



INSTITUTION PROJECT  
CENTER ITER  
ROSATOM

# Развитие Программных Комплексов Для Нейтронно-физического Моделирования Термоядерных Установок

*Д.В. Портнов, Е.С. Афанасенко, Ю.Г. Высоких*

Нейтронно-физические проблемы атомной энергетики, май 2024

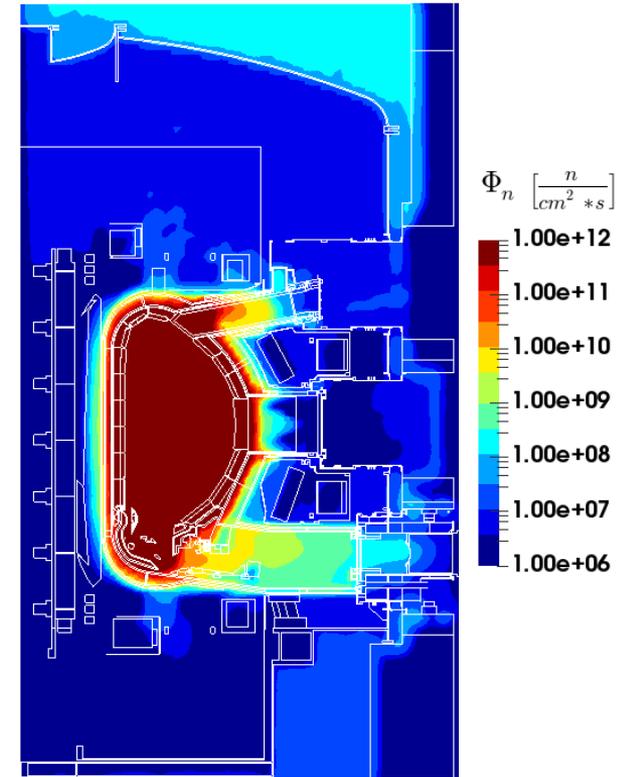
*Секция 5, Численные методы, алгоритмы и программы нейтронно-физических расчетов*

**Дмитрий Портнов** *d.portnov@iterrf.ru*

Проектный центр ИТЭР, сектор нейтронно-физического моделирования

# Основные проблемы

1. Мы обязаны использовать MCNP, поскольку это стандарт международной организации. Но у нас лицензия на MCNP закончилась в прошлом году. По этой причине (*и по целому ряду других*) нам нужны альтернативные программы для расчета **нейтронных и фотонных полей (транспорт)**.
2. Нам нужно автоматизировать **синхронизацию эквивалентных моделей** для альтернативных программ.
3. После транспорта мы рассчитываем **динамику активации материалов и поля остаточной мощности дозы**. Наши программы требуют масштабирования для работы с большими объемами данных.



ИТЭР, нейтронный поток,  $P_{\gamma}=0$

- OpenMC – современный код для моделирования полей нейтронов и фотонов, с развитой функциональностью, в частности, поддерживает расчеты с изменяющейся средой (depletion).
- OpenMC использует ту же логику спецификации геометрии CSG (Construction Solid Geometry), что и MCNP. Это упростило создание конвертеров как из MCNP, так и непосредственно из 3D моделей CAD.
- Разрабатывается в MIT с 2011 года. Open source: <https://github.com/openmc-dev/openmc> (120 основных разработчиков, 29 выпусков за 13 лет - исключительно высокий темп развития)

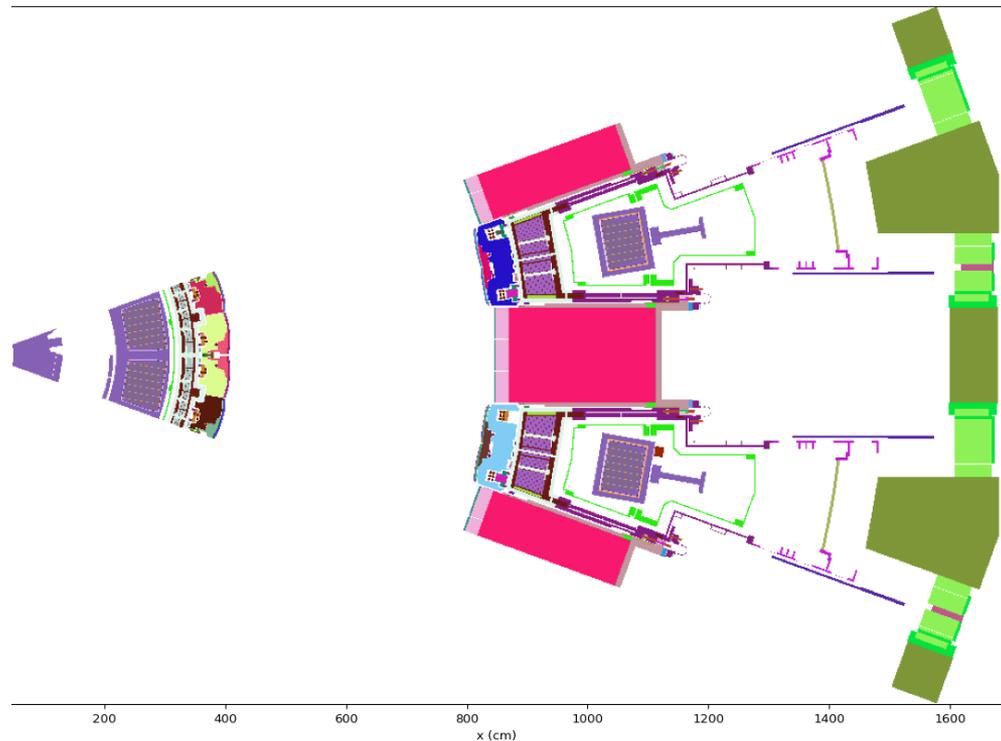
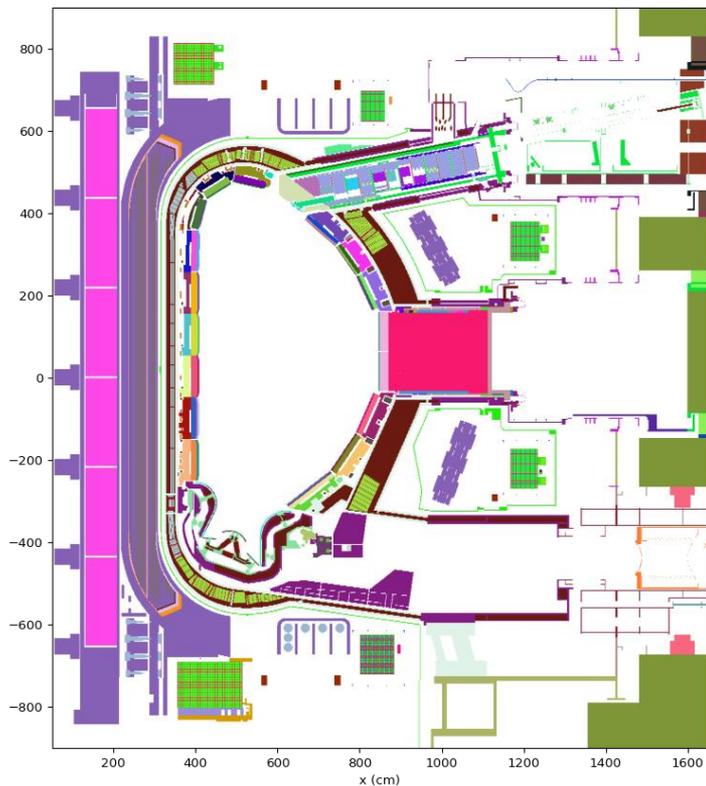
$$\frac{dN_i(t)}{dt} = \sum_j \left[ \underbrace{f_{j \rightarrow i} \int_0^\infty dE \sigma_j(E, t) \phi(E, t)}_{\text{transmutation}} + \underbrace{\lambda_{j \rightarrow i}}_{\text{decay}} \right] N_j(t) - \left[ \underbrace{\int_0^\infty dE \sigma_i(E, t) \phi(E, t)}_{\text{transmutation}} + \underbrace{\sum_j \lambda_{i \rightarrow j}}_{\text{decay}} \right] N_i(t)$$

Production of nuclide  $i$  from nuclide  $j$

Loss of nuclide  $i$

*Depletion: динамика трансмутации и распада нуклидов учтена в транспорте*

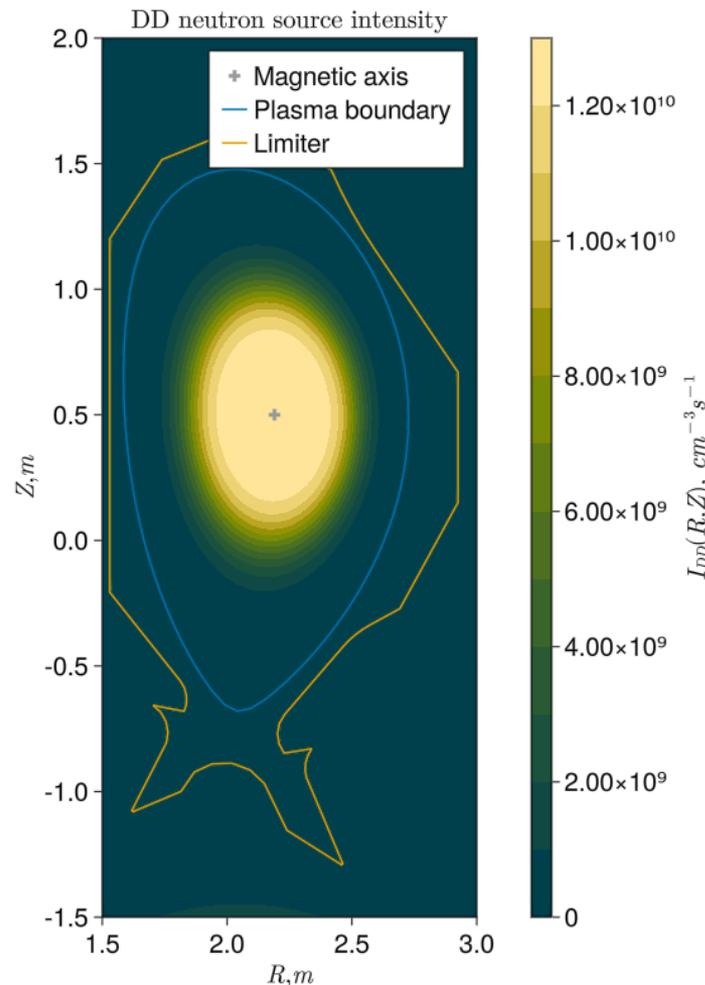
# C-модель ИТЭР в OpenMC [https://github.com/openmc-dev/openmc\\_mcnp\\_adapter](https://github.com/openmc-dev/openmc_mcnp_adapter)



