



On the Road Again ...



A. Refloch

EXA



retour sur innovation

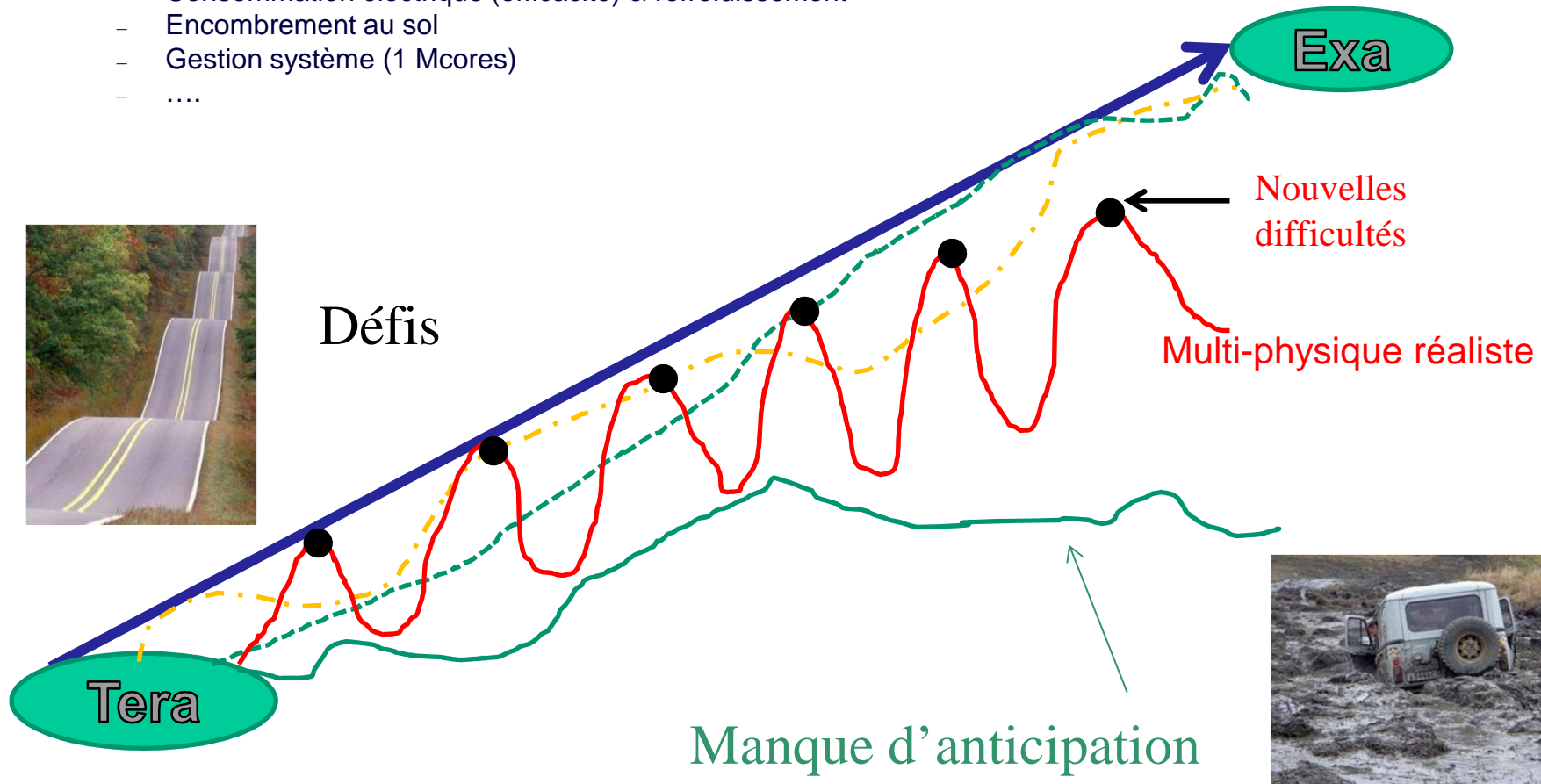
Sur la route

Considérations générales

- Tolérance aux pannes
- Consommation électrique (efficacité) & refroidissement
- Encombrement au sol
- Gestion système (1 Mcores)
-



