Investigations into **Alternative Radiation Transport Codes** for **ITER Neutronics Analysis**

Andrew Turner

Technology Department, UK Atomic Energy Authority (CCFE)

This work was funded by the RCUK Energy Programme under grant number EP/I501045.

ITER Neutronics Analysis

C-Model

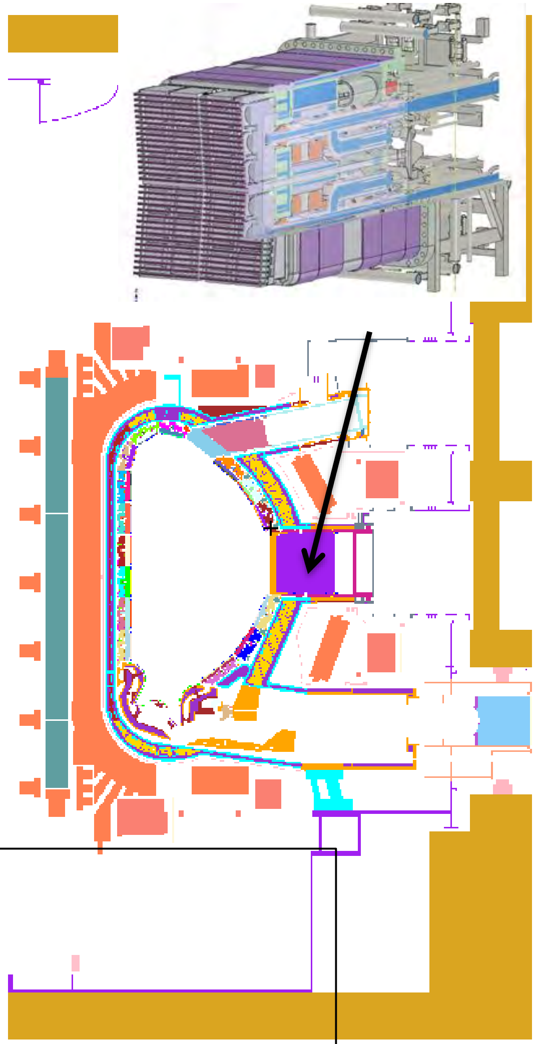
MCNP ITER reference model for nuclear analysis. Based on 'CSG' geometry. Large, complex.

- **100+ universes**
- **80,000 cells**
- **120,000 surfaces**

Nuclear analysis to support ITER design/build

Receipt of new system designs → **Creation of updated system models in MCNP** → **Integration into C-Model** →

Time-critical assessment of:
- Shielding performance
- Nuclear heating
- Activation, shutdown dose rate, etc.

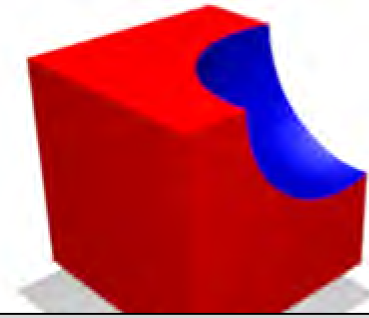


Conventional approach

MCNP

CSG

Constructive Solid Geometry



- Models require **simplification** → spheres, cylinders, cones, tori, planes…no splines.

Typical toolchain:

Detailed CAD → SPACECLAIM CORPORATION → **Simplified CAD** → SuperMC → MCNP6

Many iterations

- CAD → MCNP tools like SuperMC (FDS team, INEST, China) and MCCAD (KIT) very effective.
- **Why would we want anything different?**

Why might we need an alternative?

Time to obtain simplified CAD model that converts to CSG can be <u>months</u>.
Lengthy and uncertain task schedule.

Lost particles due to overlaps/gaps.
Manual adjustments to MCNP surfaces.

Typical ITER neutronics analysis project: 4-6 months, distributed as:

Detailed CAD → 50% → **Simplified CAD** → <5% → **Simplified MCNP** → 10% → **Clean simplified MCNP** → 5% → **C-Model with system model and materials** → 15% → **Clean C-Model ready for analysis** → 10% → **Analysis complete** → 10% → **Reporting**

Lost particles due to surfaces coincident with universe boundary.
Manual adjustments to 100s of MCNP surfaces. Non-trivial.

Hand modifications individually not significant but have to be repeated for <u>every</u> conversion from CAD.
Design optimisation/parameter studies in CAD become difficult.

How can we achieve this?

Reduce this time using mesh-based geometry.

No requirement for manual modifications: RT code overlap/gap geometry tolerance

Detailed CAD → 50% → halved? → **Simplified CAD** → <5% → **Simplified MCNP** → 10% → **Clean simplified MCNP** → 5% → **C-Model with system model and materials** → 15% → **Clean C-Model ready for analysis** → 10% → **Analysis complete** → 10% → **Reporting**

No requirement for manual modifications: RT code supporting coincident surfaces and clipping of geometry.

Enables:

Fast-response analysis capability – becomes more important as ITER is being built.

Rapidly modify designs and re-analyse: optimisation, parameter studies, closer support to design teams.

Potential solutions investigated

Three RT codes were compared as part of this study

MCNP6v1.0	DAG-MCNP5v1.6	Serpent-2.1.27

Geometry supported:

MCNP6v1.0	DAG-MCNP5v1.6	Serpent-2.1.27
Unstructured volume mesh (UM).	Unstructured surface mesh (US).	Unstructured surface mesh (US).
CSG		~~Unstructured volume mesh (UM).~~
		CSG

Comparison study

Investigate:

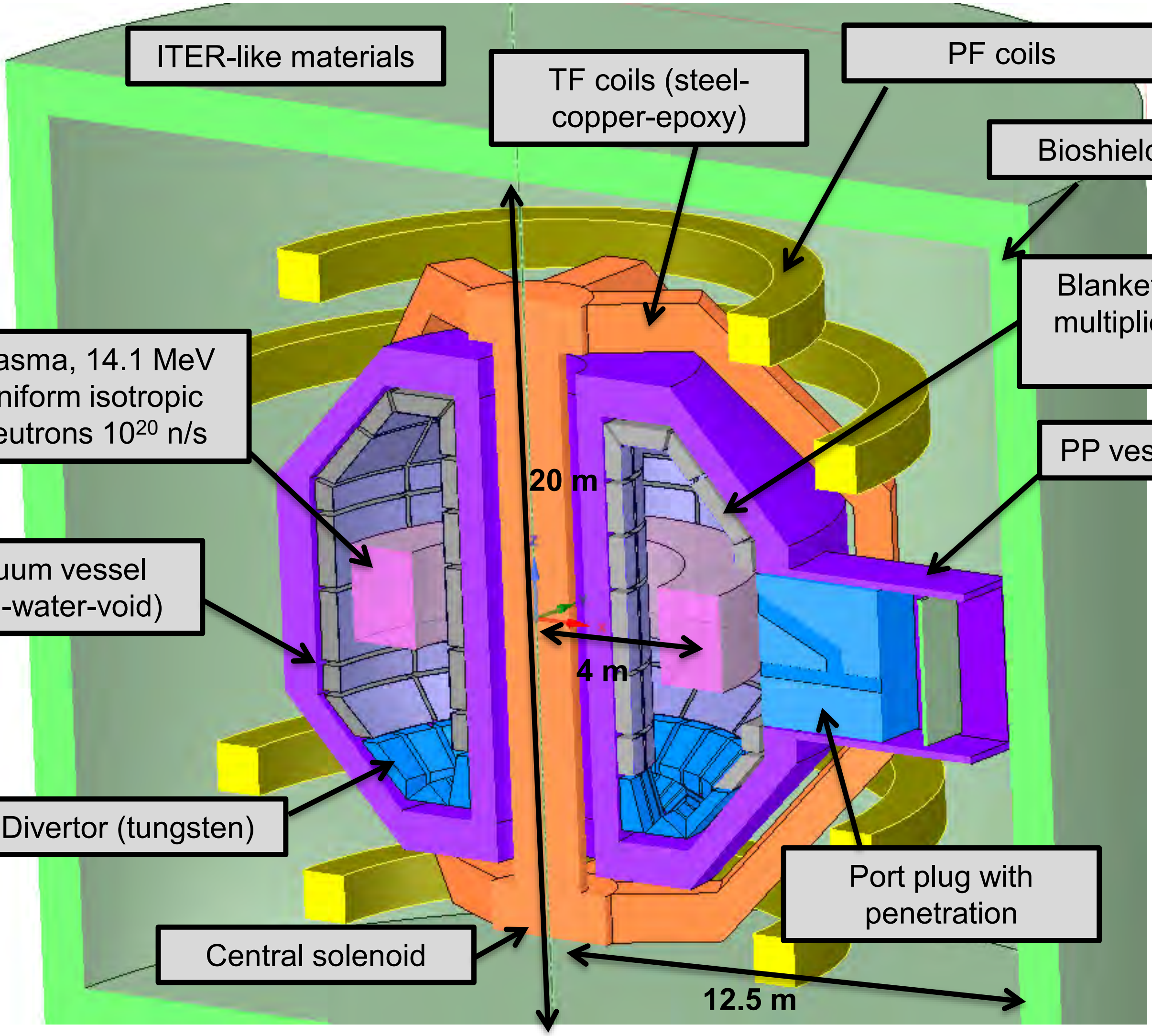
Computational practicality: speed, memory.	**'Correct' results** (i.e. close to MCNP-CSG)	**User practicality:** ease of use, missing features etc.

Comparison model :

- Able to convert to CSG and mesh model variants.
- Simple and quick to run.
- ITER-relevant size and features

→ Developed the 'Octamak' model. (octagonal tokamak…).

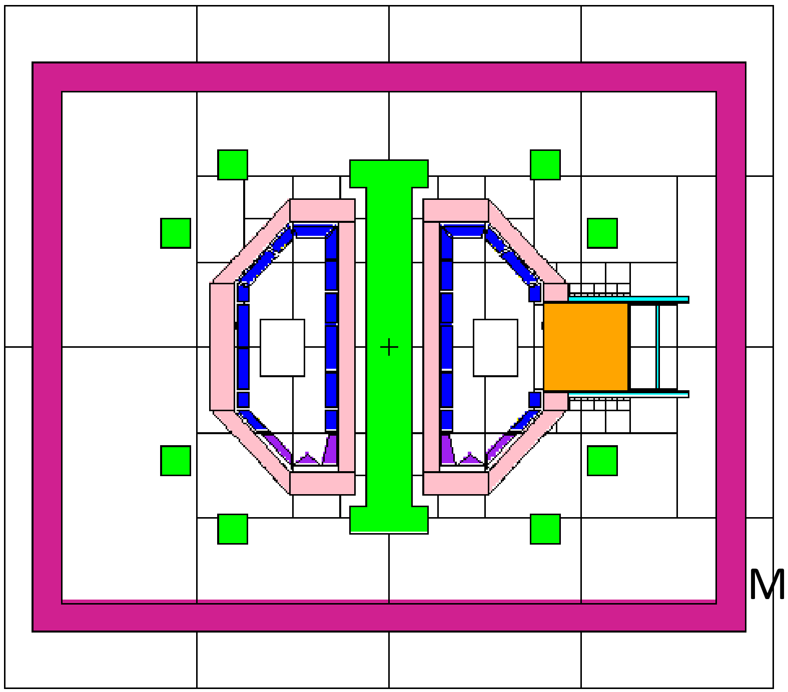
Octamak model



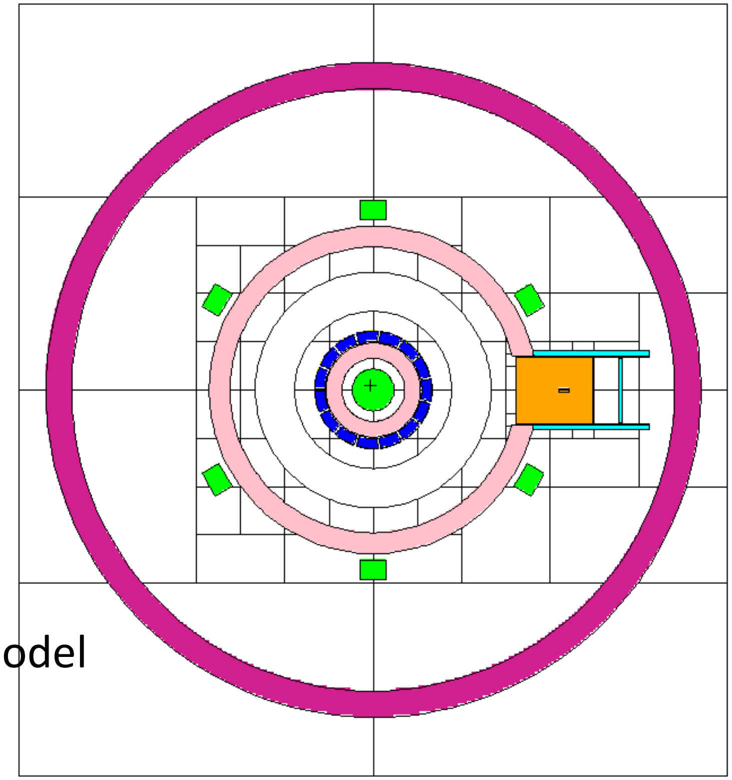
Octamak-CSG

Utilised SUPERMC/MCAM[1,2] (FDS team, INEST, China)

Approximately ~1000 cells/surfaces.



MCNP model



Identical Serpent-2 CSG model created by simple find-replace operations (Serpent supports near-identical set of CSG).

