



POLYTECHNIQUE  
MONTRÉAL

# Schémas de calculs en physique des réacteur

A. Hébert

2017/11/20

# Plan de l'exposé

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

Schémas de calcul  
Distributions de l'EPM  
Plan d'assurance qualité particulier  
Le système PyNjoy  
Les codes DRAGON5 et DONJON5  
Le langage CLE-2000  
L'environnement APOLLO2  
L'environnement APOLLO3  
Schémas REP double-niveaux  
Résultats à burnup nul

# Schémas de calcul

## Schémas de calcul

Distributions de l'EPM  
Plan d'assurance qualité particulier  
Le système PyNjoy  
Les codes DRAGON5 et DONJON5  
Le langage CLE-2000  
L'environnement APOLLO2  
L'environnement APOLLO3  
Schémas REP double-niveaux  
Résultats à burnup nul

Un calcul basé sur un système de code tel que DRAGON5/DONJON5, APOLLO2/CRONOS2 ou APOLLO3 nécessite

- Des logiciels programmés dans un langage de haut niveau et maintenus sous AQ (système de contrôle de version, système de fiches d'interventions, tests de non-régression)
  - ◆ DRAGON5/DONJON5: ANSI-C et Fortran-2003
  - ◆ APOLLO2/CRONOS2: ÉSOPE et Fortran-77
  - ◆ APOLLO3: C++ et Fortran-2003
- Des **schémas de calcul** programmés dans un langage de script intermédiaire permettant d'automatiser des enchainements de calculs typiques d'une *utilisation industrielle* des logiciels
  - ◆ DRAGON5/DONJON5: CLE-2000
  - ◆ APOLLO2/CRONOS2: Gibiane
  - ◆ APOLLO3: C++ et/ou Python (architecture UML de type Descartes)
- Des données spécifiques à une utilisation industrielle précise. Par exemple:
  - ◆ Calcul d'un assemblage  $17 \times 17$  UOX avec grappe AIC insérée, en évolution avec prise en compte de contre-réactions
  - ◆ Calcul d'un coeur complet de réacteur avec rechargement combustible sur plusieurs cycles et prise en compte des contre-réactions en régime permanent.
  - ◆ Calcul des sections efficaces du réacteur CABRI en prévision d'un calcul transitoire.

# Distributions de l'EPM

Schémas de calcul

Distributions de  
l'EPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux  
Résultats à burnup  
nul

L'École Polytechnique de Montréal (EPM) met à disposition une suite de logiciels en physique des réacteur.

- **PyNjoy 2012:** Système basé sur le code NJOY-2012 de Los Alamos National Laboratory (LANL) pour le traitement des évaluations de données nucléaires et la production de bibliothèques isotopiques pour le code DRAGON.
- **DRAGON5:** Code de réseau pour les calculs en mode fondamental.
- **DONJON5:** Code de simulation coeur entier.

Ces distributions sont disponibles sur le site <http://www.polymtl.ca/merlin/>:

## About Merlin

*\$Updated: 2013/07/27\$*



## The Merlin page

The Merlin page is a collection of resources related to nuclear reactor physics, including computer codes Dragon and Donjon.

1. [The Version4 Dragon/Donjon distribution.](#)

Version4 is a complete and consistent distribution of the legacy Dragon and Donjon codes. This distribution is generally compiled in 32-bit mode.

2. [The Version5 Dragon/Donjon distribution.](#)

Version5 is a 64-bit clean distribution of the Dragon and Donjon codes.

3. [The PyNjoy 2012 system.](#)

PyNjoy 2012 is a cross-section library production system build around the Njoy 2012 code from Los Alamos National Laboratory.

4. [The Reactor Physics Archive webpage.](#)

This page provides a link toward technical reports, student dissertations, computational schemes and sample data files written by graduate students over years.



Schémas de calcul

Distributions de  
l'EPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux  
Résultats à burnup

nul

Les codes DRAGON5 et DONJON5 sont disponibles sur la page [Version5](#):

## What is the Version5 project

Version5 is a 64-bit clean distribution of the reactor physics codes developed at École Polytechnique de Montréal. [1, 2] Release 5.0.1 is iso-functional with Version4 distribution at release 4.1.0. The Ganlib5 kernel is mostly programmed in ANSI-C language and the computational modules are programmed in Fortran 2003. An introduction of reactor physics is presented in Ref. [3]. Other frequently asked questions are answered in our [faq](#) page.



## Version5 components

The available components in Version5 distribution are:

- Ganlib5 developer's guide ([PDF](#))
- Dragon5 user's guide ([PDF](#))
- Donjon5 user's guide ([PDF](#))
- Dragon/Trivac object structure guide ([PDF](#))
- CLE-2000 tutorial ([PDF](#))
- NXT module theory guide (geometry numbering) ([PDF](#))
- Version5 beta archive. To expand the archive, type "tar xvfz version5\_v5.0.1.tgz".

tagged version 5.0.1	<a href="#">tgz</a>	2014/12/17	
tagged version 5.0.2	<a href="#">tgz</a>	2016/02/02	<a href="#">what's new</a>
tagged version 5.0.3	<a href="#">tgz</a>	2017/02/24	<a href="#">what's new</a>

- Open-source Draglibs in XMAS or SHEM binary formats. If you want to download such a multigroup cross-section library in Draglib format, please go [here](#).
- Open-source Wimlib libraries from the [WLUP](#) project. These libraries are distributed in ascii format. To convert them to binary, you have to use the willie utility, available in [for.src](#).
- C++ classes
  - [Skin++ classes in Version5](#)
  - [Yacs++ classes in Version5](#)

## Development versions

If you need an alpha or beta development version set between two tagged versions, please go [here](#).

## Issue report form

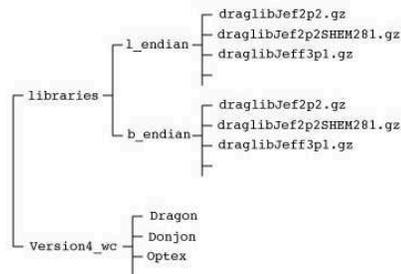
If you want to issue a bug report, an assistance request or a development suggestion, please go [here](#).

Les bibliothèques d'entrée au code DRAGON sont disponibles sur la page [Draglib](#):

## Draglib Download Page

*\$Updated: 2008/05/26\$*

Draglib files on this page are open-source cross-section libraries compatible with Dragon Version 3.06 (and upward) and Dragon Version 4. They can be downloaded from this page and laid in a directory structure similar to



Open-source libraries are available in various formats. Libraries from Jef 2.2, ENDFB-VI rel. 8, Jeff 3.1, ENDFB-VII rel. 0/1 and Jeff 3.1.x evaluations were processed using NJOY-99. Libraries from Jeff 3.2 evaluation were processed using NJOY-2012. Np-237 in Jeff 3.1 libraries was taken from Jeff 3.0. Open-source Draglibs are available with the following energy meshes:

- XMAS-172 is a legacy energy mesh defined jointly by UK and France. It is used in many countries.
- SHEM-281 is an energy mesh defined by Alain Santamarina at CEA (France). It is used in REL-2005 computational schemes adopted at CEA and EDF.[1]
- SHEM-361 is an energy mesh defined by Alain Santamarina and Alain Hébert and obtained by refining the group structure of SHEM-281 in the resolved energy domain, above 22.5eV.[2]
- SHEM-295 is an energy mesh defined by Alain Hébert at École Polytechnique de Montréal. Is obtained from SHEM-361 by defining coarse energy groups between 4.6 and 22.5 eV.
- SHEM-315 is an energy mesh defined by Alain Hébert at École Polytechnique de Montréal. Is obtained from SHEM-295 by refining the energy groups between 1 and 10 keV and by moving some energy limits so that SHEM-315 is a superset of the ERANOS 33-group mesh.

