POLYTECHNIQUE
MONTRÉAL

DRAGON and DONJON: a legacy open-source reactor physics project at Polytechnique Montréal

A. Hébert

2022/06/23

Table of contents

Introduction
Key dates
The DRAGON lattice code
The DONJON full-core simulation code
Conditions of use
Quality assurance
Using DRAGON5
Conclusion
Ressources

Introduction

- The Institut de Génie Nucléaire (IGN) at Polytechnique Montréal is supporting R&D activities in Reactor Physics for 50 years.
- Reactor physics codes DRAGON5 and DONJON5 are developed under Open-Source license since 1993. The Version5 package contains both DRAGON5 and DONJON5.
  - DRAGON5 is a lattice code, based on deterministic and multigroup solutions of the Boltzmann equation. It use NJOY-generated cross sections at input.
  - DONJON5 is a full core simulation tool compatible with DRAGON5.
- In the future, we plan to extent the use of DRAGON5 to medical applications related to radiotherapy.
- The IGN was innovating on four aspects:

1983 DRAGON was the first is currently the only deterministic lattice code developed in a university.
40-year experience.

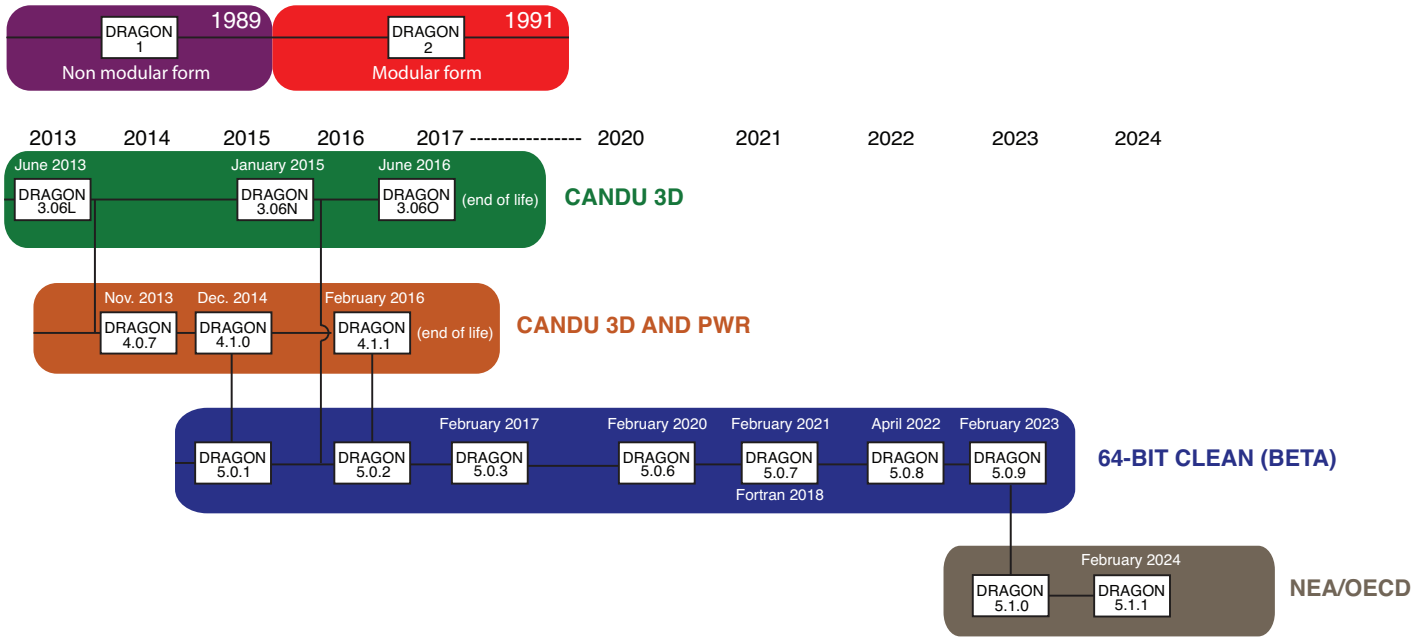
1989 DRAGON2 is an Open-Source project developed under the GNU Lesser General Public License (LGPL). Still the only Open-Source option.
34-year experience.