- 40° модель токамака-реактора ИТЭР, конвертированная из MCNP в OpenMC (**114 285** ячеек в базовой модели C-model, E-lite 360°- сотни тысяч ячеек)

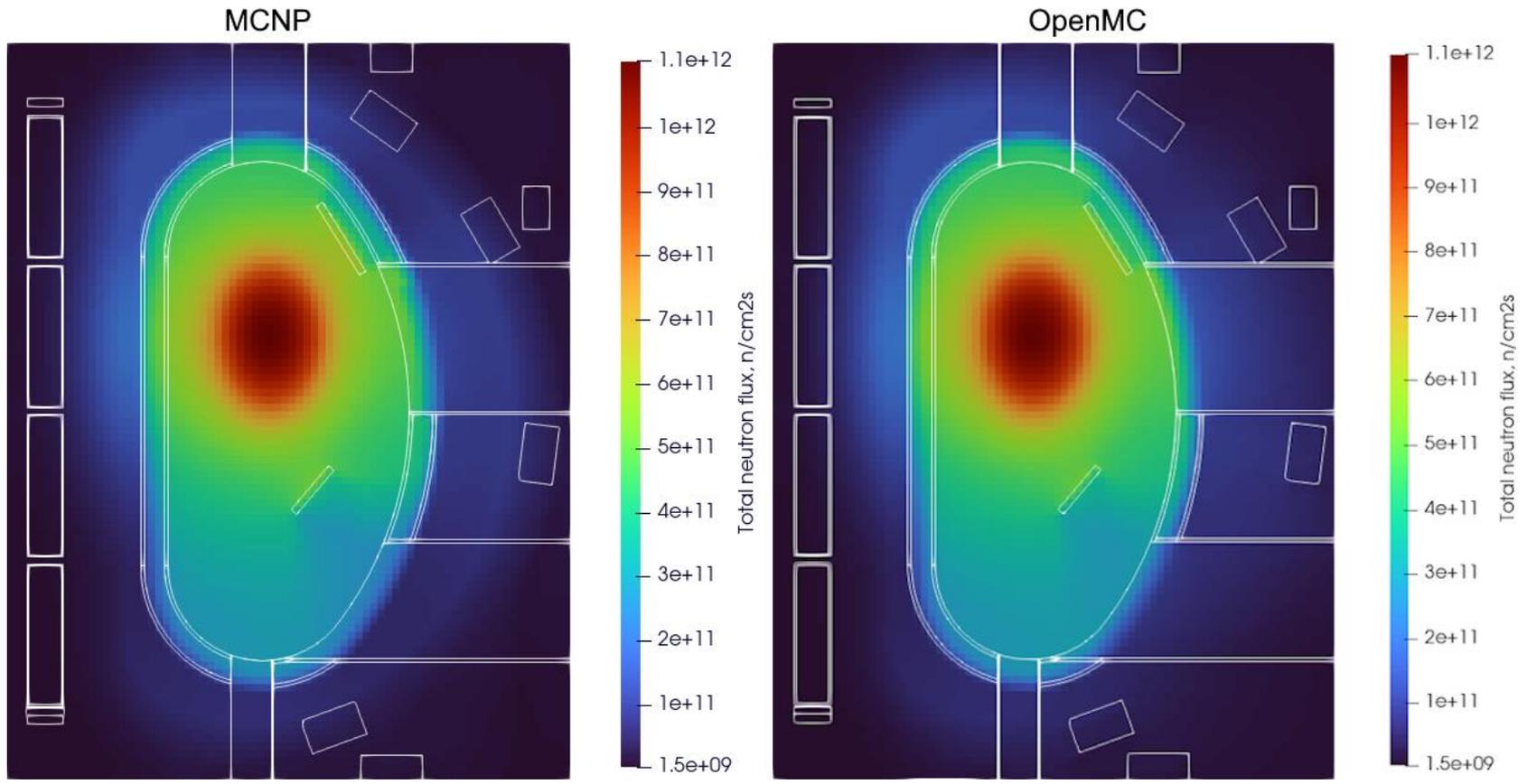
# Нейтронный источник

- Разработано ПО:
  - EQDSK Reader <https://github.com/dvp2015/EQDSKReader.jl>
  - Нейтронный источник <https://github.com/dvp2015/TokamakNeutronSource.jl>
  - Конвертер <https://github.com/MC-kit/parametrized-tokamak-source.git>
- Мы можем воспроизвести нейтронный источник по заданному в EQDSK файле состоянию плазмы или по параметризованному представлению, как это сделано в ИТЭР.
- Была разработаны модели (OpenMC, MCNP) источника для равновесной плазмы в TRT для значений тороидального поля на магнитной оси 8Т, ток 5МА.
- Была разработана OpenMC модель источника ИТЭР



DD источник для TRT

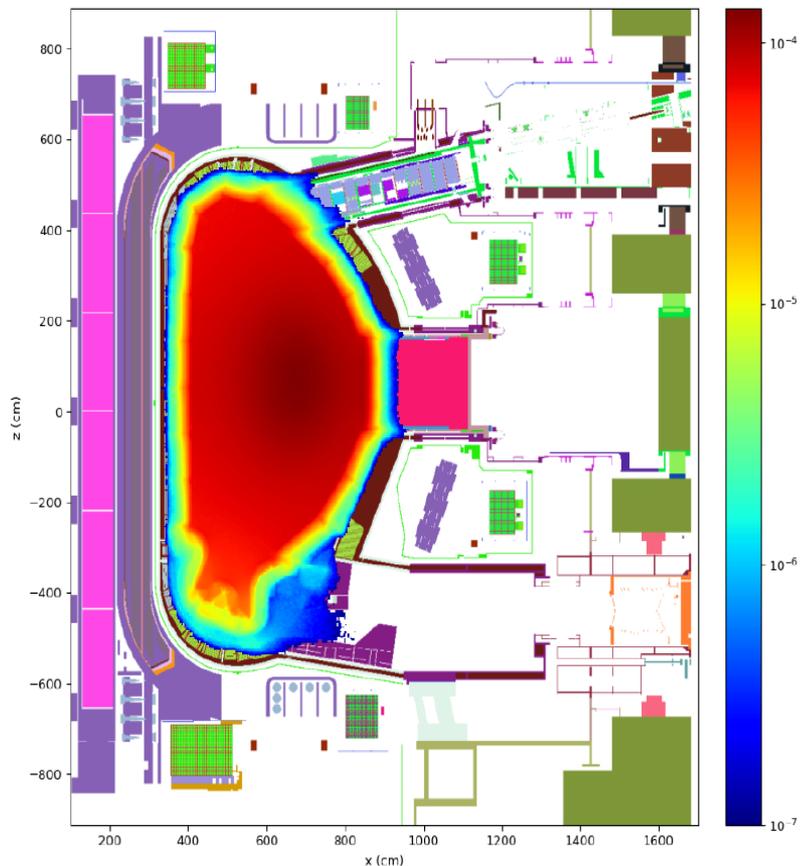
# Нейтронный поток TRT: MCNP vs. OpenMC



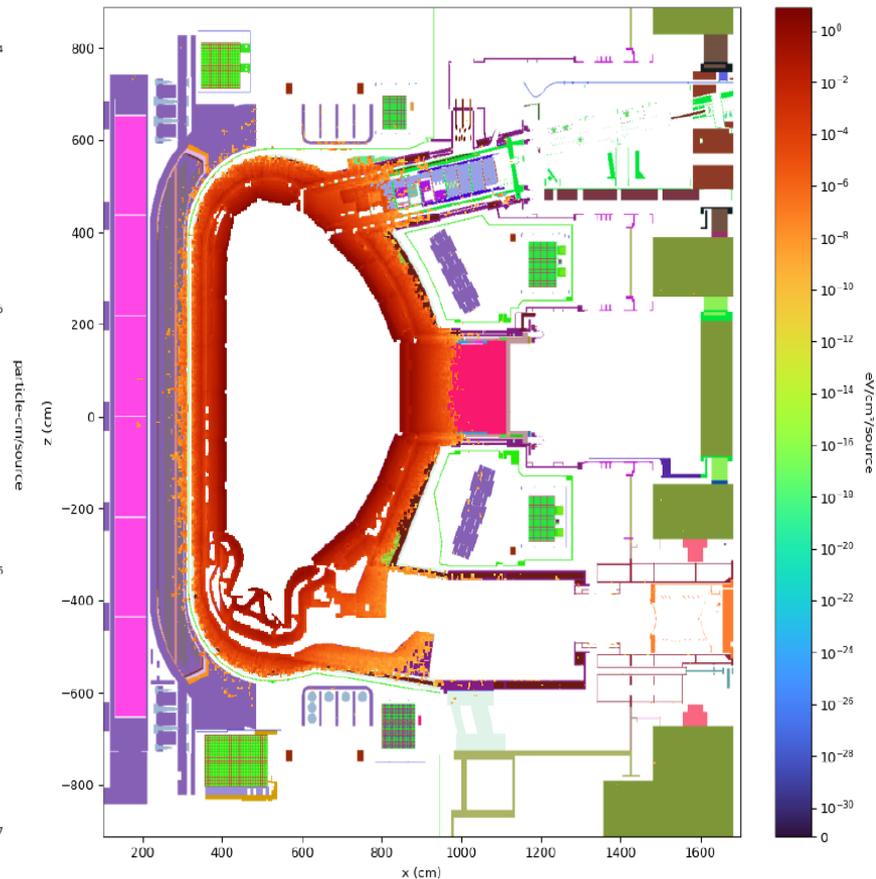
Невооруженным глазом различий не видно, однако, они есть и с этим нам предстоит разбираться.