Draglibs exist in the following formats:

- Little-endian libraries. These libraries are used on Intel, AMD and Alpha systems.

	XMAS-172	SHEM-281	SHEM-295	SHEM-361	SHEM-315
Jef 2.2	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	
ENDFB-VI rel. 8	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	
Jeff 3.1	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	<a href="#">gz</a>	
ENDFB-VII rel. 0	<a href="#">gz</a>				
ENDFB-VII rel. 1	<a href="#">gz</a>				
Jeff 3.1.1	<a href="#">gz</a>				
Jeff 3.1.2	<a href="#">gz</a>				
Jeff 3.2	<a href="#">gz</a>				

# Distributions de l'EPM

Des version de développement intermédiaires des codes DRAGON5/DONJON5 sont disponibles sur la page [Development](#):

## Version5 development page

*\$Updated: 2014/12/19\$*

### Download tables

- Version5 alpha development versions. These distributions are incremental versions set during the initial development of the code. To expand the archive, type "`tar xvfz Version5.0.0_ev532.tgz`".

development version 5.0.0 (ev532)	<a href="#">tgz</a>	2013/07/27
development version 5.0.0 (ev718)	<a href="#">tgz</a>	2014/07/05
development version 5.0.0 (ev804)	<a href="#">tgz</a>	2014/09/02
development version 5.0.0 (ev903)	<a href="#">tgz</a>	2014/12/15

- Version5 beta development versions. These distributions are incremental versions set between official tagged versions. To expand the archive, type "`tar xvfz Version5.0.1_ev49.tgz`".

development version 5.0.1 (ev49)	<a href="#">tgz</a>	2015/02/08
development version 5.0.1 (ev90)	<a href="#">tgz</a>	2015/03/22
development version 5.0.1 (ev136)	<a href="#">tgz</a>	2015/06/09
development version 5.0.1 (ev176)	<a href="#">tgz</a>	2015/07/10
development version 5.0.1 (ev349)	<a href="#">tgz</a>	2015/10/12
development version 5.0.1 (ev399)	<a href="#">tgz</a>	2015/11/27
development version 5.0.2 (ev714)	<a href="#">tgz</a>	2016/11/08
development version 5.0.2 (ev750)	<a href="#">tgz</a>	2016/12/17
development version 5.0.2 (ev753)	<a href="#">tgz</a>	2016/12/21
development version 5.0.3 (ev812)	<a href="#">tgz</a>	2017/03/20
development version 5.0.3 (ev935) <sup>1</sup>	<a href="#">tgz</a>	2017/08/14
development version 5.0.3 (ev969)	<a href="#">tgz</a>	2017/10/02

<sup>1</sup>: The new Microlib and Multicompo specifications are implemented in versions ev816 and above.

Schémas de calcul

Distributions de  
l'ÉPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux  
Résultats à burnup  
nul

## Politique d'utilisation

- **L'ÉPM ne propose pas de schémas de calcul sous AQ.** C'est la responsabilité des organismes utilisateurs de les développer.
- Les schémas de calcul développés en langage CLE-2000 restent la propriété des organismes qui les développent
- Les codes ÉPM sont développés et utilisés sous license LGPL.
- Tout organisme peut développer sa propre distribution.
- La demande d'intervention est le moyen naturel d'intégrer avec les développeurs de l'ÉPM.

# Plan d'assurance qualité particulier

Schémas de calcul  
Distributions de  
l'EPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux  
Résultats à burnup  
nul

- **Contrôle de version des composantes du projet**
  - ◆ Un repository Subversion unique contient tout le projet
  - ◆ Le repository contient les sources, les scripts de configuration, les tests de non-regression, la documentation  $\text{\LaTeX}$  et les fiches d'intervention
  - ◆ tag des versions figées (versions de la page html)
- **Demandes d'intervention et développement en cycle**
  - ◆ les fiches d'intervention sont gardées dans le repository
  - ◆ des scripts de pre- et de post-commit sont incluses dans le repository pour aider la maintenance
  - ◆ [outil web/CGI](#) de saisie des fiches d'intervention
- **Gestion de configuration des codes DRAGON5, TRIVAC5 et DONJON5**
  - ◆ utilisation de scripts UNIX simples
  - ◆ les PC lambdas sont supportés à travers Cygwin et Visual Studio

Schémas de calcul  
Distributions de  
l'EPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux

Résultats à burnup  
nul

## Exemple de fiche d'intervention

Card-index: issue000088

-----  
On Mar 17, 2017, at 12:22 PM, SALINO Vivian <vivian.salino@irsns.fr> wrote:  
Geoffrey a essayé d'utiliser le modèle de fuite hétérogène ECCO. Durant  
cela, il a rencontré un problème, plus précisément une incompatibilité du  
modèle ECCO avec la récupération du courant au bord pour l'équivalence  
SPH (JOUT).

Le message d'erreur, produit avec le jeu de données ci-joint, est:  
"EDIJOU: ARM KEYWORD MUST BE SET IN ASM: MODULE."

Est-ce une incompatibilité "fondamentale" ou s'agit-il d'un bug?

-----  
develop

Mon Mar 20 12:44:05 2017

subversion revision=810

issue000088: Correct an incompatibility between ECCO and JOUT options  
M /trunk/Dragon/src/EDIJOU.f  
-----

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier

## Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

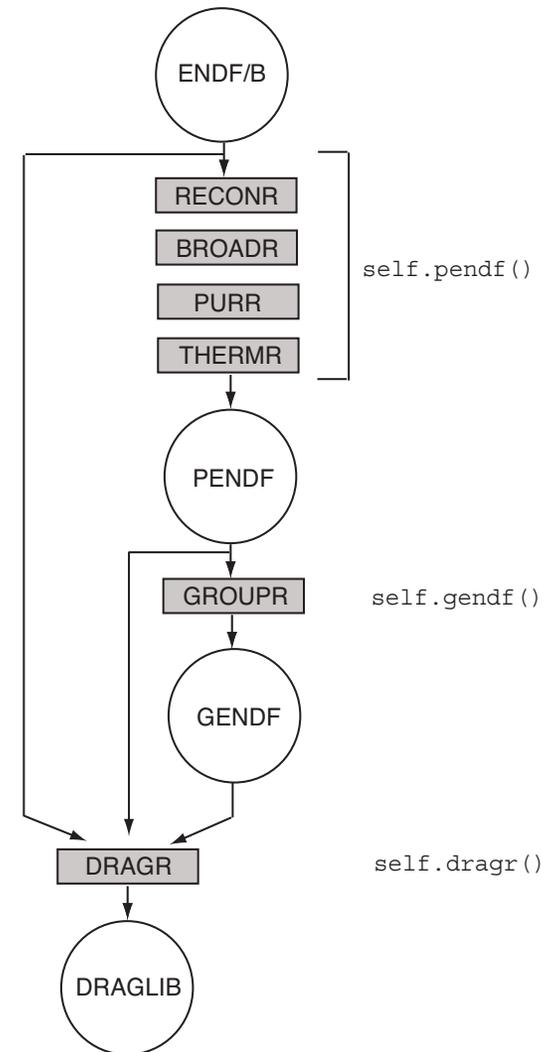
Schémas REP

double-niveaux  
Résultats à burnup  
nul

Disponible sur la page [PyNjoy-2012](#).

