

DE LA RECHERCHE À L'INDUSTRIE



# Apports d'Arcane pour Troll :

un code multiphysique 3D pour la  
simulation de plasmas créés par laser

*Rencontres Arcane du 05 mars 2020*

[cedric.enaux@cea.fr](mailto:cedric.enaux@cea.fr)

[www.cea.fr](http://www.cea.fr)

CEA/DAM/DIF, 91297 Arpajon, France

**TROLL = code d'hydrodynamique radiative 3D pour la FCI**

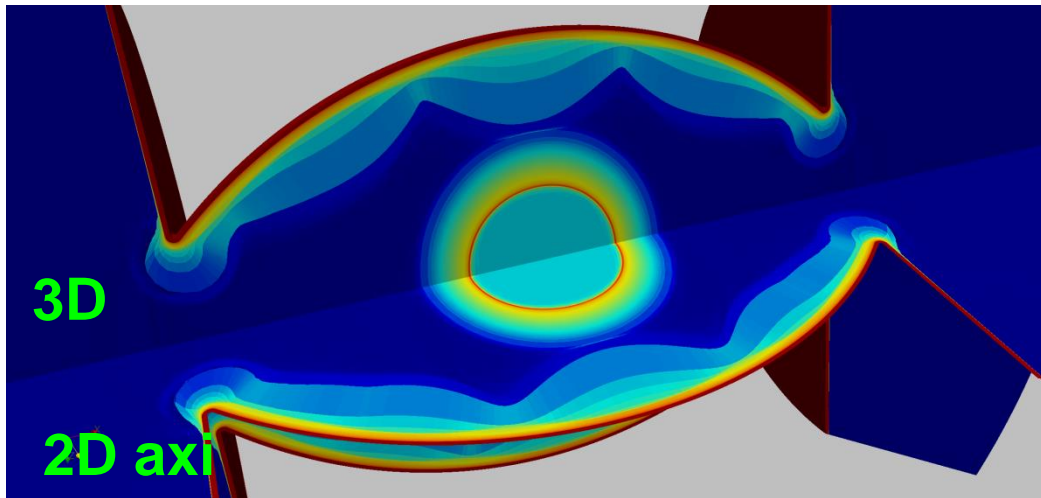
**Troll dans ses grandes lignes :**

- premiers développements il y a 20 ans
- pensé et codé 3D et parallèle dès le début
- s'appuie sur plateforme Arcane (partagée avec IFPEN depuis 2007)
- mobilise actuellement une quinzaine d'ingénieurs
- **aujourd'hui :**
  - ~ 1.5 millions de lignes de C++
  - ~ de nombreux utilisateurs en production
  - ~ 234 millions d'heures CPU consommées en 2019 sur Tera1000

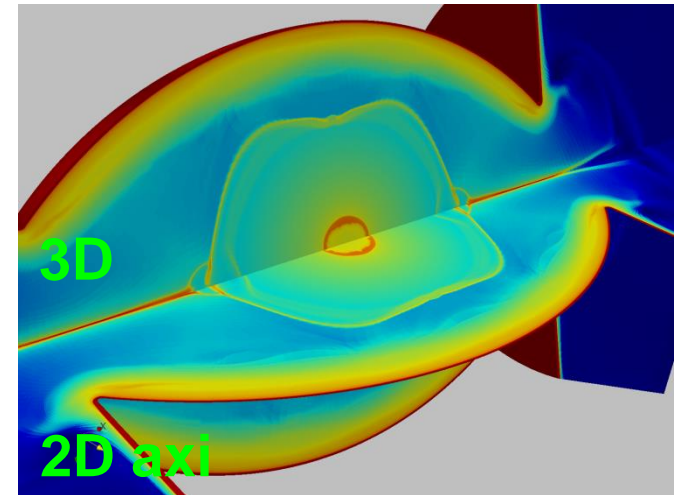


# Simulation Troll d'expérience FCI : validation du 3D dans sa limite 2D axisymétrique

Cartes de masse volumique à différents instants,  
qui montrent la dynamique de l'implosion de la capsule :