# Предварительные результаты для ИТЭР (OpenMC)

Total neutron flux, n-cm/src



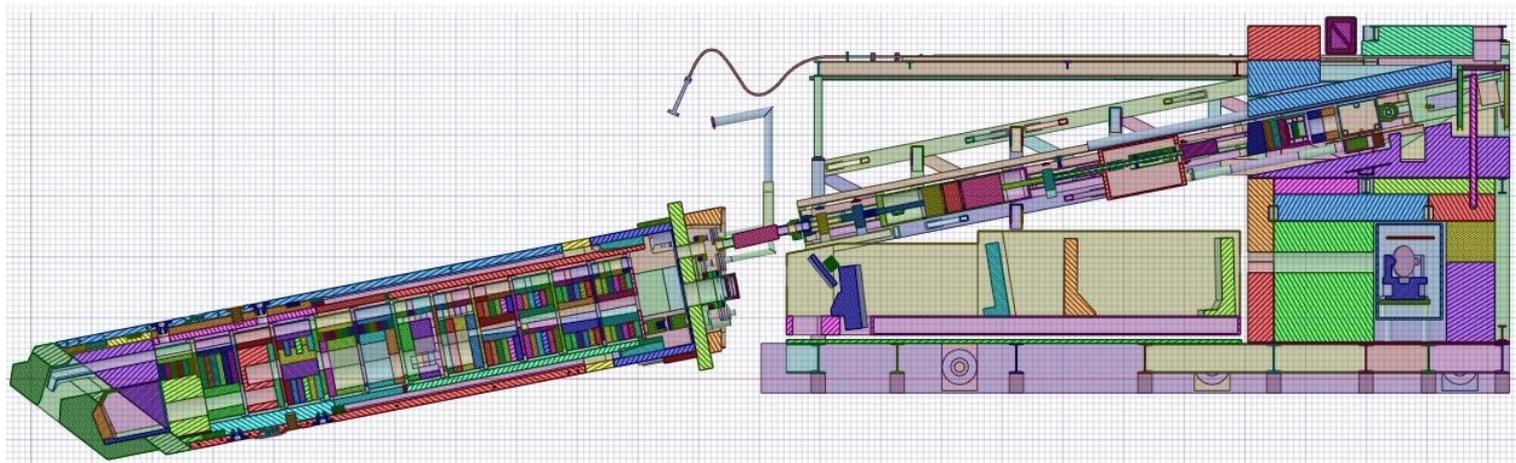
Heating, eV/cm<sup>3</sup>/src



# Расчет активации

**FISPACT** – это программа, которая для заданного материала, по заданному сценарию облучения и спектру нейтронов, рассчитывает динамику изотопного состава и интенсивности источника фотонов (*учитывает ~4000 нуклидов*).

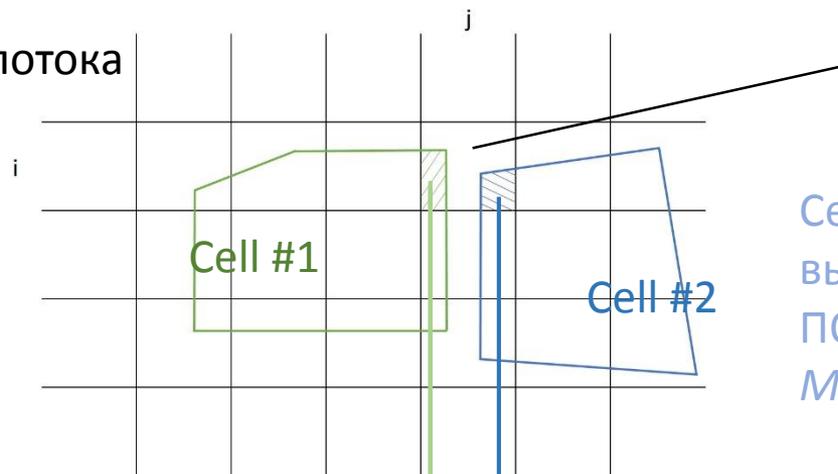
По сути, **R2S** (Rigorous 2-Step) для каждого пересечения ячейки модели Монте-Карло и ячейки (вокселя) нейтронного меша создает входные данные, прогоняет FISPACT и представляет результат в разной степени агрегации: по пересечениям, ячейкам, компонентам.



*Заглушка и межпортовое пространство верхнего порта №8 + нейтронный меш*

# Пересечения ячеек-воксел и результаты

Меш нейтронного потока



$i,j,k$  - воксел

Cell-Voxel пересечения  
вычислены и **упрощены**  
ПО **mckit** (*Quine-  
McClaskey, Espresso*)

Таблица нуклидов

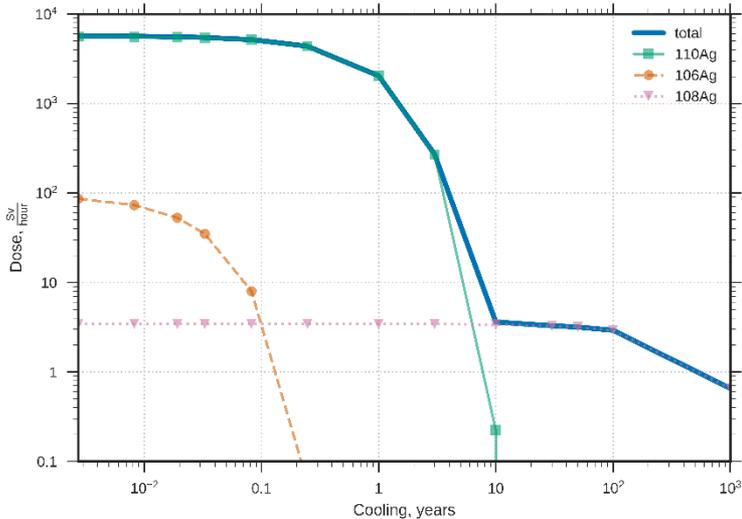
| cell | timestep | zai   | atoms | activity | ... |
|------|----------|-------|-------|----------|-----|
| ...  |          |       |       |          |     |
| 1    | 45       | 10030 | ....  |          |     |
| ...  |          |       |       |          |     |
| 2    | 45       | 10030 | ...   |          |     |
| ...  |          |       |       |          |     |

ФОТОННЫЙ ИСТОЧНИК

| timestep | cell | i   | j | k | rate |
|----------|------|-----|---|---|------|
| ...      |      |     |   |   |      |
| 45       | 1    | ... |   |   |      |
| ...      |      |     |   |   |      |
| 45       | 2    | ... |   |   |      |
| ...      |      |     |   |   |      |

# Что в результате?

Таблица нуклидов:  
динамика,  
химические особенности и т.д.

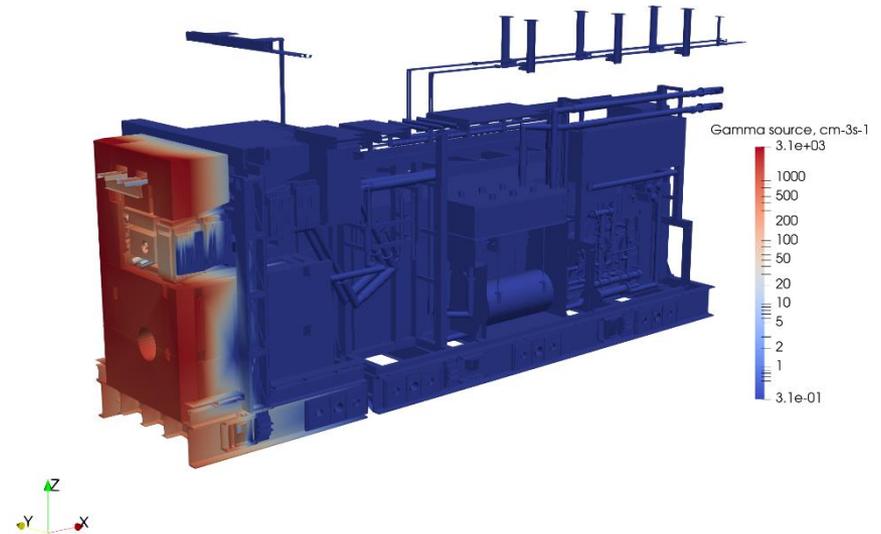


Динамика вклада в дозу нуклидов серебра

## Фотонный источник и SDEF (с vtr-file)

C total gamma intensity = 2.65482e+10

SDEF PAR=2 EFF=0.001 **CEL=D1** ERG=FCEL D2 X=FCEL D3 Y=FCEL D4 Z=FCEL D5  
SI1 L 10288 10288 10288 10288 10288 10288 10288 10288 10288 10288