Le système comporte 3 composantes:

- module (Fortran 90) de post-traitement `dragm`
- script (Python) `PyNjoy.py` pour encapsuler les appels aux modules NJOY
- un fichier de données par évaluation/bibliothèque



# Le système PyNjoy

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier

## Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux  
Résultats à burnup  
nul

## 1. Instanciation d'un objet:

```
from PyNjoy import *  
jef2p2 = PyNjoy()
```

## 2. Définition des variables d'instance:

```
jef2p2.evaluationName = "Jef2.2"  
jef2p2.nstr = 22  
jef2p2.iwt = 4  
jef2p2.legendre = 1  
jef2p2.hmat = "U238"  
jef2p2.mat = 9237  
jef2p2.evaluationFile = "$HOME/evaluations/Jef2.2/tape7"  
jef2p2.fission = 2 # fission with delayed neutrons  
jef2p2.ss = (2.76792, 1.22773e5)  
jef2p2.potential = 11.1710  
jef2p2.dilutions = ( 1.e10, 94.5, 56.3, 33.6, 20.0, 11.9, 7.1, 4.2 )  
jef2p2.temperatures = ( 293., 550., 900., 1200. )
```

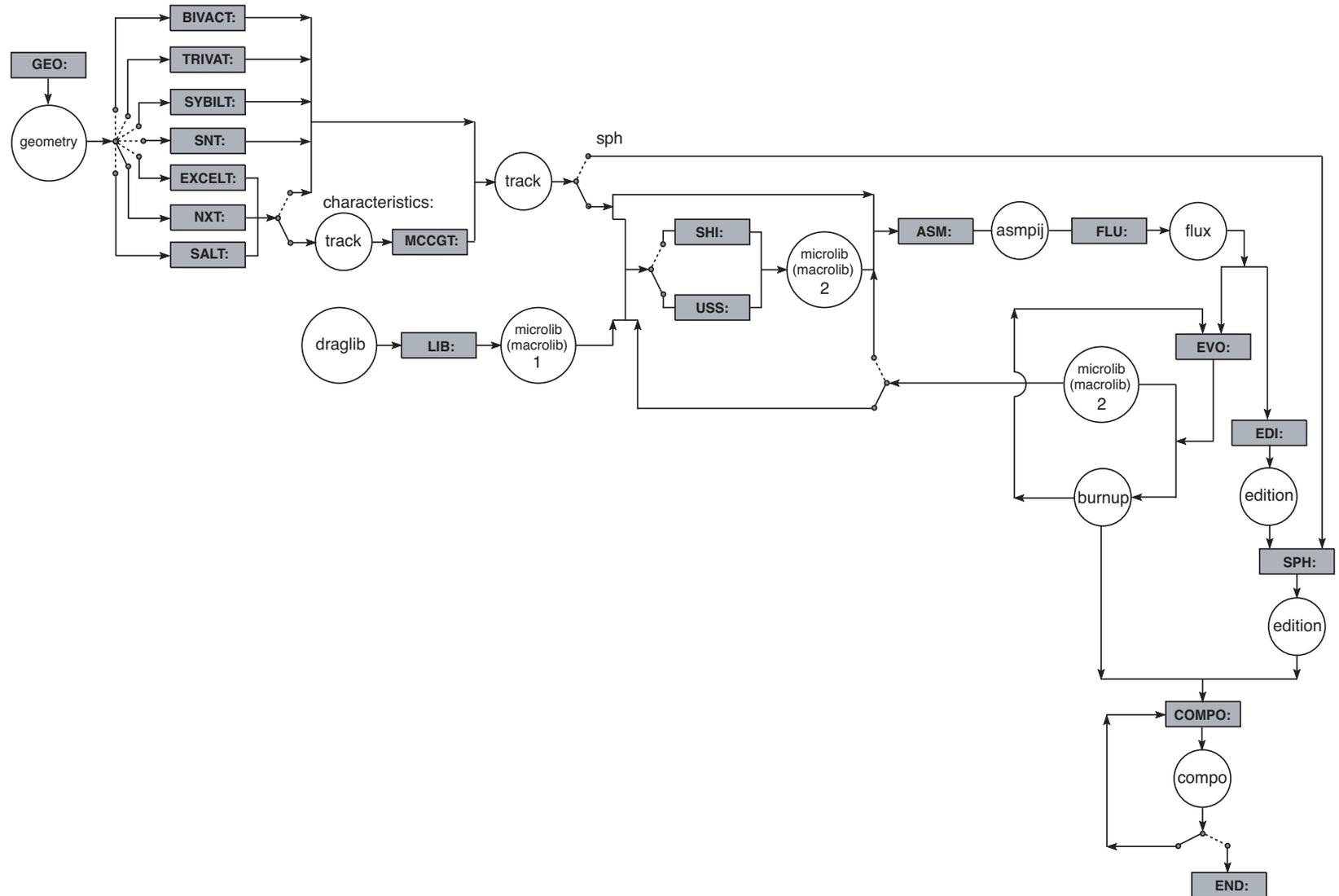
## 3. Activation de la méthode:

```
jef2p2.pendf()
```

# Les codes DRAGON5 et DONJON5

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Flot de données DRAGON5



# Les codes DRAGON5 et DONJON5

Schémas de calcul  
Distributions de  
l'EPM

Plan d'assurance  
qualité particulier

Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP

double-niveaux

Résultats à burnup  
nul

## Modules importants

- Les modules DRAGON5 sont en **bleu** et les modules DONJON5 sont en **vert**.
- Les modules DRAGON5 peuvent être utilisés par DONJON5.
- Les codes DRAGON5 et DONJON5 sont une collection de modules indépendants, sans connexions entre eux et sans effets de bord.

**GEO:** Entrée de la géométrie

**SYBILT:, SNT:, NXT:, SALT:, etc.** Tracking de la géométrie (numérotation, calcul des connexions et des trajectoires pour la méthode PIJ ou MOC)

**BIVACT:, TRIVAT:** Numérotation de la géométrie pour une méthode d'éléments finis

**LIB:** Fabrication de la microlib (sections efficaces microscopiques par isotope)

**SHI:, USS:, TONE:** Calcul d'autoprotection des résonances

**ASM:** Calcul des matrices de PIJ ou des quantités multigroupes non-itératives

**BIVACA:, TRIVAA:** Calcul des matrices d'éléments finis dans DONJON5

**FLU:** Calcul du flux dans DRAGON5

**FLUD:** Calcul du flux dans DONJON5

**EDI:** Homogénéisation et condensation

**SPH:** Équivalence SPH

**EVO:** Solution des équations de Bateman (calcul de burnup)

**COMPO:** Création d'une multicompo

**NCR:** Interpolation/accès d'une multicompo

**RESINI:, USPLIT:, MACINI:, NEWMAC:, etc.** Gestion de la fuelmap dans DONJON5

**INIKIN:, KINSOL:** Cinétique espace-temps

**THM:** Thermohydraulique simplifiée

# Les codes DRAGON5 et DONJON5

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy

Les codes DRAGON5  
et DONJON5

Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Modules sponsorisés par l'IRSN

**M2T:** Création d'un fichier APOTRIM

**CHAB:** Modification de l'information contenue dans une MICROLIB (semblable au module CHABINT : d'APOLLO2).

**SAP:** Création d'un fichier SAPHYB

**SENS:** Création d'un fichier SDF pour le module JAVAPENO de SCALE. Calcul de sensibilité utilisé par Cédric Laville.

**DUO:** Analyse perturbative d'une Microlib test par rapport à une Microlib de référence (semblable au module CLI0 : d'APOLLO2).

**SIM:** Simulation du rechargement par cycle dans un REP (écrit par V. Salino et A. Hébert à partir d'une spécification IRSN).

