UNIVERSITY OF CAMBRIDGE

Nuts and Bolts of Monte Carlo Neutron Transport:

Overview of algorithms and architecture of SCONE Monte Carlo code

Mikolaj Adam Kowalski

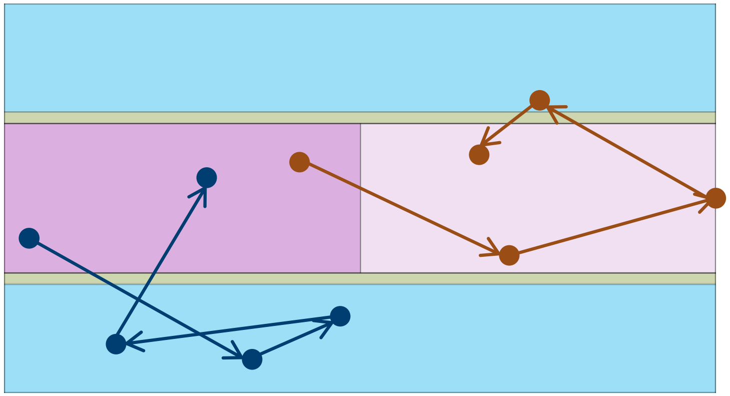
19.11.2019

Engineering - Energy, Fluid dynamics and Turbo-machinery

Monte Carlo Particle Transport

Usual student introduction:

- Model each particle from its "birth" until its "death"
- Simulate physical processes with random numbers
- Statistical estimates of interesting quantities can be obtained
- **Very accurate**
- **Very Computationally Expensive**



- Sometimes becomes replacement for reality!
- Ignores mathematical basis for Monte Carlo
- Tendency to treat codes as "black boxes"
- Nice interface of modern MC codes allows to perform calculations without taking all approximations into account

Goal: Provide quick "behind the curtain" overview of Monte Carlo in Reactor Physics
Undermine trust in MC results

Plan

PART I:
Overview of basic Monte Carlo Algorithms and Challenges

PART II:
Problem of Neutron Clustering

PART III:
SCONE Monte Carlo Code

Not In Equal Proportions

Useful Resources and References

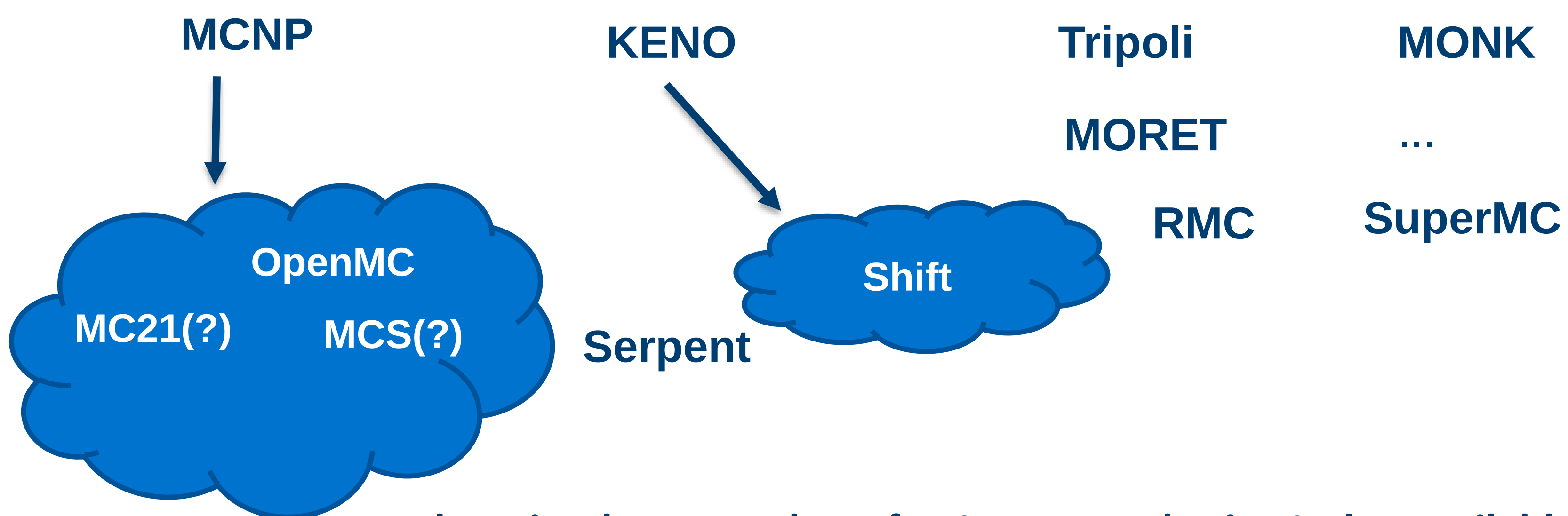
To write a Monte Carlo Code:

- MCNP4 Manual Chapter 2: Most information required to write a MC code
- MCNP4 Manual Appendix F: Specification of ACE format of Nuclear Data

To Learn More:

- Forest Brown, "Fundamentals of Monte Carlo Particle Transport", la-ur-05-4983
- I. Lux and L. Koblinger, "Monte Carlo Particle Transport Methods: Neutron and Photon Calculations"
- L. Devroye, "Non-Uniform Random Variate Generation"