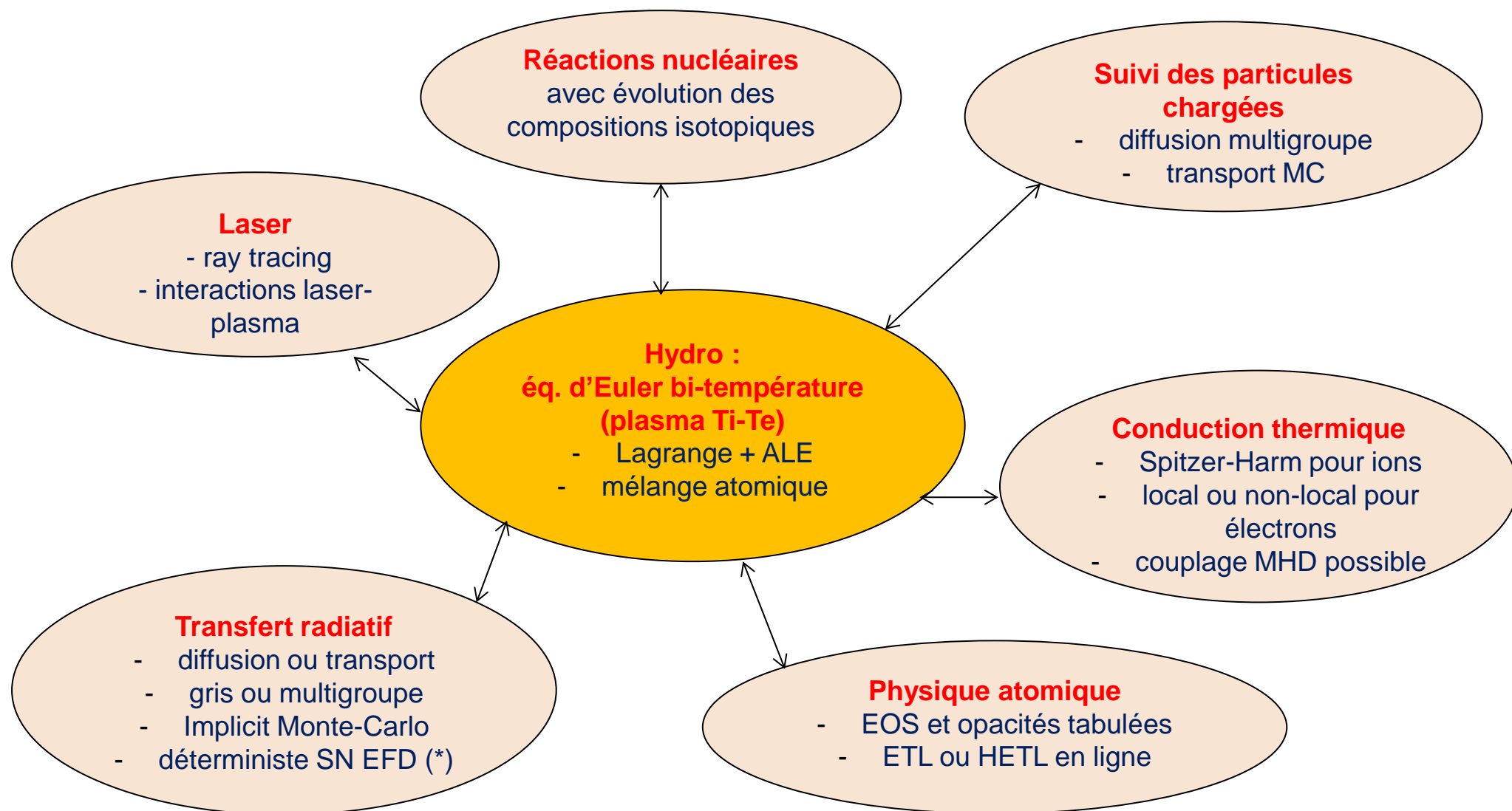


1ère phase du calcul :  
hydro Lagrange



2<sup>e</sup> phase du calcul :  
hydro ALE

# Panorama des modules du code Troll



(\*) = dév. en cours

## Configurations parallèles classiques des simulations 3D du code Troll :

- entre 64 et 512 sous-domaines (dépend de la finesse du maillage)
  - entre 1 et 4 réplicats pour ray tracing et transport Monte-Carlo
- => typiquement autour de 512-1024 processus.