**ROD:** Gestion des barres de contrôle dans un REP (écrit par G. Tixier à partir d'une spécification IRSN).

**SCR:** Interpolation d'un fichier SAPHYB

**D2P:** Interface DONJON5-PARCS (écrit par J. Taforeau).

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5

## Le langage CLE-2000

L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

- CLE-2000 est un **macro-langage** utilisé pour superviser le flot de donnée entre les modules des codes DRAGON et DONJON. Il permet de **mettre au point les schémas de calcul** et de gérer les fichiers de données.
- CLE-2000 permet de manipuler des variables (définition et utilisation) ainsi que des instructions de contrôle du flot de donnée (boucles, instructions de branchement conditionnels, etc.).
- CLE-2000 permet de manipuler les fichiers et les structures de données utilisées par les modules des codes DRAGON et DONJON.
- CLE-2000 permet de contrôler la procédure d'exécution et d'appeler les modules des codes DRAGON et DONJON. Une procédure CLE-2000 peut en appeler une autre.
- Pour les utilisations plus spécialisées (couplage multiphysique), il est possible d'appeler une procédure CLE-2000 depuis une fonction ANSI C.
- CLE-2000 a été conçu par le Prof. Robert Roy de l'EPM (Département d'Informatique et de Génie Logiciel).

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
**Le langage CLE-2000**  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Règles générales

- Une ligne de code CLE-2000 ne peut dépasser 132 caractères ASCII
- Seuls les 72 premiers caractères ASCII sont significatifs
- Tous les caractères ASCII suivant un “\*” ou un “!” sont considérés commentaires
- Tous les caractères ASCII entre “(” et “)” sont considérés commentaires
- La casse des caractères est significative
- Le caractère espace “ ” est le séparateur. Il est obligatoire.
- Le script principal (point d'entrée) a un suffixe “.x2m”. Les scripts appelés ont un suffixe “.c2m”.
- Si un script est contenu dans un fichier UNIX “monscript.x2m” ou “monscript.c2m”, son nom CLE-2000 est “monscript”. Ce nom ne peut pas comporter plus de 12 caractères.
- Toute variable CLE-2000 doit être définie avant de pouvoir être utilisée.
- Une procédure CLE-2000 peut appeler une (ou plusieurs) procédure CLE-2000 (incluant elle-même) ou un (des) **module** des codes DRAGON5/DONJON5.
- Les variables CLE-2000 sont transmises à un module ou à une procédure par les symboles “<<...>>” (envoi) et “>>...<<” (récupération).
- Il est également possible d'échanger des fichiers et/ou des objets LCM avec un module ou avec une procédure CLE-2000.
  - ◆ Les objets LCM sont des structures de données formées de **tables associatives** et/ou de **listes hétérogènes**.
  - ◆ Les objets LCM sont gardées en mémoire ou sur fichier à accès direct (XSM)
  - ◆ Les objets LCM sont accessibles en **lecture seule, création** et/ou **modification** depuis les modules des codes DRAGON5/DONJON5.

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
**Le langage CLE-2000**  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Variables CLE-2000

CLE-2000 permet de déclarer des variables de différent types. Par exemple:

```
REAL a b c r ;
```

## Calculateur de CLE-2000

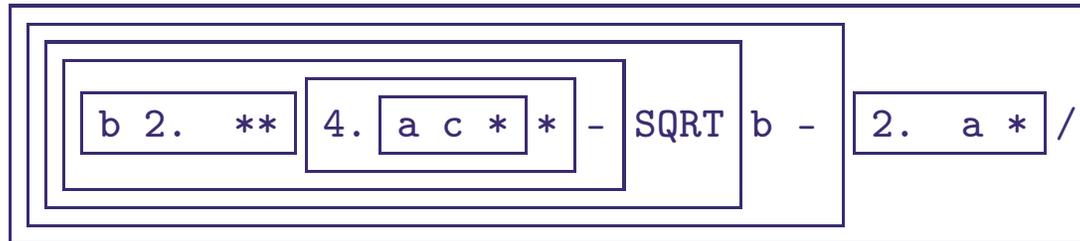
CLE-2000 possède un calculateur (commande **EVALUATE**) qui opère en notation polonaise inversée (RPN). Par exemple, pour calculer

$$r = \frac{1}{2a} \left( -b + \sqrt{b^2 - 4ac} \right)$$

on écrira

```
EVALUATE r := b 2. ** 4. a c * * - SQRT b - 2. a * / ;
```

qui s'évalue de la façon suivante:



# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
**Le langage CLE-2000**  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Calcul récursif du factoriel d'un nombre en CLE-2000

- Le script fact doit être copié dans un fichier ASCII nommé fact.c2m.
- La signature du script fact est: `fact :: <<n>> >>n_fact<< ;`

```
!  
! Example of a recursive procedure.  
!  
! input to "fact": *n*  
! output from "fact": *n_fact*  
!  
INTEGER    n n_fact prev_fact ;  
:: >>n<< ;  
IF n 1 = THEN  
    EVALUATE n_fact := 1 ;  
ELSE  
    EVALUATE n := n 1 - ;  
    ! Here, "fact" calls itself  
    PROCEDURE fact ;  
    fact :: <<n>> >>prev_fact<< ;  
    EVALUATE n_fact := n 1 + prev_fact * ;  
ENDIF ;  
:: <<n_fact>> ;  
QUIT " Recursive procedure *fact* XREF " .
```

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5

## Le langage CLE-2000

L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Exécution du script fact en CLE-2000

```
*  
* Calling the recursive "fact" procedure:  
*  
* input to "fact": *n*  
* output from "fact": *n_fact*  
*  
* use to compute n!  
*  
PROCEDURE fact          ;  
INTEGER n := 8          ;  
INTEGER n_fact          ;  
fact :: <<n>> >>n_fact<< ;  
ECHO "FACTORIAL:" n $Bang_S "=" n_fact ;  
QUIT " Program *xfact* XREF " .
```

## Résultat:

```
>|FACTORIAL: 8 ! = 40320
```

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
**Le langage CLE-2000**  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Appel d'un module DRAGON5/DONJON5

(OutDS) := **Module:** (InDS) :: (data) ;

- (OutDS) est une liste de fichiers et/ou d'objets LCM produits ou modifiés par le module **Module:**.
- (InDS) est une liste de fichiers et/ou d'objets LCM accédés en lecture seule ou modifiés par le module **Module:**.
- Un item accédé en modification doit apparaitre à la fois dans (OutDS) et dans (InDS).
- L'ordre des items est important (voir le guide d'utilisation).

```
LINKED_LIST MOSTELC VOLTRK LIBRARY ;  
SEQ_BINARY TRKSPC ;  
INTEGER an2d := 8 ;  
REAL densur := 20.0 ;  
...  
VOLTRK TRKSPC := NXT: MOSTELC ::  
    TISO <<an2d>> <<densur>> ;  
LIBRARY := SHI: LIBRARY VOLTRK TRKSPC ;
```

# Le langage CLE-2000

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5

## Le langage CLE-2000

L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

### Boucles et instructions conditionnelles

```
LINKED_LIST DONNEES LIBRARY TRACKSS ;
MODULE UTL: GREP: TONE: ;
REAL Tend BUend ;
REAL Fuelpwr := 36.8 ;
...
INTEGER maxstep := 4 ;
DONNEES := UTL: ::
    CREA BURN <<maxstep>> = 0.0  9.375  18.75  37.5 ;
INTEGER istep := 0 ;
WHILE istep maxstep < DO
    EVALUATE istep := istep 1 + ;
    ECHO "istep: " istep "/" maxstep ;
    GREP: DONNEES :: GETVAL 'BURN' <<istep>> >>BUend<< ;
    EVALUATE Tend := BUend Fuelpwr / ;
    IF istep 1 = THEN
        LIBRARY := TONE: LIBRARY TRACKSS :: EDIT 3
        MXIT 2  ARM GRMIN 52 ;
    ENDIF ;
ENDWHILE ;
```

# L'environnement APOLLO2

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000

L'environnement  
APOLLO2

L'environnement  
APOLLO3

Schémas REP  
double-niveaux  
Résultats à burnup  
nul

- Le code de réseau APOLLO2 possède un **macro-langage** nommé **GIBIANE** pour superviser le flot de donnée. Il permet de **mettre au point les schémas de calcul** et de gérer les fichiers de données.
- GIBIANE est semblable à CLE-2000 mais est plus complexe et plus puissant.
- Le CEA a mis au point un schéma de calcul REP double-niveaux nommé **REL-2005** dans l'environnement GIBIANE.