Reactor Physics MC Code Landscape

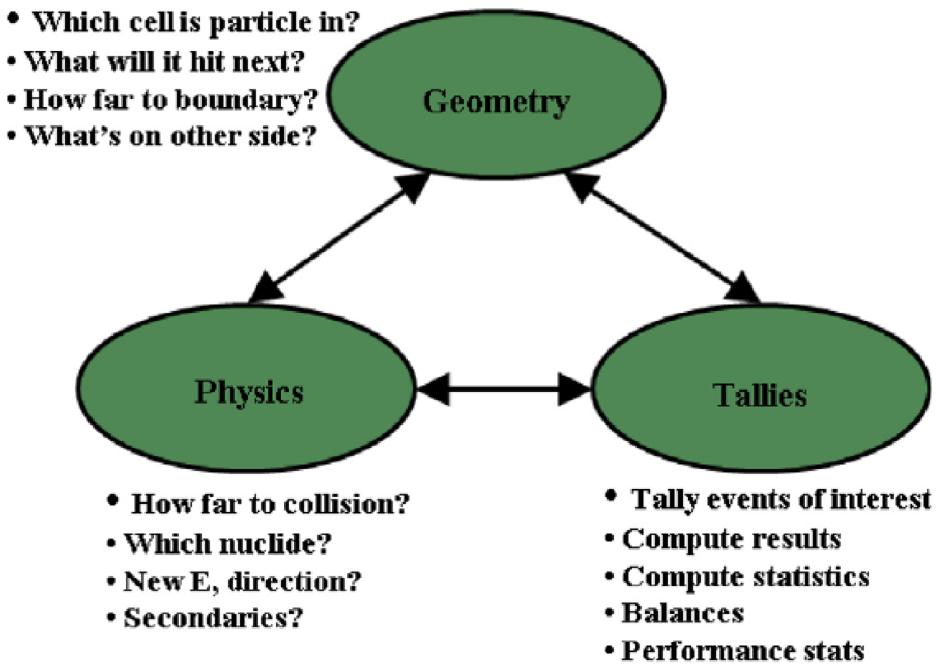


There is a huge number of MC Reactor Physics Codes Available

- Most of the codes are difficult to obtain
- Source Code is rarely available
- Little explanation of the methods in documentation
- Curiously, ALL Monte Carlo Codes are bound to an institution (LANL, Oak Ridge, VTT, CEA etc.)
- Not the case for accelerator physics: Geant4

Functional Decomposition of MC Code

Forrest Brown Functional Decomposition:



T. Adams *et al.*, 'Monte Carlo Application ToolKit (MCATK)', *Annals of Nuclear Energy*, vol. 82, pp. 41–47, Aug. 2015.

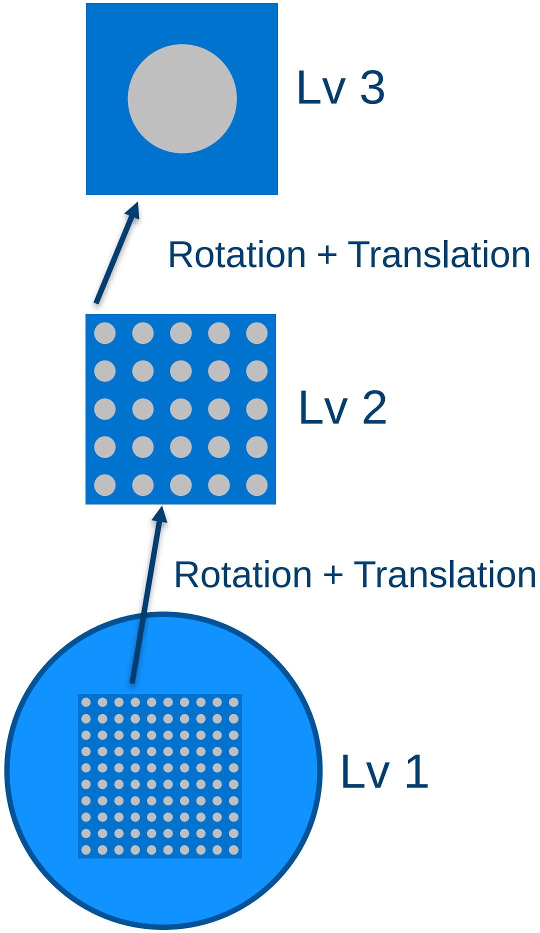
Geometry

- Often explanation is a bit confused
- Every MC code uses CSG, but in a slightly different way

Geometry Responsibilities:
- Find Material at coordinates
- Find distance to next cell
- Domain Boundary and BC

"Universe-Centric" model of geometry:
- Geometry is composed of number of levels
- Each level defined entire space
- Transition between levels are associated with transformations

In "Universe-Centric" view geometry is composed of two orthogonal problems:
- In-Universe representation of space
- Representation of universe-nesting structure



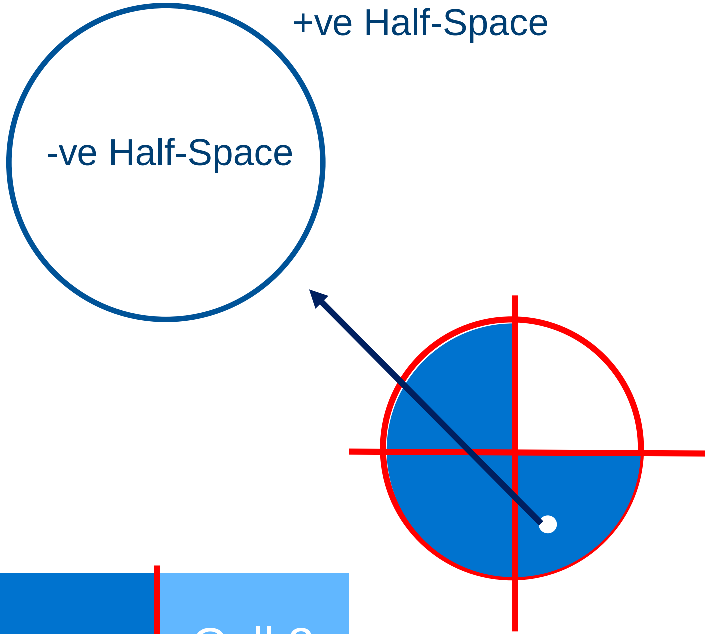
Constructive Solid Geometry

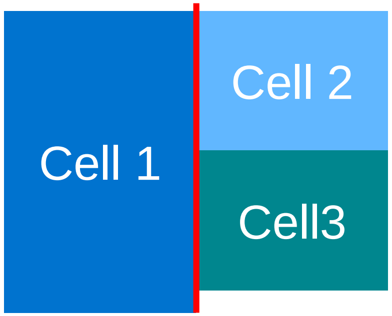
Constructive Solid Geometry:
- Divide space into cells
- Each cell contains:
    - Material
    - Transition to another level

Cell is defined by a Boolean expression of Half-Spaces:
- Intersections most efficient
- Union introduces ambiguity (non-convex, non-compact cells)

Cell-to-cell transitions are a problem
- Need to find cell neighbour
- Floating point arithmetic
- Ambiguity at the surface