1996 DRAGON3 become an Industrial Standard Toolset (IST) component used by the CANDU Owner's Group (COG) across the world for the representation of CANDU reactivity devices in 3D lattice geometry.

2003 DRAGON4 is developed under strict Quality Assurance (QA) procedures.
20-year experience.

Introduction

- DRAGON5 evolved from a modular Fortran-77 project in 1983 to a Fortran-2013 project in December 2014 and recently to a modern Fortran-2018 project in February 2021.
- The DRAGON5 project will reach an important milestone at the beginning of 2023.
- The actual Beta version is hosted on a dedicated server at Polytechnique Montréal.
- After 2 years of alpha and 8 years of beta status, it will reach the production status.



Key dates

1983 Start of DRAGON1 development on MVS (IBM) and NOS-BE (CDC) operating systems.

1989 End-of-life of DRAGON1 and start of DRAGON2 (modular version) development.

- DRAGON2 is the first version developed for the UNIX operating systems (available on Cray/UNICOS and NextStep).
- DRAGON2 is the first version developed under the LGPL.

1991 End-of-life of DRAGON2.

1993 Start of DRAGON3 (IST version) development.

1996 First release of DRAGON3 (IST version).

2003 Start of DRAGON4 (CANDU/PWR version) development.

- DRAGON4 is the first version developed under QA.

2006 First release of DRAGON4 (CANDU/PWR version).

2010 Start of DRAGON5 (64-bit clean Fortran 2003 alpha version) development.

2014 First release of DRAGON5 (64-bit clean Fortran 2003 beta version).

2016 End-of-life of DRAGON3 and DRAGON4.

2022 Release of DRAGON5 version 5.0.8

2023 Expected release of DRAGON5 production version 5.1.0

The DRAGON lattice code

DRAGON is now a full-feature lattice code with the following capabilities:

- solution techniques of the Boltzmann transport equation (BTE) based on collision probabilities (PIJ), method of characteristics (MOC) and discrete ordinates method (SN)
- solution technique of the Boltzmann-Fokker-Planck (BFP) equation using the discrete ordinates method for applications in radiotherapy.
- access to microscopic cross-section libraries in various formats (DRAGLIB, WIMSLIB, APOLIB, MATXS, etc.)
- resonance self-shielding models (equivalence and subgroup)
- burnup calculations with the solution of the Bateman equations
- leakage and diffusion coefficients calculation
- superhomogénéisation (SPH) capabilities
- homogenization and condensation of cross sections and diffusion coefficients
- production of burnup-dependent multigroup cross section libraries with local-parameter branching
- capability to design computational schemes (such as two-level schemes).

Important remark:

- DRAGON and DONJON codes are a collection of independant modules, each of them performing a single task.
- The distribution also includes tools to generate computational schemes and multiphysics components.

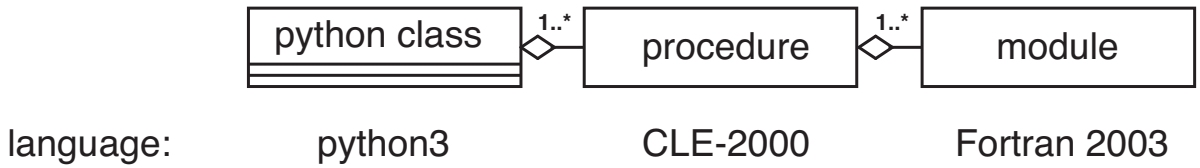
The DONJON full-core simulation code

DONJON5 is a full-core simulation code with the following capabilities:

- solution of the neutron diffusion equation
- solution of the simplified Pn (or SPn) equation
- multi-parameter interpolation in the cross-section database
- micro-depletion of particular isotopes (Xe, Sm, etc.)
- Boron critical control
- simplified thermo-hydraulics (steady state and transient) for CANDU and PWR
- management of reactivity devices for CANDU and PWR
- simulation of refuelling strategies in CANDU and PWR, including in-line fuelling in CANDU reactors
- time averaged CANDU core calculations
- pin-flux reconstruction capability in PWR
- 3D neutron kinetics
- capability to design computational schemes (burnup cycles, accident scenario, etc.).