## Exemple de code GIBIANE

```

ARGUMENT IMPRES/'ENTIER' TSTR*'TABLE' TOPT*'TABLE' TRES*'TABLE' ;
SI ( NON ( EXISTE IMPRES ) ) ;
    IMPRES = 0 ;
FINSI ;
IMP = IMPRES - 1 ;
*
MESSAGE = 'ERREUR DANS LA PROCEDURE SAPHYB_STD_S' ;
***-----
TRACE = ' &NOTRACE ' ;
SI ( IMPRES GE 1 ) ;
    WRITE: 'LISCONS' 'EXECUTION DE LA PROCEDURE SAPHYB2_STD_S' ;
    TRACE = ' &TRACE ' ;
FINSI ;
*
TXTB2 = CONCAT: ' &KEFF TRES.''KEFF'' &KINF TRES.''KINF'' ' ;
                ' &B2 TRES.''B2'' ' ;
*
SI ( NON ( EXISTE TOPT.'VERS_BIB' ) ) ;
    TOPT.'VERS_BIB' = 'CEA93' ;
FINSI ;
SI ( NON ( EXISTE TOPT.'BIBLIO' ) ) ;
    TOPT.'BIBLIO' = 7 ;
FINSI ;
*
RESPROC TRES TSTR TOPT ;
*
FINPROC ;

```

# L'environnement APOLLO3

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

APOLLO3 possède trois niveaux d'interfaces utilisateur. Les deux premiers niveaux permettent de [mettre au point les schémas de calcul](#).

- **Classes C++ nommées UDM/OS:** Un ensemble de classes mises à disposition des utilisateurs experts. Un API **orienté objet** est formé des classes UDM (User Data Model) et des classes OS (Objets Significatifs).
  - ◆ une application métier (ou schéma de calcul) est développée en langage C++ en accédant les classes UDM/OS.
  - ◆ offre les services d'un langage de programmation (variables, branchements conditionnels, boucles).
  - ◆ compilation nécessaire (architecture offerte dans l'espace modélisateur qui est livré).
- **environnement Python:** Semblable à l'environnement C++ précédent (mapping Python des classes UDM/OS).
  - ◆ pas de nécessité de recompiler.
  - ◆ mapping C++/Python basé sur Boost.Python
  - ◆ capacités de programmation du langage Python.
- **environnement UDF:** Un *User Data File* (UDF) est un fichier du type “mot clé-valeur”.
  - ◆ pas de capacités de programmation (pas de variables, pas de branchements conditionnels, pas de boucles).
  - ◆ prise en main du code simplifiée.
  - ◆ limite les erreurs d'utilisation mais souplesse limitée.
  - ◆ informations et résultats du code accessibles dans le fichier de sortie.

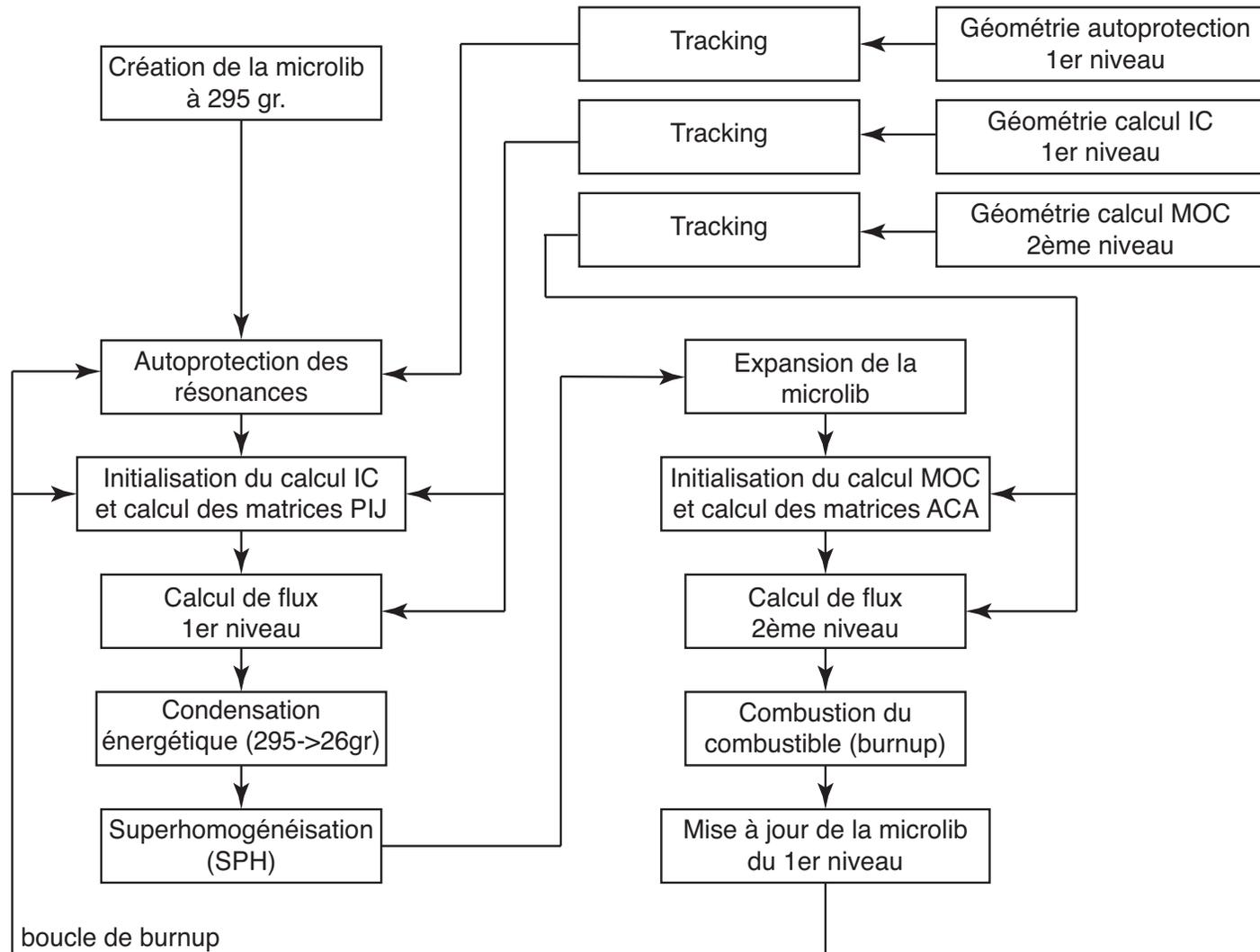
# Schémas REP double-niveaux

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

Travaux réalisés dans le cadre de maîtrises commandités par EDF/R&D et CEA/SPRC. Il s'agit d'un **schéma de calcul** d'assemblage à **deux niveaux**

