



POLYTECHNIQUE
MONTRÉAL

Les géométries du calcul de réseau

A. Hébert

2023/05/17

Table des matières

Introduction
Modèles
géométriques
R&D
Quadratures
Tracking
Conclusion
Ressources

Introduction
Modèles géométriques
R&D
Quadratures
Tracking
Conclusion
Ressources

Introduction

- Le traitement des géométries est un enjeu majeur en physique des réacteurs nucléaires.
- Nous allons présenter le traitement et l'utilisation des géométries présentes lors du **calcul de réseau**.
- Trois types de représentation géométriques sont couramment utilisées en France et au Canada
 1. Géométries gigognes (Apollo1/2/3, Dragon1/2/3/4/5, solveurs “coeur”)
 2. Géométries surfaciques (Apollo2/3, Dragon5)
 3. Géométries combinatoires (MCNP, Serpent, Tripoli, Moret)

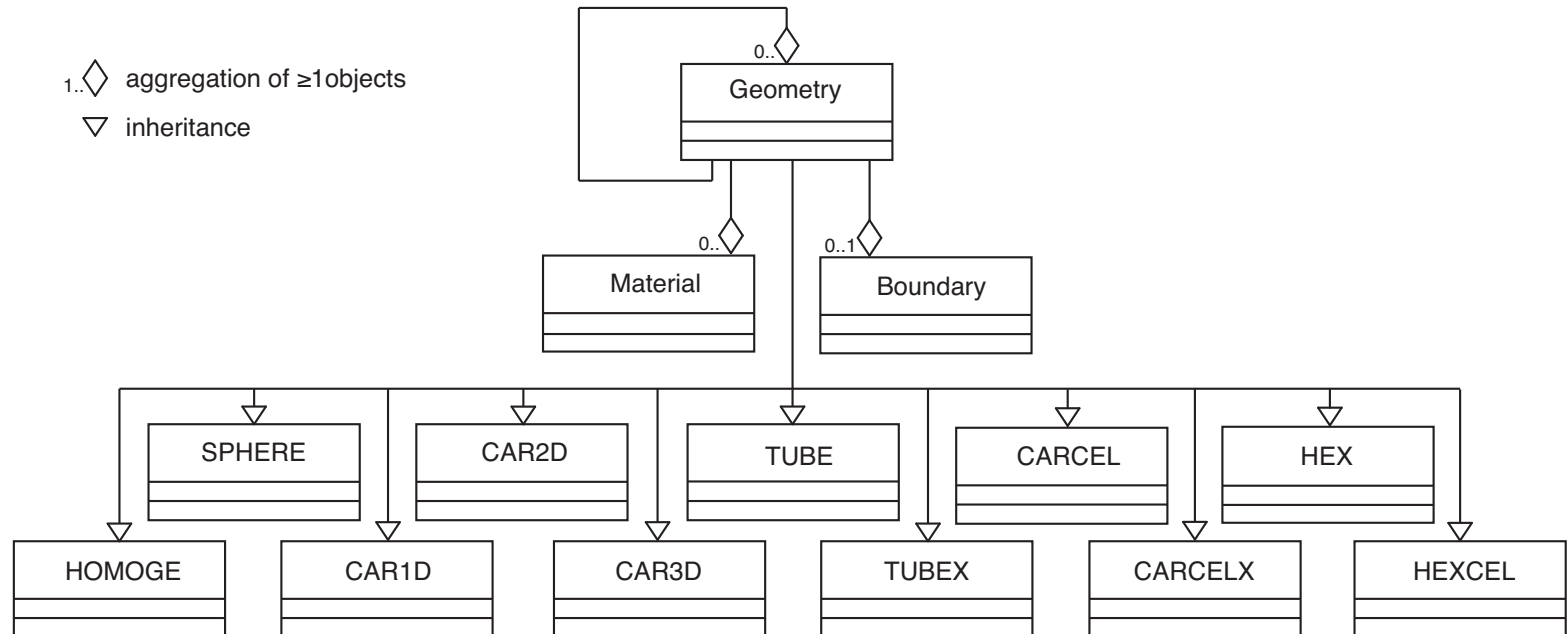
Définitions:

1. Une géométrie **native** est une géométrie définie avec les mot-clés-valeurs du code de réseau ou du code de coeur.
2. Une géométrie **non-native** est une géométrie définie avec un outil externe (Silène, Salomon, Charm ou Alamos).

Modèles géométriques

Géométries gigognes

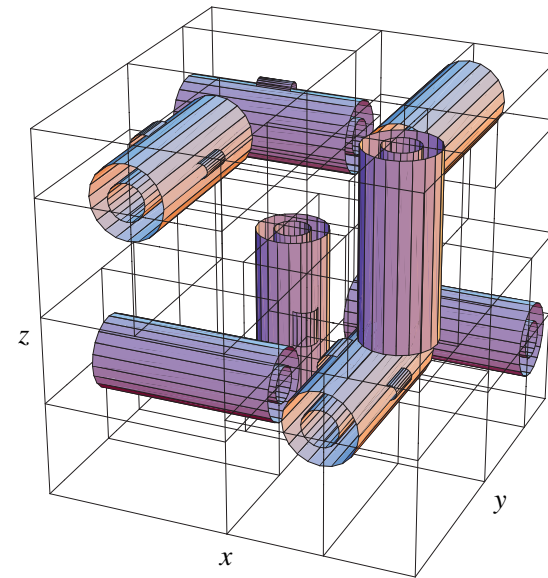
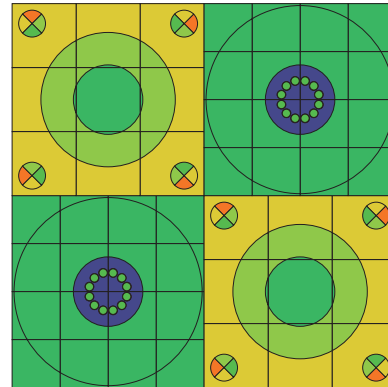
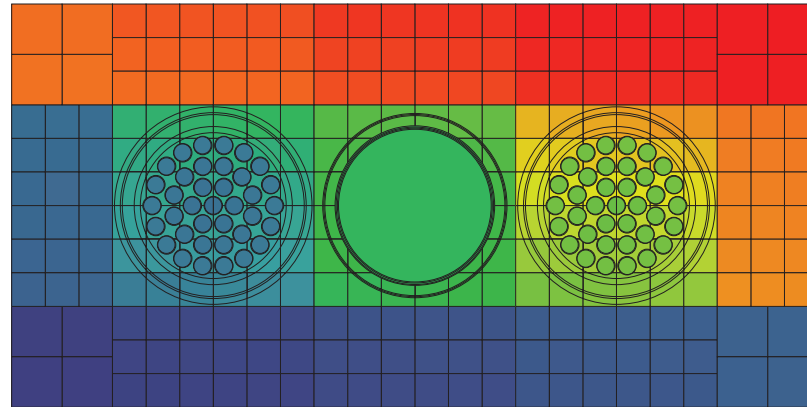
- Introduites par Zarko Stankovsky dans Apollo1 en 1979. Utilisées dans Dragon (voir diagramme UML).
- Idéales pour unifier les modèles géométriques des codes de réseau et de coeur-entier.
- Présentent des limitations pour représenter certains designs (boitiers d'assemblages hexagonaux, géométries RJH, géométries de réflecteur, etc.)
- Adaptées aux calculs 1D, 2D et 3D.
- Géométries potentiellement récursives.
- Géométries natives.