Défis



Nouvelles
difficultés

Multi-physique réaliste

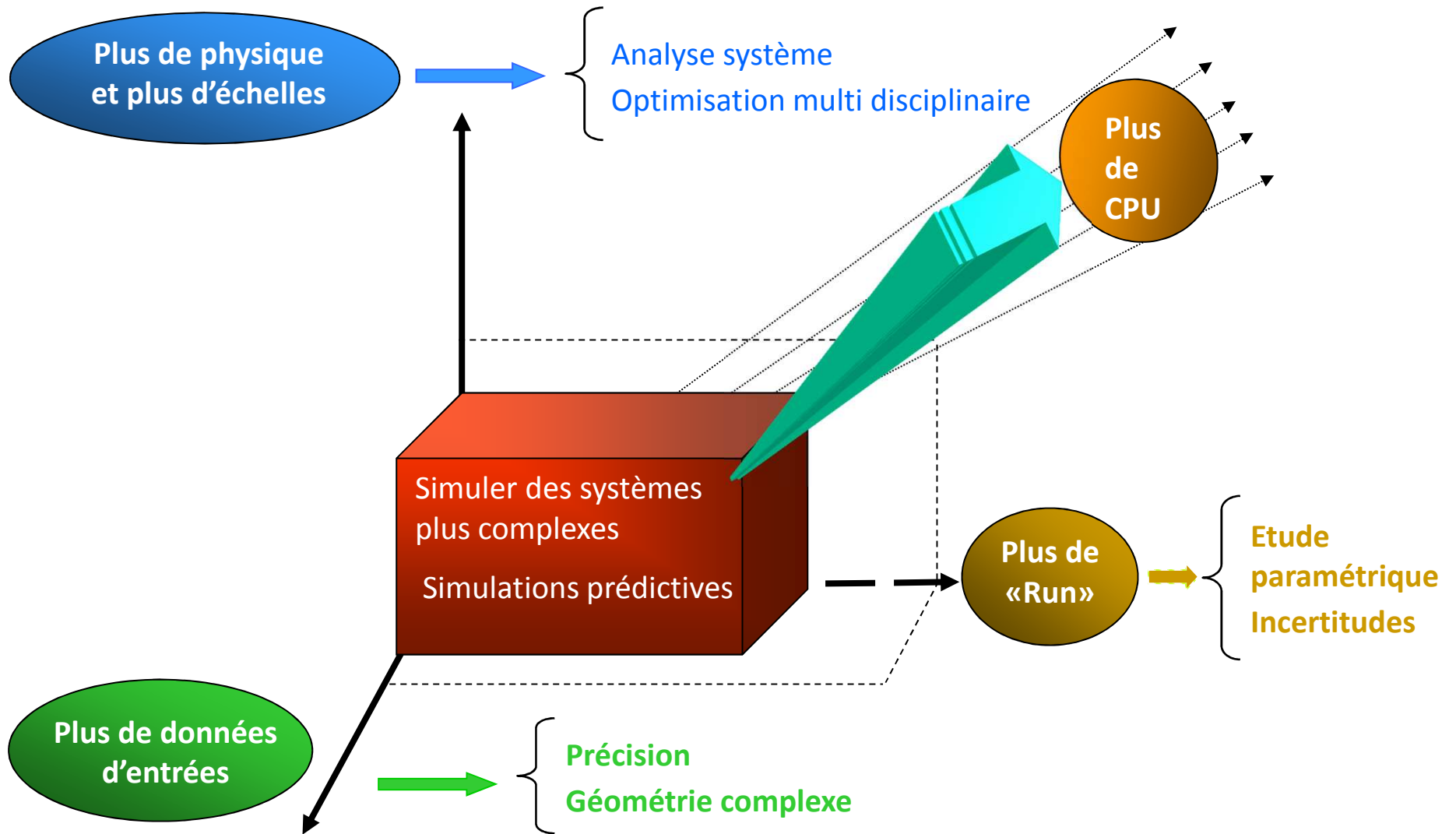
Manque d'anticipation



ONERA

THE FRENCH AEROSPACE LAB

HPC : 1 cible – 3 grands axes



Multi physique réaliste : **Nouvelles difficultés à prendre en compte**

- Géométrie complexe : intérêt : temps de réalisation des maillages
-> maillage non structuré -> numérique et localité des données (mémoire & IO)
- Multi-physique
 - conditions pure aérodynamique (1 espèce, gaz parfait) +
 - conditions réactive (multi-espèces, cinétique chimique, gaz réel) +
 - conditions multi-phases (dispersée, séparée, mixée) +
 - conditions multi-physiques (aéroacoustique, interaction fluide-solide (transfert de chaleur, mécanique), ...)
- Solveurs Lagrangien : Parallélisme – Mémoire
- Couplage (de codes mono physique : algo de couplage, équilibrage)
- Parties mobiles : remaillage automatique, partitionnement en cours de calcul



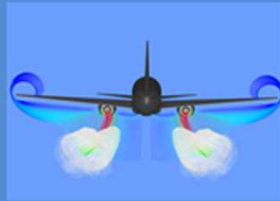
Défi Exa => HPC à tous les niveaux

Bien plus que la Σ des défis HPC mono physique

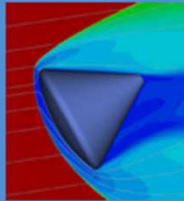
CEDRE[®] en quelques mots

- Code de référence de l'Onera pour énergétique et la propulsion
- Maillage non structuré généraux
 - "Nombre quelconque de faces par cellule, nombre quelconque de noeuds par face"
- Parallèle 'classique'
 - MPI
 - Décomposition de domaines (Utilisation de découpeur de maillage)
- Multi-physique
 - Interne (différent solveur pour différent système physique)
 - Charme: Navier-Stokes, turbulent, réactive, gaz réel, multi-phases
 - Sparte: phases dispersée (Lagrangien)
 - Spirée: phases dispersée (Eulerien)
 - Astre: rayonnement (Monte-Carlo)
 - Réa: rayonnement (ordonnées discrètes)
 - Acacia: thermique
 - Film: films liquides
 - Externe (Bibliothèque de couplage : CWIPI, OpenPalm)
- Approche Zonale – gestion des interfaces