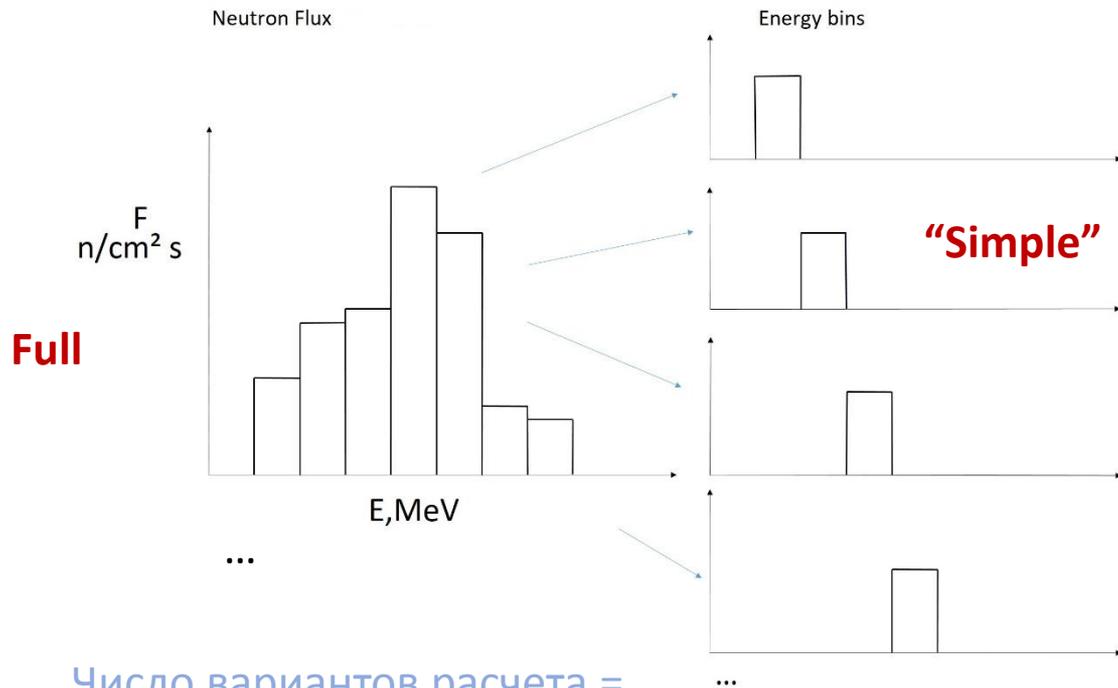
Распределение интенсивности гамма-источника на оборудовании в припортовой камере экваториального порта 11

# “Simple” декомпозиция по энергетическим группам

Нейтронный поток

Декомпозиция по группам

Материалы модели



Material 1

```
M100 6012.31c 1.16743e-004 $C 12 weight(%) N/A ab(%) 98.93
6013.31c 1.26266e-006 $C 13 weight(%) N/A ab(%) 1.07
25055.31c 1.56467e-003 $MN 55 WEIGHT(%) 1.8000
```

Material 2

```
M200 14028.31c 7.05139E-003 $Si 28 weight(%) 15.5876 ab(%) 92.23
14029.31c 3.58049E-004 $Si 29 weight(%) 15.5876 ab(%) 4.68
14030.31c 2.36027E-004 $Si 30 weight(%) 15.5876 ab(%) 3.09
```

Х

Material 3

```
M400 1001.31c 6.33910e-002 $H 1 AMOUNT(%) 2.0000 AB(%) 99.99
1002.31c 7.29080e-006 $H 2 AMOUNT(%) 2.0000 AB(%) 0.01
8016.31c 3.15453e-002 $O 16 weight(%) N/A ab(%) 99.76
8017.31c 1.20164e-005 $O 17 weight(%) N/A ab(%) 0.04
```

...

Число вариантов расчета =  
Число групп x число  
материалов

А потом все нужно  
**аккуратно** посчитать для  
каждого пересечения

Число вариантов расчета =  
числу пересечений

# Polars – экспорт результатов FISPACT в файлы parquet

1. Polars – это высокопроизводительная библиотека (Rust) для работы с таблицами.
2. Пример: данные FISPACT 50Gb (для *небольшой задачи*), в файлах parquet занимают 300Mb (1:170).
3. Файлы parquet эффективно загружаются в БД.

The image shows a terminal window with three screenshots of a file system. The first screenshot shows a directory listing of files named 'case=1' through 'case=170'. The second screenshot shows the command 'du -s' being executed, with the output '50G' circled in red. The third screenshot shows the directory listing of files in the 'intermediate' directory, including 'gbins.parquet', 'nuclide.parquet', 'rundata.parquet', 'time\_step\_times.parquet', 'timestep.parquet', and 'timestep\_gamma.parquet'. The output of 'du -s' in this directory is '288M', also circled in red.

<https://docs.pola.rs/>

<https://parquet.apache.org/docs/overview/motivation/>

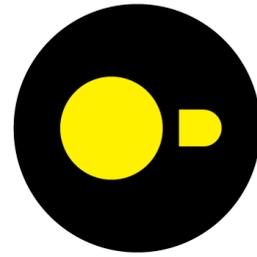
# DuckDB – для анализа данных

1. Polars не только быстрый, но и представляет богатую функциональность, включая SQL сессии *(пример внизу)*.
2. Однако с DuckDB работать удобнее *(не нужно думать об индексации, есть мощный аналитический функционал, оптимизатор запросов и т.д.)*.

```
aligned_to_old = (  
  iss_cells  
  .select("iss_cell")  
  .with_columns(pl.col("iss_cell")  
  .cast(pl.UInt32).alias("cell"))  
  .join(new_data, on="cell", how="left")  
  .select(  
    pl.all().exclude("iss_cell")  
  )  
)  
aligned_to_old
```

shape: (24, 5)

| cell   | Activity, Bq/g | H3, Bq/g    | Contact Dose, Sv/h | PS Dose, Sv/h |
|--------|----------------|-------------|--------------------|---------------|
| u32    | f64            | f64         | f32                | f32           |
| 190506 | 278449.123471  | 24.184862   | 0.014855           | 8.6548e-9     |
| 190885 | 30506.496866   | 1181.469102 | 0.001662           | 9.3427e-10    |
| 191136 | 35644.784636   | 1123.327434 | 0.001696           | 9.7856e-10    |



# DuckDB

<http://duckdb.org/>

*Объединение двух таблиц в Polars: работает, но, если без SQL сессии, то код требует немало труда, и сложность быстро растет с числом объединений, фильтров и т.д.*

# **Заключение и выводы**

1. Разработаны **средства синхронизации** моделей MCNP и OpenMC, включая модель нейтронного источника. Мы в состоянии обеспечить эквивалентные модели.
2. **Приложение R2S** теперь в состоянии решать более сложные задачи.
3. Выбранные нами технологии для R2S **отлично работают для задач малого и среднего масштаба** – на отдельной рабочей станции. Предстоит сделать шаг к большому масштабу, т.е. к распределенным вычислениям.
4. Базы данных – это не только средство для анализа данных, но также и **средство коммуникаций**. Документов Word, Excel для этого недостаточно. Нужны базы данных, которые можно интегрировать между собой. Соответственно, нужны **согласованные**:
  - схемы данных,
  - процедуры загрузки и верификации,
  - программное обеспечение,
  - средства визуализации и навигации по иерархиям атрибутов,
  - организация версионного хранения
  - и **многое еще...**

# Благодарности

1. **Роману Родионову** – первому автору r2s и библиотеки mckit (<https://github.com/MC-kit/mckit>).
2. Команде **FISPACT** (<https://fispact.ukaea.uk>) за мощное и надежное средство вычислений активации и за предоставленное обучение.
3. **SpaceClaim, SuperMC, GEOUNED** – наши основные средства для работы с CAD моделями
4. **OpenMC** – за обучение и консультации
5. Проектный Центр ИТЭР и НИЦ Институт Курчатова за вычислительные мощности.

# Спасибо!

# Вопросы?

**Дмитрий Владиславович Портнов**

Тел.: +7 (910) 405 51 73  
E-mail: [d.portnov@iterrf.ru](mailto:d.portnov@iterrf.ru)  
[www.iterrf.ru](http://www.iterrf.ru)

**26.05.2024**





...

**Дополнительные  
слайды**

**1.**

# **OpenMC vs. MCNP...**

**альтернатива для «термоядерных» расчетов нейтронного транспорта**

# Входные данные для аналитического описания нейтронного источника (2)

- Параметризация магнитных поверхностей (МП) плазмы:

$$\begin{cases} R = R_p + \Delta_{sh}(\Psi) + a(\Psi)(\cos(t) - \delta(\Psi)\sin^2(t)) \\ Z = Z_p + a(\Psi)k(\Psi)\sin(t) \end{cases}$$

- Таблицы профилей параметров геометрии МП  $\Delta_{sh}(\Psi)$ ,  $a(\Psi)$ ,  $\delta(\Psi)$ ,  $k(\Psi)$
- Таблицы профилей ионной плотности и температуры  $n_i(\Psi)$ ,  $T_i(\Psi)$

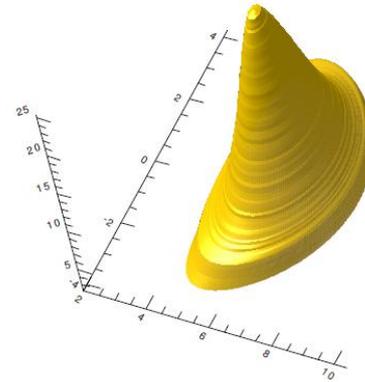


Fig.1-1 Ion temperature profile

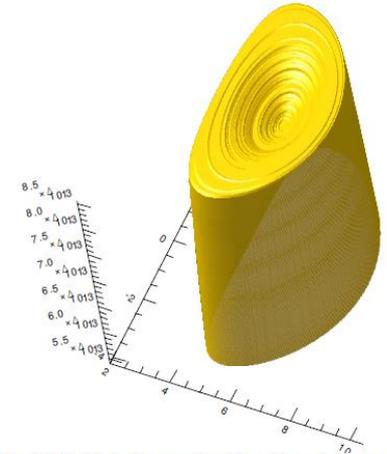
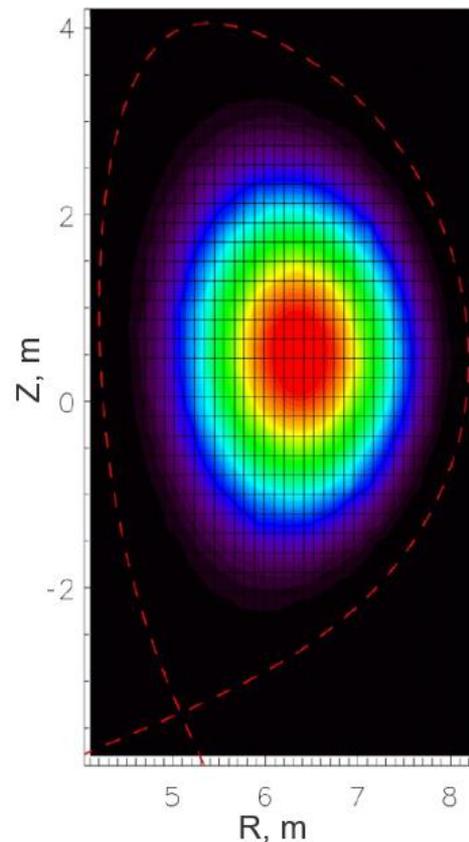