Modèles géométriques

Exemples de géométries gigognes

Colored by Region

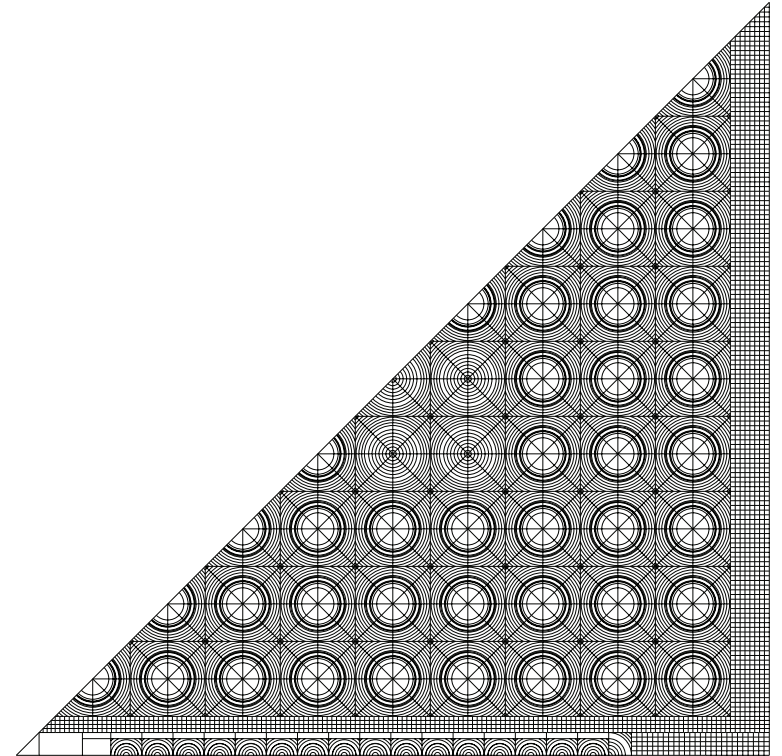
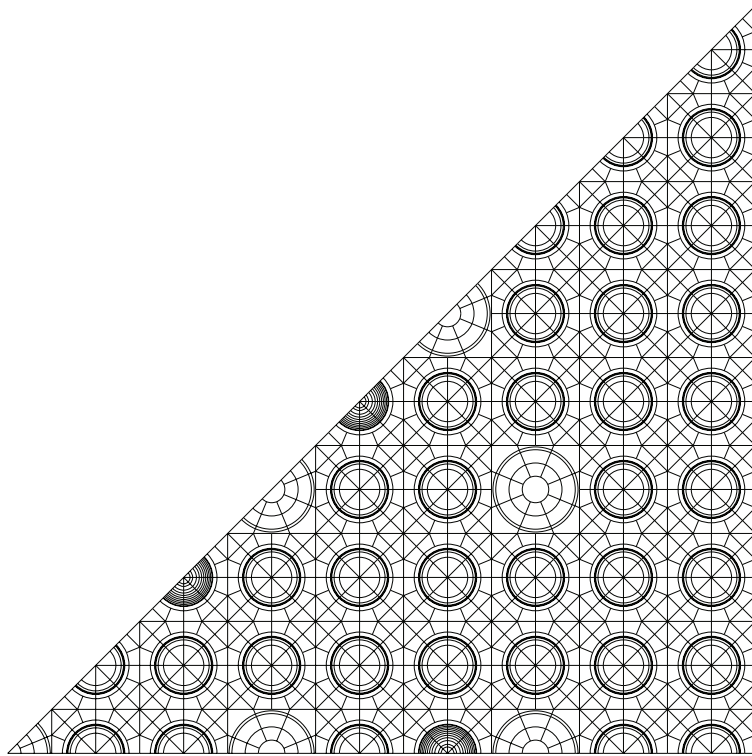


- Introduction
- Modèles géométriques
- R&D
- Quadratures
- Tracking
- Conclusion
- Ressources

Modèles géométriques

Exemples de géométries gigognes

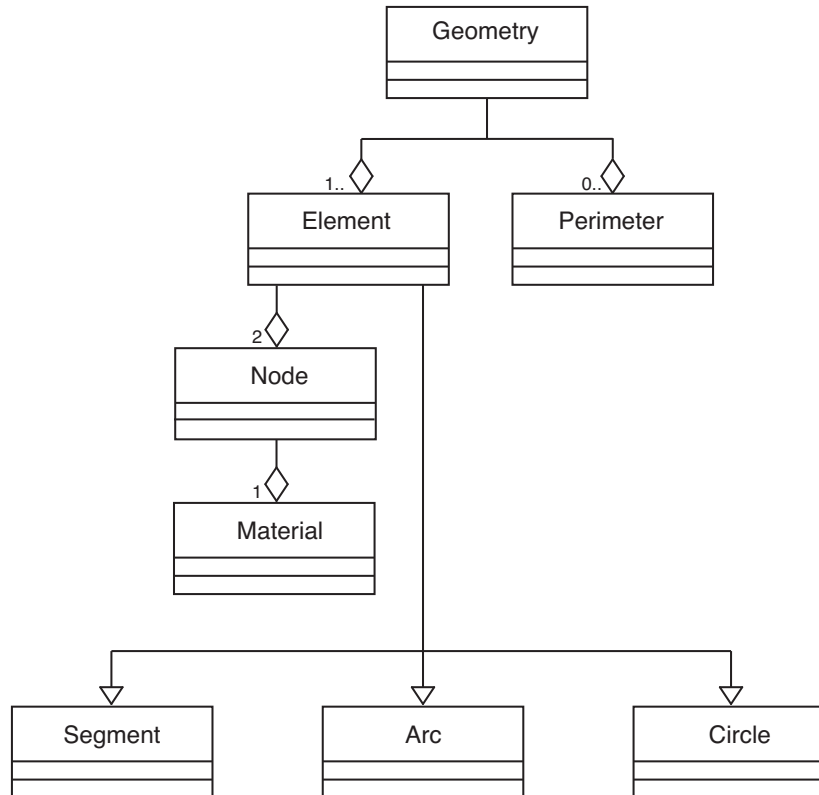
- Introduction
- Modèles géométriques**
- R&D
- Quadratures
- Tracking
- Conclusion
- Ressources



Modèles géométriques

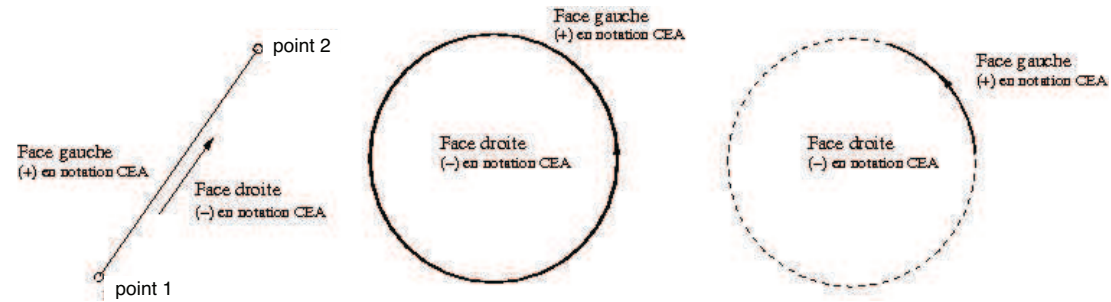
Géométries surfaciques

- Introduites par Richard Sanchez dans Apollo2 en 1993. Utilisées dans Dragon5 (voir diagramme UML).
- Limitées aux géométries 2D.
- Ne présentent aucune limitation pour aucun design 2D.
- Géométries natives (pour les plus simples) ou non-natives.
- Peuvent être décrites en langage PostScript directement.