Cell neighbour look-up
- Cell based
- Common surface based

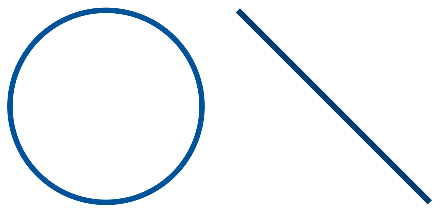




Constructive Solid Geometry – What is a surface

Simple surfaces
Described with a single equation

- Plane
- Cone
- Sphere
- Cylinder
- Paraboloid

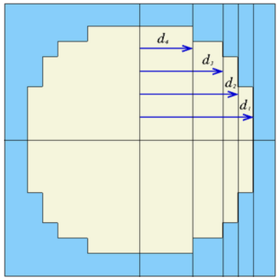


Only simple surfaces:

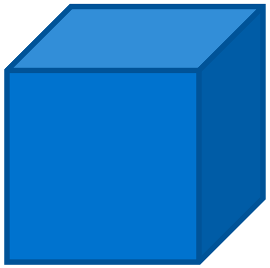
- MCNP
- OpenMC

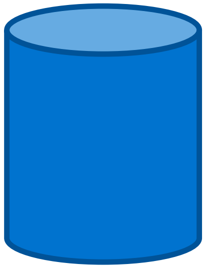
Codes that use body surfaces:

- Serpent
- KENO (?)
- MONK



Source:
Serpent Wiki





Body surfaces
Cannot be described with single equation

- Box
- Truncated Cylinder
- Cruciform Prism

Body surfaces make geometry definition easier

- Require more sophisticated algorithms (e.g. distance to axis aligned box)

Often Discussion of Geometry Ends Here

Constructive Solid Geometry

Often the arrangement of the universe is not random:

- Pin cell
- Fuel lattice

Special properties of the universe can be used to simplify:

- Cell search
- Distance search
- Neighbour look-up

Universes are <u>Polymorphic</u> (in O-O terminology)

Some codes ignore this and relay on optimisation of generic solutions:

- MCNP
- OpenMC

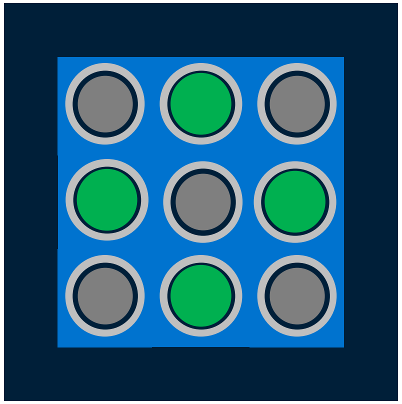
Some codes employ it:

- Serpent
- SCONE

Open question:
Is it worth paying the price of dynamic dispatch in exchange for simplified universe routines?

Vast amount of Problems can be composed of special-universes only

Graph Representation of Geometry



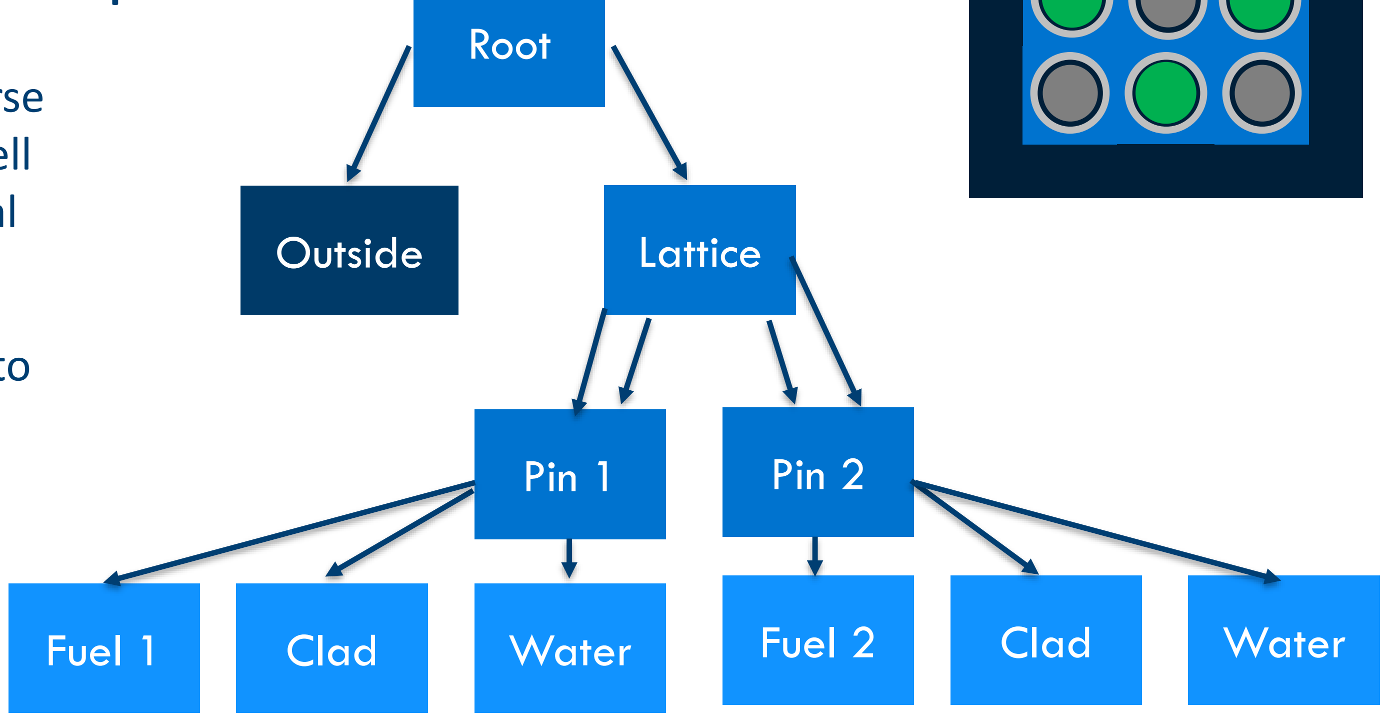
Problem:
How to assign individual ID to each instance of a cell

Geometry structure can be represented as <u>Directed Acyclic Graph</u>

- Each Node is a universe
- Each Transition is a cell
- Each Leaf is a material

Solve the problem:

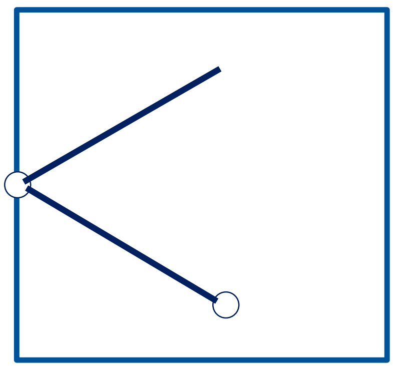
- Decompose Graph into <u>tree</u>
- Enumerate leaves
- Can use any existing Computer Science Solutions



In SCONE no material information is stored in cell definition
Material information is contained in separate structure with the DAG representation

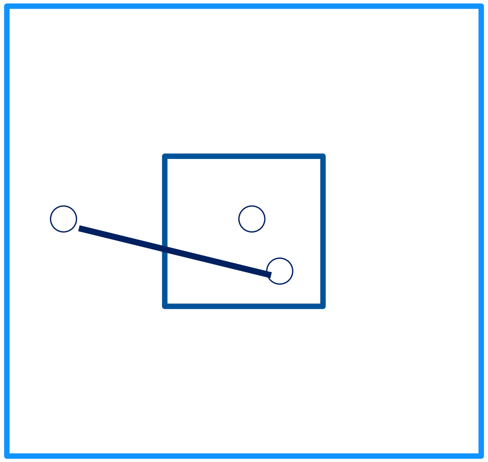
Boundary Conditions

Explicit Treatment



- Explicitly stop particle at the boundary
- Apply reflection or Translation

Co-ordinate Transformation

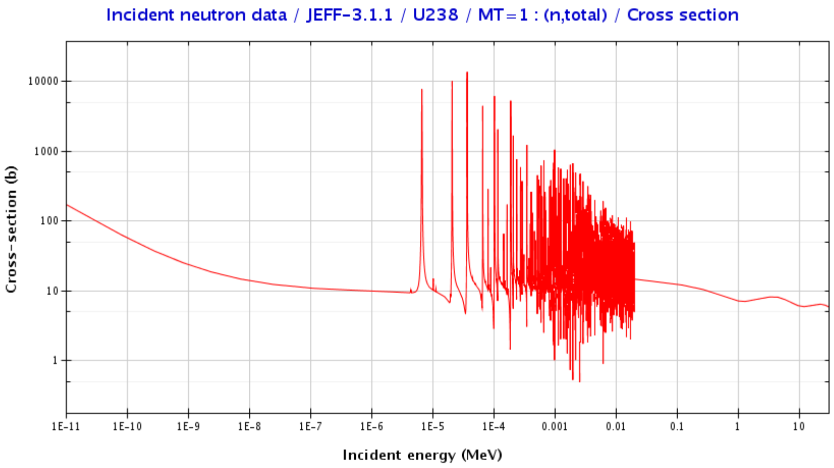


- Allow particle to leave the geometry
- "Fold" the space by applying reflections and transitions
- Efficient with Delta-Tracking

Nuclear Data

Two types of data are needed for transport

- Reaction Cross-Sections
- 2nd-ary Angle (μ) and Energy Distributions



Cross-Sections

- Stored in lin-lin interpolated table
- Each Nuclide has its own energy grid
- Lot of points needed to capture resonances

2nd-ary Distributions

- Many different formalisms →Source of branching
- Most data stored in a table format. Linear/Histogram interpolation
- Some are explicit distributions (e.g. Maxwellian Energy-Spectrum, Delta-probability)
- Some formalisms are rarely used (Watt-fission spectrum → replaced with table)
- Support for all the distributions and their variation is the main challenge in writing a physics section of a MC code

ACE Format

Header	Misc. Info: Mass, Temperature etc.
IZ-AW	Empty Legacy Entry
NXS	Int[16]: Info about data e.g. • Number of energy points • Number of precursor groups
JXS	Int[32]: Pointers to data on XSS e.g. • Location of XS data
XSS	Real[*]: Table with the data. Stored as real numbers.

Most codes use ACE formatted data

- OpenMC uses HDF5
- In near future might be replaced by GNDS
- Even if is replaced ACE-style pre-processing of data will be used (e.g. μ PDF as table not Legendre Moments)

ACE Format

Header

```
 26056.21c    55.454400  2.5852E-08   12/10/04
26-Fe-56 from FENDL-2.1(JEFF-3.0) NJOY 99.90 NDS/IAEA Nov2004             mat2631
```

Legacy

```
     0         0.       0         0.       0         0.       0         0.
     0         0.       0         0.       0         0.       0         0.
     0         0.       0         0.       0         0.       0         0.
     0         0.       0         0.       0         0.       0         0.
```

NXS

```
  824565    26056    53036       75       36      326        2        0
       0        0        0        0        0        0        0        0
```

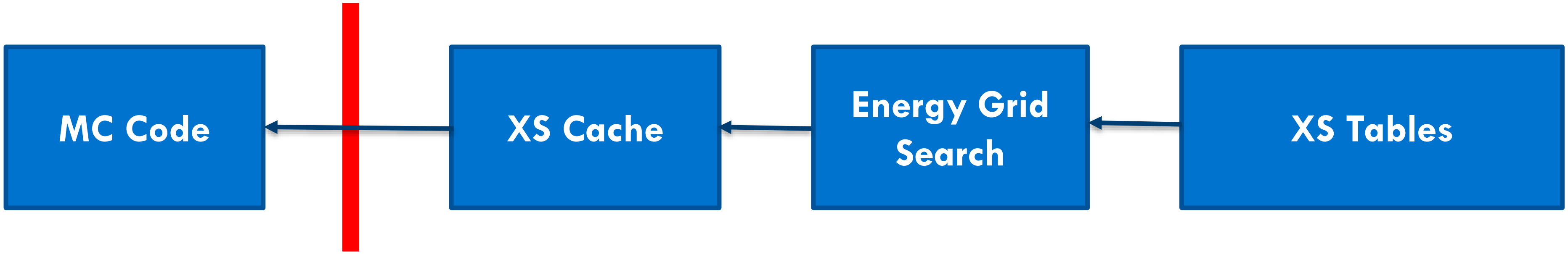
JXS

```
       1        0   265181   265256   265331   265406   265481   491758
  491795   652191   652227   692574   745610   745936   746262   749180
  749506   749506   749832   784730        0   784792        0        0
       0        0        0        0        0   784793   784795   784797
```

