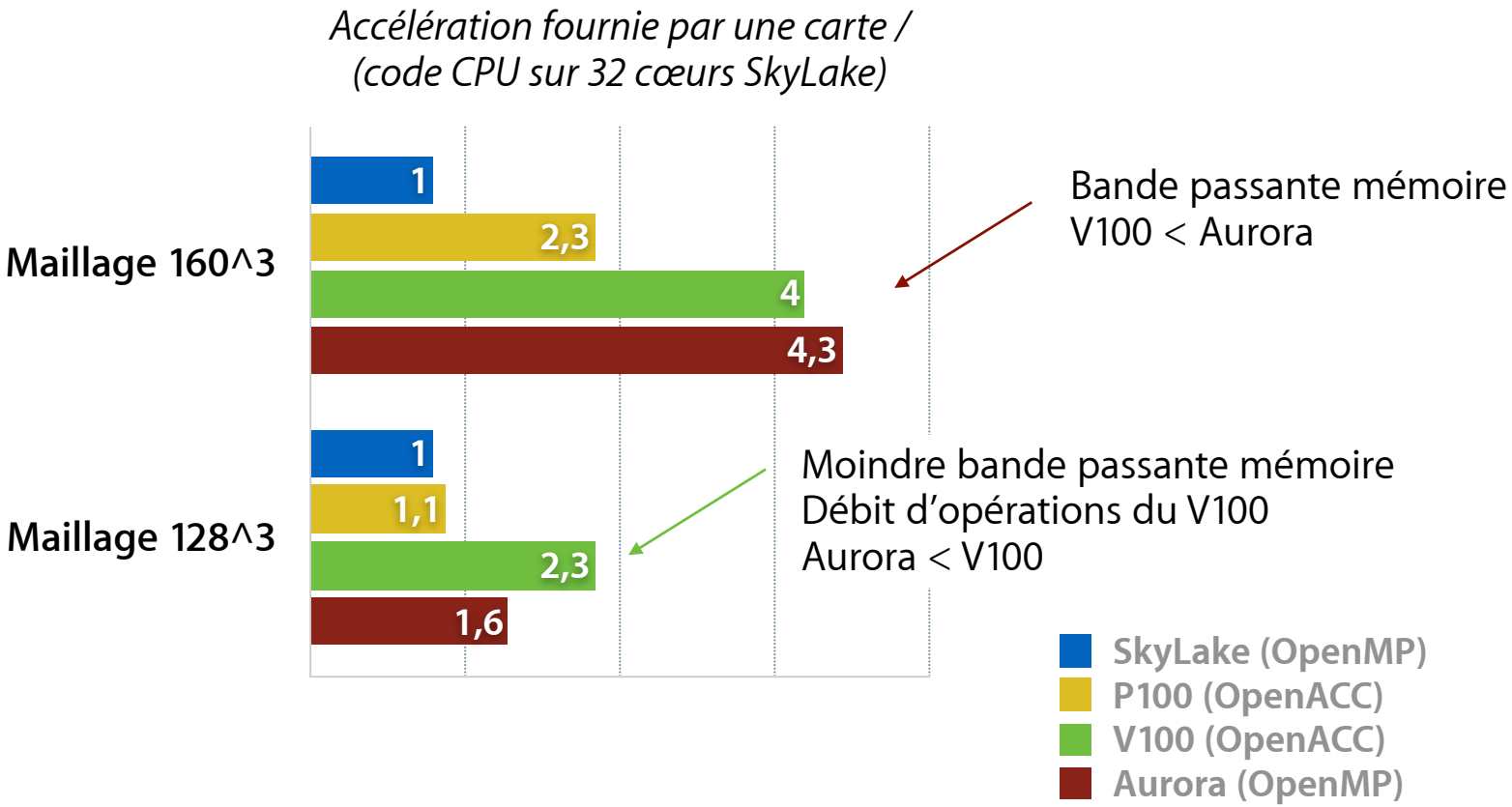
- Accélérateurs adressés par les langages et API standard (C/C++, FORTRAN, MPI, OpenMP), largement maîtrisés par la communauté
- FFT optimisée
- Préparer/quantifier un travail ultérieur de portage sur GPU

- Traitement de données : en cours d'évaluation au CRIANN (Spark …)

- Performance énergétique
  - Argument d'investissement de DWD (institut météorologique allemand)
    - https://www.nec.com/en/press/201906/global_20190617_01.html