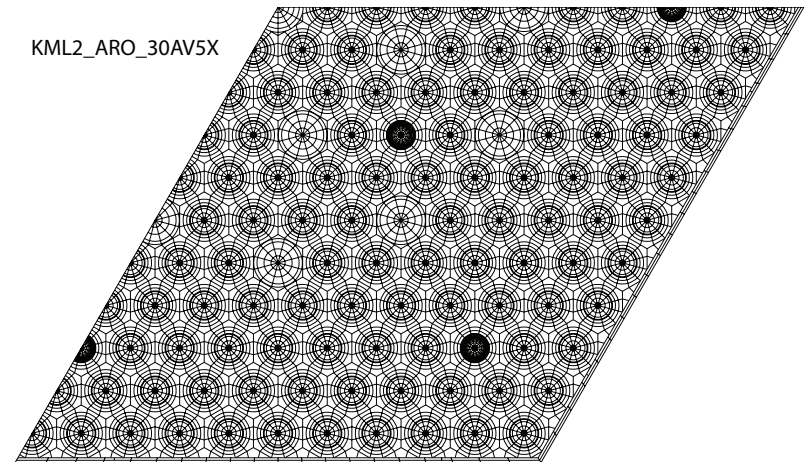
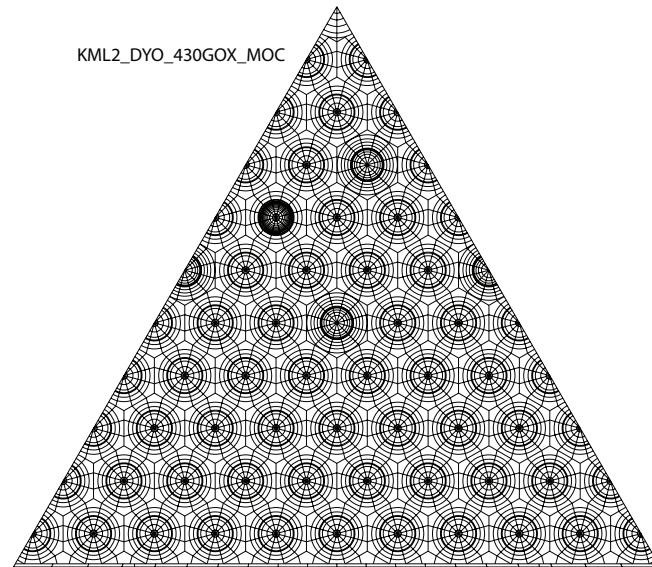


Modèles géométriques

Éléments surfaciques



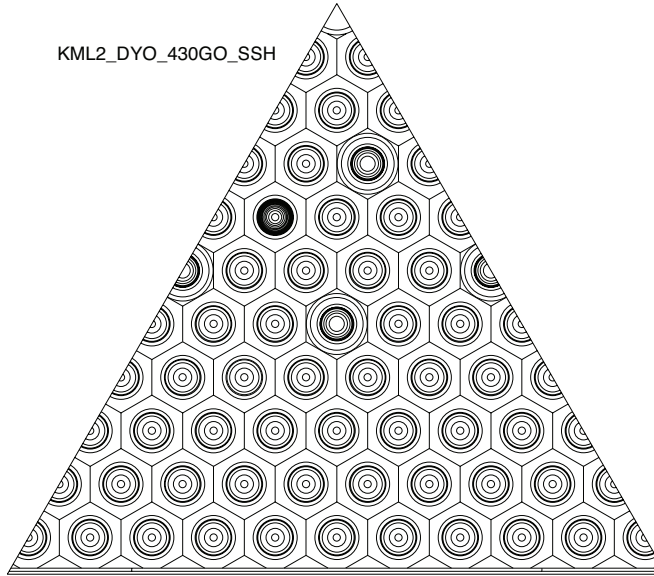
Exemples de géométries surfaciques



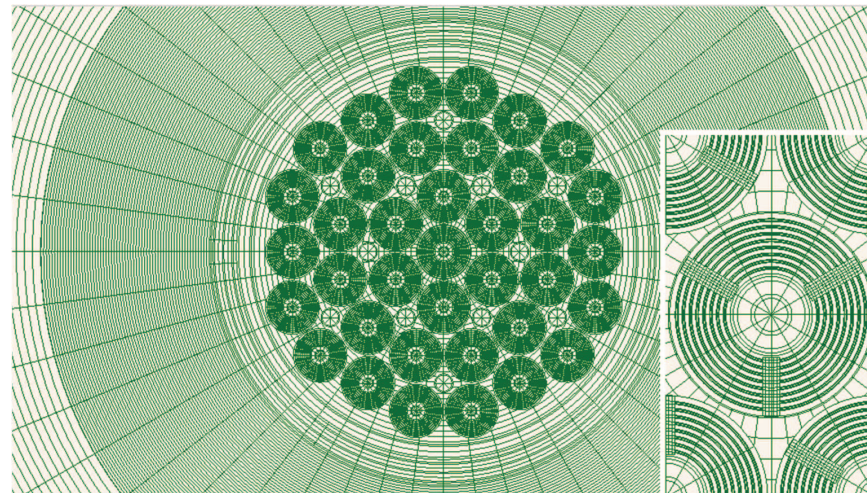
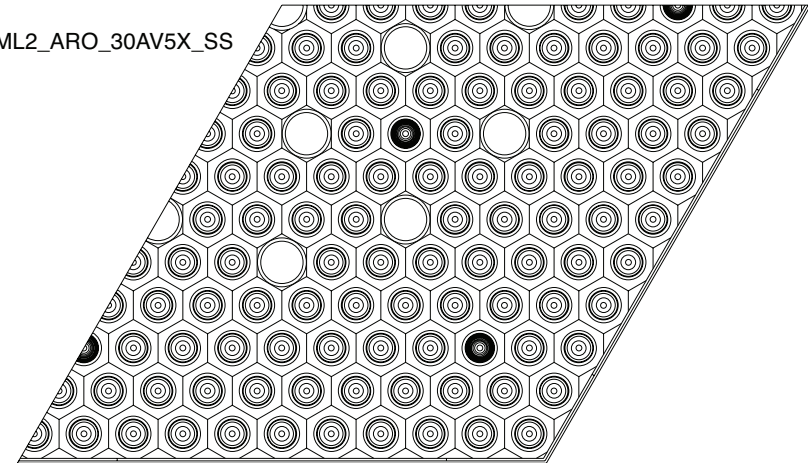
Modèles géométriques

