Towards DRAGON Version4

A. Hébert

Institut de génie nucléaire
École Polytechnique de Montréal
September 10, 2006
What is Version4?

Version4 is a new **distribution** of the reactor physics computer codes at GAN. Its components are:

- **DRAGR** module in NJOY and Python script `PyNjoy.py`
- **Ganlib** tools (CLE-2000, LCM/XSM API)
- **Modules** (calculation operators) of the following codes:
  - **Dragon**: lattice code
  - **Trivac**: reactor (full core) code
  - **Donjon**: simulation of reactor operation – **end of 2006**
  - **Optex**: reactor design optimization – **current 2007**
- **Configuration scripts** (sh), non-regression tests, JEF2.2/XMAS standard library, \LaTeX documentation, etc.

Version4 is **not** a complete replacement for Version3.
Motivations for building this distribution are:

- We want to introduce support for cross-section library production with NJOY.
- State-of-the-art ACR1000 modelization needs some advanced capabilities not available in Version3.
- We want to avoid duplication of similar capabilities and improve interoperability.
- We want to adopt a more consistent development model for our reactor physics computer codes.
- After 12 years of development, the Dragon flow diagram needs some cleaning.
- We did our best to avoid changing anything in the user’s interface.
Advanced capabilities in Version4

- Jef-2.2 XMAS (172-group) Draglib-formatted libraries
- capability to produce Dragon libraries with NJOY
- **NXT**: module (2D/3D new-generation Excell tracking)
- self-shielding **USS**: module based on the subgroup equations
- isotropic streaming model **ECCO** in **FLU**: (for space-dependent diffusion coefficient calculations)
- asymptotic SPH method for reflector model
- SPH method with simplified PN Thomas-Raviart finite elements in 2D
Advanced capabilities in Version4

- multi-parameter COMPO database (creation and interpolation)
- simplified PN Thomas-Raviart finite elements in 3D for full core models in Trivac (Cartesian 3D)
- capability to use the characteristic method for self-shielding, leakage, flux and SPH calculations
- availability of the double-heterogeneity model (Bihet) with Sybil, Excell (PIJ) and NXT (PIJ)
- discrete ordinates capabilities in 1D and 2D geometries (new SNT: module)
- availability of the current iteration method with the interface current (IC) method in Sybil
Capabilities under active development

- **NXT**: geometries
  - MERG GEOM (equigeom) capability
  - mergings
  - cylindrical boundaries and hexagonal geometries
  Developed in Version3 and copied in Version4

- **NXT**: inline tracking with the method of characteristics
  (available with **EXCELT**: in Version3)

- Thomas-Raviart-Schneider (i.e., hexagonal) simplified PN capabilities in 3D (for the qualification of some ZED2 experiments)
### Avoiding duplication

#### Donjon-Dragon duplication

<table>
<thead>
<tr>
<th>Version 3</th>
<th>Version 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GEO</strong> : in Donjon, <strong>GEOD</strong> : in Donjon</td>
<td><strong>GEO</strong> : in Dragon</td>
</tr>
<tr>
<td><strong>MACD</strong> : in Donjon, <strong>MAC</strong> : in Dragon</td>
<td><strong>MAC</strong> : in Dragon</td>
</tr>
<tr>
<td><strong>BIVACT</strong> : in Donjon and Dragon</td>
<td><strong>BIVACT</strong> : in Dragon</td>
</tr>
</tbody>
</table>

#### Duplication of flux solution modules (power iteration)

<table>
<thead>
<tr>
<th>Version 3</th>
<th>Version 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FLU</strong> : $P_{ij}$ and IC</td>
<td><strong>FLU</strong> : all methods</td>
</tr>
<tr>
<td><strong>MOCC</strong> : cyclic characteristics in 2D</td>
<td>(based on <strong>MOCC</strong> : )</td>
</tr>
<tr>
<td><strong>MCU</strong> : characteristics in 3D</td>
<td></td>
</tr>
</tbody>
</table>
Avoiding duplication

Duplication of system matrix assembly modules

<table>
<thead>
<tr>
<th>Version3</th>
<th>Version4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASM</strong>: $P_{ij}$ and IC</td>
<td><strong>ASM</strong>: all types of assemblies</td>
</tr>
<tr>
<td><strong>EXCELL</strong>: 3D $P_{ij}$ (in-line tracking)</td>
<td></td>
</tr>
</tbody>
</table>

Dragon seminar at PHYSOR 2006
### Improving interoperability

<table>
<thead>
<tr>
<th>Resonance self-shielding</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version 3</strong></td>
</tr>
<tr>
<td><strong>SHI</strong>: Stamm’ler model</td>
</tr>
<tr>
<td>- $P_{ij}$ and IC (non-iterative)</td>
</tr>
<tr>
<td><strong>USS</strong>: subgroup model</td>
</tr>
<tr>
<td>- $P_{ij}$ and IC (iterative or not)</td>
</tr>
<tr>
<td>- characteristics</td>
</tr>
<tr>
<td>- SN</td>
</tr>
</tbody>
</table>
Improving interoperability