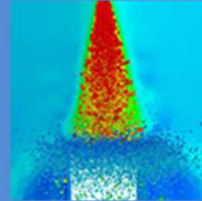
Applications CEDRE



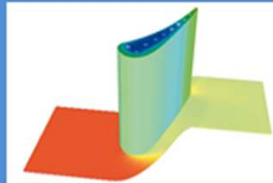
Trainée de condensation



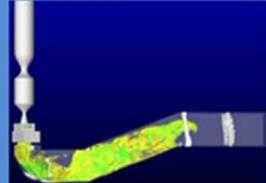
Véhicule hypersonique



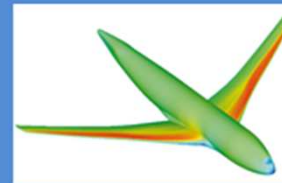
Diphase



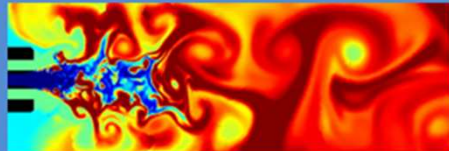
Couplage aérothermique



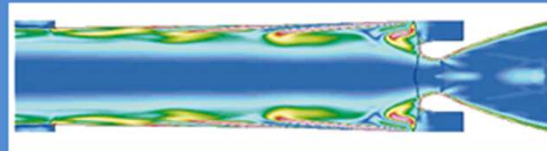
Onde de souffle



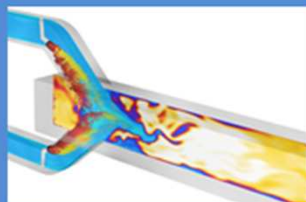
Aérodynamique



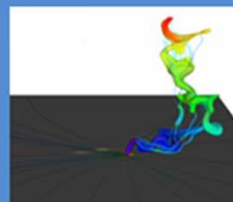
Propulsion liquide



Moteur à propergol solide



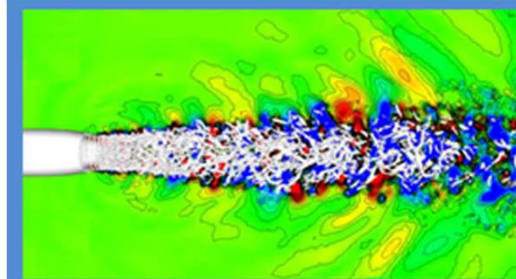
Statoréacteur



Foudre - Calcul couplé
CEDRE-Saturne_EDF



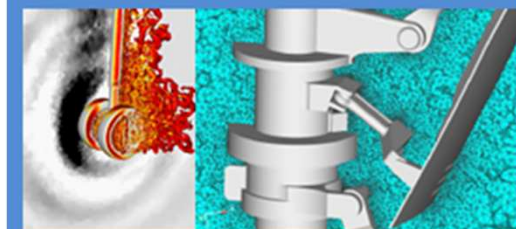
Cavité aéroacoustique



Bruit de jet



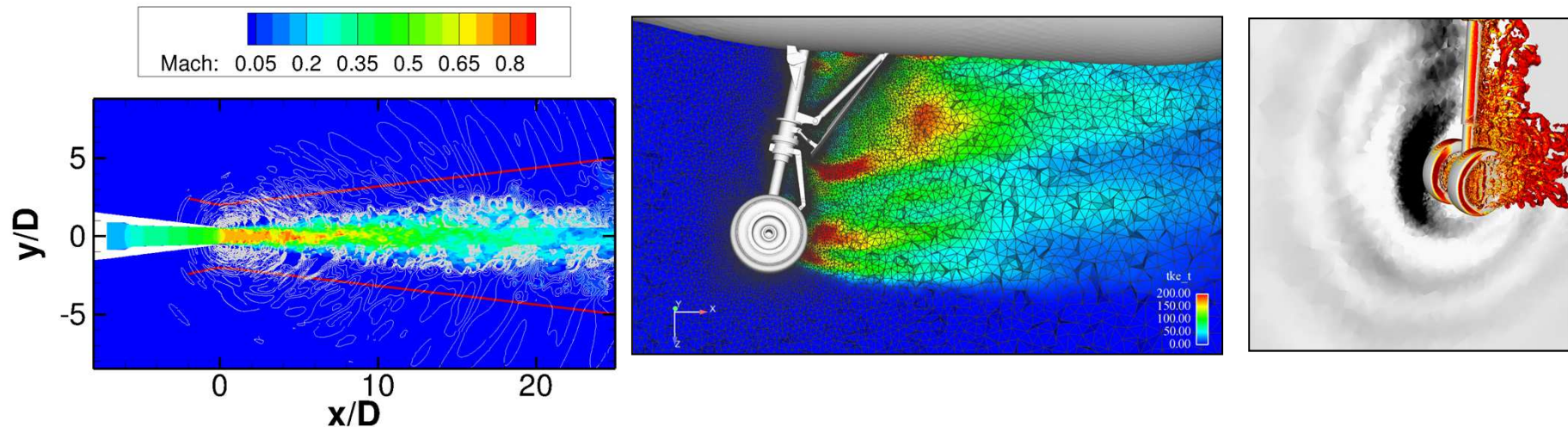
Chambre de combustion



Trains d'atterrissage

1 exemple d'application LES Aeroacoustique

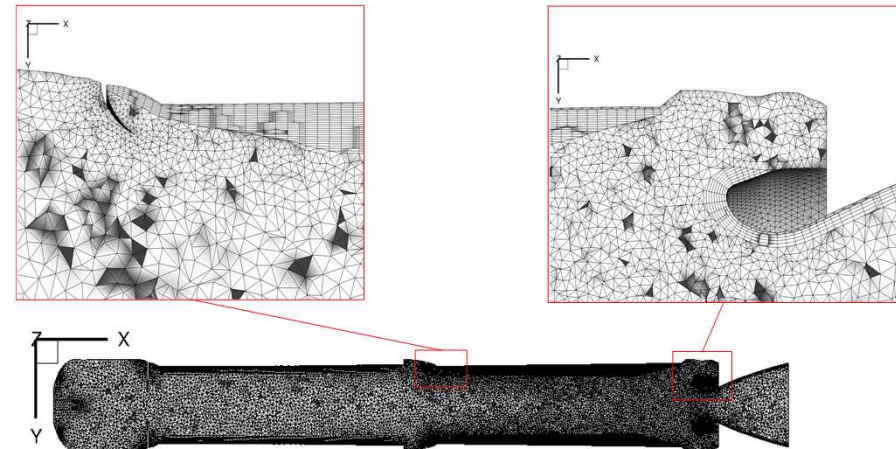
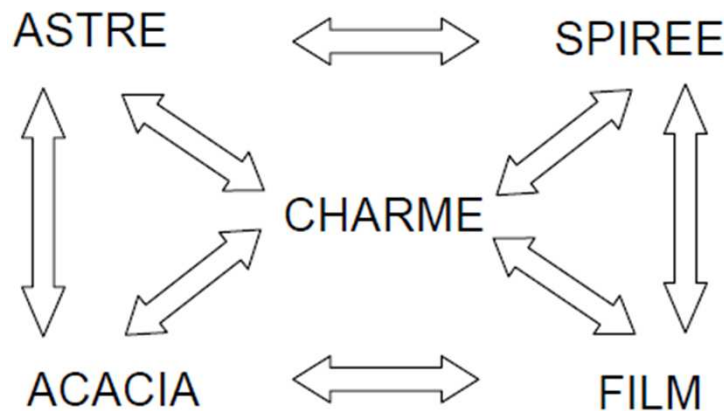
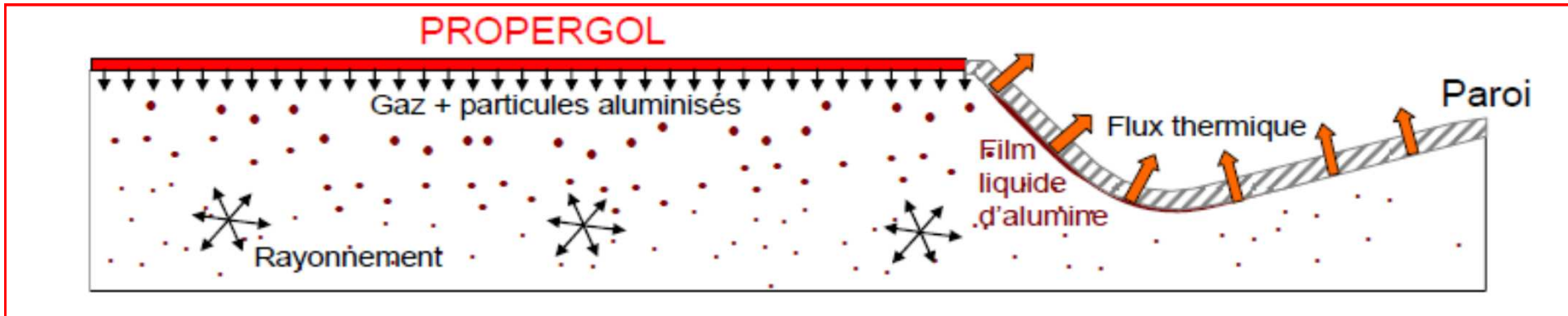
- **Bruit de Jet – réduction du bruit des avions**
 - approche hybride : cedre + code Onera acoustique