Exemples de géométries surfaciques

KML2_DYO_430GO_SSH



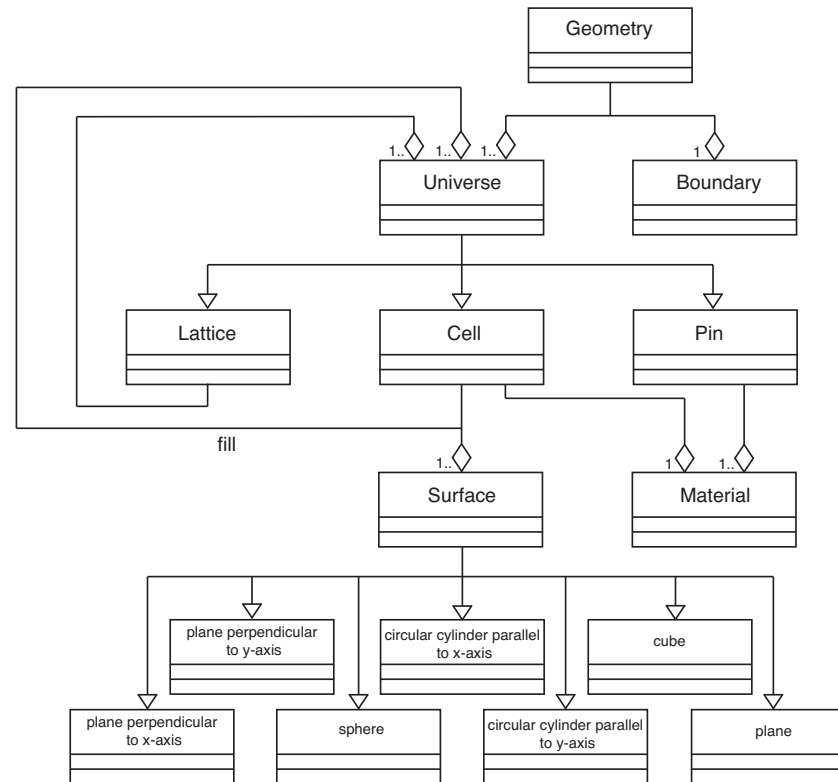
KML2_ARO_30AV5X_SS



Modèles géométriques

Géométries combinatoires

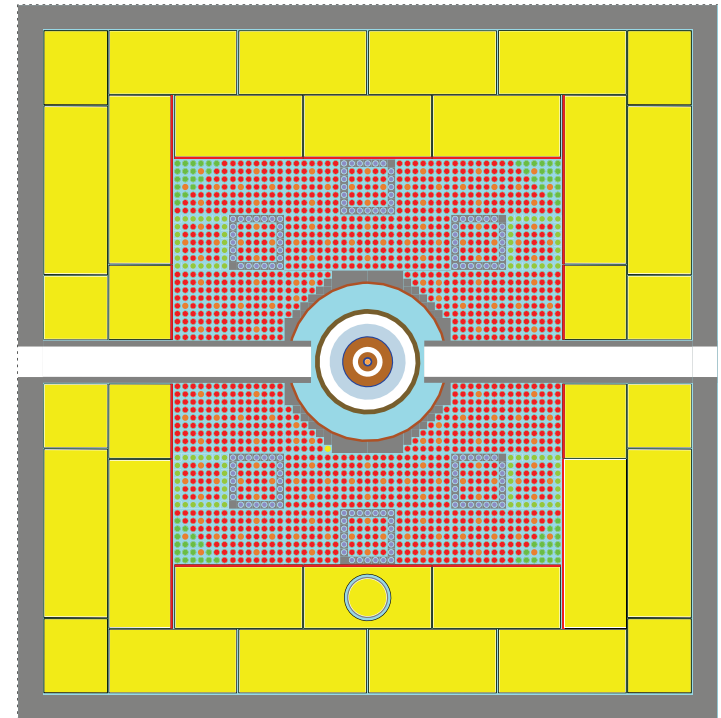
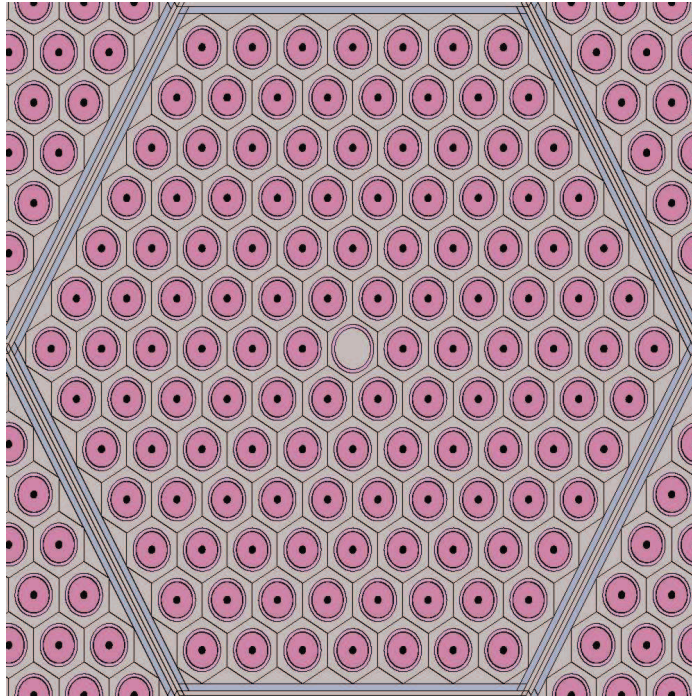
- Principalement utilisées avec les codes Monte-Carlo en 3D.
- La performance de calcul est sensible à la façon dont ces géométries sont décrites.
- Les surfaces peuvent être **combinées** à l'aide des opérateurs booléens & (intersection), | (union), et ~ (complément). Le code Serpent n'utilise que l'intersection (voir diagramme UML).
- ◆ Un point est à l'intérieur d'une **Cell** si et seulement s'il est à l'intérieur (ou à l'extérieur des **Surface** négatives) de toutes les **Surface** de la liste.



Modèles géométriques

Exemples de géométries combinatoires

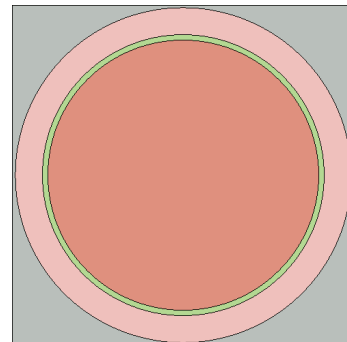
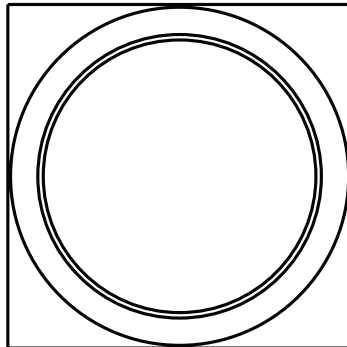
- Introduction
- Modèles géométriques**
- R&D
- Quadratures
- Tracking
- Conclusion
- Ressources



Modèles géométriques

Exemples de géométries combinatoires

- Peut être obtenue par la conversion d'une géométrie surfacique avec le module G2MC: de Dragon5.
 - ◆ Node \Rightarrow Cell et Element \Rightarrow Surface
 - ◆ La géométrie combinatoire obtenue est non-réursive.
- Peut être construite en suivant les instructions du guide d'utilisation Serpent \Rightarrow Le calcul sera plus rapide.



% Serpent combinatorial geometry generated by module G2MC:

% cell cards

```
cell 1 0 MIX_4 -8 7
cell 2 0 MIX_3 6 -7
cell 3 0 MIX_2 5 -6
cell 4 0 MIX_1 -5 -6
cell 5 0 outside 8
```

% surface cards

```
surf 5 cyl 0.63 0.63 0.50
surf 6 cyl 0.63 0.63 0.52
surf 7 cyl 0.63 0.63 0.62
surf 8 cuboid 0.0 1.26 0.0 1.26 -1.0E+10 1.0E+10
```

% Serpent combinatorial geometry:

```
pin 1
MIX_1 0.50
MIX_2 0.52
MIX_3 0.62
MIX_4
```