XSS

```
  1.00000000000E-11  1.03125000000E-11  1.05859400000E-11  1.08593800000E-11
  1.11328200000E-11  1.14062500000E-11  1.16796900000E-11  1.19531300000E-11
  1.25000000000E-11  1.28125000000E-11  1.31250000000E-11  1.34375000000E-11
  1.37500000000E-11  1.43750000000E-11  1.50000000000E-11  1.56250000000E-11
  1.62500000000E-11  1.68750000000E-11  1.75000000000E-11  1.81250000000E-11
  1.87500000000E-11  1.93750000000E-11  2.00000000000E-11  2.09375000000E-11
  2.18750000000E-11  2.28125000000E-11  2.37500000000E-11  2.46875000000E-11
  2.56250000000E-11  2.65625000000E-11  2.75000000000E-11  2.84375000000E-11
  2.93750000000E-11  3.03125000000E-11  3.12500000000E-11  3.21875000000E-11
  3.31250000000E-11  3.40625000000E-11  3.50000000000E-11  3.59375000000E-11
```

Accessing Cross-Sections



Cross-Section Access is the main bottleneck in MC Calculations

- Can take as much as <u>85% of runtime</u> (based on OpenMC)!
- Monte Carlo Code is effectively a glorified table interpolator!

Evaluating Macroscopic XSS requires loop over many nuclides

- Interpolated Macroscopic XSS need to be cached
- In burned core can have 100s of nuclides
- Energy grid search acceleration is necessary to avoid 100s of binary searches

Unified Energy Grid and Hashing

Unified Energy Grid (Serpent)

1. Create Union of energy grids of all nuclides
2. Interpolate all XSs on the new grid
3. Only single energy grid search is necessary
4. Allows to pre-calculate macroscopic XSS

Energy Grid Hasing (MCNP6, OpenMC)

1. Divide energy grid into uniform bins (in log of energy)
2. For each nuclide find index of energy location just before a bin
3. To access XS find a bin with arithmetic operation
4. Perform short linear search from the stored index for a nuclide

- Unified energy grid consumes a lot of memory (1.6 GB for depleted core problem)
- Offers significant acceleration (~53% for depleted core problem)
- Hashing gives ~24% speed-up for 6.9 MB
- Trade-off between performance and memory usage
- Most Monte Carlo Problems are limited by memory

J. A. Walsh, P. K. Romano, B. Forget, and K. S. Smith, 'Optimizations of the energy grid search algorithm in continuous-energy Monte Carlo particle transport codes', *Computer Physics Communications*, vol. 196, pp. 134–142, Nov. 2015.

Neutron Clustering

Eigenvalue Problems

There are no critical problems!

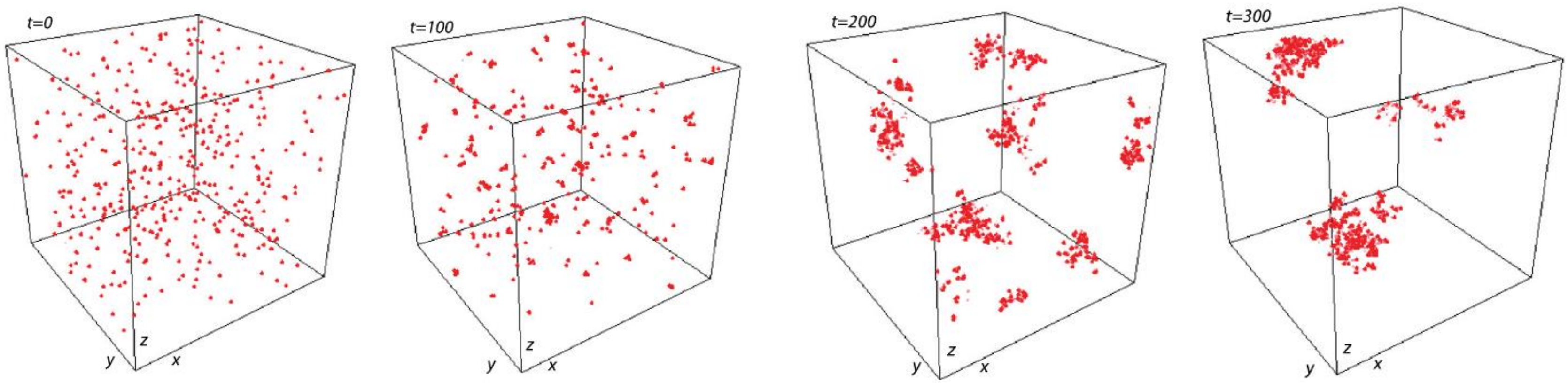
- Every problem requires renormalisation of population
- On average each fission particle causes 1 fission for next generation
- "Gambler Ruin" phenomena → Expectancy = 1, but variance increases with generations
- Eventually neutron families die-out

Clustering is a Physical Phenomena
Attempts are made to detect it in ZPR

For most problems:

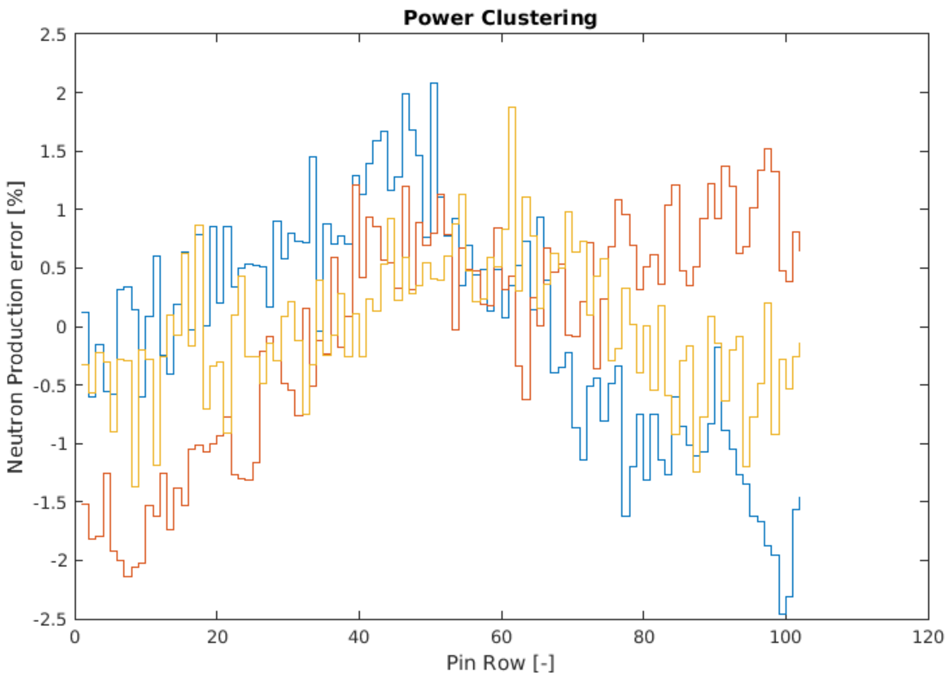
Family extinction rate > Diffusion through the domain