- Désigné par le nom du dernier étudiant y ayant contribué: **schéma Canbakan**
- Disponibles sur la page <http://www.polymtl.ca/merlin/canbakan.htm>
- Schéma CLE-2000 programmé avec les modules de DRAGON5
- Pas de modèle de fuites (on utilise TYPE K) ni de contre-réactions (le burnup est le seul paramètre)
- Le premier niveau est basé sur le maillage **SHEM-295**. Le solveur est la **méthode à courants d'interfaces** (IC) avec **itérations flux-courant**
  - ◆ Le calcul d'autoprotection utilise la méthode des sous-groupes avec tables de probabilités CALENDF (méthode SPM, mot clé PT)
- Le second niveau est basé sur un maillage à **26 groupes**. Le solveur est la **méthode des caractéristiques** (MOC)
  - ◆ Discrétisation spatiale de type “moulin à vent” avec description explicite de la lame d'eau
  - ◆ Conditions frontières de réflexion spéculaire (tracking cyclique)
  - ◆ Scattering  $P_0$  corrigé transport ou  $P_1$
  - ◆ Accélération synthétique par la méthode ACA de Igor Suslov
- Le calcul de référence est un calcul Monte-Carlo réalisé avec le code SERPENT2
  - ◆ 2 000 000 neutrons x 1000 cycles à chaque pas
  - ◆ résolution des équations de Bateman par la méthode CRAM

## Flot de données



Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
**Schémas REP  
double-niveaux**  
Résultats à burnup  
nul

# Schémas REP double-niveaux

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Assemblage UOX – calculs à burnup zéro

	SERPENT2		DRAGON5		
	analog	implicit	295-group	295/26-grp no sph	295/26-grp sph
pin cell	<b>1.37572 ± 0.00093</b>	<b>1.37539 ± 0.00036</b>	<b>1.374360</b> -103 pcm	<b>1.372913</b> -248 pcm	<b>1.374826</b> -56 pcm
17 × 17 assembly	<b>1.37053 ± 0.00094</b>	<b>1.36876 ± 0.00038</b>	<b>1.368348</b> -41 pcm	<b>1.367015</b> -175 pcm	<b>1.368985</b> 22.5 pcm

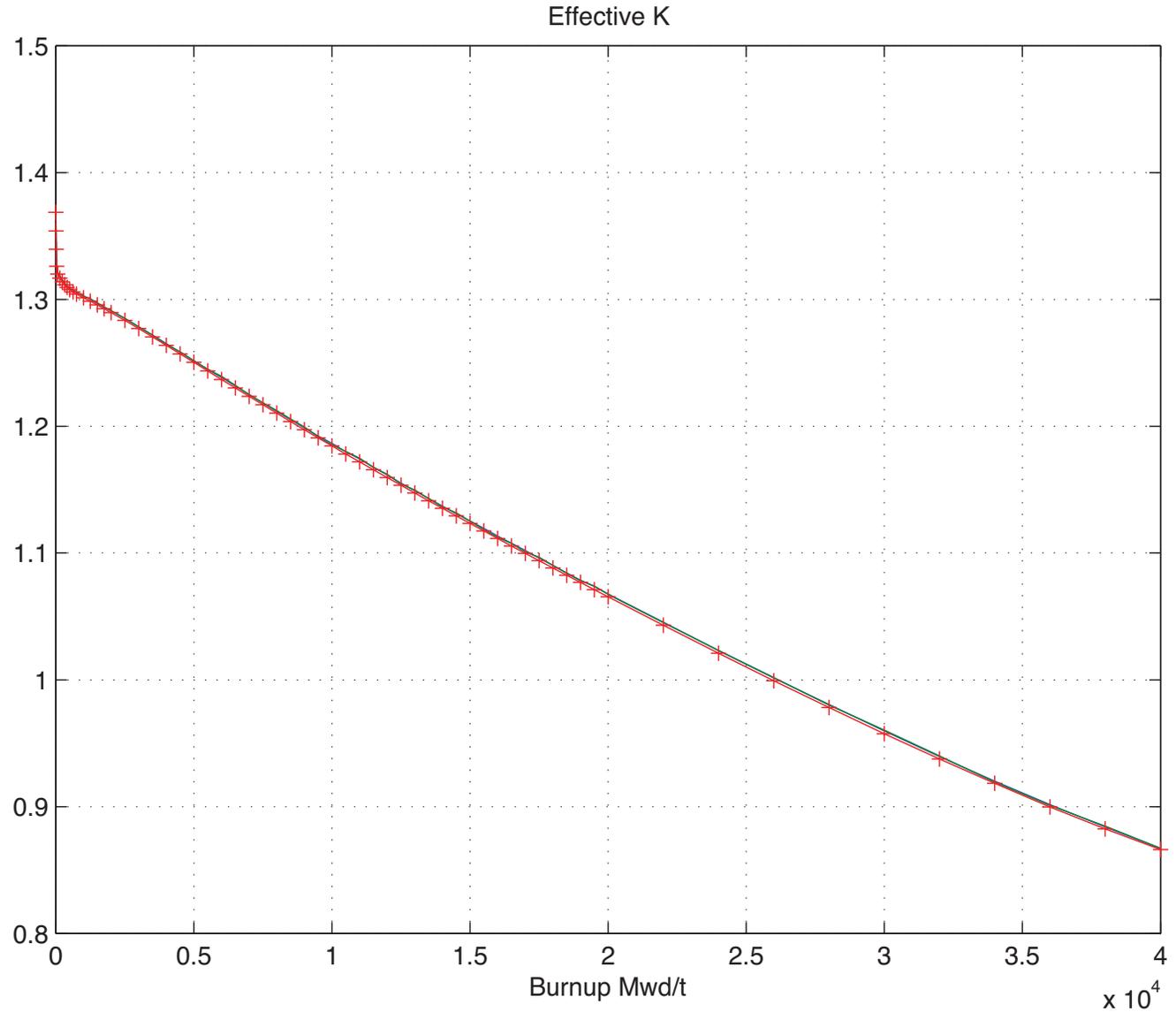
## Assemblage UOX – temps de calcul en évolution

	SERPENT2	295-group	295/26-grp no sph	295/26-grp sph
Temps	173 heures	138 heures	4 heures	8 heures