 - 250 Mcells / 960 cores
- **Bruit de train d'atterrissage**
 - Géométries complexes : 70 Mcells + ZDES



M. Lorteau et al "Recent progress in LES computation for aeroacoustics of turbulent hot jet. Comparison to experiments and near field analysis", AIAA 2014-3057, 20th AIAA/CEAS Aeroacoustics Conference, 16-20 June 2014, Atlanta, Ga, USA

L. Sanders et al., "Further flow and noise predictions of the Gulfstream PDCC nose landing gear based on the CEDRE unstructured solver", 19th AIAA/CEAS Aeroacoustics Conference, 27-29 May 2013, Berlin, Germany
F. de la Puente, et al., "On LAGOON nose landing gear CFD/CAA computation over unstructured mesh using a ZDES approach", AIAA 2014-, 20th AIAA/CEAS Aeroacoustics Conference, 16-20 June 2014, Atlanta, Ga, USA

Application Propulsion Solide



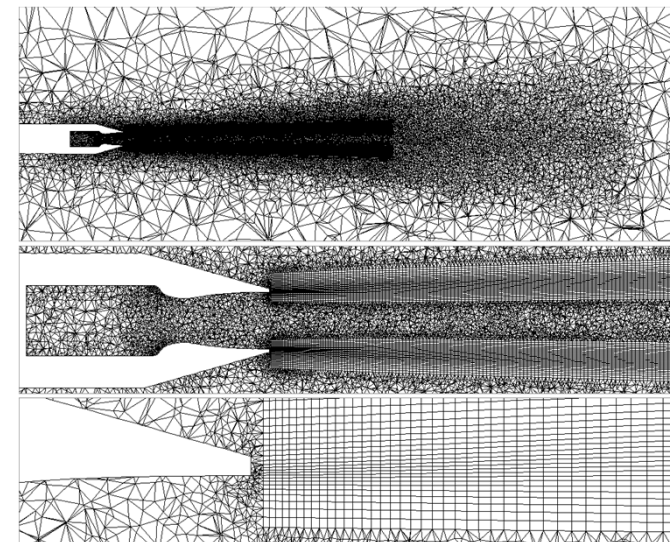
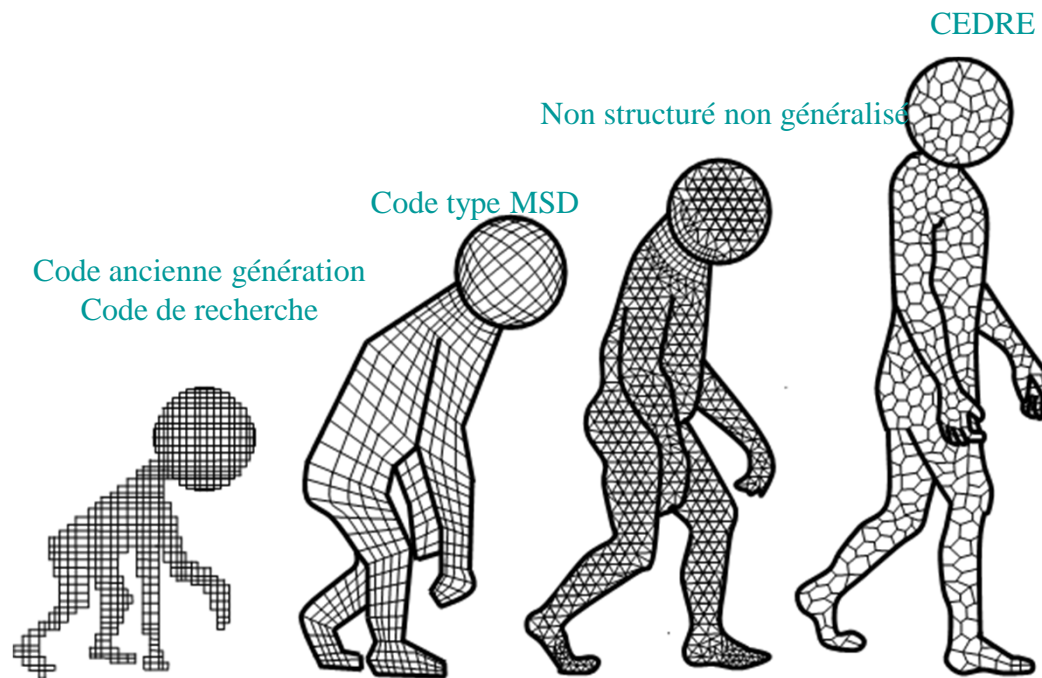
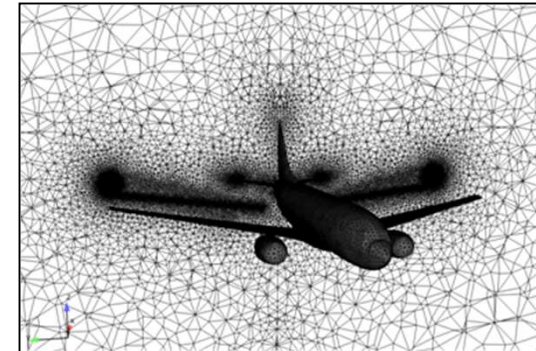
- 1er calcul* : Écoulement avec film d'alumine
- 2ème calcul* : Écoulement + rayonnement
- 3ème calcul* : Écoulement+film+rayt+thermique parois