Aurora : architecture : illustration

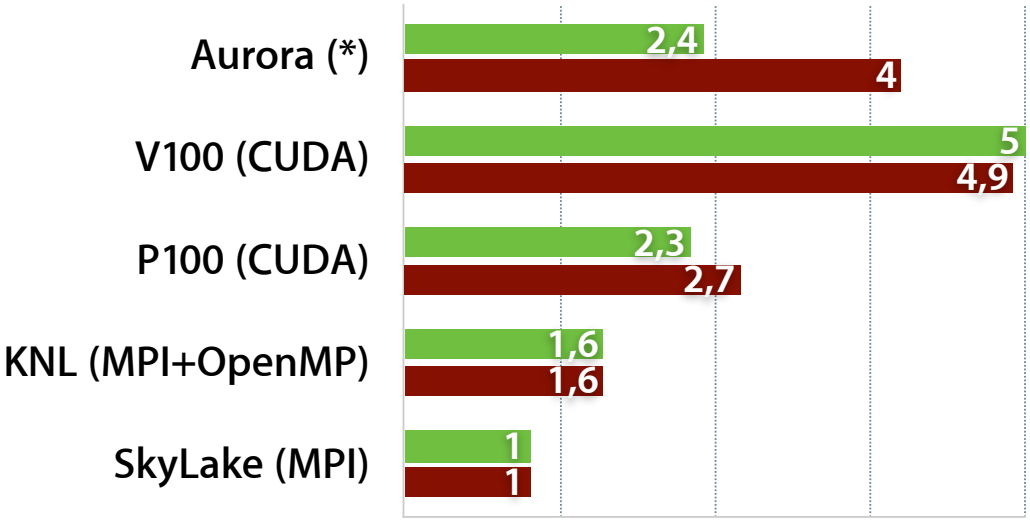
Noyau Poisson 3D (double précision, Jacobi)



Aurora : architecture : illustration

Traitement d'image 2D (double précision)

Accélération fournie par une carte ou un KNL / (code CPU sur 32 cœurs SkyLake)



Application D2F : détection de fissure
Vectorisation sur KNL,
suivie d'un portage *direct* sur Aurora ((*) OpenMP)

Schémas d'ordre élevé, pour lesquels la librairie SCA (Stencil Code Accelerator : https://www.hpc.nec/documents/sdk/SDK_NLC/UsersGuide/sca/c/en/index.html) de NEC fournirait une optimisation à tenter (au prix d'une programmation 100% spécifique)

D2F (9 M pixels) RSEG (4,5 M pixels)

Application RSEG : recalage et segmentation conjoints
Vectorisation sur KNL,
suivie d'un portage sur Aurora en *un jour* ((*) MPI)
Schémas du premier ordre

- P100 ≲ Aurora < V100, sans API spécifique sur Aurora (C ici, OpenMP, MPI)

HPC

Chantiers initiés

CFD

Application	Labo	Travaux effectués / en cours
YALES2, CFD/LES	CORIA	Modèle de chimie et diffusion de YALES2, pour un nombre d'espèces allant jusqu'à 60 - Objectif : déport sur les VE (« offload ») des sous-programmes de chimie et de diffusion d'espèces, le reste du modèle étant gardé sur CPU - Portage natif initié, pour analyse des sous-programmes précédents
GPS, Fluide quantique	Pprime / LMRS	GPS (base FFTW) porté en natif *- Gain (serveur NEC (2 VE)) / (1 serveur (28 cœurs) Broadwell) = 2, maillage 256^3 :* modéré, pour cause de non vectorisation en direction y ; amélioration possible, mais serait spécifique à l'architecture *- Accélération linéaire de 1 à 2 VE*