# Schémas REP double-niveaux

## Assemblage UOX – calcul à deux niveaux

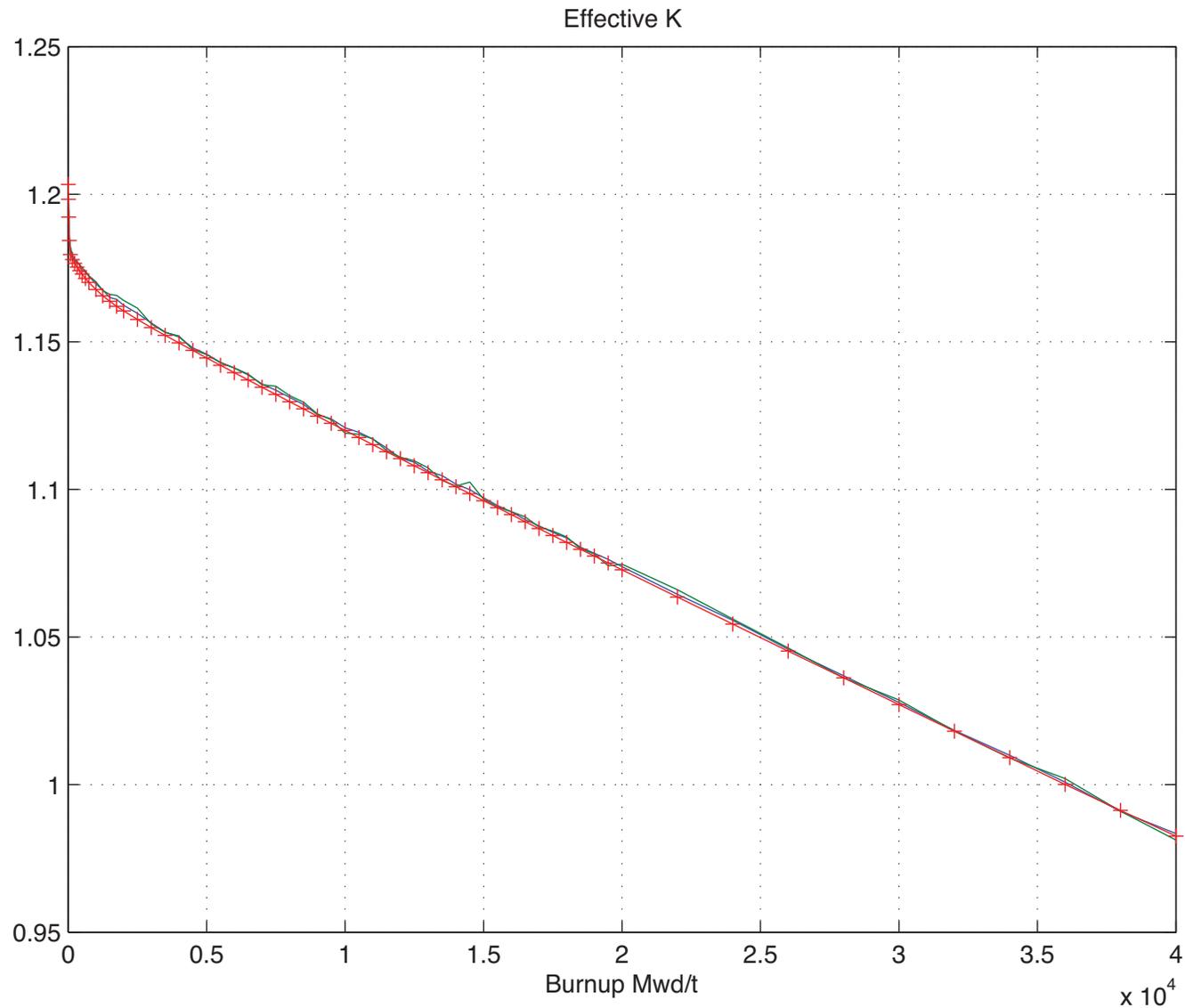
Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
**Schémas REP  
double-niveaux**  
Résultats à burnup  
nul



# Schémas REP double-niveaux

## Assemblage MOX trizoné- calcul à deux niveaux

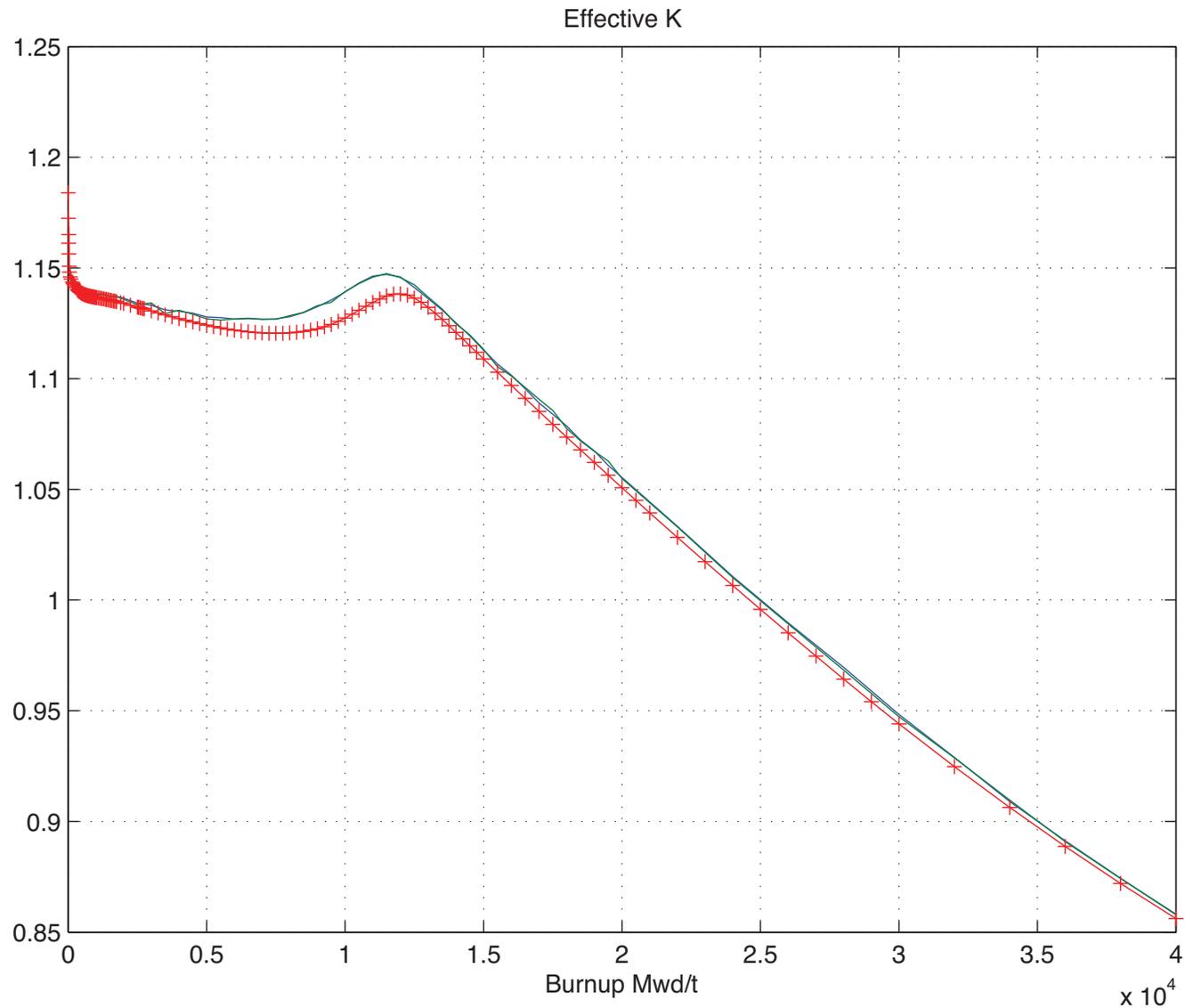
Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
**Schémas REP  
double-niveaux**  
Résultats à burnup  
nul



# Schémas REP double-niveaux

## Assemblage UOX avec gadolinium- calcul à deux niveaux

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
**Schémas REP  
double-niveaux**  
Résultats à burnup  
nul



# Résultats à burnup nul

Schémas de calcul  
Distributions de l'EPM  
Plan d'assurance qualité particulier  
Le système PyNjoy  
Les codes DRAGON5 et DONJON5  
Le langage CLE-2000  
L'environnement APOLLO2  
L'environnement APOLLO3  
Schémas REP double-niveaux  
Résultats à burnup nul

- Les résultats UOX présentés ici ont été refaits en ajoutant du Bore dans le caloporteur.
- Des méthodes alternatives d'autoprotection ont été testées:
  - ◆ SPM: Subgroup projection method (recommandée)
  - ◆ ESSM: Embedded self-shielding method (utilisée par le code POLARIS de SCALE6)
  - ◆ Tone: (Tone's method, aussi disponible dans APOLLO3)
  - ◆ Tone/SPH: (Tone's method avec correction SPH)

$k_{\text{eff}}$  and absolute discrepancies in pcm.