Conditions of use

- Chaining of modules and generation of computational schemes are made using the scripting language CLE-2000.
- Encapsulation of multiphysics components into python3 classes is possible using the PyGan API (based on distutils utility).
- Polytechnique Montréal doesn't provide validated computational schemes nor multiphysics components for production use.
- Computational schemes are entirely written with CLE-2000 scripting syntax.
- Computational schemes are specific to each type of reactor or application and contains the intellectual property (IP) of the users.
- Computational schemes are not subject to the LGPL license of DRAGON.
  - This distinction is possible because DRAGON5 and DONJON5 are released under the lesser form of the license and not under the more restrictive GPL version of it.
  - The drawback of this approach is that users need to learn the capability to build their own computational schemes to use the code.
  - This requires much more know-how than using competing codes such as CASMO5.



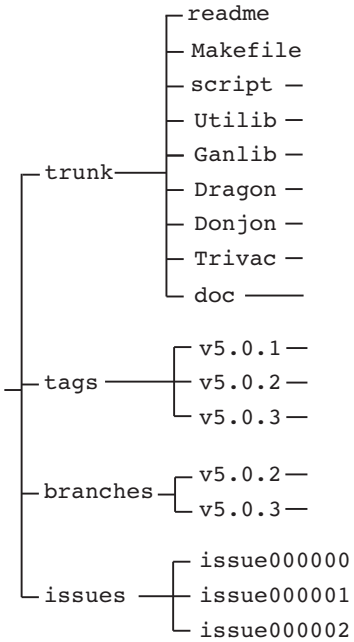
Conditions of use

- The long-term maintenance issue is important in this context as current CANDU reactors are expected to remain in operation until 2060.
- An agreement has been negotiated between COG and Polytechnique Montréal to provide the required commitment.
- The long-term maintenance issue is closely related to the quality assurance (QA) issue, as DRAGON5 and DONJON5 are developed under strict QA procedures.
  - All interactions between DRAGON5 and DONJON5 users and the development team and all modifications to the code resulting from these interactions are registered and supervised by the QA system.
  - Such an approach is essential in an academic context where student contributions could potentially affect the code stability.
  - Strict adherence to the QA rules is essential to maintain code stability in the long term.
- Five aspects of development procedures are described:
  1. version control of the project components
  2. issue tracking, spiral development management, and continuous integration
  3. configuration management of the code.
  4. continuous integration.
  5. distribution of tagged versions.

Quality assurance

Quality assurance (QA) is a combination of procedures:

1. Version control is the art of managing changes to information.
   - In 2003, we selected Subversion (svn) an open-source version control system (VCS) widely available at that time.
   - Subversion is a widely used system used to keep track of the historical evolution of the project.
   - We use svn for the totality of Version5 components. The information in the repository is organized with a directory structure, as shown in the figure.
   - The repository contains
     - Fortran 2018 and ANSI C sources
     - tagged version (issued once a year)
     - Makefiles for `gmake` and configuration python scripts
     - Bash and python scripts
     - basic CLE-2000 procedures
     - non-regression tests for the continuous integration procedure
     - LaTeX documentation
     - issue tracking information (QA database for accepted increments).



Quality assurance

2. Issue tracking management (QA) is provided using pre- and post-commit python scripts based on the pysvn API
   - Pre- and post-commit scripts have been added to the repository to validate commit operations and to automatically perform the second automatic issue-tracking commit.
   - Both scripts are written in python and are based on the pysvn application programming interface (API).
   - This information is recovered as a directory named `issues_wc` containing a set of card index, each of them representing a development issue.
   - If an issue submission form is accepted, the information relative to the issue is stored in file `issuennnnnn`. This file is the card index (fiche d'intervention in French) characterizing the development request.