Results in particle clustering & Failure of Source Convergence



E. Dumonteil *et al.*, ‘Particle clustering in Monte Carlo criticality simulations’, *Annals of Nuclear Energy*, vol. 63, pp. 612–618, Jan. 2014.

Particle Clustering



- Clustering results in unreliable statistical error estimates
- Necessary to ensure that cycle population is large enough
- Every problem will eventually experience clustering

Neutron Production in lattice of 102 PWR pins
3 Independent MC Runs

No reliable metrics to detect clustering were created
Can look at the variance in Centre-Of-Mass of population as some indication

What is SCONE

Stochastic
Calculator
Of
Neutron Transport
Equation

- Particle Transport Monte Carlo Code for Nuclear Engineering Applications
- Academic Focus: Target audience →Master's and PhD Students
- Designed for modification: Object-Orientation, well-defined abstractions
- Use: Teaching, New Algorithms Prototyping
- **Prioritise modifiability over performance**

Language: Fortran 2008

- Easy to learn; Informative Compiler Errors; Easy to read standard
- Offers good performance
- Well-established (generally supported, OpenMP, OpenACC etc.)

Original Motivation: Variable Fidelity Calculations

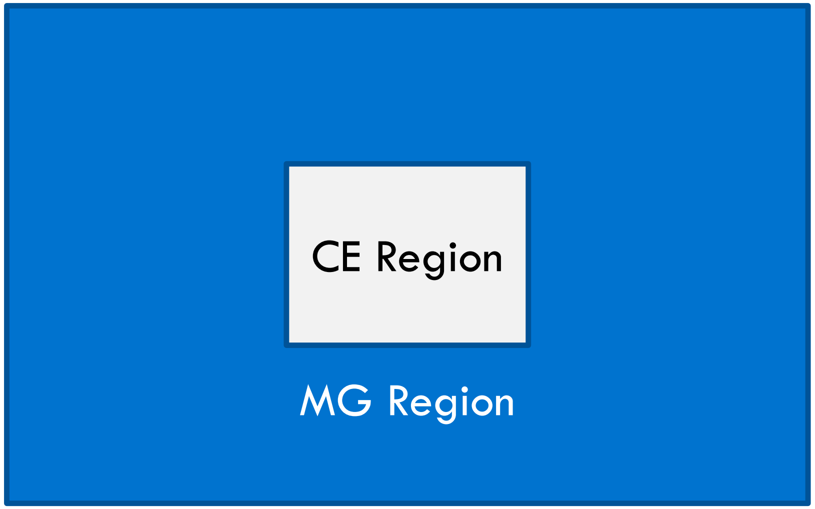
Motivation:

- Computational resources are a limited resource
- Accurate results are not required everywhere in a problem
    - Temperature margin evaluation → accuracy matters for few pins close to the limit
- In general, reduction of fidelity accelerates calculation

Multi-Group Monte Carlo:

- ~4x faster execution speed then CE
- Consumes less memory
- Easier to vectorise (thus optimise for GPUs and Intel AVX etc.)

Question:
Is it possible to accelerate MC by using lower fidelity in regions where accuracy is less important?



Is there a need for SCONE?

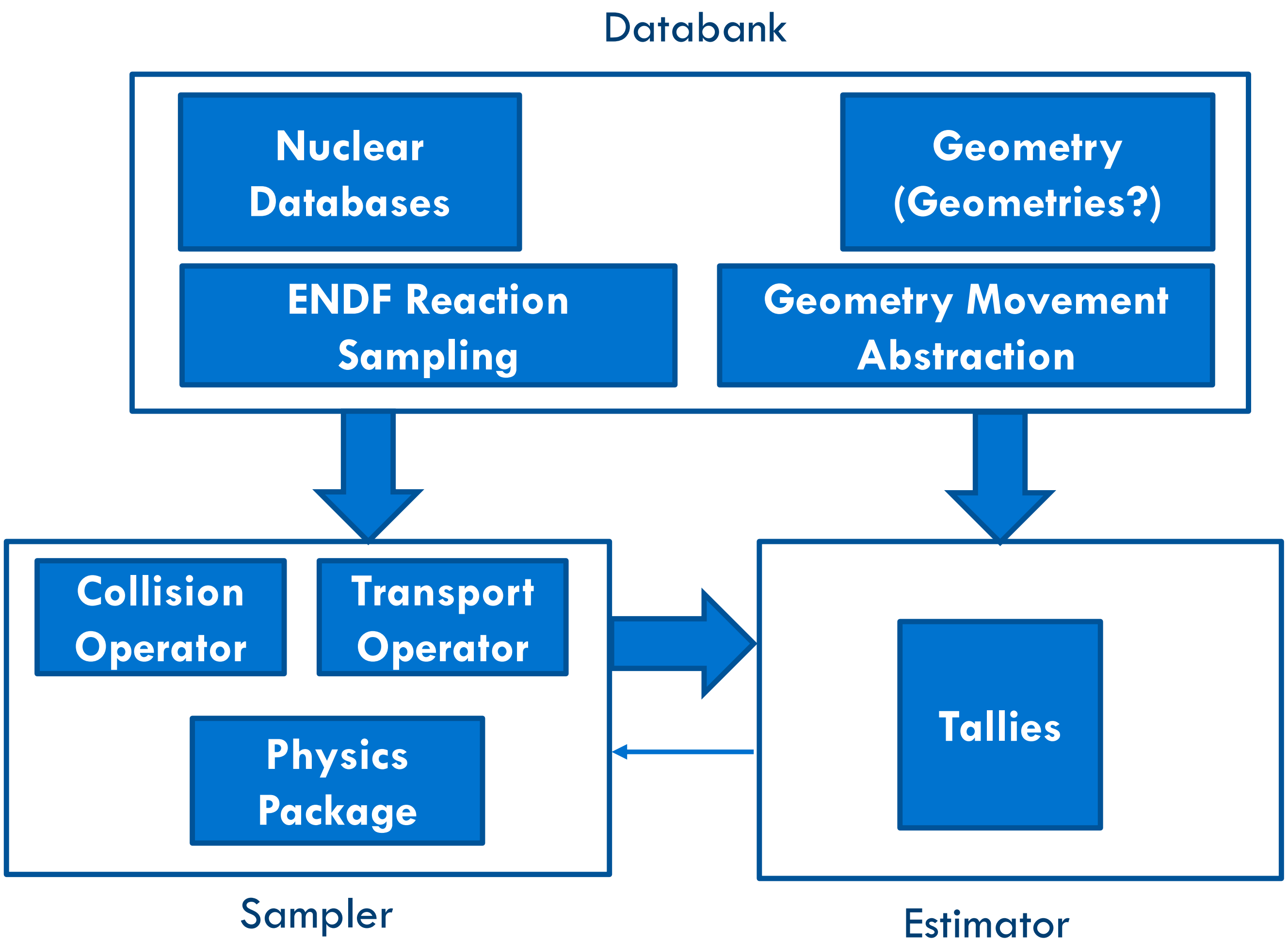
Why don't just use OpenMC?