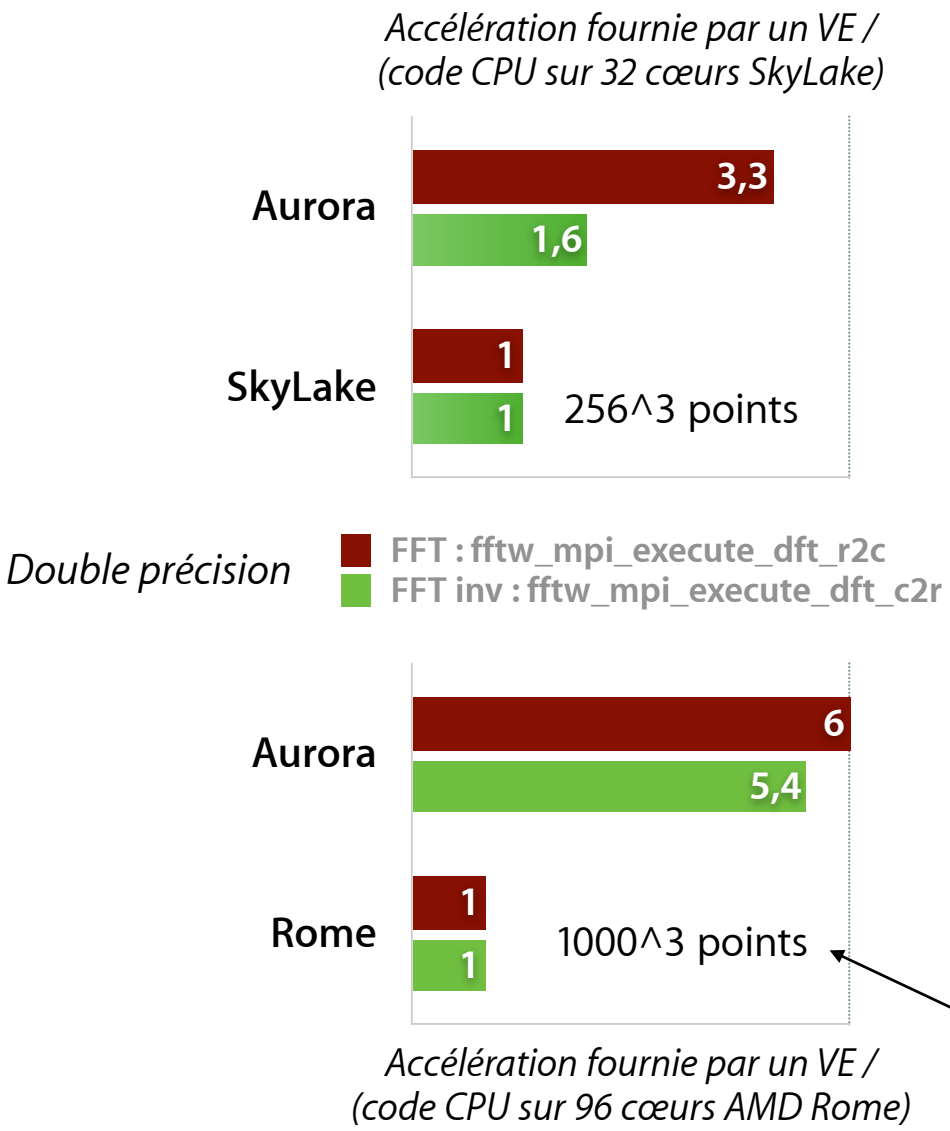
Chantiers initiés

Simulation atomistique

Application	Travaux effectués / en cours
Quantum Espresso (QE)	QE 6.3 déjà porté sur Aurora par NEC - 8 VE = (2 nœuds x86 AMD Rome) pour cas test à 92 atomes - Cas test à 408 atomes en cours d'analyse QE 6.4 en cours de portage par NEC
VASP (Vienna Ab initio Simulation Package)	Version 5.4 de VASP déjà optimisée par NEC pour Aurora - Cas test d'un partenaire du CRIANN en cours d'analyse

Chantiers initiés

FFT 3D MPI, laboratoire GPM



- Code test du GPM
  - Noyau FFT, préalablement au portage sur Aurora de l'application PFC (Phase Field Crystal) en cours de réécriture
- Librairie FFT parallèle (MPI)
  - Sur CPU : FFTW 3.3.x
  - Sur Aurora : aslfftw3 de nlc
- Taille de système ciblé par GPM

Chantiers initiés

LBM, laboratoire LMRS

Lattice Boltzmann Method (LBM)

- Méthode qui s'appuie sur la physique statistique et sur l'équation de Boltzmann pour simuler le comportement des fluides
- Mécanisme de collision et de propagation appliqué à un ensemble de particules d'échelle intermédiaire

  – *stream* : $f_i(\vec{x} + \vec{e}_i\delta_t, t + \delta_t) = f_i^t(\vec{x}, t + \delta_t)$

  – *collide* : $\Omega_i^t(\vec{x}, t + \delta_t) = f_i(\vec{x}, t) + \frac{1}{\tau_f}(f_i^{eq} - f_i)$

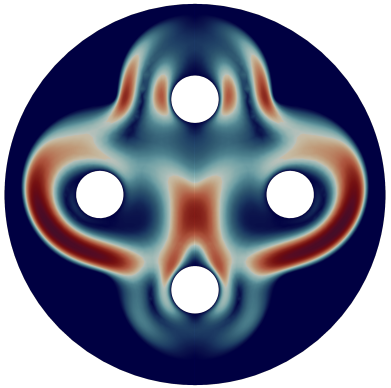
avec :

  – $f_i^{eq} = \rho w_i \left(1 + \frac{1}{3}(\vec{u} \cdot \vec{e}) + \frac{9}{2}(\vec{u} \cdot \vec{e})^2 - \frac{3}{2}(\vec{u} \cdot \vec{u})\right)$