Quality assurance

2. Example: Cyclic tracking in hexagonal geometry

```
Card-index: issue000251
-------------------------------------------------------
develop
Sat Mar  5 07:41:32 2022
subversion revision=2345
issue000251: Implement cyclic tracking of equilateral triangular geometry
             in module SALT:
M /trunk/Dragon/src/XELTSA.f
A /trunk/Dragon/data/salmacro_proc/TDCM54.c2m
M /trunk/Dragon/src/SAL_GEOMETRY_MOD.f90
M /trunk/Dragon/src/SALACG.f90
M /trunk/Dragon/data/salmacro.x2m
M /trunk/Dragon/src/SAL_AUX_MOD.f90
M /trunk/Dragon/src/g2s_generatingSAL.f90
M /trunk/Dragon/src/NXTTCG.f
M /trunk/Dragon/src/SALTCG.f
M /trunk/doc/IGE335/Section3.90_salt.tex
M /trunk/Dragon/src/g2s_g2s.f90
M /trunk/Dragon/src/XELTS2.f
A /trunk/doc/IGE335/hex_tspc.eps
A /trunk/doc/IGE335/cart_tspc.eps
M /trunk/Dragon/src/SALTLC.f90
A /trunk/Dragon/data/salmacro_proc/439NR_SSH.sal
M /trunk/Dragon/src/SAL_TRAJECTORY_MOD.f90
M /trunk/Dragon/data/salmacro_proc/TDCM50.c2m
M /trunk/Dragon/src/NXTQAC.f
M /trunk/Dragon/data/salmacro.access
-------------------------------------------------------
develop
Thu Mar 10 07:29:24 2022
subversion revision=2353
issue000251: Forgot to commit a file
M /trunk/Dragon/src/XCWSCL.f
-------------------------------------------------------
```