Mais pour répondre à certaines questions spécifiques, des simulations « grand challenge » peuvent être menées. Par exemple :

## Caractéristiques d'une simulation 3D menée en 2019 avec Troll sur Tera1000-2 :

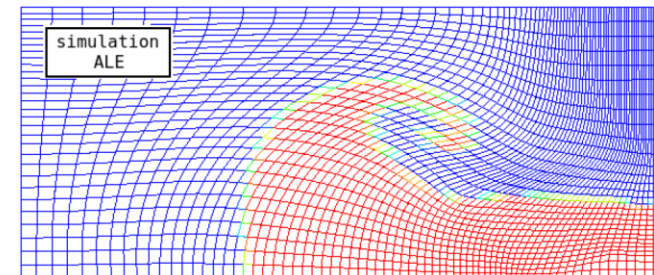
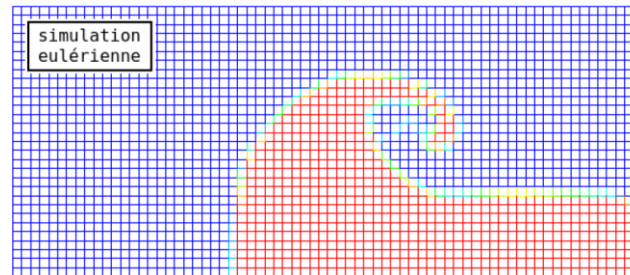
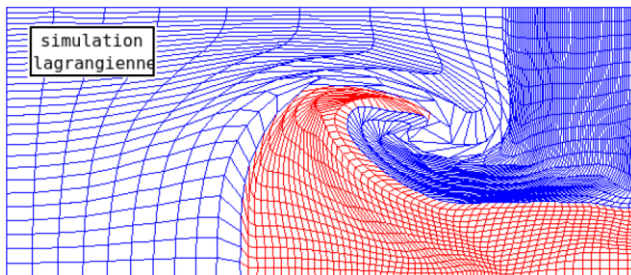
- 23 millions de mailles (calcul Lagrange + ALE),
- 4.1 milliards de particules Monte-Carlo,
- 16 384 sous-domaines (pas de réplication),
- 32 768 cœurs alloués, soit 512 nœuds,
- 24 millions d'heures de calcul,
- 25 personnes impliquées à divers niveaux.

## Principaux points appréciés par l'équipe Troll :

1) Arcane encaisse l'effort **d'adaptation à l'environnement** (machine et versions OS et MPI notamment), et facilite ainsi le portage du code Troll, de façon pérenne.

2) Arcane gère et encapsule tous les aspects de **gestion des maillages non-structurés (connectivité maillage)**, de façon **générique en 2D / 3D** :

-> cet aspect est au cœur du projet Troll, dont l'hydro est ALE



3) Arcane gère les structures des **matériaux/constituants (connectivité matériau)**

### Principaux points appréciés par l'équipe Troll :

4) Arcane gère les couches basses du **parallélisme** :

- encapsulation des appels MPI et/ou TBB bas niveau
- mise à disposition d'un « parallel\_manager » supportant :

**décomposition de domaines**

et/ou **réplication de domaines**

et/ou **multithreading** (incluant un mode répétable)