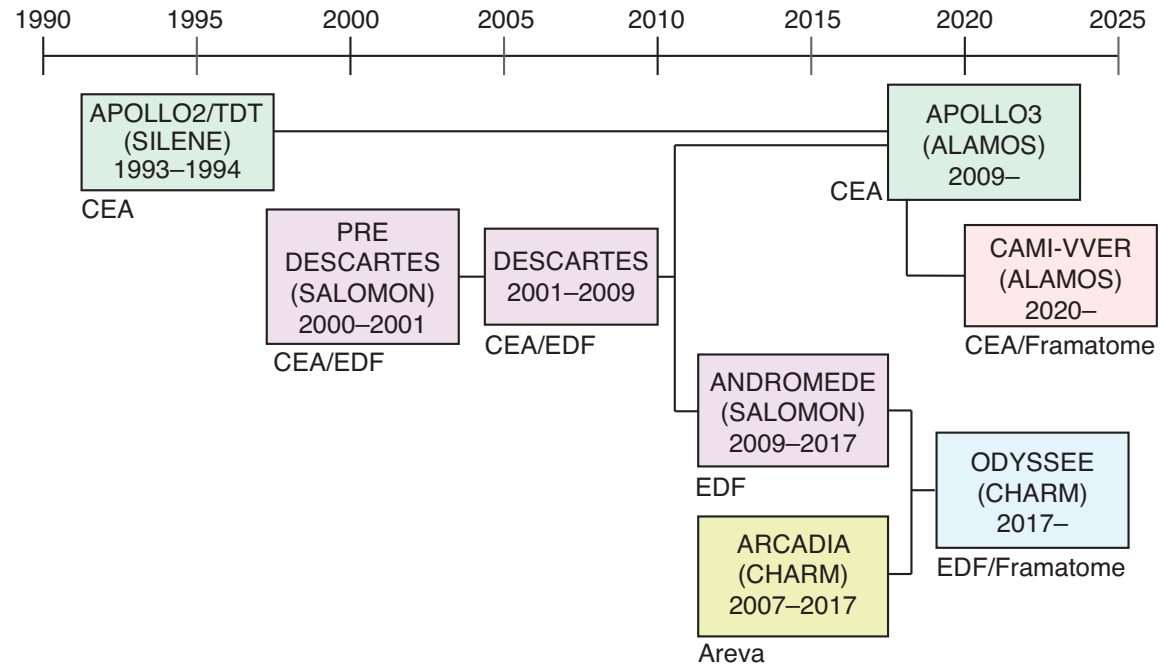
```
surf 10 sqc 0.0 0.0 0.63
cell 1 0 fill 1 -10
cell 2 0 outside 10
```


Le développement des géométries surfaciques pour le calcul de réseau est le résultat de décennies de R&D en France.

Introduction
Modèles
géométriques

R&D

Quadratures
Tracking
Conclusion
Ressources



Développement initial 1993–1994

Le développement initial des géométries surfaciques comporte deux composantes:

SILENE: Outil graphique de saisie d'une géométrie surfacique non-native. Silène a été programmé par Zarko Stankovsky en langage Java.

R. Sanchez and Z. Stankovsky, SILENE & TDT: A code for CP calculations in XY geometry, CEA-CONF-11953 (1993).

TDT: Outil de génération de trajectoires (tracking) d'une géométrie surfacique dans l'environnement APOLLO2. TDT a été programmé par Richard Sanchez, assisté d'une thésarde au Ph. D. (Naïma Lyoussi-Charrat).

N. Lyoussi-Charrat, Calcul de transport neutronique dans le code APOLLO2 par la méthode ..., thèse Ph. D. (1994).

- Cette contribution SILENE/TDT est limitée aux géométries non-natives.
- La contribution de Naïma Lyoussi-Charrat sur les probabilités de collision n'a pas été intégrée à APOLLO2. Par contre, la méthode des caractéristiques basée sur TDT a été implémentée.

Prototypage Pré-Descartes 2000–2001

- Le projet Pré-Descartes avait pour but de réaliser des prototypes rapides de solutions innovantes et audacieuses en amont du projet Descartes dont le début était fixé à Septembre 2021.
 - ◆ Un prototype rapide est une expérimentation informatique menée sur une courte période de temps et basée sur des composantes Open-Source (Salome, Python2, Dragon2).

Les analyses suivantes ont été menées:

1. Conception du prototype G2S : pour transformer une géométrie gigogne 2D en géométrie surfacique ou une géométrie surfacique en géométrie combinatoire.
 - \approx 13,000 lignes de Fortran-90
2. Conception du prototype SALT : pour effectuer un tracking 2D sur une géométrie surfacique. Ce tracking est compatible avec la méthode des probabilités de collision (PIJ) et avec la méthode des caractéristiques (MOC).

X. Warin, Notice théorique de la méthode des caractéristiques 2D et du générateur de trajectoires SALT (2002).

- \approx 7,000 lignes de Fortran-90

3. Conception du prototype SALOMON pour générer une géométrie surfacique non-native en utilisant le module GEOM de SALOME.

Y. Pora, Notice théorique de l'outil de création de géométries surfaciques basé sur SALOME (2001).

- \approx 1,900 lignes de Python2

Situation actuelle 2017–

1. La nouvelle interface graphique, ALAMOS, spécifiquement développée pour APOLLO3 avec une interface de programmation d'application (API) Python, est le successeur de SILÈNE.

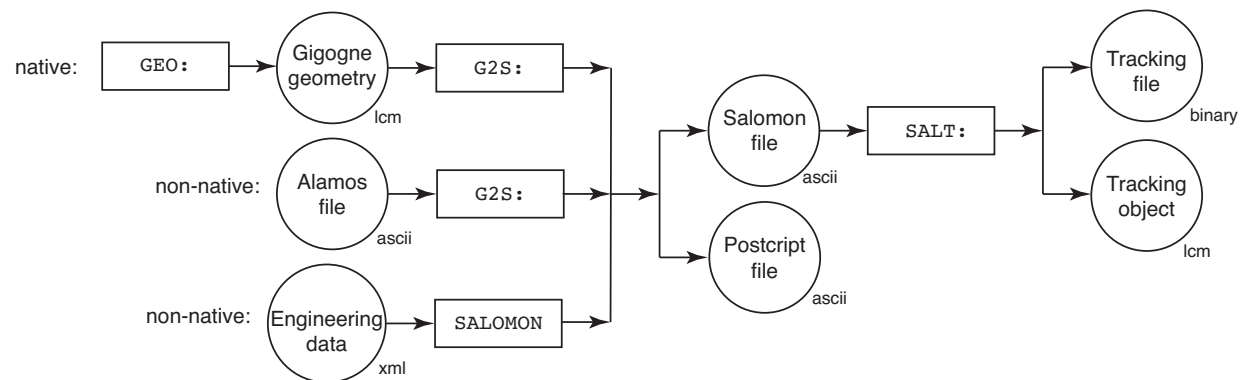
D. Tomatis *et al.*, Overview of SERMA's Graphical User Interfaces for Lattice Transport Calculations (2022).

2. Le code APOLLO3 permet de traiter les géométries surfaciques natives et non-natives (générées par SILÈNE ou par ALAMOS) avec la méthode des probabilités de collision (PIJ), avec la méthode des caractéristiques (MOC) ou avec la méthode aux ordonnées discrètes (solveur MINARET) si les éléments surfaciques sont des segments de droite.

J.-F. Vidal *et al.*, New reference Apollo3 calculation scheme for light water reactors, (2020).