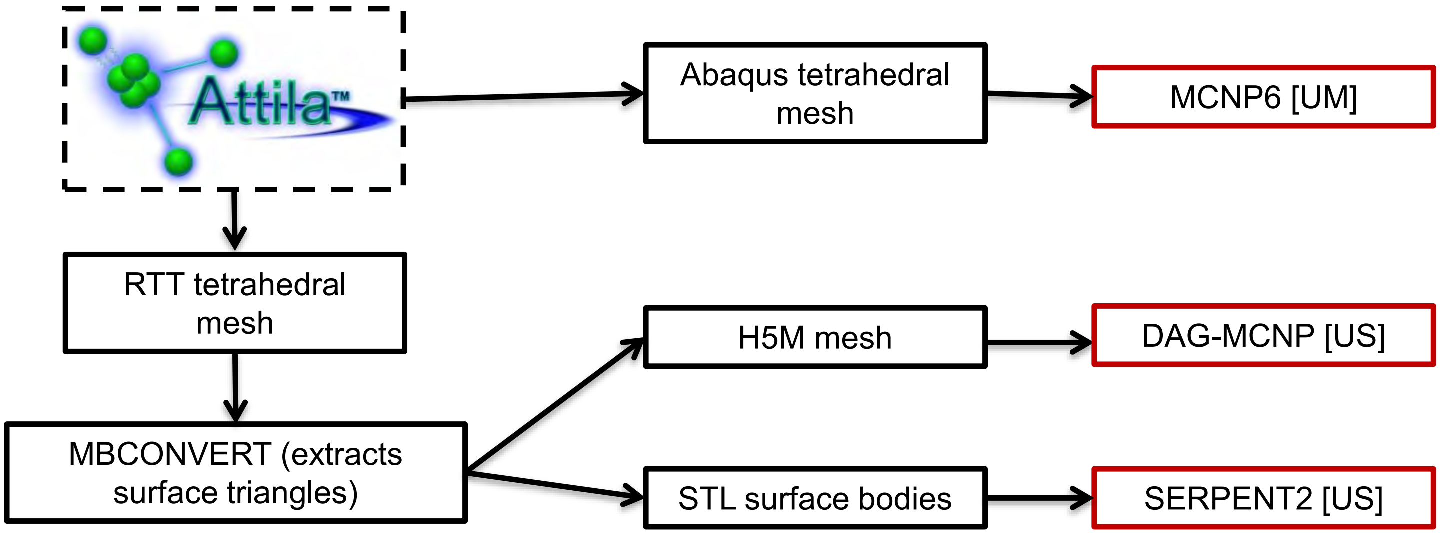
[1] Y. Wu, FDS Team. CAD-based interface programs for fusion neutron transport simulation, Fusion Eng. Des. 84 (2009), 1987-1992
[2] Y. Wu, J. Song, H. Zheng, et al. CAD-Based Monte Carlo Program for Integrated Simulation of Nuclear System SuperMC, Ann. Nucl. 82(2015) 161-168.

Meshing workflow

Attila was selected as the 'hub' mesher

→ To create identical mesh geometry (same surface facets) for all 3 RT codes.

(in practice, DAG-MC geometry would be created in CUBIT/Trelis, and Serpent-2 STL files from CAD software, this approach is just to ensure a fairer comparison).



Increasing detail ↓

	Surfaces/facets	Cells/elements
CSG	541	1,795
Mesh 1	31,420	43,763
Mesh 2	182,011	606,857
Mesh 3	591,364	3,001,000
Mesh 4	1,266,494	11,162,939

Various mesh resolution settings used, to test the effect of mesh count on performance

Tally comparisons for mesh 1 provided here…no significant change in results with resolution.

Tallies, settings

FENDL-3.0 materials used in all RT codes

Neutron transport only

- At the time of this study, no neutron-photon transport in Serpent-2.
- Under active development.

Analogue transport (no VR)

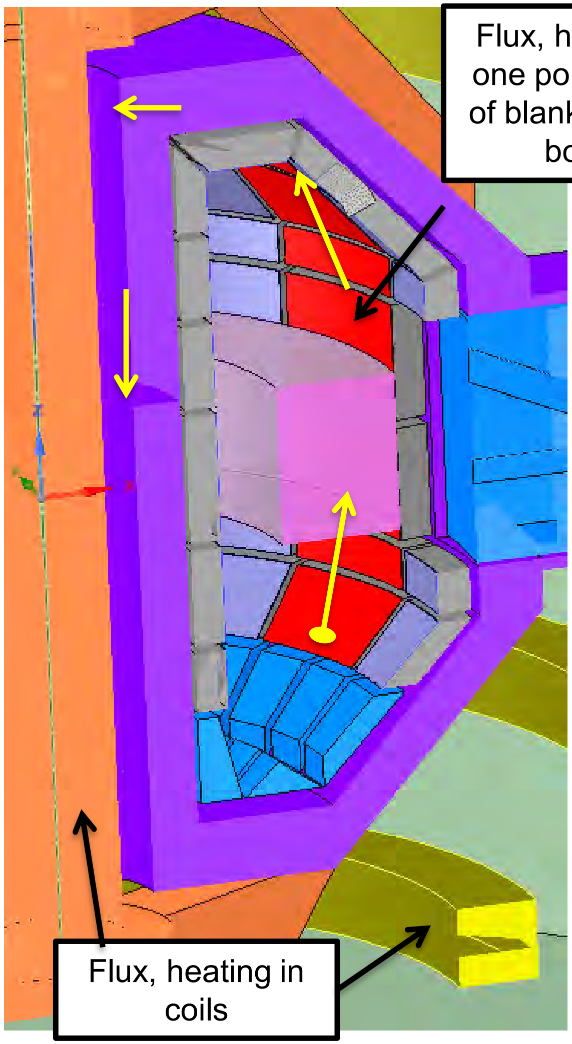
- At the time of this study, no weight window support in Serpent-2.

(It has since been added to Serpent-2)

- Densities were reduced in the vacuum vessel to permit practical calculation without VR.

MCNP-CSG ran to 10^9 histories (for reduced statistical error in the reference scenario). All other cases ran to 10^8 histories