Streaming models (space-dependent diffusion coefficients) in module **FLU**:  

<table>
<thead>
<tr>
<th>Version 3</th>
<th>Version 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Isotropic streaming:</strong></td>
<td><strong>Isotropic streaming (ECCO):</strong></td>
</tr>
<tr>
<td>not available</td>
<td>$P_{ij}$ and IC (iterative or not)</td>
</tr>
<tr>
<td></td>
<td>$S\ P_n$ approximation</td>
</tr>
<tr>
<td></td>
<td>characteristics</td>
</tr>
<tr>
<td></td>
<td>SN</td>
</tr>
</tbody>
</table>

**Anisotropic streaming (HETE):**  

- $P_{ij}$ in Excell and NXT  

**Anisotropic streaming (HETE):**  

- $P_{ij}$ in Excell and NXT
## Improving interoperability

SPH equivalence in module **EDI**:

<table>
<thead>
<tr>
<th>Version3</th>
<th>Version4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Types of macro-calculations:</td>
<td>Types of macro-calculations:</td>
</tr>
<tr>
<td>$P_{ij}$ and IC (non-iterative)</td>
<td>$P_{ij}$ and IC (iterative or not)</td>
</tr>
<tr>
<td>diffusion approximation</td>
<td>diffusion approximation</td>
</tr>
<tr>
<td></td>
<td>$SP_n$ approximation</td>
</tr>
<tr>
<td></td>
<td>characteristics</td>
</tr>
<tr>
<td></td>
<td>SN</td>
</tr>
</tbody>
</table>
A more consistent development model

- Version control of the project components
  - A single Subversion repository holds the complete project
  - The repository contains sources, configuration scripts, non-regression tests, \LaTeX\ docs and issue-tracking info.
  - LGPL subsets are available for download

- Issue tracking and spiral development management
  - The issue-tracking data is kept in the Subversion repository
  - Pre- and post-commit Python scripts are hooked in the repository to help issue-tracking
  - A web/CGI tool is available to all users for submitting issues

- Configuration management of the codes Njoy, Dragon, Trivac, Donjon and Optex.
  - Simple UNIX install scripts are used
  - PCs are supported through Cygwin
The system is made of 3 components:

- **DRAGR**, a post-treatment Fortran 77 module
- **PyNjoy.py**, a Python script encapsulating NJOY modules
- one data-file per evaluation/library
Draglib production with NJOY