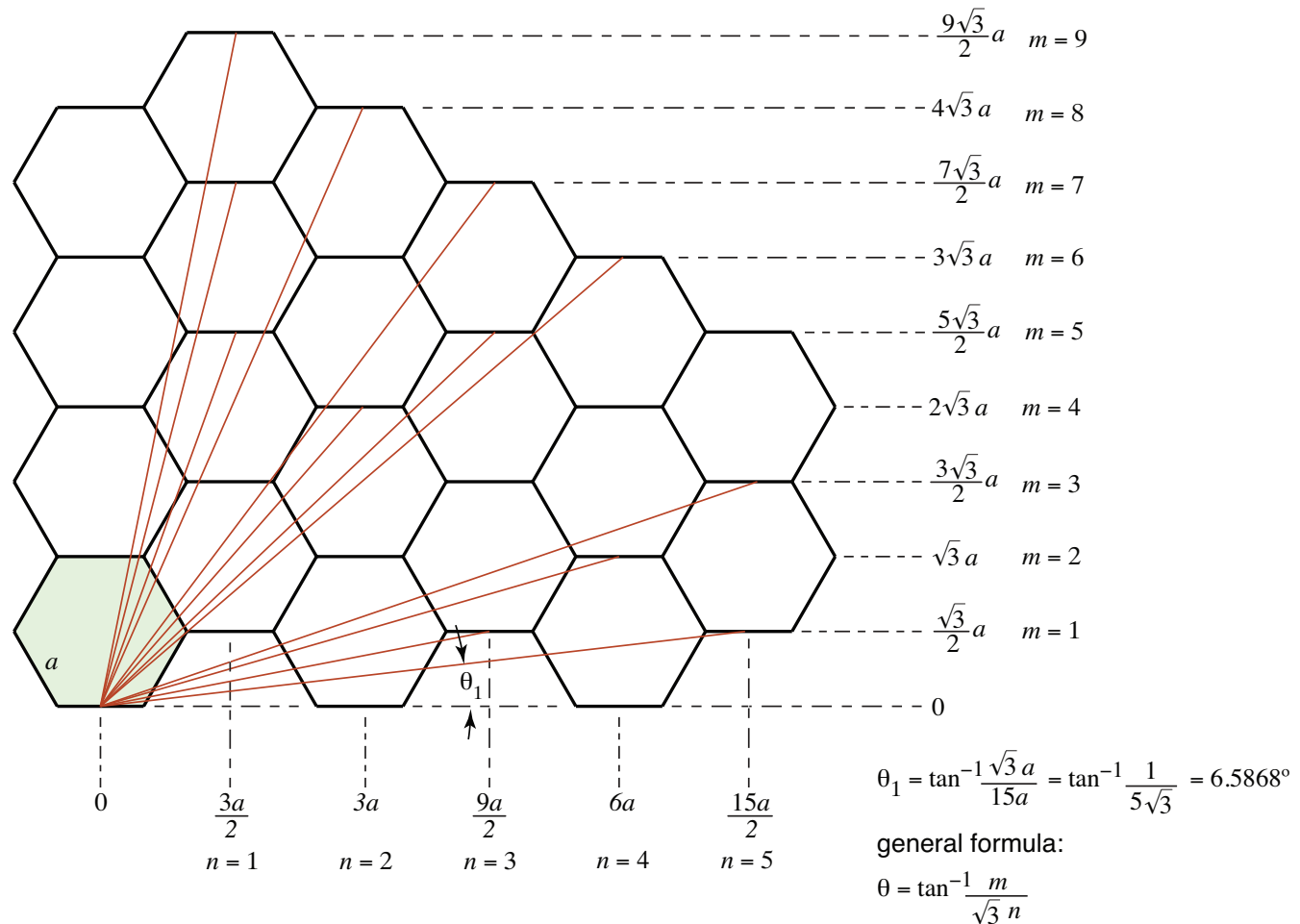
3. Le code DRAGON5 permet de traiter les géométries surfaciques natives et non-natives (générées par SALOMON ou par ALAMOS) avec la méthode des probabilités de collision (PIJ) ou avec la méthode des caractéristiques (MOC). De nouvelles symétries ont été ajoutées au module SALT : pour pouvoir traiter les assemblages VVER.

A. Hébert, DRAGON5 and DONJON5, the contribution of ÉPM to the SALOME platform, (2020).



Quadratures

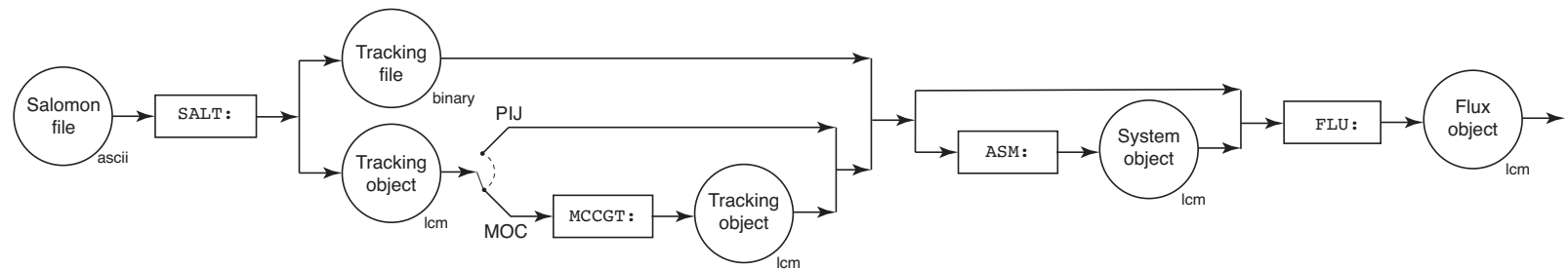
- Les quadratures **uniformes**, utilisées avec le mot clé TISO, sont semblables à celles utilisées avec une méthode aux ordonnées discrètes (SN).
- Les quadratures **cycliques**, utilisées avec le mot clé TSPC, sont déterminées de façon à représenter des caractéristiques de longueur infinies.



Tracking

Le calcul de flux neutronique par la méthode des probabilités de collision (PIJ) ou par la méthode des caractéristiques (MOC) est réalisée en plusieurs étapes:

1. Création d'une géométrie surfacique native ou non-native.
2. Tracking de la géométrie surfacique (par exemple, avec le module SALT: de Dragon5).
3. Vérification optionnelle du tracking (par exemple, avec le module TLM: de Dragon5).
4. Calcul des matrices de probabilités de collision ou calcul de l'information amont pour la méthode des caractéristiques.
5. Calcul d'autoprotection ou calcul de flux neutronique par la méthode des puissances.



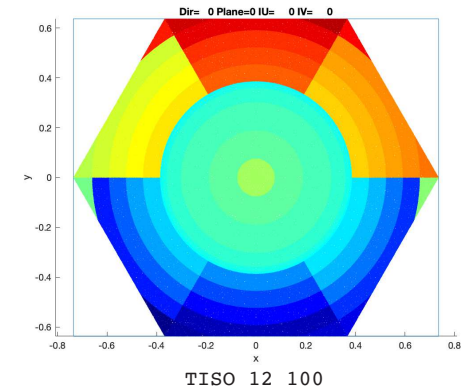
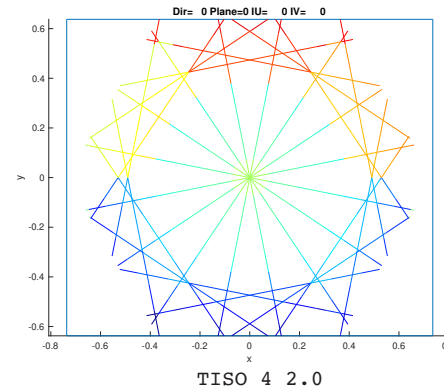
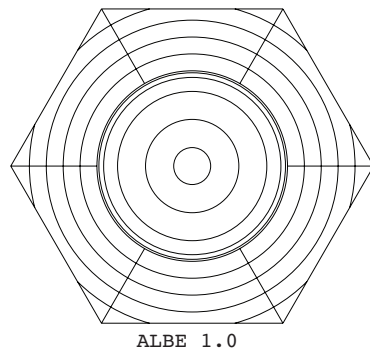
Définitions:

1. Le **domaine de calcul** est la géométrie surfacique définie à l'étape 1. Les conditions frontières de la classe `Perimeter` permettent de copier cette géométrie par réflexion spéculaire, rotation et/ou translation. L'imposition d'une réflexion isotrope (ou **blanche**) est possible, à condition que le côté ne coupe pas de crayons circulaires.
2. Une **track** est la projection d'une caractéristique (3D) sur le **domaine de calcul** (2D) en prenant soin de satisfaire les conditions frontières. Une track est **bidirectionnelle**.