- Transport() function is not virtual → There is ONE way to do the calculation
- From architecture (it seems):
  - Priority of OpenMC: Fast & Scalable Calculations of Reactor Problems
  - Not a priority of OpenMC: Supporting implementation of "wacky" (often not very useful in practice) ideas
  - E.G. Does not support delta-tracking in its current implementation
- NOTE: Not a criticism of OpenMC, but an observation that its priorities seems to be much different from SCONE's

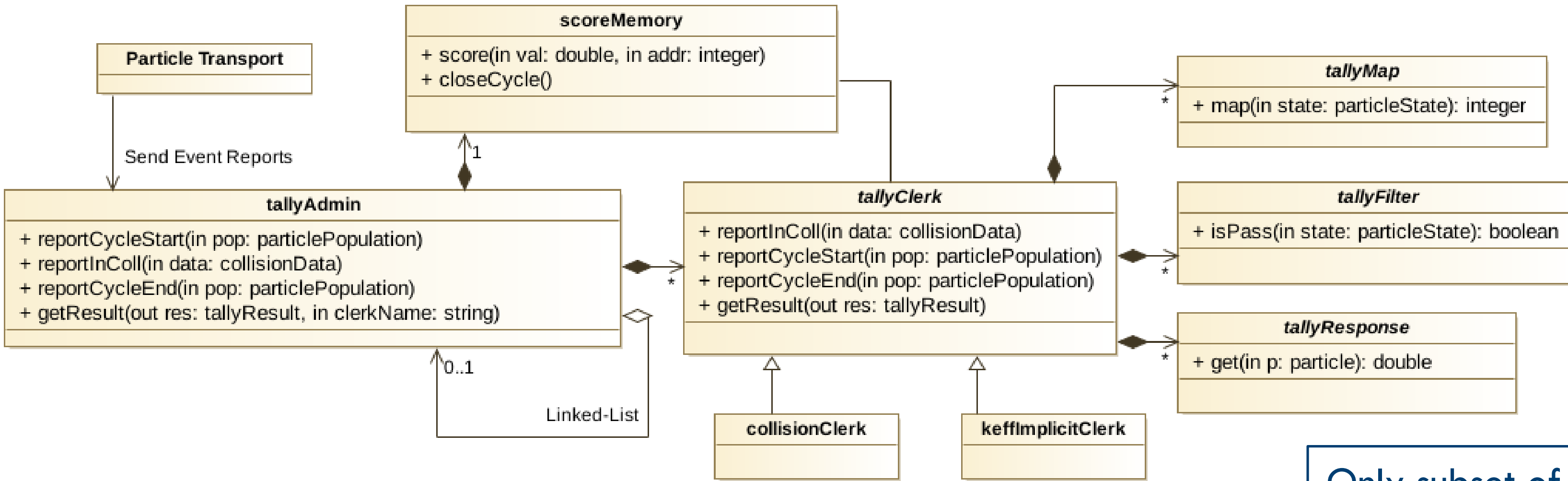
How does SCONE fit?:

- Goal: Challenging to use, <u>Easy to modify,</u> Somewhat slow to execute
- Expose the user to some gritty details of MC methods in input files (similarly to OpenFOAM)
- Allow maximum flexibility in defining calculation sequences
- Define clear abstraction for interaction with key components (Nuclear Data, Geometry, Tallies).
- Try to optimise for speed of : *Idea → Prototype Implementation;* **not** *Input → Result*

Alternative High Level Decomposition



Design of Tallies



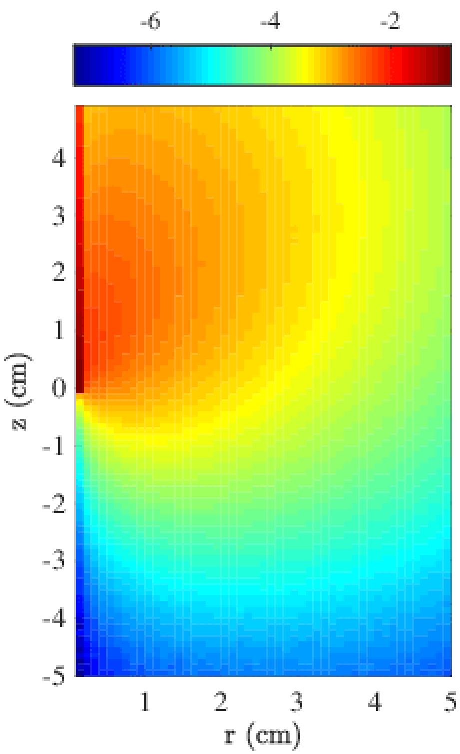
Only subset of the methods is shown

Requirements:

- Accept events reports
- Return some of the results
- Concentrate relevant code into single class

In order to create new result estimator it is sufficient to write just a tallyClerk
No modification elsewhere in the code is required
Little knowledge about the inner workings of other sections is needed

Student Involvement



Log$_{10}$(Photon Flux)

1 MeV Beam

Iron Cylinder

MEng Project to implement Photoatomic Photon Transport

- Very successful in short time (6 months)
- Showed that it is possible for Master's students to contribute to the development
- Positive feedback from the student on SCONE

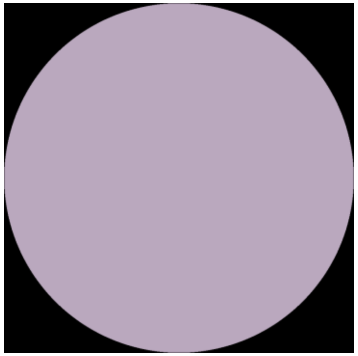
Lessons learned:

- Need for Code Reviews from the start
  - Students tend to stay quiet
  - Can spend a lot of time struggling with problems easy to correct if they ask for help
  - Necessary to enforce good style
- Need for detailed step-by-step tutorials to the SCONE
  - Save time for the supervisors

More Projects Offered to Students

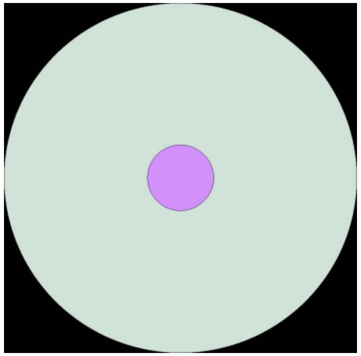
Comparison against Serpent

Benchmarks



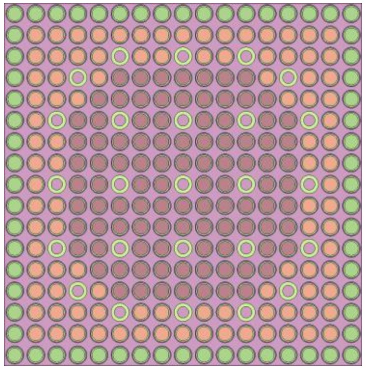
Jezabel

Metallic Pu Sphere



Popsy

Metalic Pu Sphere with natural U reflector



C5G7 based PWR MOX assembly

Heterogeneous arrangement of:

- 4.3wt%
- 7.0wt%
- 8.3wt%

MOX pins with guide tubes

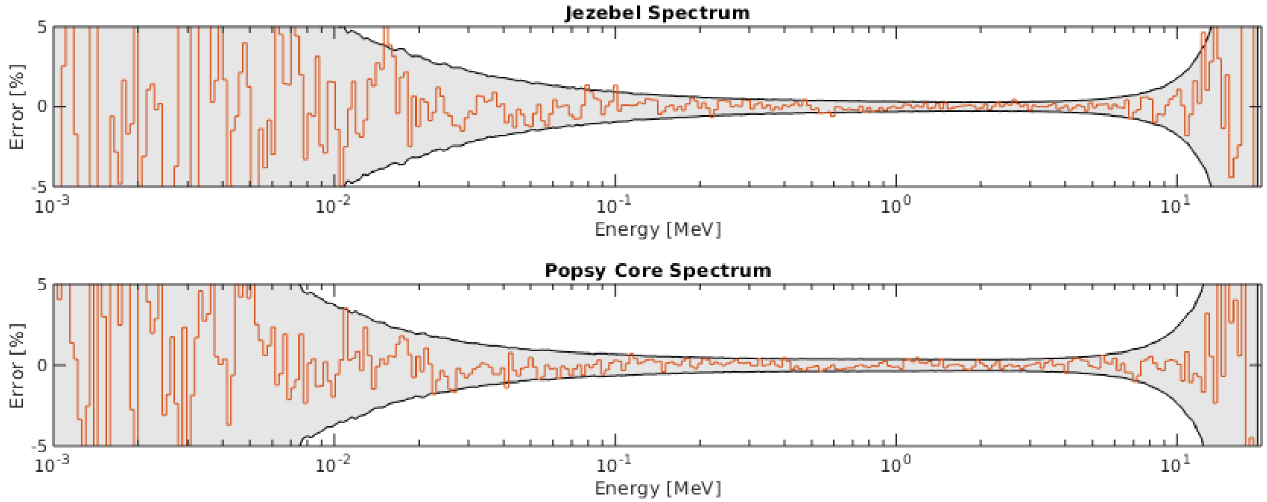
Without:

- Unresolved resonances tables
- S(α,β) tables

Serpent opt 4; set gcu -1

Comparison against Serpent

Comparison of runtime & criticality between SCONE and Serpent				
Case	SCONE k-eff	SCONE Runtime [s]	Serpent k-eff	Serpent Runtime [s]
Jezebel	1.0015±9pcm	257	0.99999± 10pcm	765
Popsy	1.00307±11pcm	3250	1.00376± 11pcm	5036
MOX	1.19871±6pcm	9367	1.19872± 6pcm	7091



Error in flux spectrum in fast benchmark cases. Grey region represents extent of 2σ

Good agreement in terms of speed & accuracy

Future of SCONE

Preparing for open source release:

- In near future (planned February 2020)
- Need to clean-up repository
- Need to choose a licence (MIT? GPL3?)

Further Work:

- Parallelisation
- Missing Physics (S(α,β), Unresolved Resonances)
- Investigate advanced calculations
- Continuously improve the design

Problems:

- Lack of menpower (It's just me!)
- Patchy general documentation
- Lack of funding for future development

Some interest in collaboration:

- University of Bath Mathematic Dept.

Happy to share access to the repositry. Send me an e-mail: *mak60@cam.ac.uk*

><((((*> ><((((*> ><((((*> ><((((*> ><((((*> ><((((*>

Thank you for your attention!

<*))))>< <*))))>< <*))))>< <*))))>< <*))))>< <*))))><