Cell tallies set up in blankets and coils…

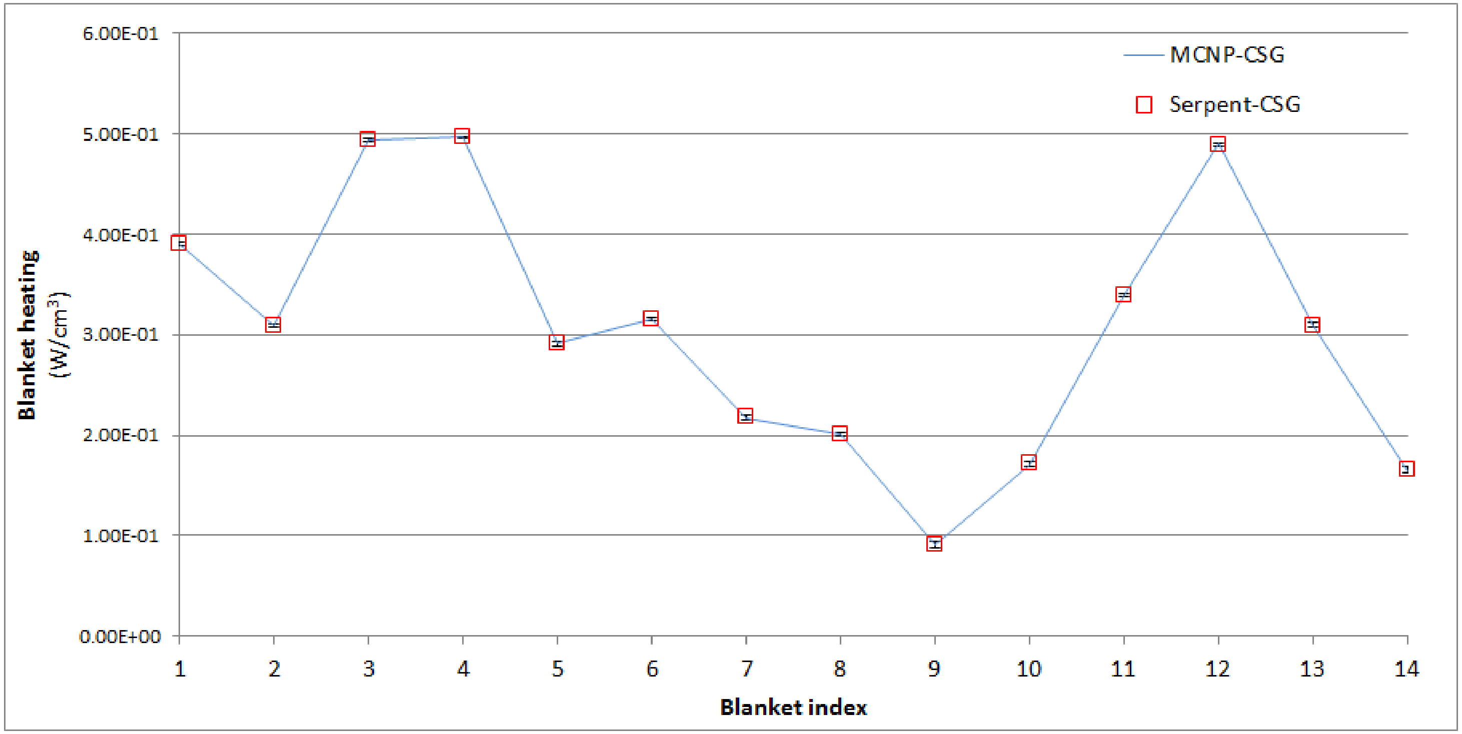


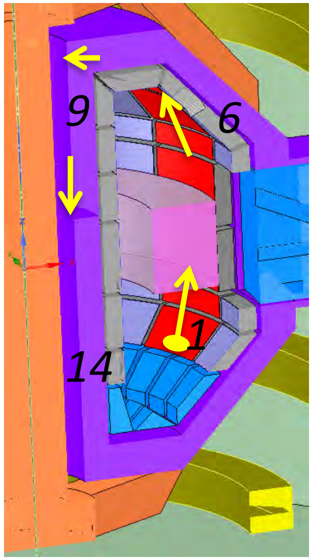
Flux, heating in one poloidal set of blankets (rear body)

Flux, heating in coils

Results – comparison of **tally values**

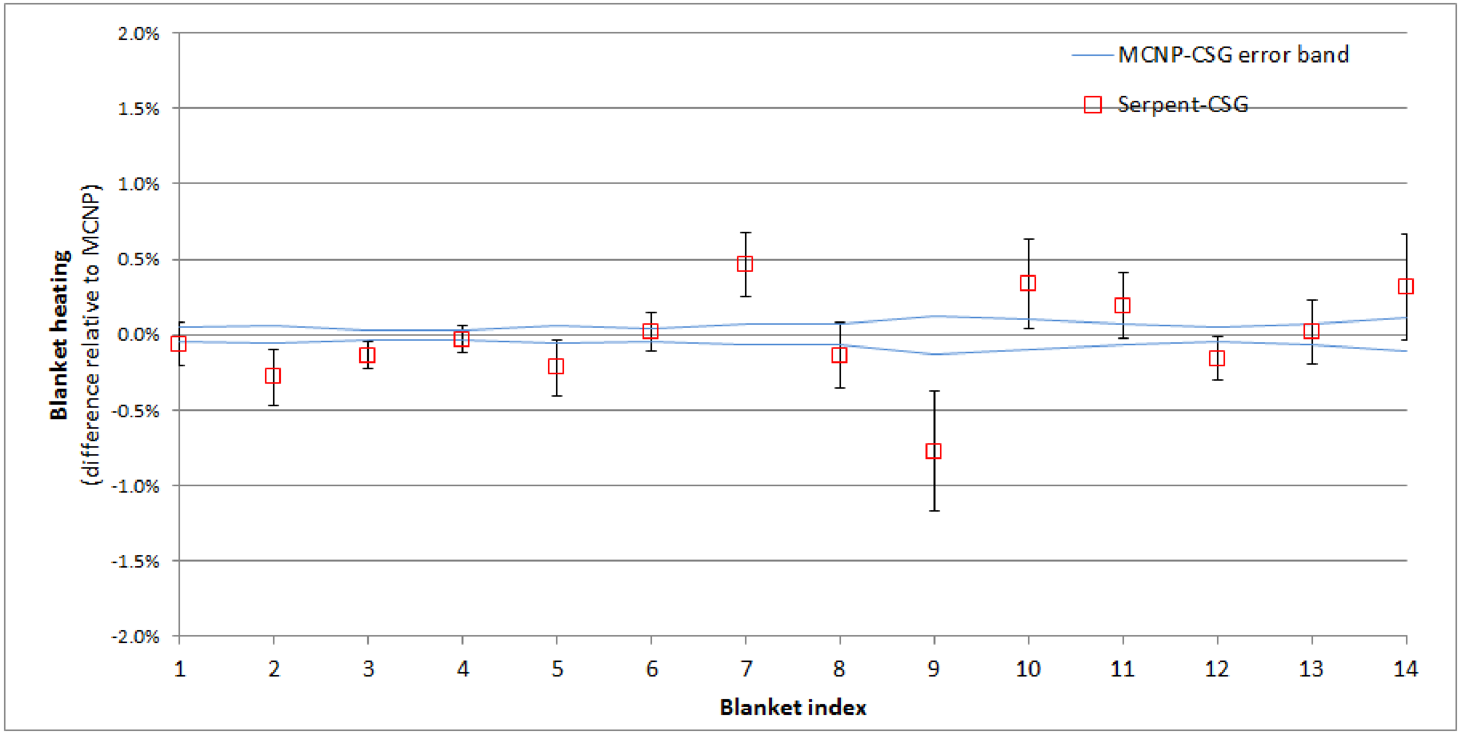
Neutron heating in rear layer of blanket – MCNP (CSG) vs Serpent2 (CSG)