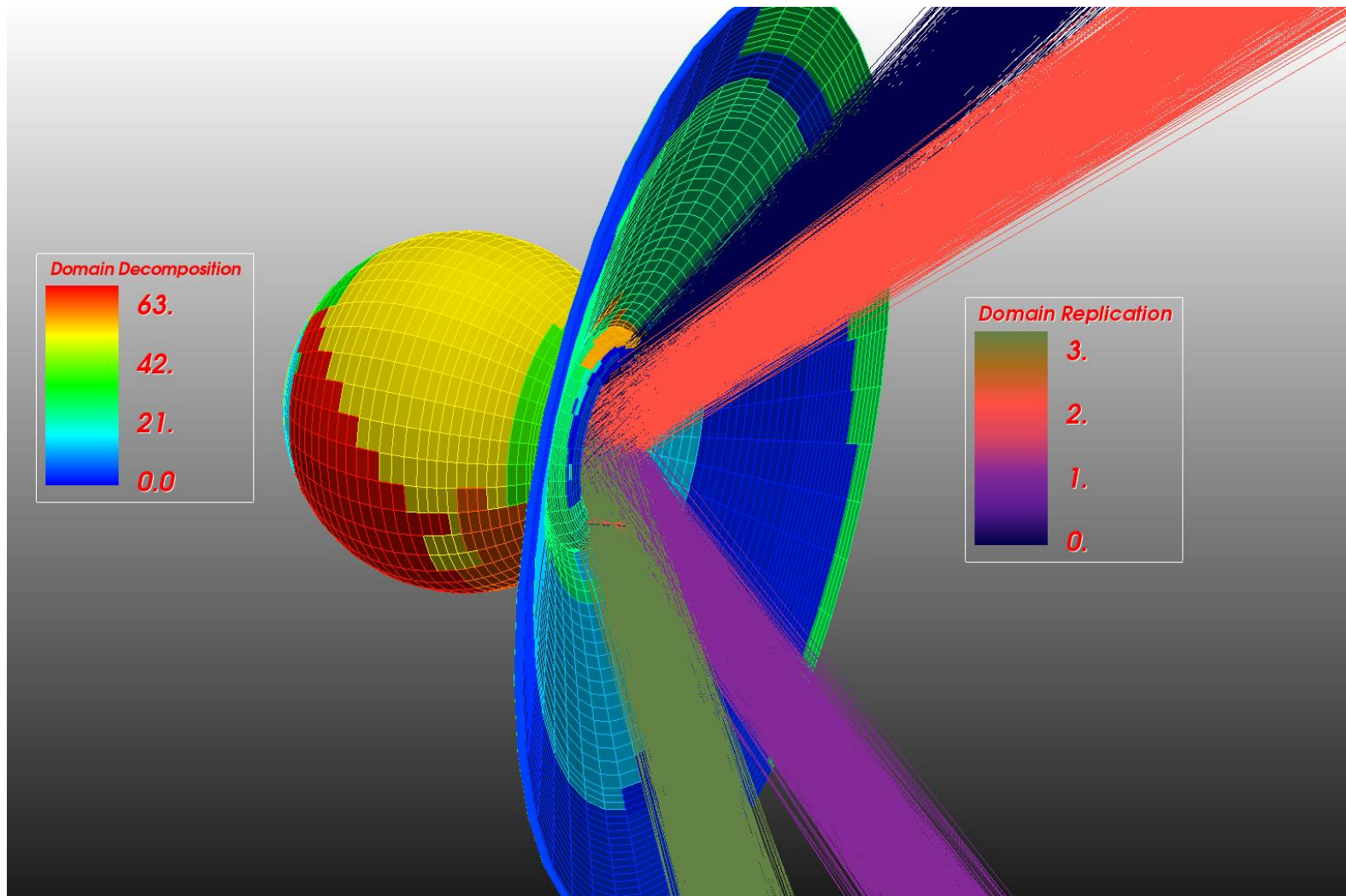
*Exemple : mise à jour d'une variable par simple .synchronize()*

### Remarques sur le parallélisme dans Troll :

- **MPI décomposition de domaines utilisé systématiquement**, pour tous les modules du code
- **MPI réplication de domaines utilisé optionnellement**, et seulement pour les modules de transport Monte-Carlo et ray tracing laser
- TBB multithreading déployé pour le transport Monte-Carlo, mais **très peu utilisé** car généralement moins efficace que MPI



# Exemple d'une simulation Troll avec 256 processus MPI : 64 sous-domaines x 4 réplicats



Répartition des sous-domaines et réplication pour le ray tracing



### Principaux points appréciés par l'équipe Troll :

#### 5) Arcane gère **l'équilibrage de charge** :

- partitionnement dynamique du maillage (via ParMetis)
- migration des données entre sous-domaines
- depuis Arcane 2.15 : mode répétable ajouté au service Metis

#### 6) Arcane fournit plusieurs **outils pour le développement et le debug** :

- trois modes d'exécution de Troll : release, check, debug
- outil stdenv\_verif : pour comparer 2 exécutions
- contrôle des débordements de tableaux (mode check)
- gestion des sorties listing (notamment par processeur, etc.)
- mécanisme de détection des Floating Point Exception (via initialisation à NaN de toutes les variables flottantes)
- etc.