Fig.1-2 D+T density profile. The dip in the center of plasma is caused by plasma quasi-neutrality condition and the higher density of helium ions on

# Профиль источника ДТ-нейтронов ИТЭР

MCNP (E. Polunovskiy paper)



OpenMC (OpenMC-plotter)

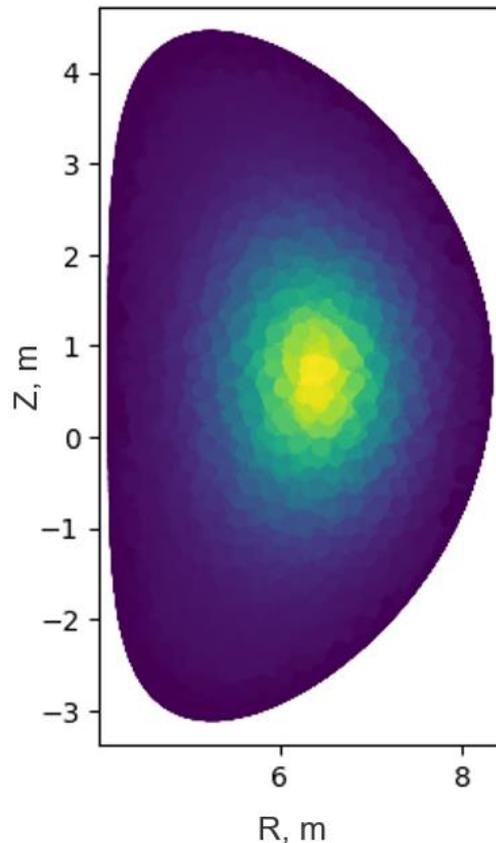
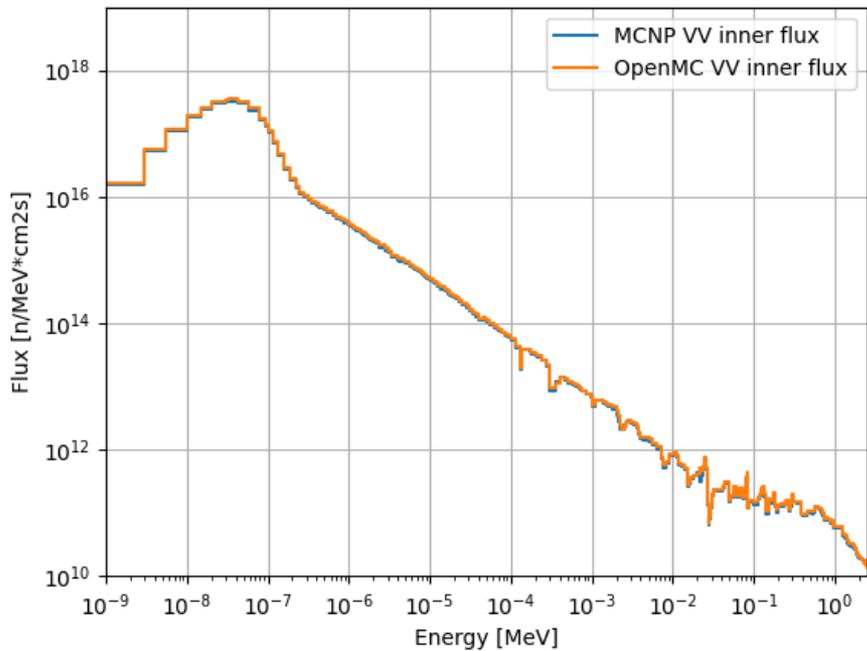


Таблица параметров плазмы

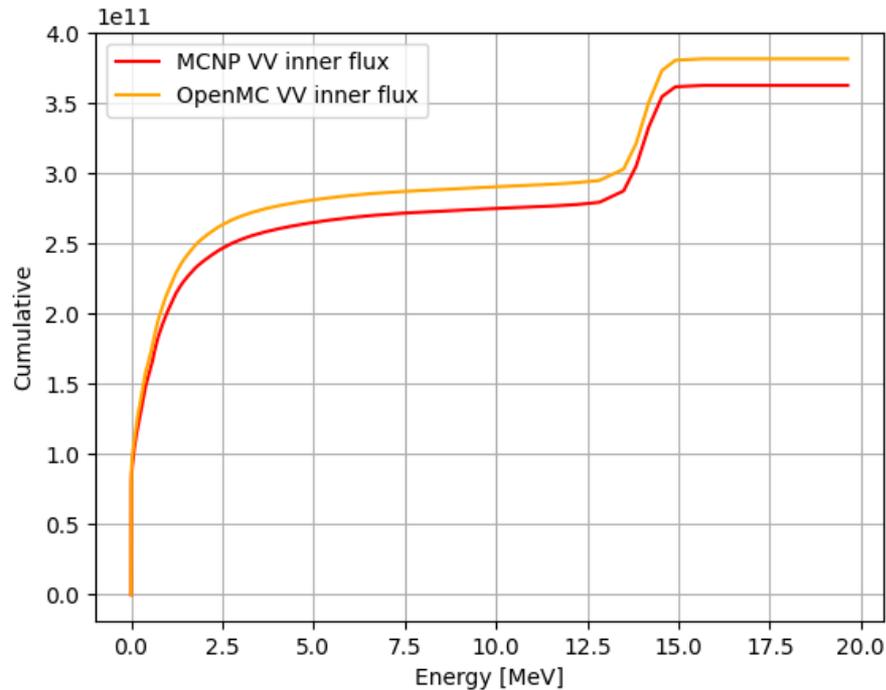
| $\sqrt{\Psi}$<br>normalized | $a$<br>[m] | $\Delta_{SH}$<br>[m] | $K$      | $\delta$  | $T_i$<br>[keV] | $N_{e,r}$<br>[ $10^{19} \text{ m}^{-3}$ ] |
|-----------------------------|------------|----------------------|----------|-----------|----------------|---|
| 6.16E-03                    | 1.400E-02  | 1.81E-01             | 1.45E+00 | 0.000E+00 | 2.12E+01       | 8.17E+00                                  |
| 2.18E-02                    | 4.442E-02  | 1.80E-01             | 1.45E+00 | 9.694E-03 | 2.12E+01       | 8.17E+00                                  |
| 3.22E-02                    | 7.484E-02  | 1.80E-01             | 1.45E+00 | 1.163E-02 | 2.12E+01       | 8.17E+00                                  |
| 4.79E-02                    | 1.053E-01  | 1.80E-01             | 1.45E+00 | 1.551E-02 | 2.12E+01       | 8.17E+00                                  |
| 5.83E-02                    | 1.255E-01  | 1.80E-01             | 1.45E+00 | 1.939E-02 | 2.12E+01       | 8.17E+00                                  |
| 6.87E-02                    | 1.560E-01  | 1.80E-01             | 1.45E+00 | 2.520E-02 | 2.12E+01       | 8.18E+00                                  |
| 8.44E-02                    | 1.864E-01  | 1.79E-01             | 1.45E+00 | 2.714E-02 | 2.09E+01       | 8.18E+00                                  |
| 9.48E-02                    | 2.168E-01  | 1.78E-01             | 1.46E+00 | 2.908E-02 | 2.07E+01       | 8.18E+00                                  |
| 1.05E-01                    | 2.371E-01  | 1.77E-01             | 1.46E+00 | 2.908E-02 | 2.05E+01       | 8.19E+00                                  |
| 1.21E-01                    | 2.675E-01  | 1.77E-01             | 1.46E+00 | 2.908E-02 | 2.04E+01       | 8.19E+00                                  |
| 1.31E-01                    | 2.979E-01  | 1.76E-01             | 1.46E+00 | 2.908E-02 | 2.01E+01       | 8.21E+00                                  |
| 1.42E-01                    | 3.182E-01  | 1.74E-01             | 1.46E+00 | 2.908E-02 | 1.99E+01       | 8.21E+00                                  |
| 1.57E-01                    | 3.487E-01  | 1.74E-01             | 1.46E+00 | 2.908E-02 | 1.96E+01       | 8.22E+00                                  |
| 1.68E-01                    | 3.791E-01  | 1.72E-01             | 1.46E+00 | 2.908E-02 | 1.94E+01       | 8.23E+00                                  |
| 1.84E-01                    | 4.095E-01  | 1.71E-01             | 1.46E+00 | 2.908E-02 | 1.91E+01       | 8.23E+00                                  |
| 1.94E-01                    | 4.298E-01  | 1.70E-01             | 1.46E+00 | 3.102E-02 | 1.89E+01       | 8.24E+00                                  |
| 2.04E-01                    | 4.602E-01  | 1.69E-01             | 1.46E+00 | 3.490E-02 | 1.85E+01       | 8.26E+00                                  |
| 2.20E-01                    | 4.906E-01  | 1.67E-01             | 1.46E+00 | 4.072E-02 | 1.82E+01       | 8.27E+00                                  |
| 2.30E-01                    | 5.109E-01  | 1.66E-01             | 1.46E+00 | 4.459E-02 | 1.80E+01       | 8.27E+00                                  |
| 2.41E-01                    | 5.413E-01  | 1.63E-01             | 1.46E+00 | 4.653E-02 | 1.76E+01       | 8.28E+00                                  |
| 2.57E-01                    | 5.718E-01  | 1.62E-01             | 1.46E+00 | 4.653E-02 | 1.73E+01       | 8.29E+00                                  |
| 2.67E-01                    | 5.920E-01  | 1.61E-01             | 1.46E+00 | 4.847E-02 | 1.70E+01       | 8.31E+00                                  |
| 2.77E-01                    | 6.225E-01  | 1.58E-01             | 1.46E+00 | 4.847E-02 | 1.66E+01       | 8.31E+00                                  |
| 2.93E-01                    | 6.529E-01  | 1.57E-01             | 1.46E+00 | 5.041E-02 | 1.63E+01       | 8.32E+00                                  |