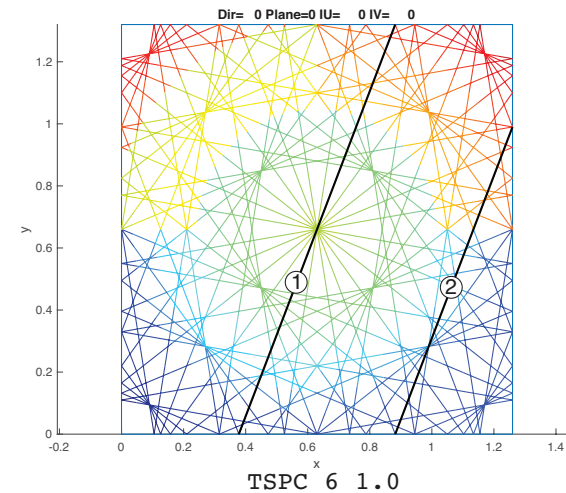
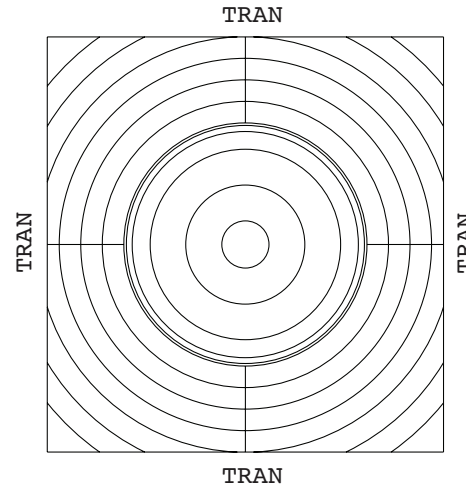
Tracking

Le module SALT : utilise un index TYPGEO afin d'adapter le type de tracking aux besoins de modélisation:

TYPGEO = 0: Un tracking **uniforme** est adapté aux conditions frontières VOID (vide) ou ALBE 1.0 (réflexion isotrope ou blanche avec albedo).



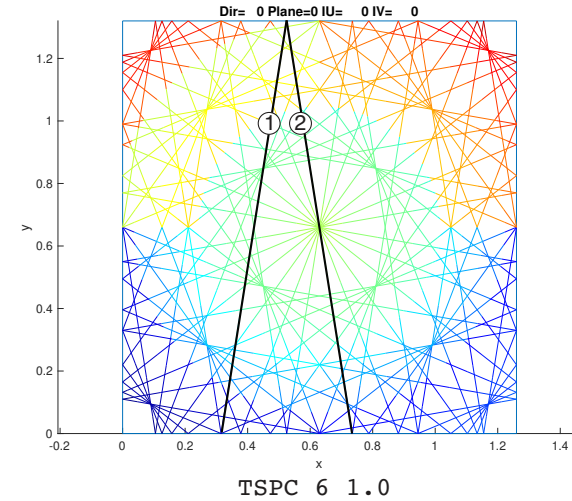
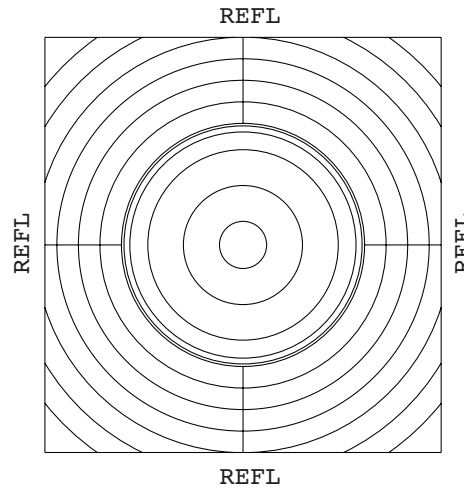
TYPGEO = 5: (**nouveau**) Un tracking **cyclique** est adapté aux conditions frontières TRAN (translation) pour une géométrie cartésienne.



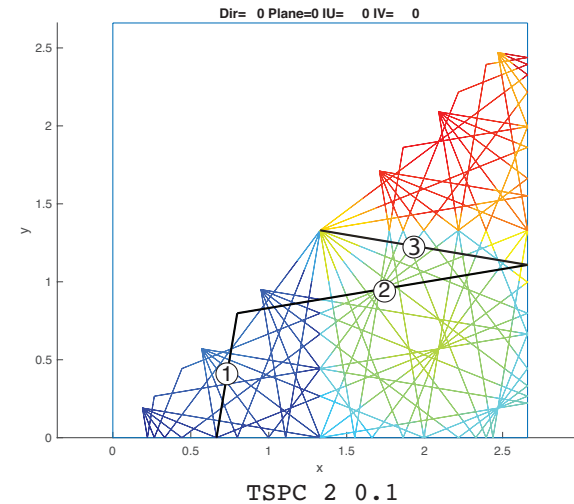
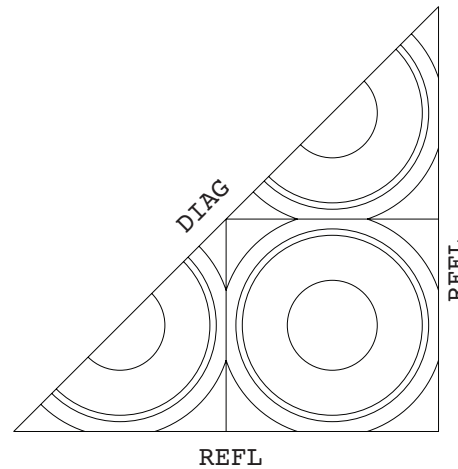
Tracking

- Introduction
- Modèles géométriques
- R&D
- Quadratures
- Tracking**
- Conclusion
- Ressources

TYPGEO = 6: Un tracking **cyclique** est adapté aux conditions frontières REFL (reflexion spéculaire) pour une géométrie cartésienne.

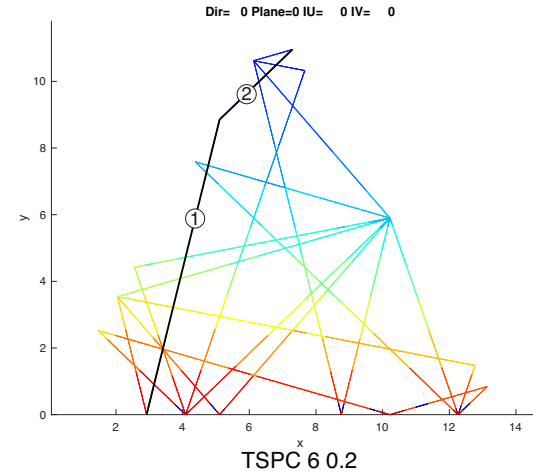
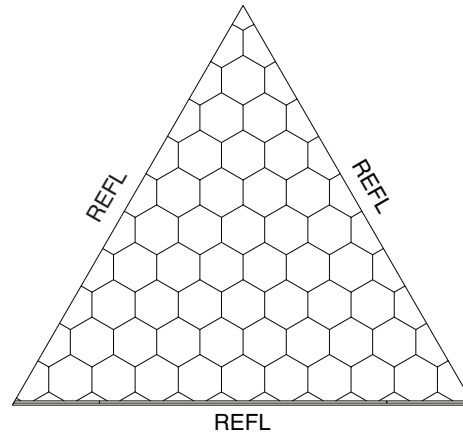


TYPGEO = 7: Un tracking **cyclique** est adapté aux conditions frontières REFL (reflexion spéculaire) pour une géométrie cartésienne de huitième d'assemblage.