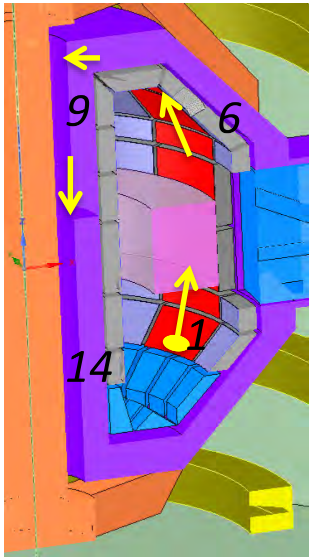




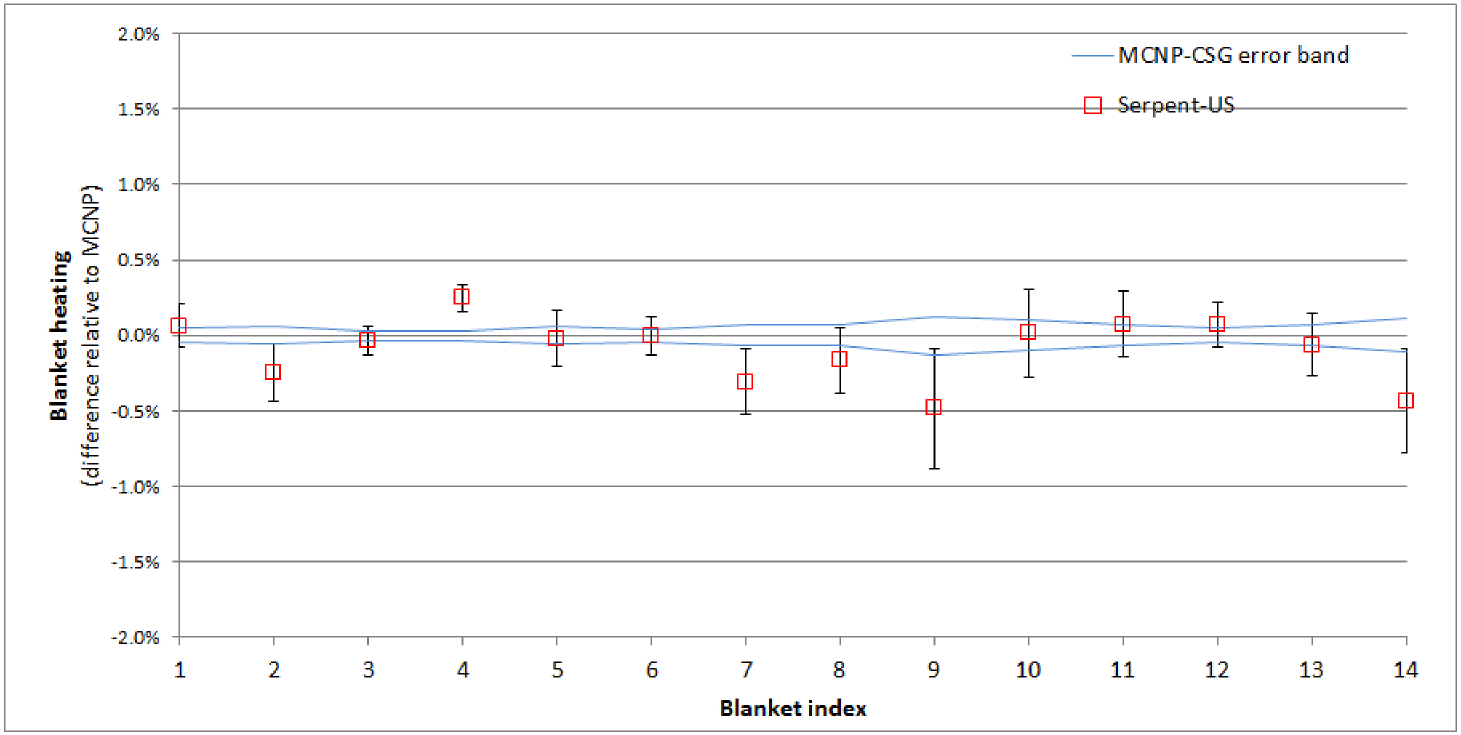
Relative results follow

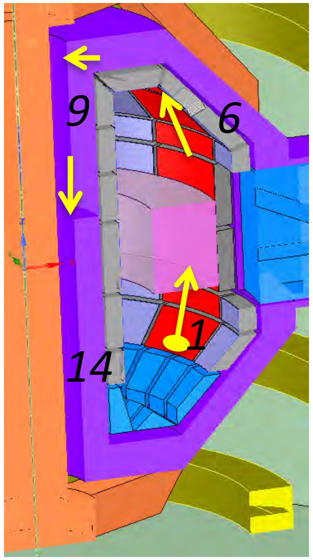
Neutron heating in rear layer of blanket – MCNP (CSG) vs Serpent2 (CSG)



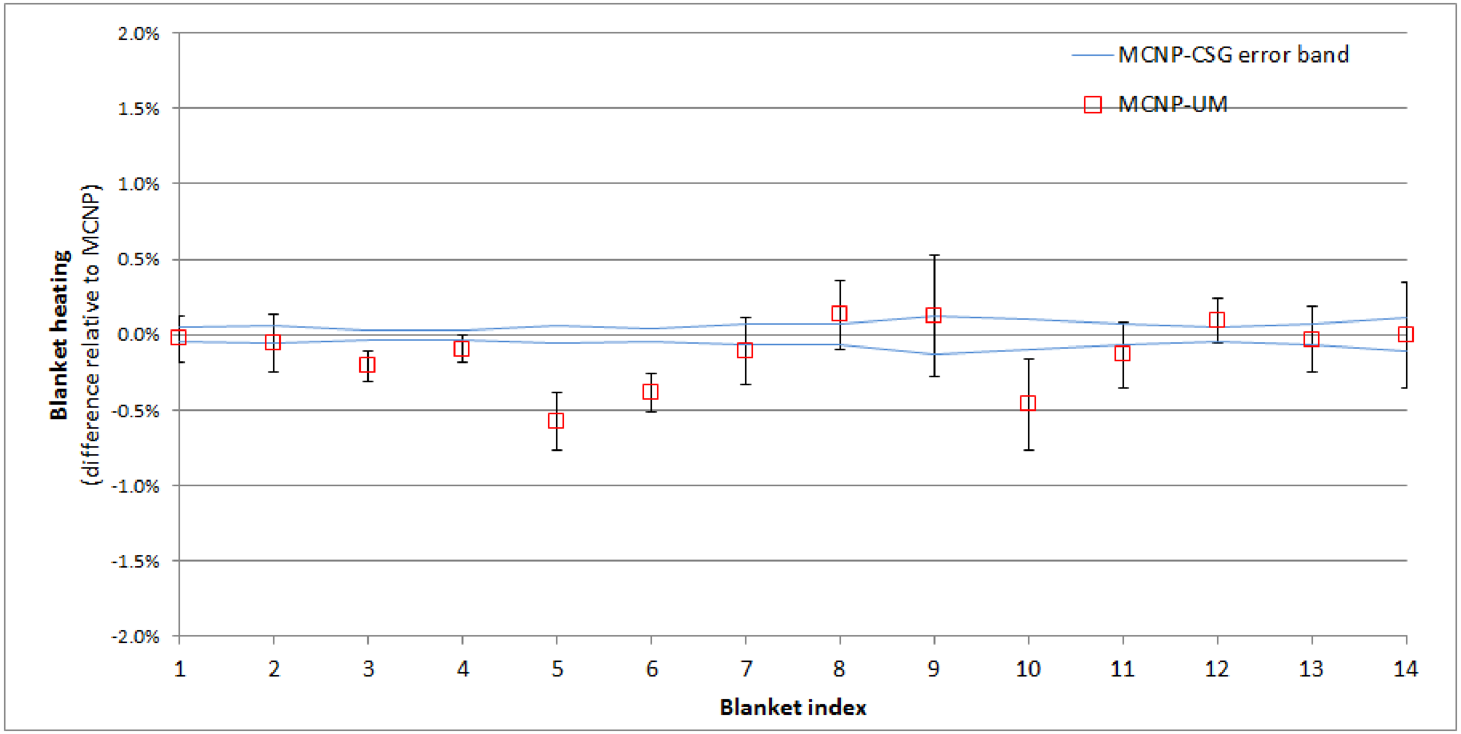


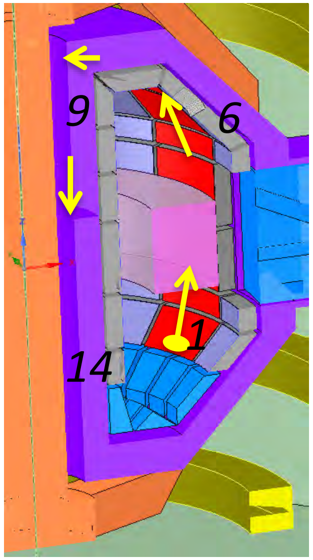
Neutron heating in rear layer of blanket – MCNP (CSG) vs Serpent2 (US)