# Нейтронный поток в вакуумной камере TRT (внутр. слой), MCNP vs. OpenMC

«ВК внутр.» спектр

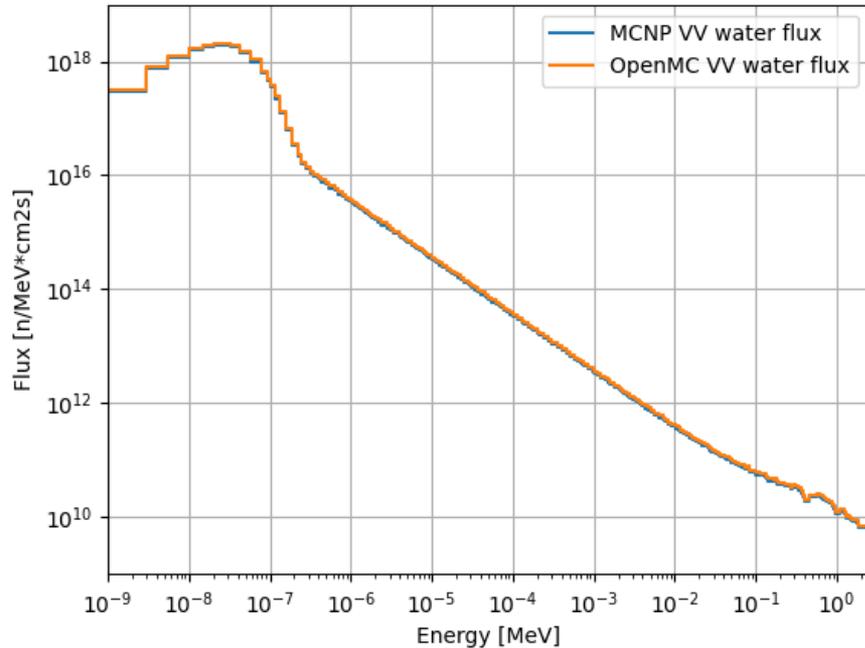


«ВК внутр.» кумулятивное распределение

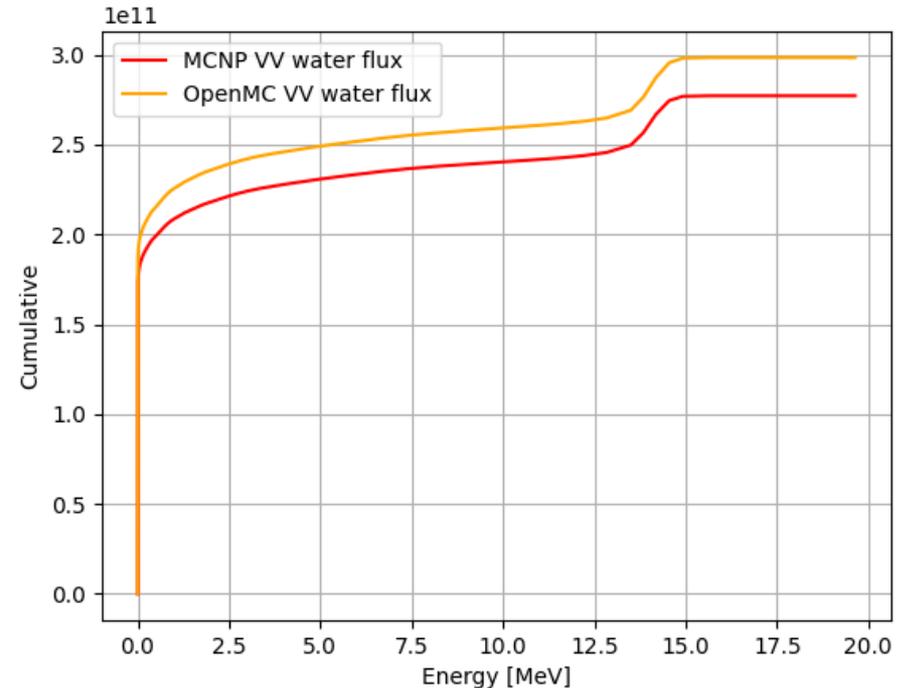


# Нейтронный поток в вакуумной камере TRT (слой «вода»), MCNP vs. OpenMC

«ВК вода», спектр



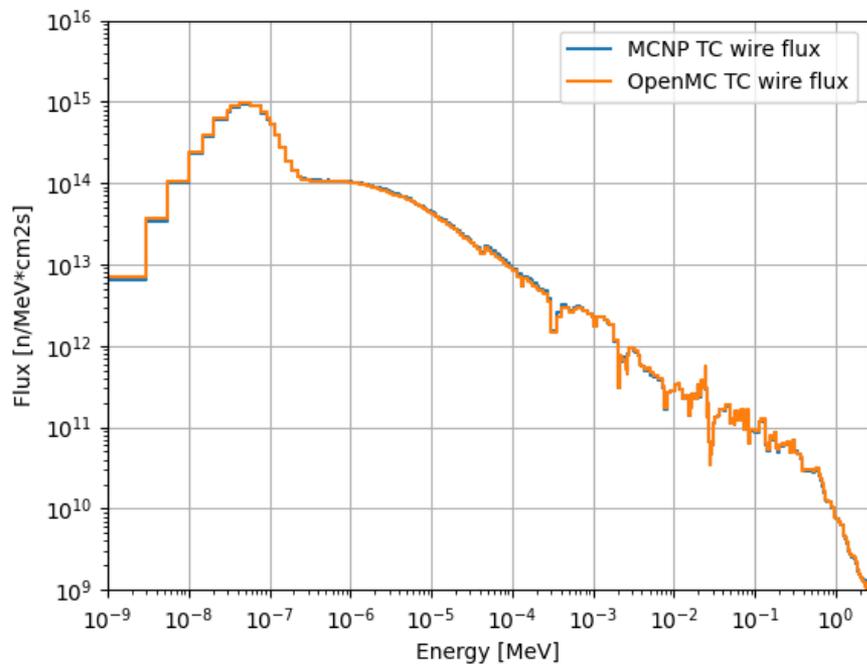
«ВК вода» кумулятивное распределение



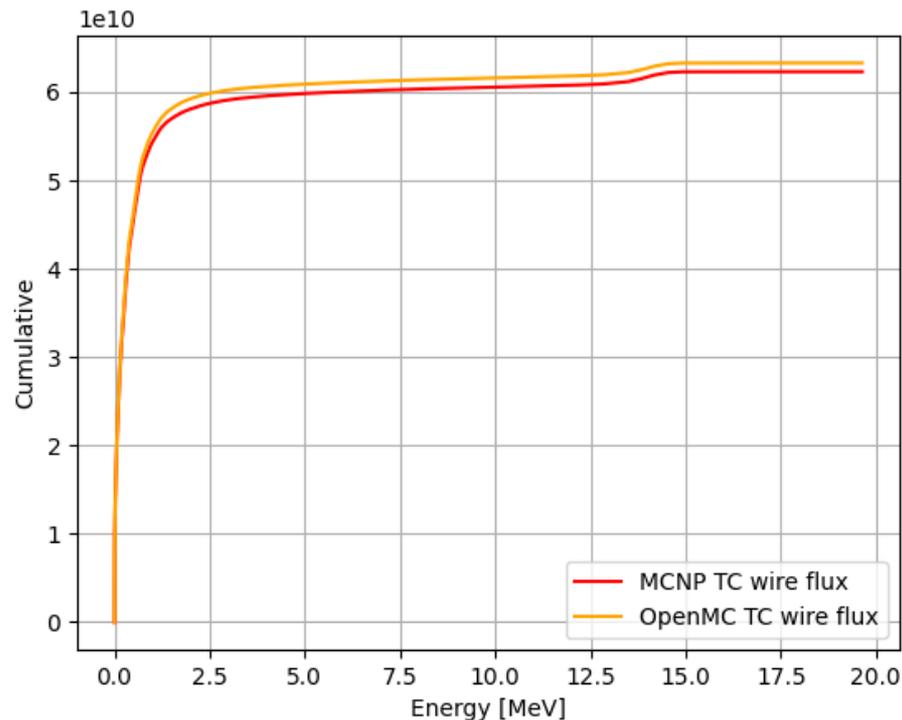
Использовали одну и ту же библиотеку (FENDL3.1), результаты различаются для тепловых нейтронов. Будем разбираться.

# Спектр нейтронного потока в катушке тороидального поля, TRT, MCNP vs. OpenMC

## Спектр в проводнике КТП



## Спектр (кумулятивно) в проводнике КТП



# 2.

## R2S – rigorous 2-step

Расчет динамики трансмутаций и распада - большой  
объем данных и вычислений

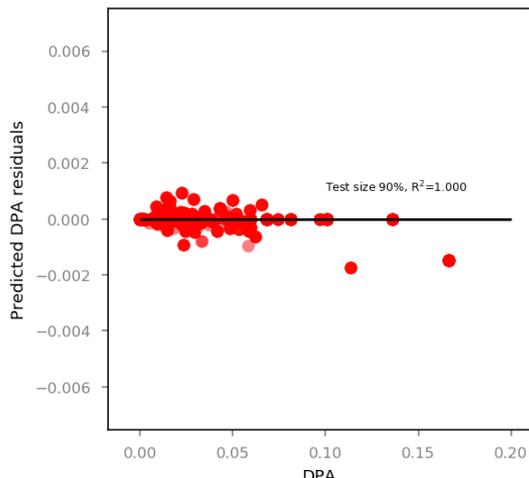
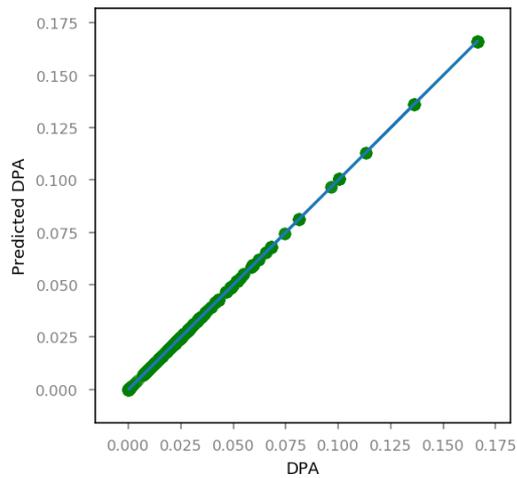
# Метод в лоб: “Full”

Просто прогнать FISPACT для каждого пересечения ячейка-воксел и собрать результаты. Применим только для тестирования (чтобы создать “референсные” результаты) и простых моделей. В реальных моделях потребовались бы миллионы прогонов FISPACT.