Quality assurance

2. Example: Cyclic tracking in hexagonal geometry

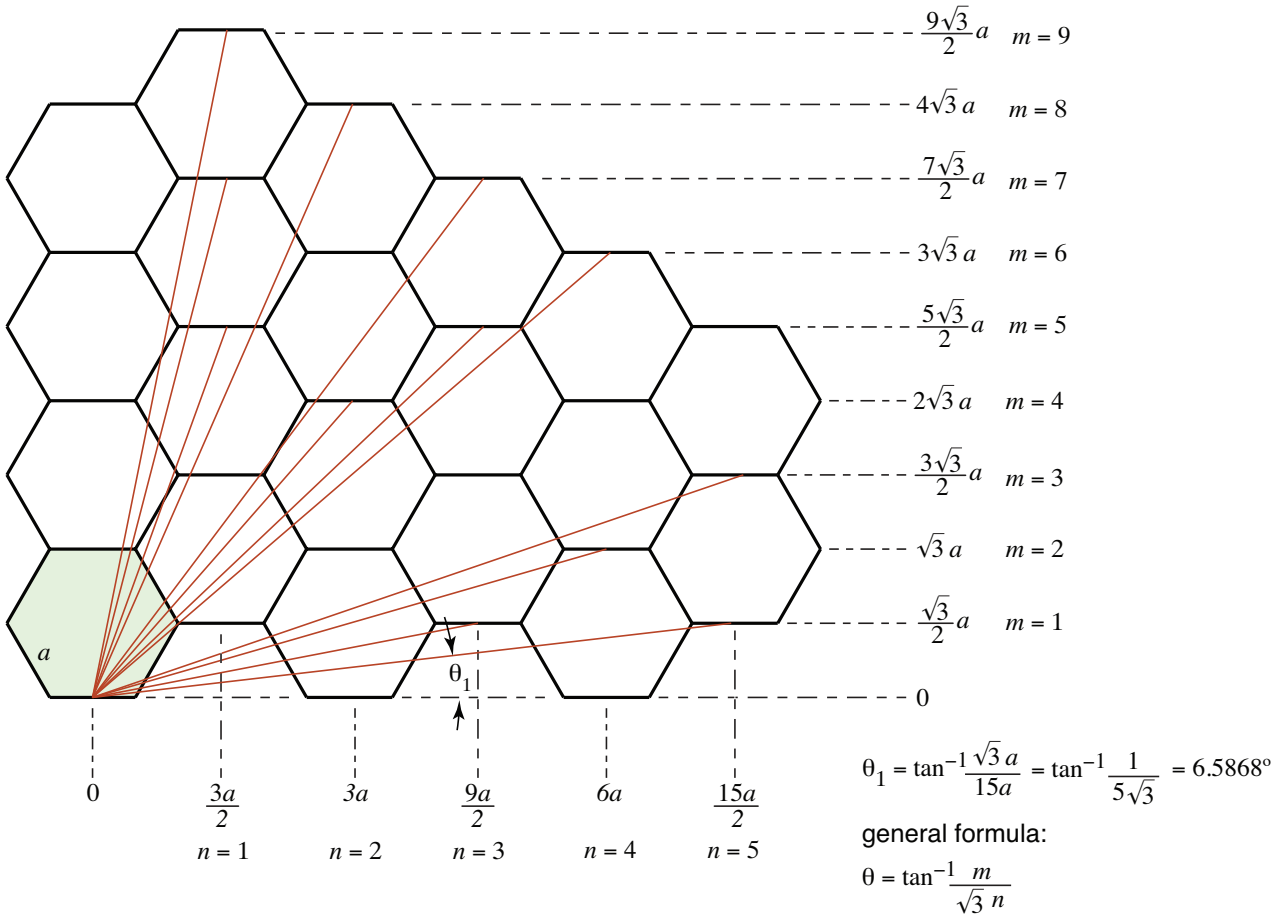


Figure 1: /doc/IGE335/hex_tspc.eps

Quality assurance

3. Configuration management is the art of assembling the project components, available in the repository, to build the end product of the project.

- In case of Version5, the end product is a set of executables for codes DRAGON, TRIVAC and DONJON on different UNIX-like operating systems (including PCs under Cygwin or WSL) and a set of PDF reports.
- The basic principle of Version5 configuration management consists of executing `make` scripts (`gmake` is used) within the user's or developer's working copy.
- For example, an executable of code DRAGON v5.0.8 can be produced using

```
tar xvfzp version5_v5.0.8.tgz
cd Version5_beta_ev2420/Dragon
make
```

  - The installation of DRAGON5 includes the installation of GANLIB5 and TRIVAC5.
  - Similarly, the installation of DONJON5 includes the installation of DRAGON5.
- Each documentation directory has its own install script.

Quality assurance

4. Continuous integration.
   - A set of selected non-regression tests are performed with the developer's working copy. A continuous integration procedure is implemented using `make tests`.
   - If these tests are conclusive, the issue is closed and an issue closing report is written, appended to the card-index named `issuennnnnn`.
   - References to the issue-related documentation are also added to the card index.
   - The issue identifier is of the form `issuennnnnn` where nnnnnn is equal to the maximum existing value plus-one, as assigned automatically by the pre-commit script.
   - At any time during the cycle, file `issuennnnnn` can be updated by the developer in charge of the issue and recommitted as

```
cd  issues_wc
svn commit -m 'issuennnnnn:' .
```

   - A cycle may require many commits. After each commits of a Version5 item, file `issuennnnnn` is automatically updated and re-committed by the post-commit script. The issue card-index trace the progress of the work done by the developer(s) to solve the issue.

Quality assurance

5. Distribution of tagged versions

- Once a year, all project increments are collected into a tagged version identified as v5.n.m. and made available on the official project website. The hyperlink “what’s new” is a list of issues references and short descriptions relative to this tagged version.