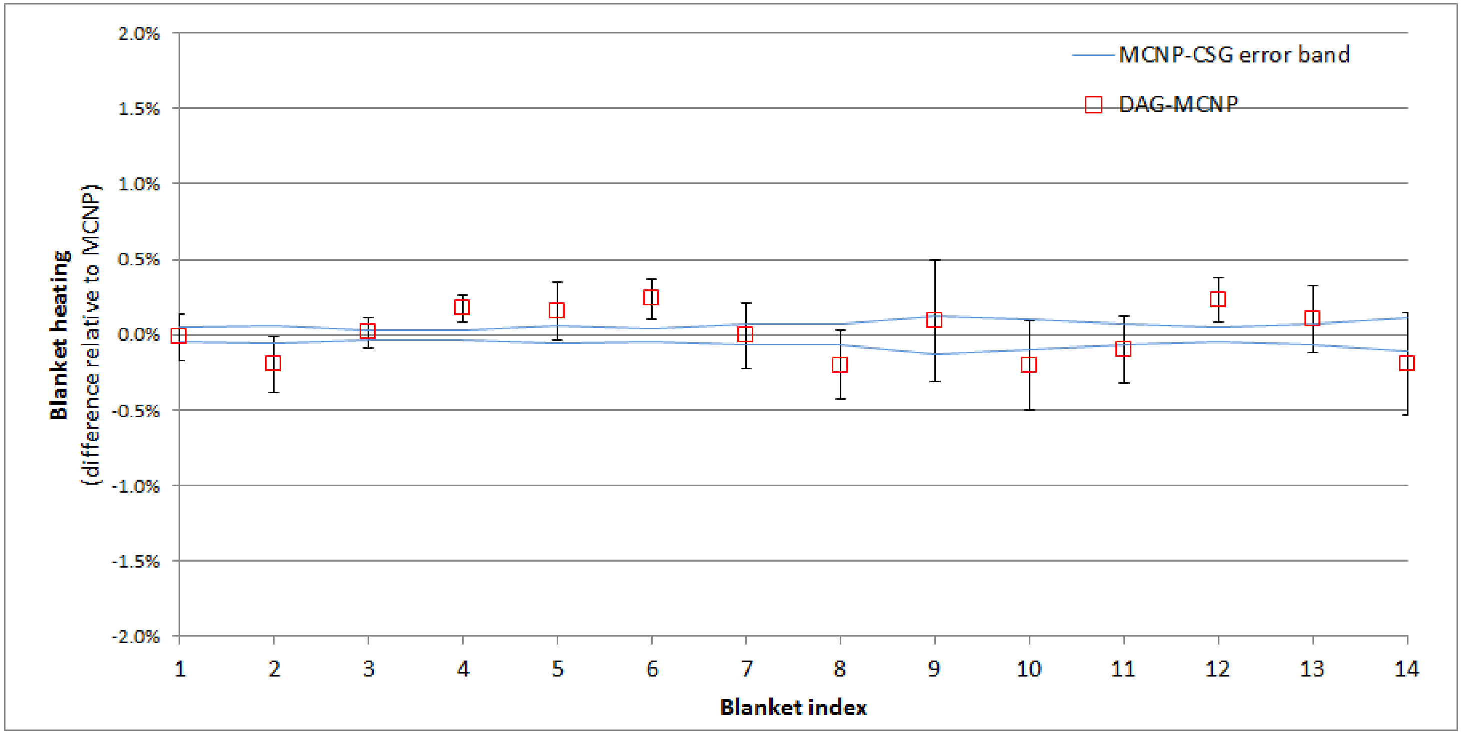


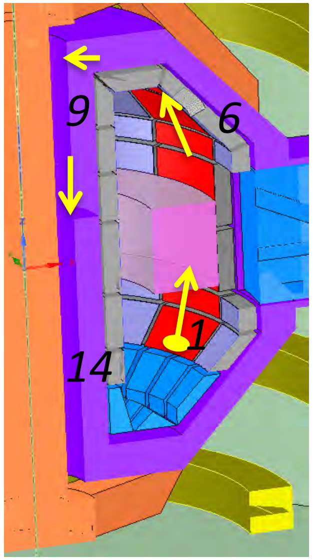
Neutron heating in rear layer of blanket – MCNP (CSG) vs MCNP (UM)





Neutron heating in rear layer of blanket – MCNP (CSG) vs DAGMC (US)





Neutron heating in SC coils

	Total heating (kW)	Statistical error	Relative to MCNP-CSG
MCNP-CSG	13.49	0.16%	1.000
Serpent-CSG	13.49	0.53%	1.000
Serpent-US	13.56	0.52%	1.005
MCNP-UM	13.52	0.49%	1.002
DAG-MCNP	13.45	0.49%	0.997

Tally values

- Mostly within 3σ, all within 1% of MCNP-CSG.
- No concerning differences observed between codes.
- All mesh-based implementations give similar results to CSG counterpart.

Results – comparison of **speed / memory**

Tally efficiency

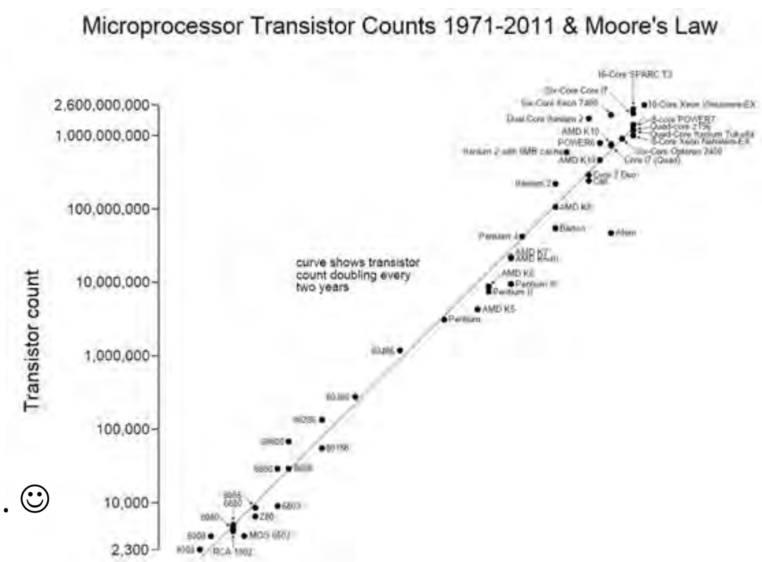
Mesh-based implementations expected to be less computationally efficient.

Bad? Maybe not. Consider:

It is a trade-off between human effort and computational effort.

Easier to parallelise MC code, than to distribute CAD simplification activities between people!