Замечание на полях:

Есть возможность уменьшить общий объем вычислений, применяя линейную регрессию. Ниже пример при тренировке модели 10% от всех результатов.

В “нормальных” условиях “похожие” нейтронные потоки должны давать “те же” результаты. Каждое слово “в кавычках” требует специального рассмотрения. Похоже на трюк, но годится для быстрых оценок.



## “Simple” метод

**Декомпозиция задачи по энергетическим группам** может уменьшить число прогонов FISPACT. При сборке результатов требуется аккуратно выполнить агрегирование по энергетическим группам – это не так просто.

$$\frac{dN_i}{dt} = \sum_{j \neq i} \lambda_{i,j} N_j - \sum_{j \neq i} \lambda_{j,i} N_i + \sum_k (\sum_{j \neq i} \sigma_{i,j}(E_k) N_j F(E_k) - \sum_{j \neq i} \sigma_{j,i}(E_k) N_i F(E_k))$$

$N_i(t)$  - число атомов нуклида  $i$  в момент  $t$ ,  $i, j \in 1 \dots M$  – номера нуклидов

$\lambda_{i \rightarrow j}$  - постоянная спонтанного перехода  $i \rightarrow j$ ,

$\sigma_{i \rightarrow j}$  - сечение индуцированного перехода  $i \rightarrow j$ ,

$F(E_k, t)$  – neutron flux vs. energy group distribution (function of time)

В нашем случае третий член уравнения доминирует над четвертым, поскольку **начальные нуклиды дают значительно больший вклад.**

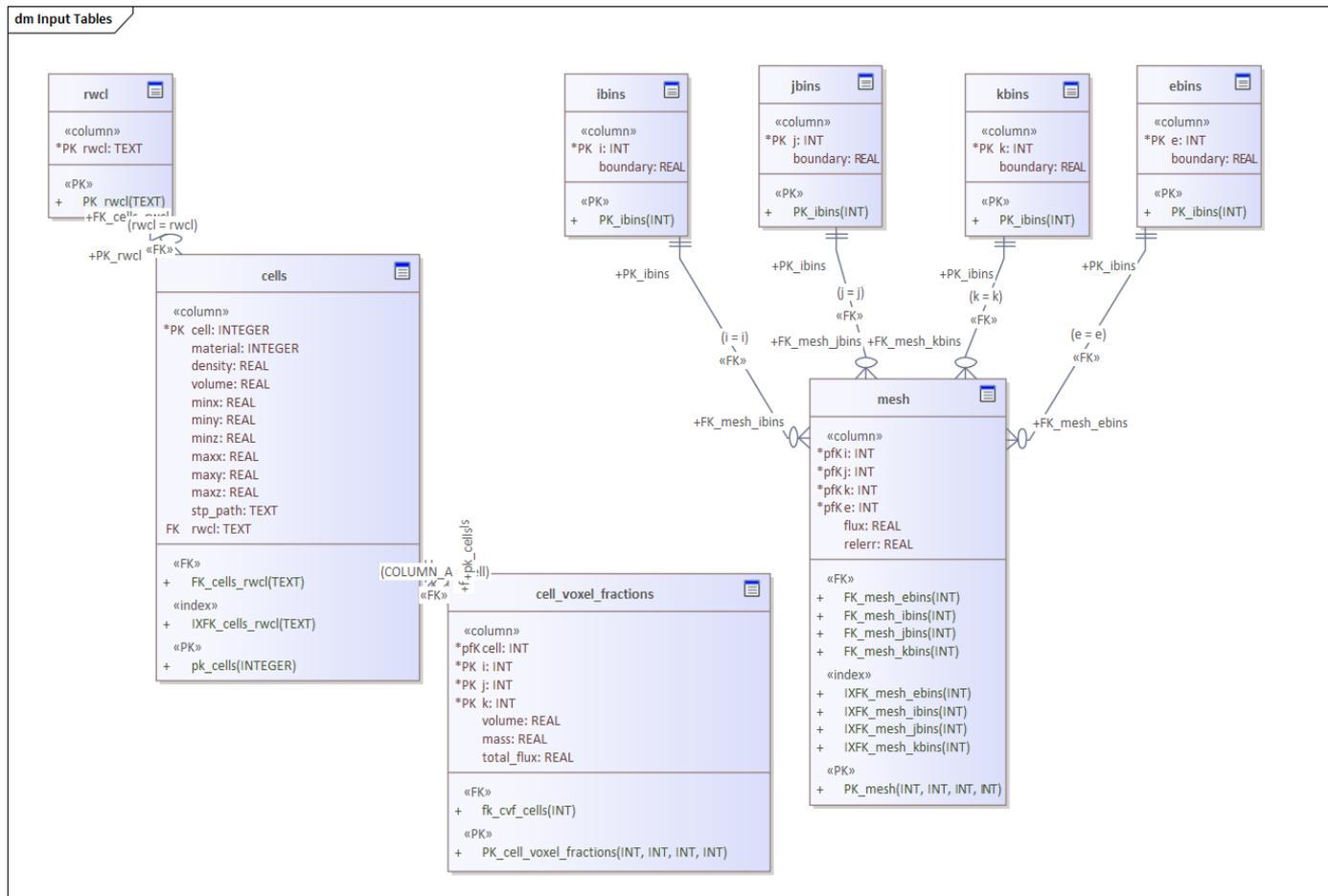
Мы **можем пренебречь** деталями взаимного влияния нейтронов разной энергии на скорость производства и выгорания нуклидов в индуцированных переходах. Тесты, и не только наши, показывают, что такой подход оправдан в «линейных» задачах.

# Схема БД (только входные данные, ~15% схемы)

Меш  
нейтронного  
потока и бины

Ячейки с мета-  
информацией:  
материал,  
плотность,  
путь в CAD,  
RWCL  
классификаци  
я...

Используем  
mapstp +  
SpaceClaim скрипт



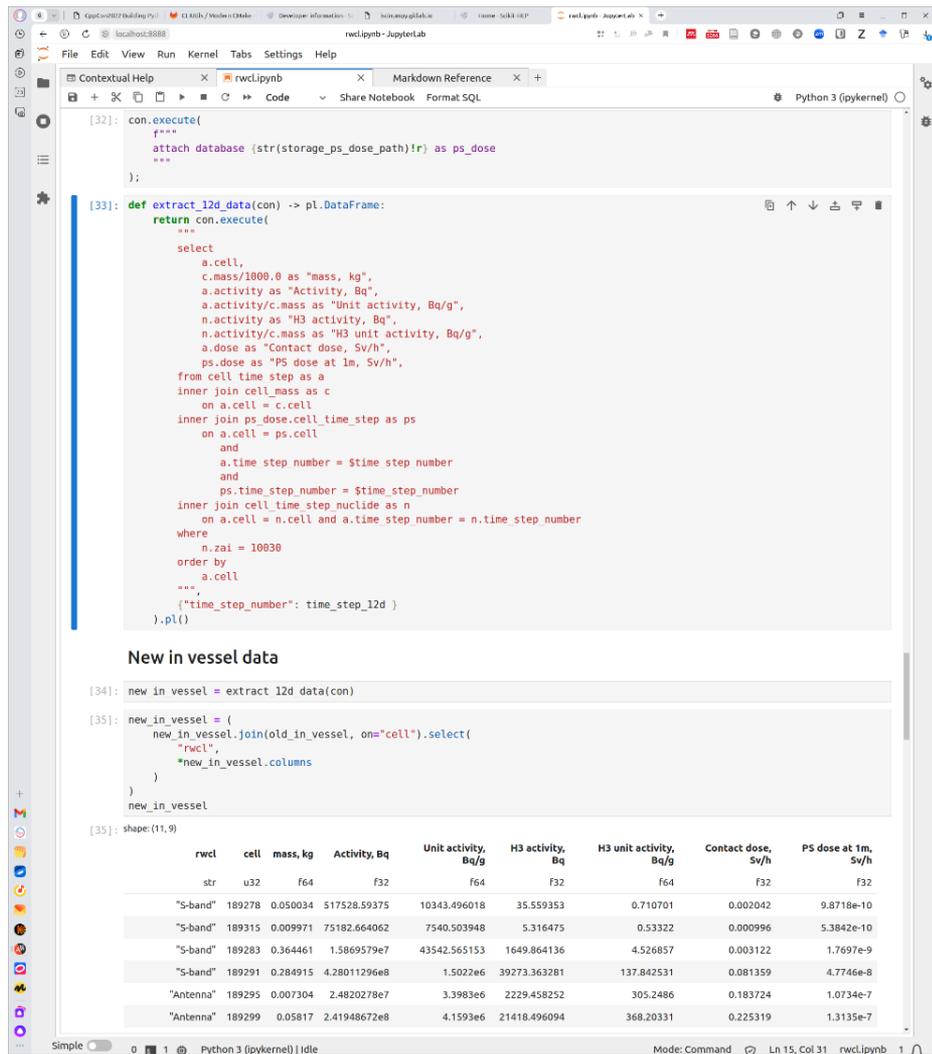
# Как работать с данными?

В нашем случае удобно представлять данные таблицами (графы и многомерные кубы также разворачиваются в таблицы).

SQL – наилучший инструмент для “произвольного” анализа данных в таблицах. Консолидация данных для RWCL\* требует минимального знания SQL (достаточно 1 дня практики).