  - Version5 beta archive. To expand the archive, type "`tar xvfz version5_v5.0.1.tgz`".

tagged version 5.0.1	tgz	2014/12/17	
tagged version 5.0.2	tgz	2016/02/02	what's new
tagged version 5.0.3	tgz	2017/02/24	what's new
tagged version 5.0.4	tgz	2018/04/22	what's new
tagged version 5.0.5	tgz	2019/01/18	what's new
tagged version 5.0.6	tgz	2020/02/01	what's new
tagged version 5.0.7[1]	tgz	2021/02/02	what's new
tagged version 5.0.8	tgz	2022/04/20	what's new

1: Version 5.0.7 is Fortran 2018 compatible.

Statistics

- Number of lines of code (as of June 2022):

  UTILIB: 14,302 including 3,693 lines of ANSI C code
  GANLIB: 29,929 including 13,083 lines of ANSI C code
  TRIVAC: 55,106
  DRAGON: 273,976
  DONJON: 78,993
  PyGAN: 2,142

- Number of commits: (as of June 2022):

  VERSION4: 2,489 representing 351 issues
  VERSION5: 2,523 representing 265 issues

- Prerequisites
  - UNIX or Linux OS
  - Fortran 2003 compiler
  - python2 or python3
  - HDF5 C API (optional)
  - LaTeX (optional)

Using DRAGON5

Installation of DRAGON5 requires:

1. Installation of DRAGON5 executable

```
tar xvfzp version5_v5.0.8.tgz
cd Version5_beta_ev2420/Dragon
make
```

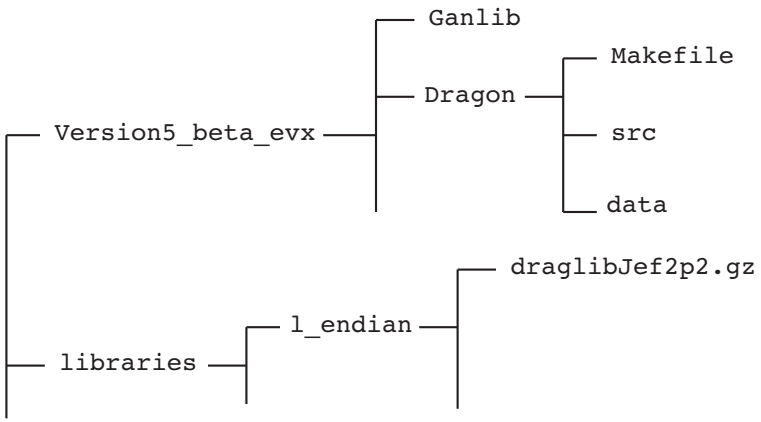
2. Installation of cross-section libraries:
   - Download little-endian draglibs from website
   - Copy libraries into directory `libraries/l_endian/`

3. Execution of non-regression tests

```
cd Version5_beta_ev2420/Dragon
make tests
```

4. Execution of a single dataset

```
cd Version5_beta_ev2420
cd Dragon
./rdragon tdraglib.x2m
```



Conclusion

- The DRAGON5/DONJON5 system of codes is an OpenSource project available for reactor physics applications.
- Can be inslalled and used easily on any UNIX-type operating system.
- Users need to construct their own computational schemes and can put a proprietary license on them.
- The next major version, labeled 5.1.0, will be issued in February 2023 and hosted on the NEA/OECD GitLab data bank.

Ressources

- Academic:
  Guy Marleau(guy.marleau@polymtl.ca)
  Alain Hébert (alain.hebert@polymtl.ca)
  Richard Chambon (richard.chambon@kinectrics.com)
- Merlin website:
  DRAGON5/DONJON5: `http://merlin.polymtl.ca`
- Archives website:
  Useful informations (including student contributions):
  `http://merlin.polymtl.ca/archives.htm`
- Textbook:
  Alain Hébert, Applied Reactor Physics,
  Presses Internationales Polytechnique,
  Thirdedition, Montréal, 2020.