Développement:

- ASTRE : Répartition P_{rad} gaz et P_{rad} gouttes
- SPIREE : Prise en compte P_{rad} gouttes
- FILM : Modèle arrachement + séparation et
- Modèle couplage thermique

Physique réaliste, effet technologique : Maillages non structurés généralisés


Possibilité de discrétiser des géométries très complexes, dans des temps de réalisation des processus de conception industriel.



Mais la localité est le secret des performances

Bonne performance implique bande passante importante, renumérotation des faces

MEMOIRE

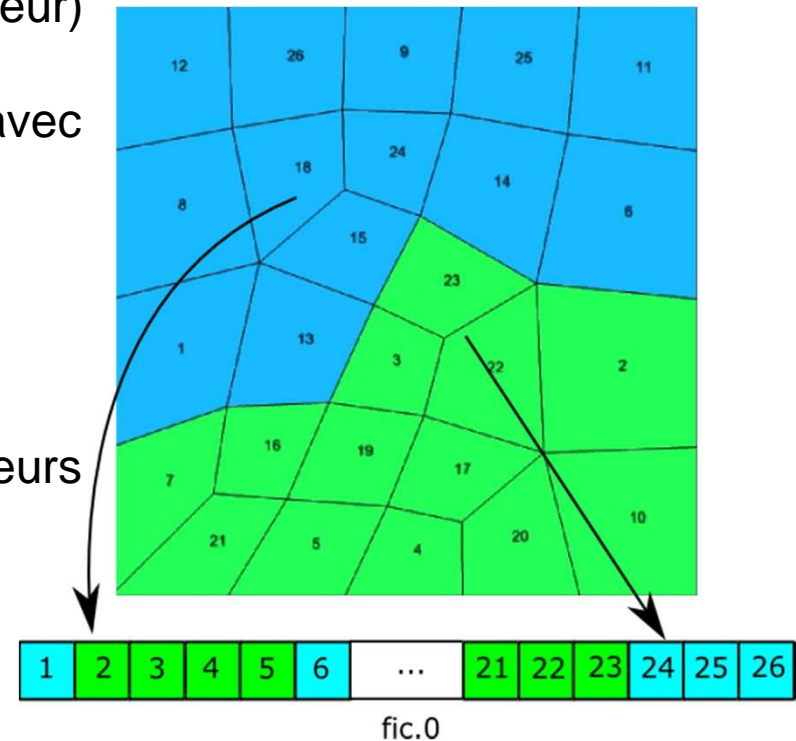
- Importance de la mémoire :
 - % du prix de la machine et % de la consommation électrique
=> mesure à l'intérieur du code
- Tendence naturelle à la décroissance de la mémoire disponible par coeur
 -  Plus de mémoire influencera le couplage multi physique

Mais (bonne nouvelle) :

- Problème de résilience : MTTI => checkpoint-restart
- Avancée dans la techno
 - NVRAM ; PRAM (parameter random access memory) au lieu de RAM
 - ReRam (HP – The Machine project)
- Impact du Big Data (besoin de grosse mémoire)

I/O

- Un grand nombre de petits fichiers (par coeur) n'était plus une solution :
 - Problème de backup et pas optimum avec « file system » parallèle (Lustre, GPFS)
- Orientation vers un fichier unique :
 - Binaire,
 - Compatibilité little-endian/big-endian,
 - Reprise sur un nombre différent de coeurs