Если, наоборот, требуется очень сложный анализ, то выбранная БД предоставляет множество мощных и удобных средств.

*\*) RWCL – Radioactive Waste Check List, таблица с характеристиками активации компонент*



The screenshot shows a JupyterLab notebook with two code cells. The first cell contains a SQL query to attach a database and execute a query. The second cell contains a Python function that returns a DataFrame with the results of the SQL query. Below the code, the output of the function is displayed as a table.

```
[32]: con.execute(
      """
      attach database (str(storage_ps_dose_path)!r) as ps_dose
      """
      );

[33]: def extract_12d_data(con) -> pl.DataFrame:
      """
      return con.execute(
      """
      select
      a.cell,
      c.mass/1000.0 as "mass, kg",
      a.activity as "Activity, Bq",
      a.activity/c.mass as "Unit activity, Bq/g",
      n.activity as "H3 activity, Bq",
      n.activity/c.mass as "H3 unit activity, Bq/g",
      a.dose as "Contact dose, Sv/h",
      ps.dose as "PS dose at 1m, Sv/h",
      from cell_time_step as a
      inner join cell_mass as c
      on a.cell = c.cell
      inner join ps_dose.cell_time_step as ps
      on a.cell = ps.cell
      and
      a.time_step_number = $time_step_number
      and
      ps.time_step_number = $time_step_number
      inner join cell_time_step_nucleide as n
      on a.cell = n.cell and a.time_step_number = n.time_step_number
      where
      n.zai = 10030
      order by
      a.cell
      """
      ("time_step_number": time_step_12d )
      ).pl()
```

**New in vessel data**

```
[34]: new_in_vessel = extract_12d_data(con)

[35]: new_in_vessel = (
      new_in_vessel.join(old_in_vessel, on="cell").select(
      "rwcl",
      *new_in_vessel.columns
      )
      )
      new_in_vessel

[35]: shape: (11, 9)
```

|           | rwcl   | cell     | mass, kg     | Activity, Bq | Unit activity, Bq/g | H3 activity, Bq | H3 unit activity, Bq/g | Contact dose, Sv/h | PS dose at 1m, Sv/h |
|-----------|--------|----------|--------------|--------------|---------------------|-----------------|------------------------|--------------------|---------------------|
|           | str    | u32      | f64          | f32          | f64                 | f32             | f64                    | f32                | f32                 |
| "S-band"  | 189278 | 0.050034 | 517528.59375 | 10343.496018 | 35.559353           | 0.710701        | 0.002042               | 9.8718e-10         |                     |
| "S-band"  | 189315 | 0.009971 | 75182.664062 | 7540.503948  | 5.316475            | 0.53322         | 0.000996               | 5.3842e-10         |                     |
| "S-band"  | 189283 | 0.364461 | 1.5869579e7  | 43542.565153 | 1649.864136         | 4.526857        | 0.003122               | 1.7697e-9          |                     |
| "S-band"  | 189291 | 0.284915 | 4.28011296e8 | 1.5022e6     | 39273.363281        | 137.842531      | 0.081359               | 4.7746e-8          |                     |
| "Antenna" | 189295 | 0.007304 | 2.4820278e7  | 3.3983e6     | 2229.458252         | 305.2486        | 0.183724               | 1.0734e-7          |                     |
| "Antenna" | 189299 | 0.05817  | 2.41948672e8 | 4.1593e6     | 21418.496094        | 368.20331       | 0.225319               | 1.3135e-7          |                     |

# Cell complexity for R2S

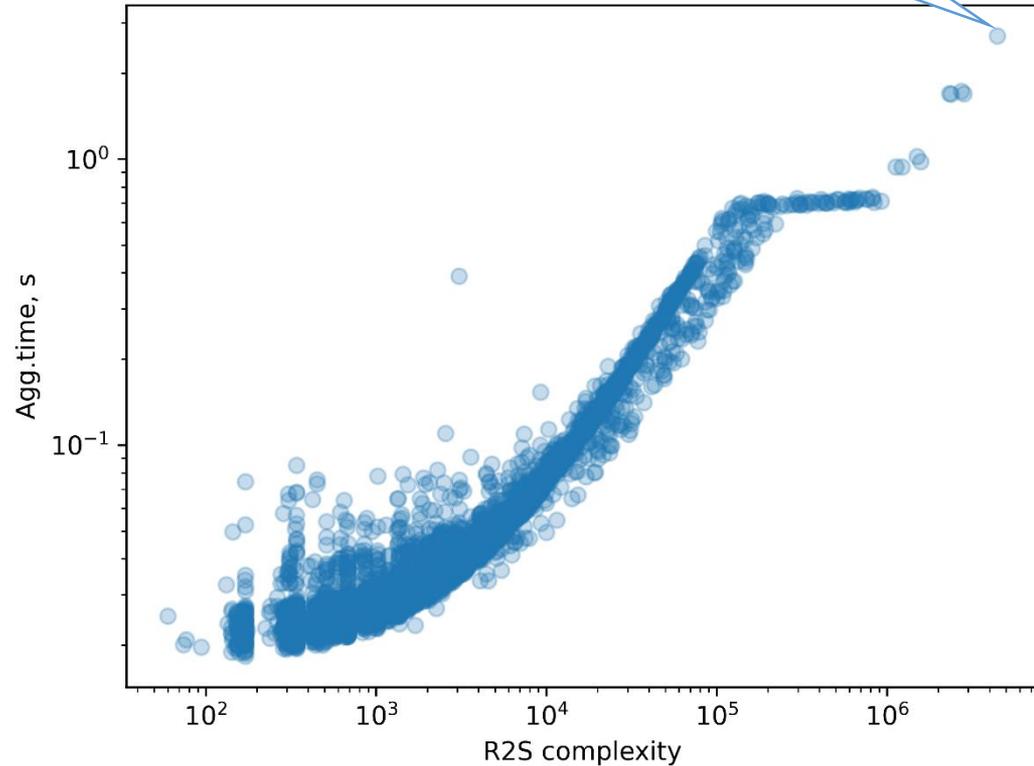
= *number of cell-voxel intersections per cell*

Some cells are large, not only computation their activation is a demanding task. The aggregation as well.

For instance, “air” cells in the Tokamak Complex model (presented here for demo purposes).

Fortunately, “air” and “water” cells are to be excluded from activation computations: SA2 irradiation scenario doesn’t make sense for these substances.

We usually extracted R2S model from the transport model. In recent scripts we produce both transport and R2S at the same time.



# Joins are expensive

1. For a simple model we can provide “perfect” aggregation: taking into account initial values, separate scaling by neutron fluxes in voxels and scaling by masses of intersections.
2. But real models are too demanding for every “join”, so, we have to allow some deviations from being “perfect”. If initial value is not zero, than the result is incorrect. However, this is fixable.
3. The application provides option to select the mode of processing “—large-volume”.

```

-- /mnt/amarano-2tb/dev/mcnp/npa-2023/npa-2023
r2s-rfda fetch --help
024-04-04 16:40:36 | INFO      | r2s-rfda, 1.4.1a0
sage: r2s-rfda fetch [OPTIONS]

Aggregate and fetch data produced by 'Run' to 'results' folder.

ptions:
--intermediate-dir TEXT      Directory to store intermediate parquet
                             files [default: (intermediate)]
-t, --threads INTEGER       the number of worker processes to use in
                             duckdb [default: (all cpus)]
--large-volume / --no-large-volume
                             Fetch large volume of data by portions
                             [default: (False)]
--help                       Show this message and exit.

```

**3.**

**What we have to do?**

# Deployment to cluster

1. We are going to apply Dask for distributed processing to get “large” application (when both disk and operational memory needs exceed a single workstation capabilities).
2. Other alternatives?

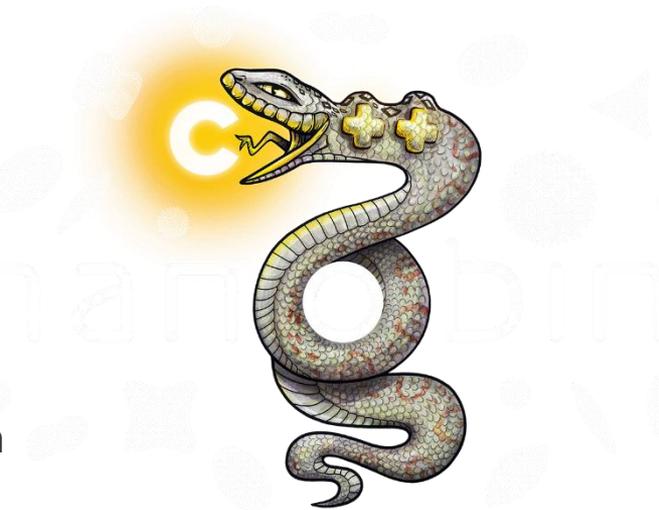


Dask distributed processing dashboard

<https://www.dask.org/>

# Own C++ FISPACT binding

1. FISPACT Python API package provided by NEA/OECD requires Python3.6
2. The recent versions of `numpy` require at least Python 3.9.
3. The `numpy` package is “must have” in Python based analytics projects, so, we need own FISPACT C++/Python binding package. We have to support at least Python 3.10.
4. We have to take into account ABI (binary compatibility) with the quite conservative toolkits available on clusters.
5. Probably, somebody already did this?



# Cell-voxel intersection blurring

1. Our cell x box intersection finding algorithm is trade off between the desired precision and performance.
2. If a cell “almost” exactly fills a given voxel edge, then due to rounding errors small ( $\sim 1e-4$  of total voxel volume) portions may appear in neighboring voxels. Currently we filter these cases on some level, but many of them still go to FISPACT runs, aggregation... The effect on the final aggregated result is negligible, so, the resources and time are just wasted.
3. We have R2S configuration options to filter out this artifacts, but it seems too rough approach. We could “de-blur” these portions returning them to dominating neighboring cell, before creating cases for FISPACT. I hope, a single SQL statement will do. But, this requires proper testing and, therefore, time.

# 3.

## Решения для средних объемов данных