Computers are becoming ever more capable - humans not so much. ☺



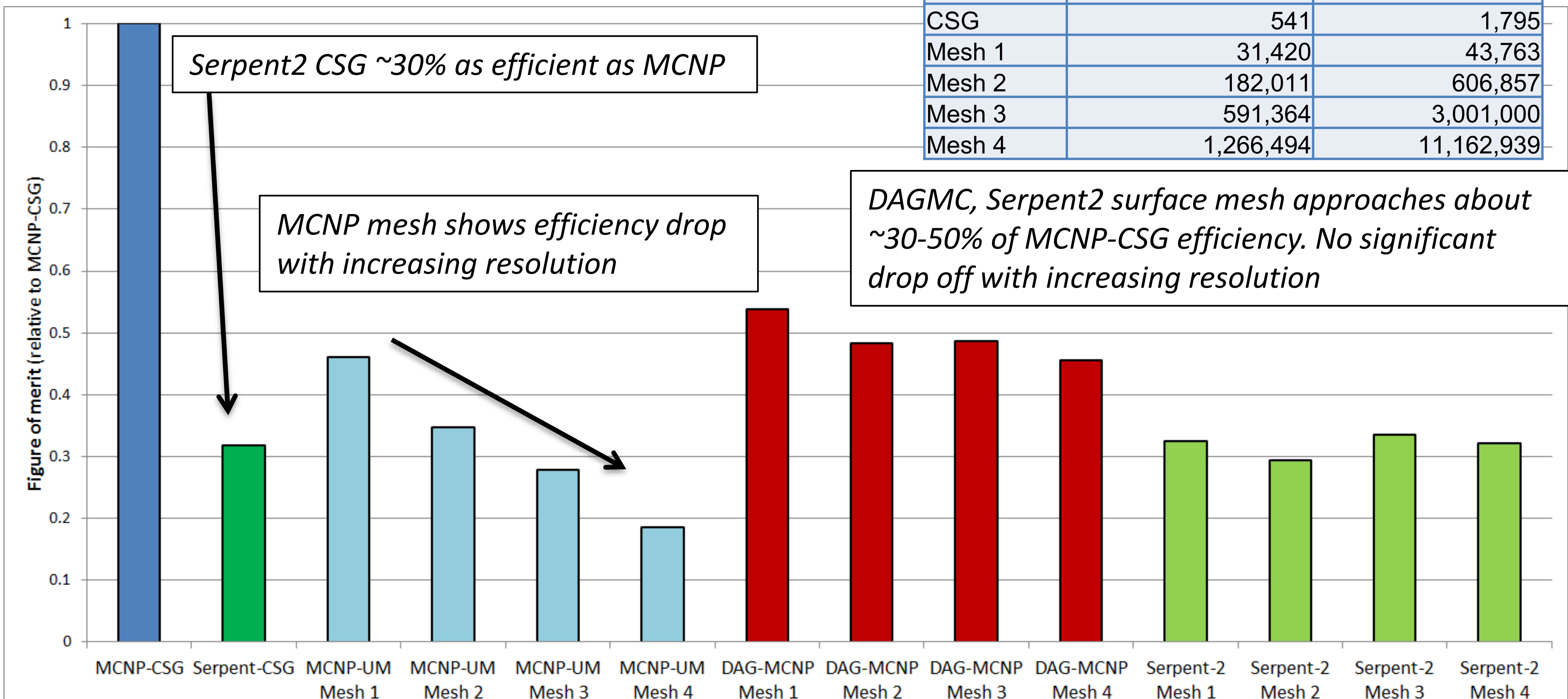
Compare tally Figure of Merit :: FoM = 1 / [err^2.time]
(statistical error reducing power per unit computation).

Tally Figure of Merit

Blanket heating tally FoM, normalised to MCNP-CSG

How does this change with increasing mesh resolution?

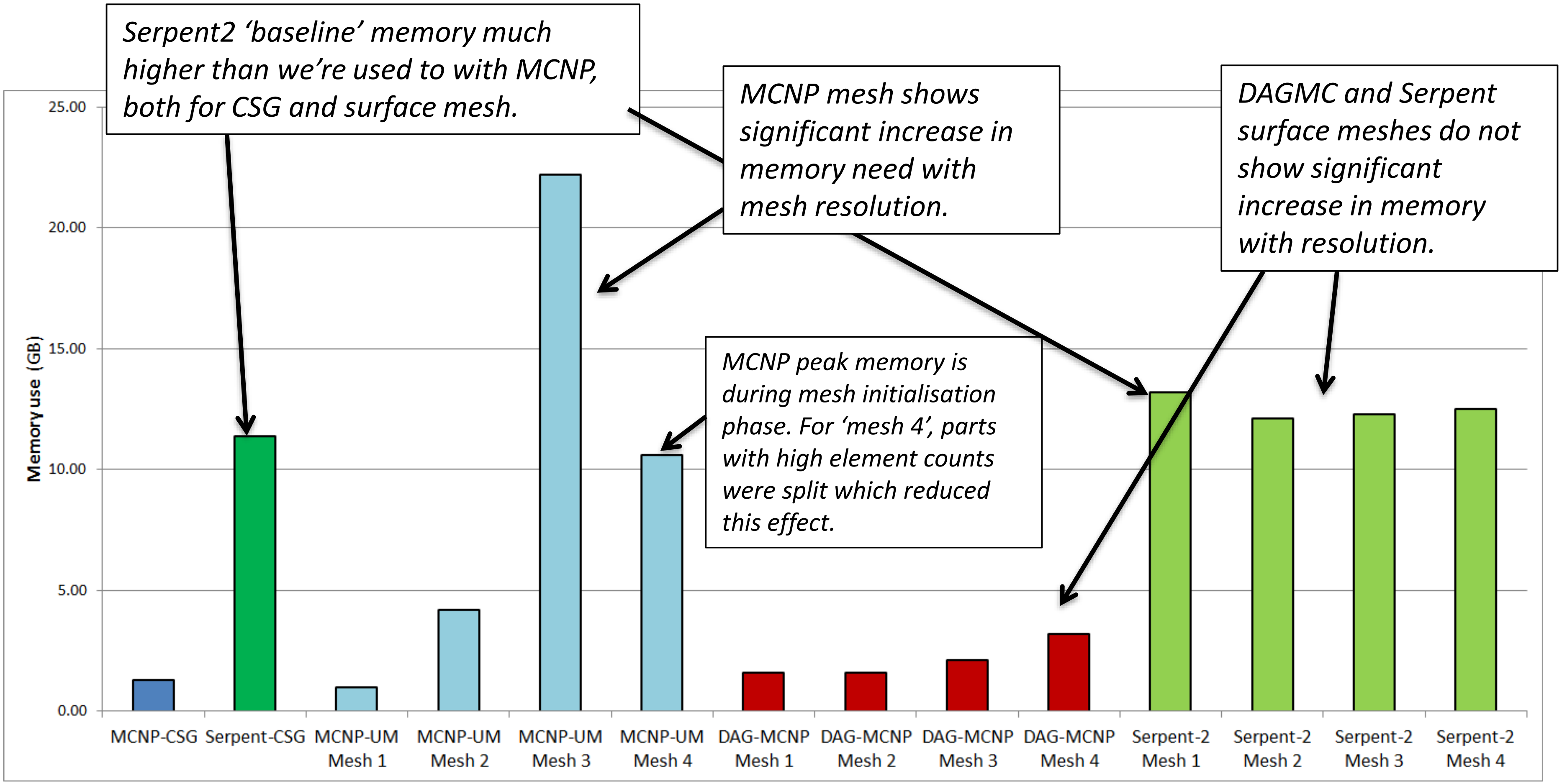
	Surfaces/facets	Cells/elements
CSG	541	1,795
Mesh 1	31,420	43,763
Mesh 2	182,011	606,857
Mesh 3	591,364	3,001,000
Mesh 4	1,266,494	11,162,939



Speed penalty seems practical.

Surface mesh approaches look particularly promising to scale to ITER size.

Computational memory requirement



Memory use seems practical.

Surface mesh approaches look particularly promising to scale to ITER size.

Results – observations of **usability / features**

Features / usability

Feature	MCNP	DAG-MCNP	Serpent2	Benefit
Implicit background void in mesh	✓	✓	✓	No need to mesh the void (conformal meshing of solids and void is hard!).
Mixed universes	~✓	X	✓	Use both CSG and mesh geometry. Re-use parts of reference CSG models.
No universe boundary restrictions	X	N/A	✓	Avoid fixing geometry errors due to interactions between model and universe boundary.
Modular geometry creation	✓	X	✓	Separately mesh components rather than the entire model. Re-mesh only modified parts.
Tolerance to overlaps	✓	X	✓	Less stringent CAD requirements. Necessary for separately meshed components.
OpenMP threading	✓	X	✓	Memory efficiency

MCNP6.1 limited to single mesh universe. MCNP6.2 perhaps not? Serpent supports arbitrary numbers of CSG and mesh universes. Can re-use existing parts of C-Model CSG. DAGMC does not support mixed geometry types - the entire model must be available in a form that can be successfully meshed.