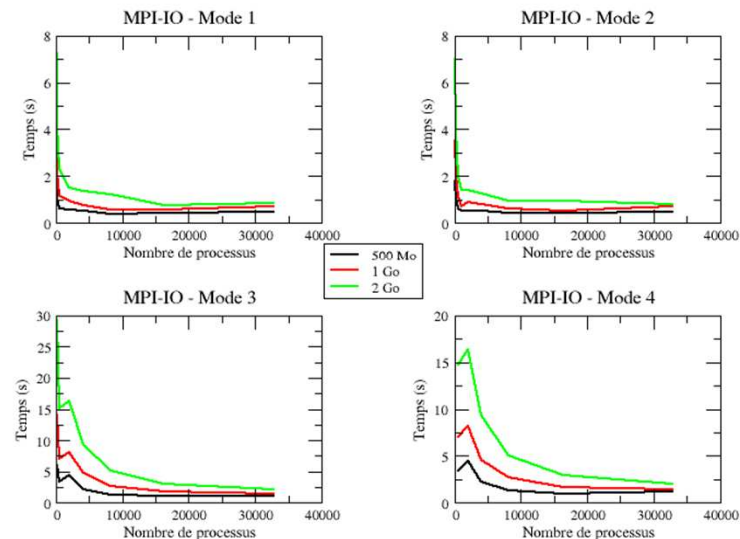
Bibliothèque PARIO

- Goulet d'étranglement

- Test sur Projet PRACE Cedre Preparatory project n°PRA0 17, 2010, computer Jugene from Juelich, 1Mh
- IO library test sur 32000 cœurs, GPFS meilleur comportement que Lustre
- IO source principal d'arrêt de production

- MPI IO

- Collective writing per bloc :
 - Mode 1 (reference)
 - Mode 2 (use of MPI_TYPE_HINDEXED)
- Interleaved writing
 - Mode 3 (use of MPI_TYPE_SUBARRAY)
 - Mode 4 (use of MPI_TYPE_HINDEXED)



IO : sujet ingrat, résultats de performance très variables en fonction de la charge de la machine, sujet mal appréhendé et documenté, pas de grands moyens d'actions du côté Utilisateur, dépendant du système,...

- Stage en cours (PARIO, HDF5, HDF5Exa, DAMARIS, SIONlib)

Multi-physique

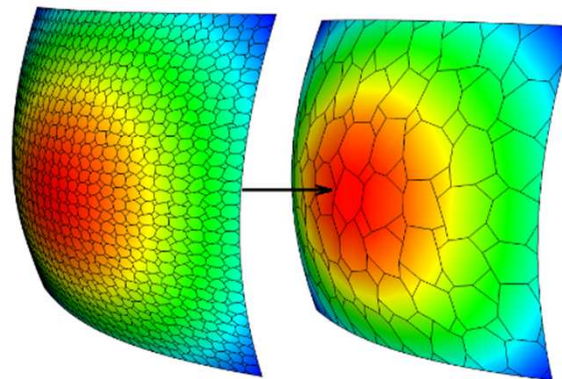
- Approche Lagrangien : parallèle par trajectoire (e.g. photons)
 - Approche Lagrangienne necessite la connaissance de tout le maillage
 - Limitation par la mémoire : une solution : OpenMP qui a permis rapidement de faire passer la limite de 3 M éléments à 20 M éléments (48Go/noeuds NHM)
 - Parallélisme multi-niveaux (disparition de la programmation homogène (pur MPI))



On sait aujourd'hui que la programmation parallèle sera hybride

CWIPI: Coupling With Parallel Interface

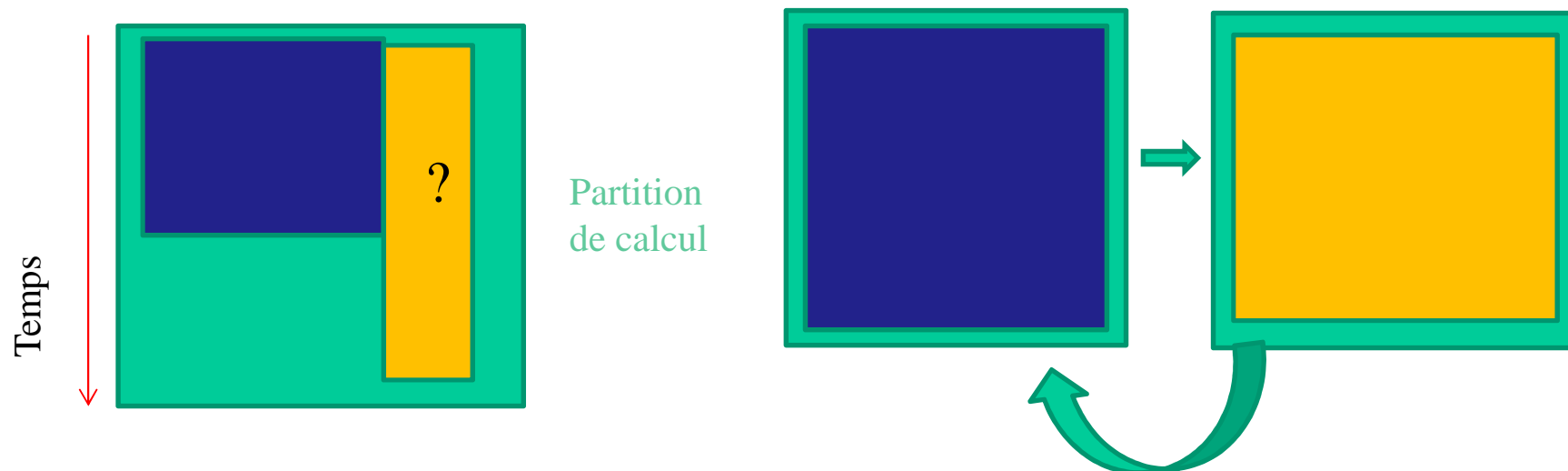
- Développé à l'ONERA par E. Quémérais
- Open source (cedre.onera.fr)
- Test sur Curie 32k coeurs – 1 Milliard de points
 - Simule le couplage entre deux codes au travers d'un échange sur une interface constituée de deux surfaces partitionnées. La discrétisation de la surface est unique à chaque code.
 - Synchrone vs asynchrone: ratio 11 à 15



Le couplage est un moyen naturel d'utiliser la puissance de l'Exascale.
Il faut donc un coupleur dans sa caisse à outils.