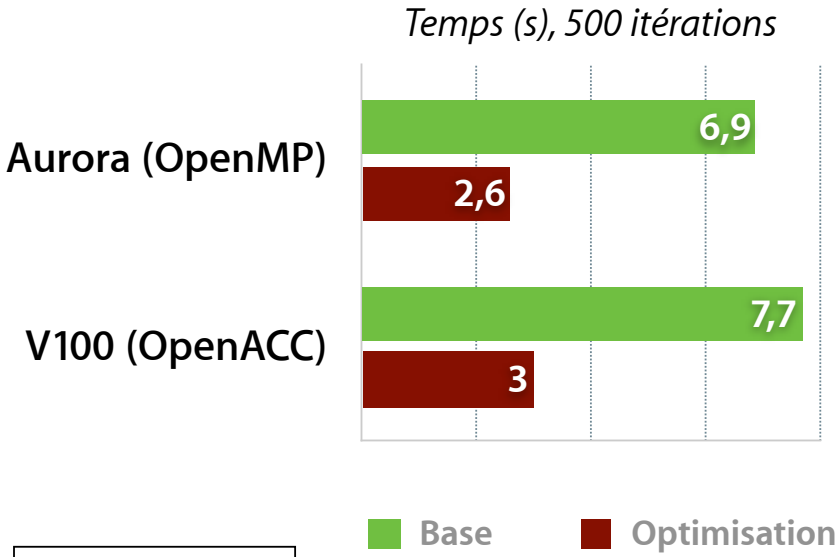
  – $\rho = \sum_i f_i(\vec{x}, t)$

  – $u = \sum_i f_i(\vec{x}, t)\overrightarrow{e_i}$

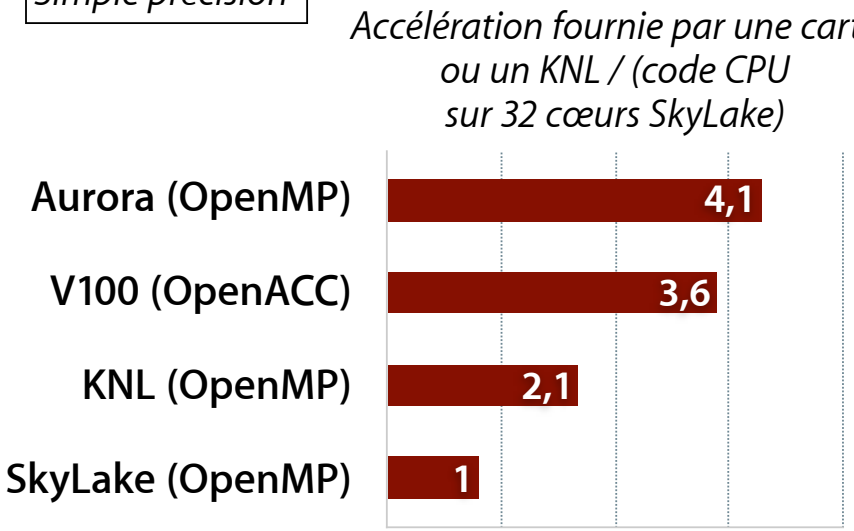
- Objectif scientifique du LMRS
  - Appliquer la LBM à la modélisation de matériaux à changement de phase
    - Thermique et écoulement, changement de phase solide-liquide



Fusion d'un matériau avec 4 tubes de chaleur (éléments finis)



Maillage 4096^2
Simple précision



Chantiers initiés

LBM, laboratoire LMRS

- Au préalable : analyse d'un noyau LBM sur accélérateurs
  - Cavité entrainée

– Base :
  - f_{eq}, Ω et f de taille $N \times N \times 9$ (D2Q9)
  - calcul f_{eq}, ρ et $\vec{u}$ à part
  - calcul de Ω à partir de f, f_{eq}, ρ et $\vec{u}$

– Optimization :
  - Ω calculé à partir de f seulement (pas de passage par un intermédiaire)
  - expressions de Ω obtenues grâce à `sympy`
  - f et Ω sont des structures de 9 tableaux $N \times N$

Le Pôle Régional de Modélisation Numérique,
le réseau régional pour l'éducation et la recherche
et la Maison Normande des Sciences du Numérique
sont des actions cofinancées par la Région Normandie, l'État et l'Union européenne

RÉGION NORMANDIE

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

Ce projet a été cofinancé par l'Union européenne.
L'Europe s'engage en Haute-Normandie avec le Fonds européen de développement régional.

CRIANN

Centre Régional Informatique et d'Applications Numériques de Normandie
www.criann.fr

@criannormandie