MCNP6 does not permit mesh geometry to be clipped by the universe, plus usual CSG coincident surface issues. Serpent2 has no issues with coincident boundaries or clipping of any geometry type.

MCNP6 and Serpent permit components within a universe to be separately meshed and accommodates overlaps. DAGMC requires the entire geometry meshed in a single file with a watertight mesh.

MCNP6 and Serpent support hybrid MPI x OMP parallelism. DAGMC as yet does not (in development).

Conclusions

Conclusions: Workflow issues

Current workflows are sub-optimal due to:

- Time spent simplifying CAD (e.g. splines)
- Time spent making hand modifications to resulting model (fix overlaps/gaps, universe-fill interactions).

Solution is to combine:

- Efficient meshing software
- RT code with mesh geometry, tolerance to overlaps, automatic void, a robust and general universe implementation.

Will enable:

- Faster model creation with increased model accuracy.
- Rapid-response analysis capabilities.
- Study alternative designs, quickly modify CAD and re-assess.

Conclusions: Results/speed/memory

Comparisons undertaken between MCNP6, DAGMC and Serpent2 for a simple tokamak-like model.

All mesh approaches found to yield believable results.

Speed / efficiency penalty was a factor of 2-5. Practical with modern computing.

DAGMC - fastest of the implementations and required least memory.

Serpent-2 was next fastest, needed more memory, but the most user-friendly implementation.

Surface mesh approaches seem best suited to ITER applications

MCNP was user-friendly when used with Attila4MC, but showed increases in run time and memory for increasing element counts.

Conclusions: usability – DAG vs Serpent

DAG-MCNP – no modular meshing, no universes

Need to have complete and meshable CAD for entire model.

Needs CUBIT/Trelis

Well-suited when CAD-based neutronics reference model exists - However ITER reference CAD does not yet exist.

Serpent-2 – supports multiple universes with mixed geometry types.

Mesh bodies as separate STL files. No need for specialist software.

Re-mesh and update only the components that have changed (even in a single universe).

Mesh components in different packages, e.g. shrink wrap mesher for problematic CAD.

Arbitrary mixtures of CSG and mesh universes - can retain CSG if desired.

Supports near-identical set of CSG surfaces to MCNP – easy to convert.

No restrictions on universe boundary coincident surfaces or clipping of mesh geometry.

Conclusions

Whilst DAGMC is more efficient, the Serpent-2 implementation is promising from an ease-of-use perspective for ITER applications.

Serpent-2 is not fully featured yet, but has a basic implementation of weight windows, and coupled N,P mode is in progress...

Chicken/egg situation for ITER where reference CAD does not yet exist.

Serpent – mixed geometry types provide a 'stepping stone' to a fully mesh-based approach (ITER neutronics currently MCNP-based). i.e.

- Retain existing MCNP-CSG universe structure of C-Model
- Insert mesh geometry for new items.
- Retain existing CSG components otherwise.

Recommendation: IO to store in analysis database new ITER system models as reference faceted geometry files (e.g. STL file)**.**

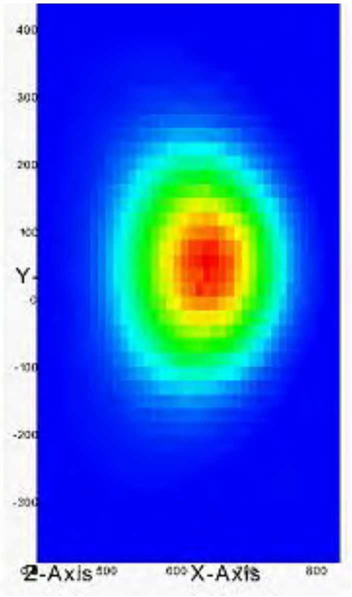
- No questions over suitable meshing resolution.
- STL conversions/import to MCNP-UM and DAG-MC from STL - possible?

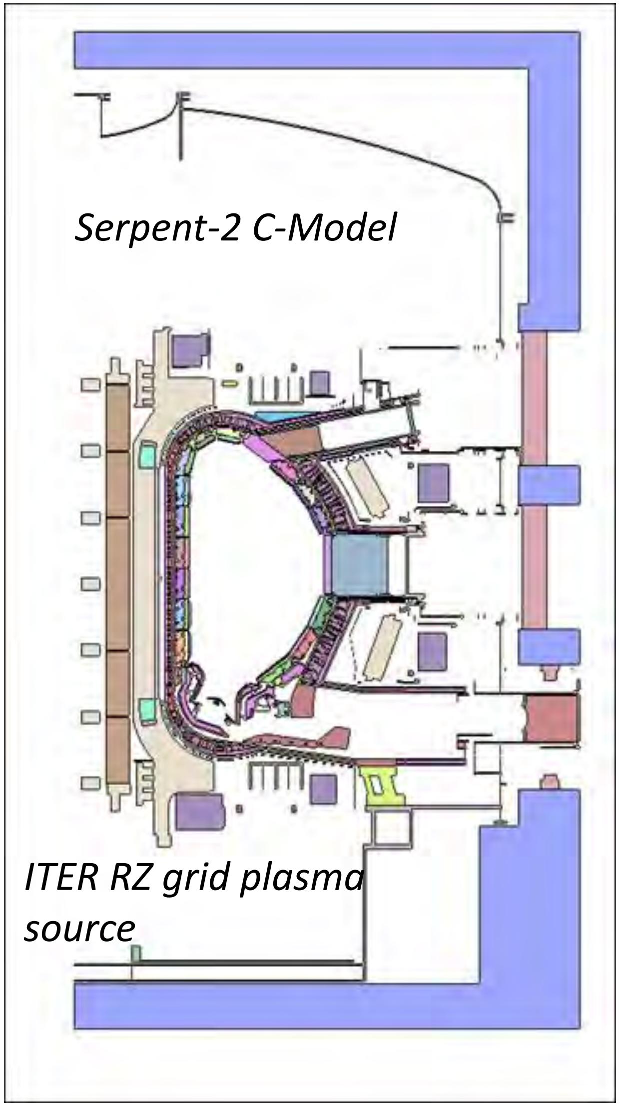
Further investigations ongoing to assess Serpent-2 for ITER applications.

Additional slides

Additional investigations 2017/2018

- Feasibility of scaling MCNP/DAG/Serpent meshes to ITER-size (10^{8-9} facets - promising, so far!).
- Benchmarking of Serpent against MCNP for simple cases and for C-Model CSG (fully converted!).
- Aiming to report at ITER neutronics meeting.





CUMULUS

CCFE's latest HPC cluster 'CUMULUS' has been installed.

- 1128 cores (at present - will be expanded to 5000+ cores).
- 32 cores, 512 GB RAM per node.
- Eventually, will be part of a cloud computing infrastructure.

- High memory, high core count ideal for mesh-based neutronics modelling.
- Coupled with an effective mesh-based radiation transport code, CUMULUS will provide an ideal platform to perform rapid nuclear analysis for ITER and beyond.

CUMULUS is opened by Eng Lim Goh and Ian Chapman (middle of photo), accompanied by the project team and senior managers

Thank you for your attention