	UOX		MOX		UO <sub>2</sub> Gd <sub>2</sub> O <sub>3</sub>		UOX+AIC	
	$k_{\text{eff}}$	$\Delta a$	$k_{\text{eff}}$	$\Delta a$	$k_{\text{eff}}$	$\Delta a$	$k_{\text{eff}}$	$\Delta a$
SERPENT2								
ANALOG	1.32276±0.00032		1.12375±0.00035		1.17240±0.00031		0.91165±0.00031	
IMPLICIT	1.32247±0.00014		1.12400±0.00016		1.17256±0.00017		0.91182±0.00019	
SPM	1.322213	-26	1.122440	-156	1.171562	-100	0.9107985	-102
ESSM	1.318249	-422	1.119247	-475	1.167947	-461	0.9056712	-615
Tone	1.318295	-418	1.119287	-471	1.167975	-459	0.9057724	-605
Tone/SPH	1.321074	-140	1.121421	-258	1.169881	-268	0.9075054	-431

# Résultats à burnup nul

Absorption rates in  $^{238}\text{U}$  and absolute discrepancies in pcm.

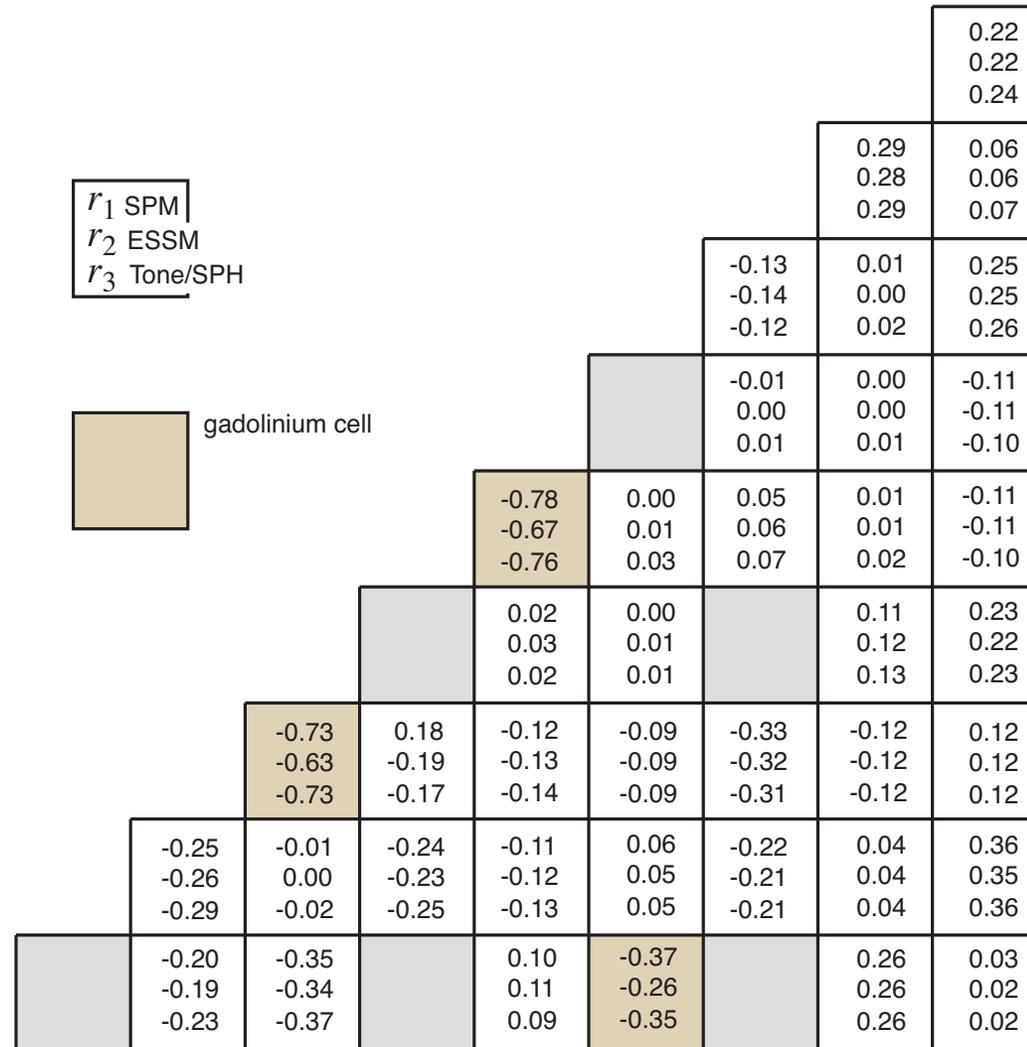
		UOX		MOX		UO <sub>2</sub> Gd <sub>2</sub> O <sub>3</sub>		UOX+AIC	
		$\tau_a$	$\Delta a$	$\tau_a$	$\Delta a$	$\tau_a$	$\Delta a$	$\tau_a$	$\Delta a$
group 1	SERPENT2	0.17370		0.19926		0.19492		0.24877	
	SPM	0.17306	-64	0.19893	-34	0.19498	6	0.24794	-81
	ESSM	0.17526	156	0.20140	213	0.19754	262	0.25135	261
	Tone	0.17523	153	0.20136	210	0.19752	260	0.25131	258
	Tone/SPH	0.17354	-16	0.19949	23	0.19604	111	0.24936	62
group 2	SERPENT2	0.03898		0.01036		0.03820		0.03469	
	SPM	0.03899	2	0.01045	9	0.03820	$\approx 0$	0.03469	1
	ESSM	0.03898	1	0.01044	8	0.03819	-1	0.03466	-2
	Tone	0.03898	1	0.01044	8	0.03819	-1	0.03466	-2
	Tone/SPH	0.03899	1	0.01045	9	0.03819	-1	0.03467	-1

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

# Résultats à burnup nul

Schémas de calcul  
Distributions de l'EPM  
Plan d'assurance qualité particulier  
Le système PyNjoy  
Les codes DRAGON5 et DONJON5  
Le langage CLE-2000  
L'environnement APOLLO2  
L'environnement APOLLO3  
Schémas REP double-niveaux  
Résultats à burnup nul

## Power map relative accuracy on Gd assembly



# Résultats à burnup nul

Schémas de calcul  
Distributions de  
l'EPM  
Plan d'assurance  
qualité particulier  
Le système PyNjoy  
Les codes DRAGON5  
et DONJON5  
Le langage CLE-2000  
L'environnement  
APOLLO2  
L'environnement  
APOLLO3  
Schémas REP  
double-niveaux  
Résultats à burnup  
nul

## Percent discrepancies of the fission reaction map.

	UOX		MOX		UO <sub>2</sub> Gd <sub>2</sub> O <sub>3</sub>		UOX+AIC	
	max	avg	max	avg	max	avg	max	avg
SPM	0.43	0.11	0.71	0.17	0.78	0.17	0.54	0.24
ESSM	0.42	0.11	0.69	0.16	0.67	0.16	0.58	0.22
Tone	0.43	0.11	0.69	0.16	0.67	0.16	0.58	0.22
Tone/SPH	0.42	0.11	0.71	0.16	0.76	0.18	0.58	0.21

## CPU time for DRAGON5 assembly calculations.

	UOX	MOX	UO <sub>2</sub> Gd <sub>2</sub> O <sub>3</sub>	UOX+AIC
SPM	4m42s	5m25s	16m32s	9m26s
ESSM	3m38s	4m24s	8m34s	6m14s
Tone	3m38s	4m24s	8m21s	6m12s
Tone/SPH	4m7s	4m50s	10m7s	7m4s