7) Arcane, c'est aussi et surtout **une équipe** :

- dont le support et la réactivité sont appréciés par l'équipe Troll
- ouverte aux échanges et aux demandes d'évolution
- exemple avec plusieurs progrès récents impactant Troll : répétabilité (partitionneur, TBB), taille des messages MPI élargie ( $> 2^{31}$ ), etc.

### Autres remarques sur fonctionnalités d'Arcane :

- *Arcane inclut des mécanismes internes de protection/reprise, mais Troll ne les utilise pas (utilisation d'un format Hercule interne CEA). Idem pour le dépouillement.*
- *Les particules Monte-Carlo sont instanciées sous la forme d'un ensemble de tableaux (1 tableau aux particules par attribut), pour des raisons de performance. Rq : concept éloigné d'autres concepts existants, comme liste chaînée par exemple.*

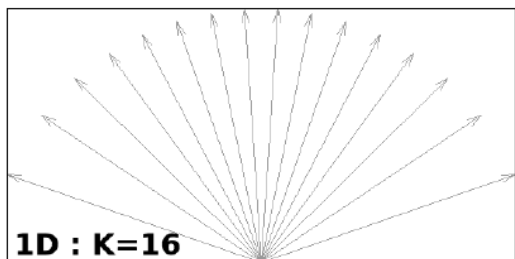
# Perspective sur le transport déterministe dans Troll (1/2)

	Transport (Monte-Carlo)	Diffusion (P1 hyperbolique)
<p><b>Milieu Opaque</b> (propagation lente et quasi-isotrope)</p> <p>Ici :</p> <ul style="list-style-type: none"> <li>- Monte-Carlo et approx. P1 sont en accord,</li> <li>- approx. P1 est recevable,</li> <li>- approx. P1 réduit le cout CPU.</li> </ul>		
<p><b>Milieu Transparent</b> (propagation rapide et directionnelle)</p> <p>Ici :</p> <ul style="list-style-type: none"> <li>- Monte-Carlo et approx. P1 sont en désaccord,</li> <li>- approx. P1 non recevable,</li> <li>- calcul transport nécessaire.</li> </ul>		

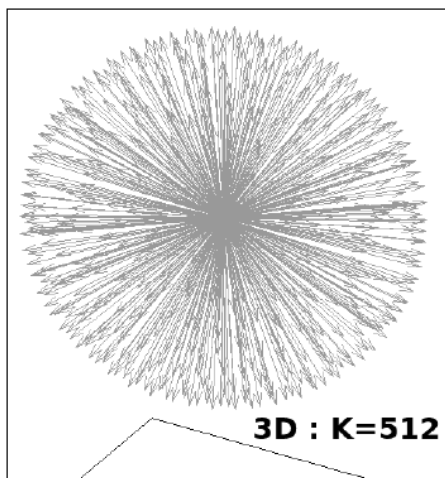
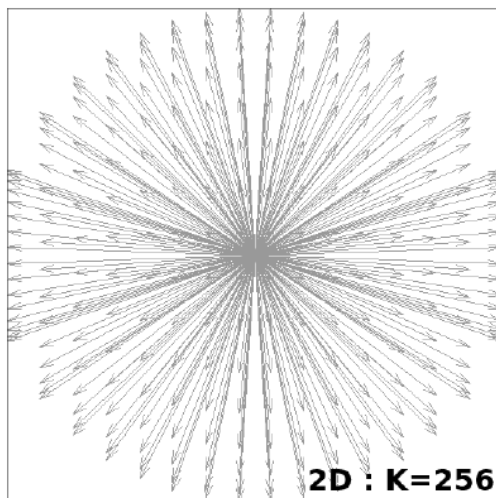
Un solveur de transport déterministe permet d'obtenir des **solutions de référence** (pas de bruit Monte-Carlo), mais son implémentation performante est un **challenge**

# Perspective sur le transport déterministe dans Troll (2/2)

**Sphère unité** approchée façon Ordonnées Discrètes, en 1D, 2D et 3D, à précision fixée :



**N=8**



*[dév. Troll en cours]*

**Nombre de variables**, par maille, pour représenter l'inconnue « intensité radiative » :

- 100 groupes de fréquence (environ),
  - multiplié par 100 directions angulaires (environ),
  - multiplié par 8 degrés de liberté en espace (éléments finis discontinus),
- => **80 000 réels** (double) par maille, dont le calcul résulte de 4 boucles imbriquées...