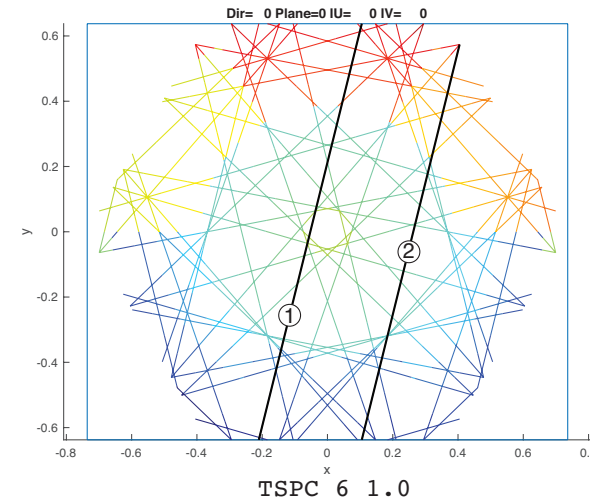
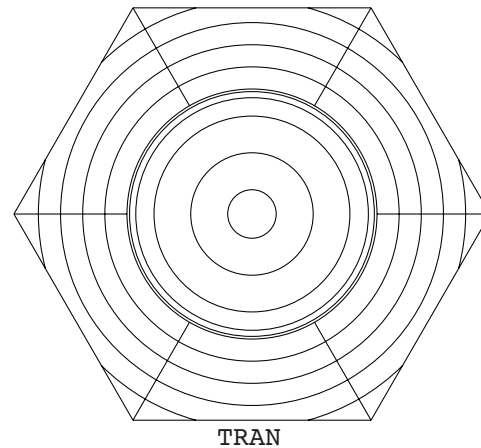


Tracking

TYPGEO = 8: Un tracking **cyclique** est adapté aux conditions frontières REFL (reflexion spéculaire) pour une géométrie de triangle isocèle SA60.

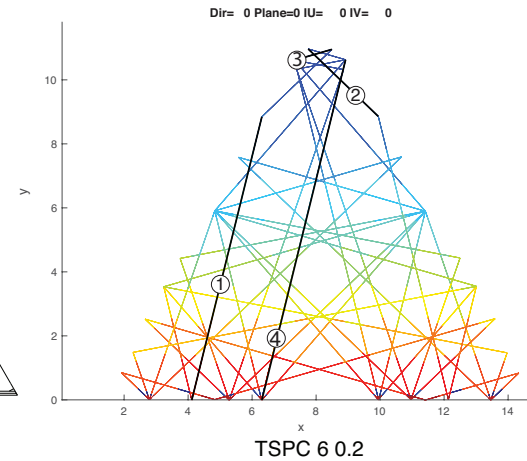
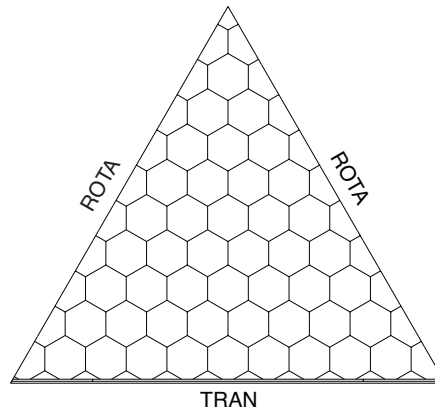


TYPGEO = 9: **(nouveau)** Un tracking **cyclique** est adapté aux conditions frontières TRAN (translation) pour une géométrie hexagonale COMP (complète).

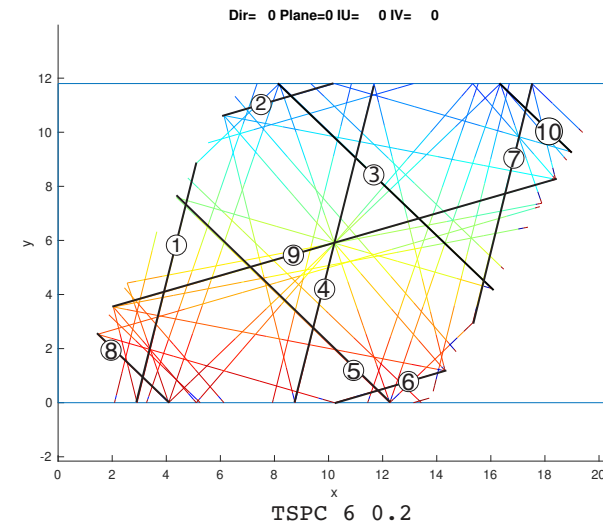
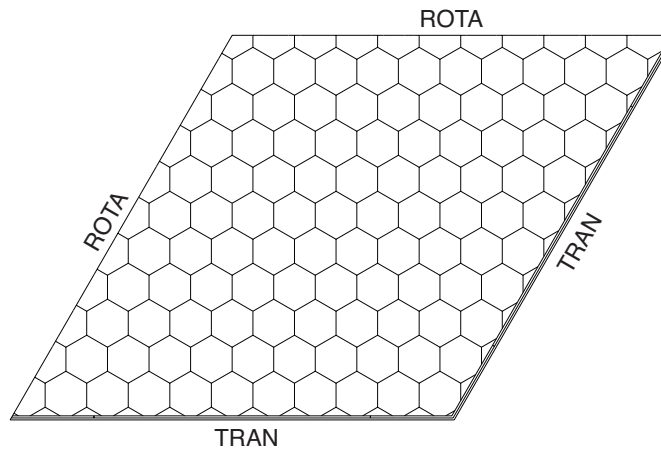


Tracking

TYPGEO = 10: (nouveau) Un tracking cyclique est adapté aux conditions frontières TRAN (translation) et ROTA (rotation de 60°) pour une géométrie de triangle isocèle RA60.



TYPGEO = 11: (nouveau) Un tracking cyclique est adapté aux conditions frontières TRAN (translation) et ROTA (rotation de 120°) pour une géométrie trapèze R120.



Conclusion

Le code Dragon5 offre la possibilité de traiter les géométries complexes des réacteurs ayant des designs variés (VVER, RJH, SMR, etc.). Il existe un besoin d'une solution Open Source pour pouvoir traiter ces géométries avec des calculs déterministes.

1. L'utilisation de géométries natives est facile, mais présente certaines restrictions pour des designs non-conventionnels.
2. L'utilisation de géométries non-natives est possible, mais Dragon5 n'offre pas d'outil semblable à ALAMOS.

Proposition de développement:

- Programmer un outil semblable à la version [pré-Descartes](#) de SALOMON pour pouvoir traiter les géométries de VVER.
- Baser cet outil sur Open CASCADE, afin de simplifier sa configuration. Open CASCADE donne accès à une boîte à outils de modélisation [B-Rep](#) (Boundary representation).

Open CASCADE Technology

■ Academic:

Alain Hébert (alain.hebert@polymtl.ca)

■ Merlin website:

DRAGON5/DONJON5: <http://merlin.polymtl.ca>

■ Archives website:

- ◆ Accès à l'information sur les développements Dragon5
- ◆ Autres contributions académiques

<http://merlin.polymtl.ca/archives.htm>

■ Textbook:

A. Hébert, Applied Reactor Physics, Third Edition, PIP, 2020.