1. Instantiating an object:
   
   ```python
   from PyNjoy import *
   jef2p2 = PyNjoy()
   ```

2. Defining instance variables:
   
   ```python
   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. Invoking a method:
   
   ```python
   jef2p2.pendf()
   ```
Cross section library treatment

- Improved DRAGLIB library support
  - in-house library creation with NJOY99 and DRAGR
  - contains detailed isotopic depletion data (with reaction-wise energy components)
  - contains autolib data for the Riemann integration and Ribon extended methods.
  - contains delayed neutron data
  - no pseudo fission products

- WIMS-D4 and MATXS library support (available in V3.05)

- NDAS library support (not available in the LGPL subset)
  - use of certified AECL libraries
  - required to improve the quality of our validation studies
Resonance self-shielding models

- Models based on the Generalized Stamm’ler method (SHI: module)
  - without distributed self-shielding effects (available in V3.05)
  - with Nordheim distributed self-shielding model (new)
  - with Riemann integration method (new)

- Models based on the subgroup method (USS: module) (new)
  - with physical probability tables (aka WIMS-7 and HELIOS)
  - Ribon extended method (with or without a model to represent mutual shielding effects)
Improved isotopic depletion

- Take into account energy produced by
  - fission (aka Dragon V3.05)
  - radiative capture (important in gadolinium and dysprosium) (new)
  - radioactive decay (important when the fuel is out-of-core) (new)

- Use a linear variation of $\langle \sigma_x \phi \rangle$ in time. NOTE: Version 3.05 is assuming a linear variation of $\langle \phi \rangle$ in time.
  - possibility to extrapolate from the preceding time step (new)

- Availability of a saturation model for small-halflive isotopes (aka Dragon V3.05)
Method of characteristics

A new set of solvers based on the method of characteristics:

- 2D/3D EXCELT: and NXT: geometries
- 2D/3D isotropic or 2D specular (aka MOCC:) boundary conditions
- scattering anisotropy to arbitrary $P_n$ order
- algebraic collapsing acceleration (ACA)
- compatible with the flux solution module used for PIJ calculations
- use of vectorial doors (DOORAV and DOORFV)

In V3.05, only the specular 2D and isotropic 3D options with $P_0$ scattering are available in specific flux-solution modules (MOCC: and MCU:).
SPH model improvements

- A greater variety of macro-calculation techniques:
  - 1D, 2D and 3D collision probabilities (PIJ)
  - 1D, 2D and 3D method of characteristics (new)
  - 1D and 2D diffusion theory
  - 1D and 2D $SP_n$ method (new)
  - 1D and 2D discrete ordinates method (new)

- Availability of the asymptotic normalization

- Use of vectorial doors ($DOORAV$, $DOORPV$ and $DOORFV$)
Flow diagram of Dragon Version 3

Flowchart showing the process flow from geometry (GEO), through lib (LIB), microlib (microlib), sph (sph), track (track), SHI (SHI), microlib (macrolib), MCU (MCU), MOCC (MOCC), and END (END) with various modules such as BIVACT, SYBILT, JPMT, EXCEL, nxt, MCU, EXCELL, SHI, microlib (macrolib), ASM, and FLU.
A multi-D reactor database

- We have generalized modules CPO: and CRE: to an arbitrary number of global / local parameters.
- build a reactor database: COMPO: generalize CPO:
- interpolate the database: NCR: generalize CRE:

**Definitions:**
- **Global parameter:** characterize the complete lattice
- **Local parameter:** characterize a unique cell in a checkerboard or supercell calculation

- Compatible with micro-depletion
- Contains delayed neutron data
- Available only in Dragon Version 4

Dragon seminar at PHYSOR 2006
set of **elementary calculations** characterized by a unique $n$–tuple of global / local parameters

each of them is a **microlib** containing data condensed over $G$ groups and homogenized over each local zone

a **table-of-content** is used to classify the elementary calculations and to relate them to global / local parameters.

---

Dragon seminar at PHYSOR 2006
Database format

- build from
  - associative tables (aka hash tables or dictionaries)
  - heterogeneous lists (aka cell arrays) (new)
- use LCM (in core memory) and XSM (direct access file) access routines
  - available in Fortran-77 and ANSI C
- Other characteristics:
  - XSM: direct access binary format (big or little endian)
  - LCM and XSM: same auto-descriptive format (similar to XML). Can be serialized.
  - the XSM associative tables are used for the Draglib object.
Database construction in **COMPO**: 1

- **Initialization call** (at the beginning of the Dragon run)
  - define the number and types of global / local parameters

- **Data gathering call** (at the end of each burnup / edition step in Dragon)
  - find the values of the global / local parameters
  - store the corresponding homogenized / condensed **microlib** object.
Initialization call:

EVALUATE FUEL1 := 3 ;
CPO := COMPO: ::
    STEP UP fuel
    COMM 'Line of comment' ENDC
    PARA 'BCON' VALU REAL
    PARA 'FTMP' TEMP LIBRARY <<FUEL1>>
    PARA 'BURN' IRRA
    PARA 'FLUB' FLUB
    PARA 'PUIS' POWR
    PARA 'XE1' CONC XE135PF LIBRARY <<FUEL1>>
    LOCA 'burn' IRRA
    LOCA 'flub' FLUB ;

Data gathering call:

CPO := COMPO: CPO EDIT BURNUP FLUX LIBRARY ::
    STEP UP fuel
    SET <<evoend>> DAY
    BCON <<BoronCont>> ;
Database interpolation in NCR:

- Multidimensional interpolation based on
  - Ceschino polynomial expansions
  - cubic Hermite polynomials

- Available functionalities:
  - interpolation at a specific parameter $n$–tuple
  - parameter-averaging (e.g., time-averaging)
  - delta-sigma contributions

- Produce a microlib or a macrolib

- Can gather parameters values from a map object in Donjon

- Micro-depletion is possible from the interpolated microlib.
Database interpolation in NCR:

Interpolation call:

MACRO2 := NCR: CPO ::

NMIX 7 MACRO COMPO CPO fuel
MIX 1 FROM 1 SET 'flub' 2.1248E-02 ENDMIX
MIX 2 FROM 2 SET 'BURN' 3.7498E+01 ENDMIX
MIX 3 FROM 3 SET 'FLUB' 2.1363E-02 ENDMIX
MIX 4 FROM 4 SET 'burn' 3.7426E+01 ENDMIX
MIX 5 FROM 5 SET 'flub' 2.1127E-02 ENDMIX
MIX 6 FROM 6 SET 'flub' 2.1289E-02 ENDMIX
MIX 7 FROM 7 SET 'BURN' 3.7498E+01 ENDMIX ;
Availability

The open-source subset of Version4 (including PyNjoy, Dragon and Trivac) is available for download. Visit:
http://www.polymtl.ca/merlin

- Actually at level v4.0.0. NXT: is identical to the version in v3.0.5B
- This is open-source; you can contribute with
  - improved configuration scripts
  - new or improved Fortran sources
  - bug report and/or development suggestions

Use our issue tracking submission form!