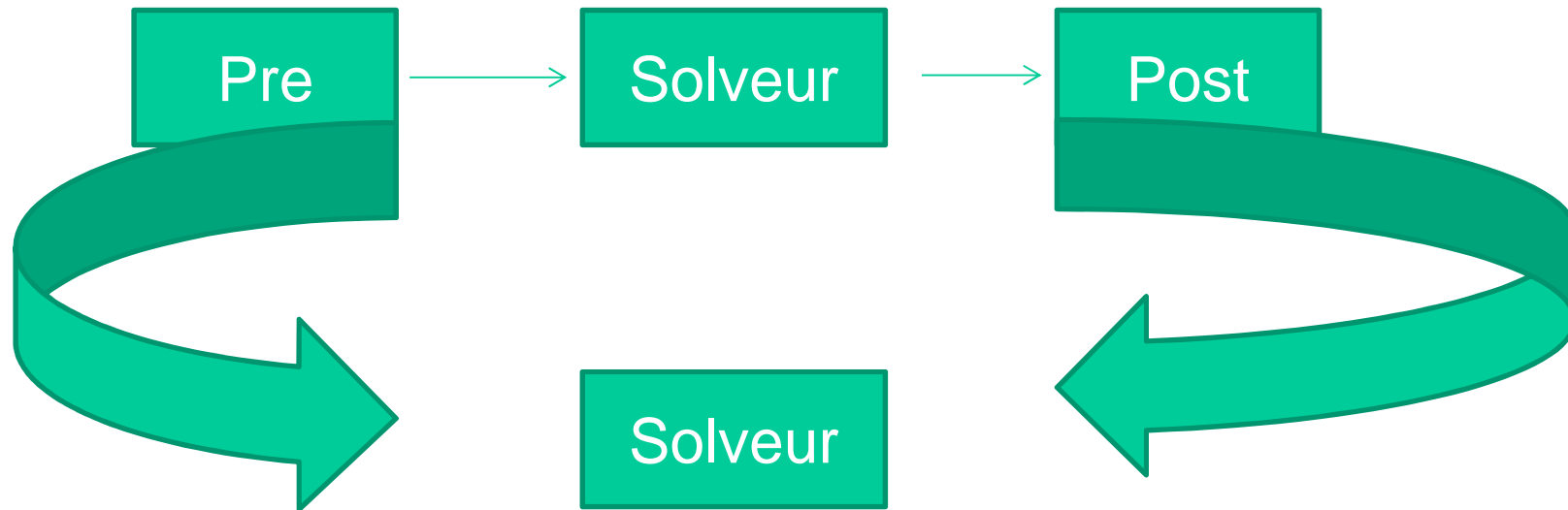
Couplage multi physique 'externe – interne'

- Couplage externe : Zebulon (Zset – code de structure)
- Couplage interne dans CEDRE :
 - Multi physique possible au sein de l'Onera
 - Assemblage au sein du code (allocation mémoire, passage par arguments, ...)
 - Toutes les infos disponibles pour divers algo de couplage
 - Cycle global, cycle par solveur



Tous les cœurs actifs: LA solution si le nombre de cœurs reste limité

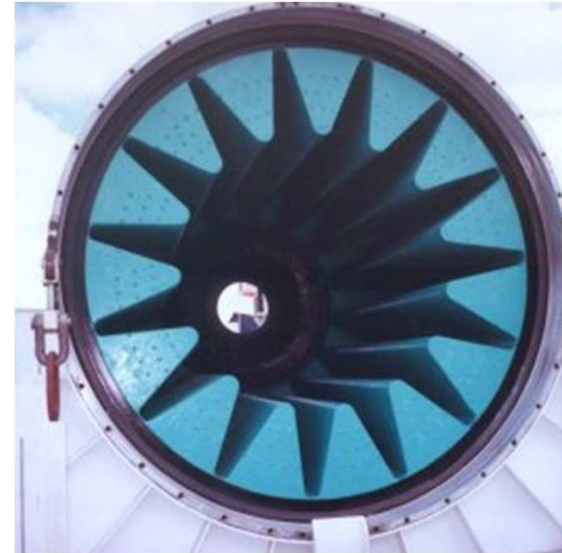
Environnement - Chaine de calcul



- Réintégration de fonctionnalités pré-post
- Analyse in-situ
- Consomme une partie non négligeable des ressources humaines

Partie Mobile

- Régression propergol
- Séparation de charge
- Protection thermique
- Braquage de tuyère
- Vannage gaz chaud
- Partie tournante
-



- => Raffinement-deraffinement, basculement de faces
 - Recollement conservatif le long d'une surface commune
 - Chevauchement conservatif de maillages

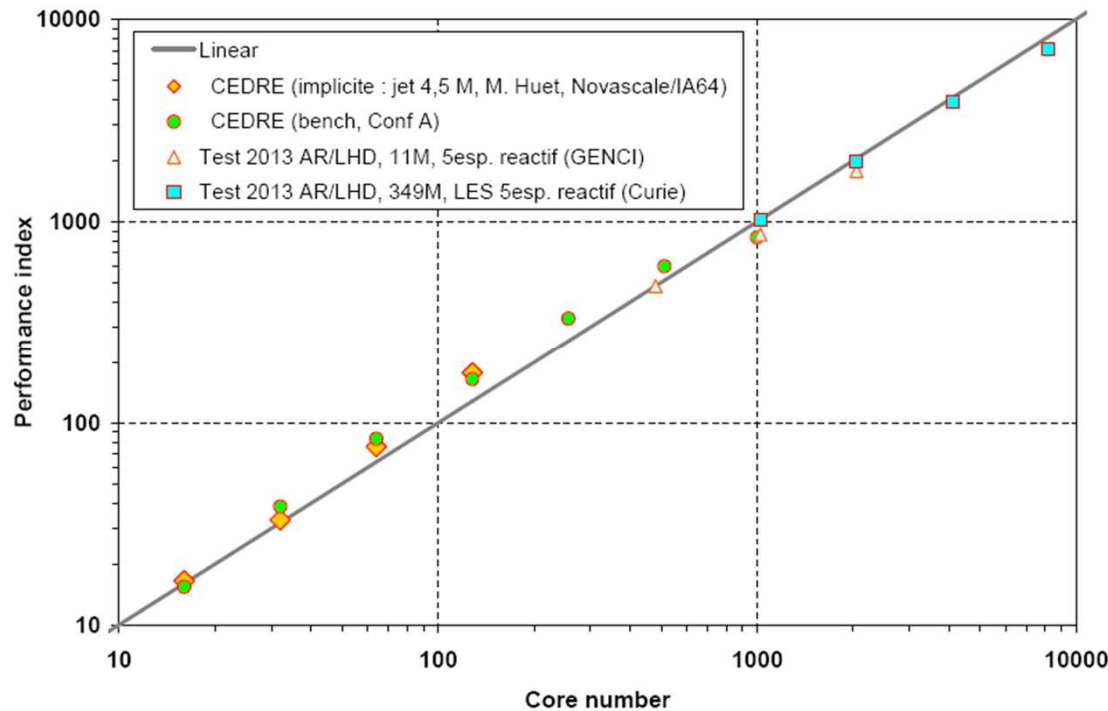


Remaillage en cours de calcul : Un vrai challenge « exascale »

Scalabilité

- *We acknowledge PRACE for awarding us access to resource CURIE based in FRANCE at TGCC, Project PA1389, 2013*

Study of combustion instabilities in a variable-length combustor by Large Eddy Simulation on 350 Mcells



Besoin d'un accès à de gros moyens de calculs **pour le DEVELOPPEMENT**




Aggregated results from several projects

Opérations collectives à éviter



Environnement de développement



Avalanche d'outils : *module available*

- *Profiler CPU et elapsed ok mais utilité ...*
- *Profiler mémoire vrai besoin*
- *Debugger : vital* 

Typologie & contraintes

- Code de recherche
 - Déjà pas simple ...
- Grand code de recherche
 - Volume du développement et la taille de l'équipe (importance du GL)
 - Evolution rapide
- Grand code de recherche utilisé par l'industrie en avant projet
 - Portabilité (aucune maîtrise sur les moyens informatique externe)
 - Pérennité (code traverse plusieurs générations de machines)
 - Validation (bit à bit ne sera plus tenable)
- Les grands code de recherche utilisé par l'industrie en BE
 - Convivialité
 - Performance pour la conception
- Avec quels moyens humains ? 
- Conséquence  réduction du nombre de codes Exa

Définir une stratégie, exemple :

- **Court terme:**
 - Optimiser les codes sur HW actuel, comprendre les évolutions, interaction avec les vendeurs
- **Moyen terme:**
 - Sur la base des codes en production, test nouveau HW, faire des benchmarks, accéder à des hardware les plus récent (exemple: MIC Intel)
 - Anticiper nouveaux développements: plus de physique, mais aussi nouvelle approche numérique (interne, collaborations ...) (sur 30 ans les avancées 'numériques' égalent celles des machines)
 - Modèle de programmation (exemple Co Array Fortran, DSL ?)
- **Long terme:**
 - Collaboration, anticiper nouvelle technologie au travers de collaboration R&D (Doctorants, MoU ...) (Difficulté du multi disciplinaire : 'informatique', numérique, physique)

En plus pratique :

- Faire passer à l'échelle les applications existantes vers le Peta puis multi-Peta-scale et l'Exa-scale
- Garder la maintenabilité des codes
- Garder la validation des codes
- Garder l'attention sur les 'à coté' (IO, visualisation, ...) nouveaux goulets d'étranglement



Quelle machine pour quelle(s) application(s) ?

- **Network intensive** → IBM BG/P → Bluebrain (EPFL)
- **CPU intensive & Network intensive** → Large SPMD simulation → Cray
- **CPU intensive & assez peu couplé** – CFD → Cluster généraliste
- **CPU intensive & naturellement parallèle** - HEP @ATLAS (CERN) → Un grand nombre d'ordinateurs portables
- **CPU vectoriel et besoin très fort en bande passante mémoire (1 octet par flop)** → NEC (SX-ACE (big core))



Orientation : Co – design

Roadmaps différentes en fonction de l'industrie :

- Aéronautique : l'Exaflop n'est pas le but ultime au moins le Zetaflop pour LES avion. Beaucoup d'applications (simulations presque indépendante), calcul stationnaire (RANS, URANS) avec modèle physique amélioré
- Combustion : modélisation de la combustion turbulente, couplage multi physique, ...
- 'Engineering' : Optimisation, simulations type Monte Carlo, ...

Conclusion_(s)

- La consommation électrique est le moteur du changement
- Ce n'est plus la recherche qui pilote le marché des supercalculateurs
- L'Exascale est un challenge à tout les niveaux (langage, compilateurs, système (gestion batch, checkpoint-restart, ...)) **ET pour les APPLICATIONS**
- Les applications devront s'adapter **rapidement** dans un contexte encore flou
- L'approche à base de directives (standardisé) est sans doute à privilégier pour les grands codes (portabilité)
- Parallélisme multi-niveaux (disparition de la programmation homogène (pur MPI))
- Il faut éviter des solutions propriétaires (pour les grands codes)
- Il faudra utiliser des bibliothèques optimisées
- Le positionnement sera encore plus difficile pour les codes de recherche utilisés par l'industrie
- La performance disponible rebat les cartes de l'écosystème
- Le multi physique complexifie encore la route vers l'Exascale dans le cadre de simulation réaliste