The KANEXT modular program for reactor physics calculations

S. Van Criekingen, R. Dagan, M. Becker and C.H.M. Broeders

+ Contributions from
M. Badea, B. Becker, A. Ponomarev and D. Christophe

What is KANEXT ?

- KArlsruhe Neutronic EXtendable Tool
- The follow-up version of the KArlsruhe PROgram System KAPROS, develope and applied in many projects at FZK since the 1970's.
- A modular multi-purpose tool for deterministic reactor calculations.
- A non-commercial tool suitable for research and education.

Why KAPROS ⇨ KANEXT ?

- To save and transfer the accumulated knowledge.
- To make it available on the newest computer platforms, improving robustness and portability, while maintaining down-compatibility (same inputs).
- To provide a professional software environment for further developments / extensions.

Characteristics of KAPROS/KANEXT

- The KAPROS/KANEXT code system is the main production tool for most of fast reactors projects in which INR is involved.

- Validation of the system was performed recently based on available KWO burn up data, and previously based on epithermal experiments at PROTEUS facility at PSI, as well as in an earlier stage by participation in several IAEA and OECD benchmarks. Recent validation is also obtained by code to code comparison to MCNP and to the ERANOS system within the EC projects HPLWR, XTADS, TRADE and IP EUROTRANS.

- In the last years, the code capabilities were applied to some extent for Light Water Reactors and for Gas Cooled Reactors. This was mainly done in the framwork of diploma theses.

- Suitable for basic tutorial studies.

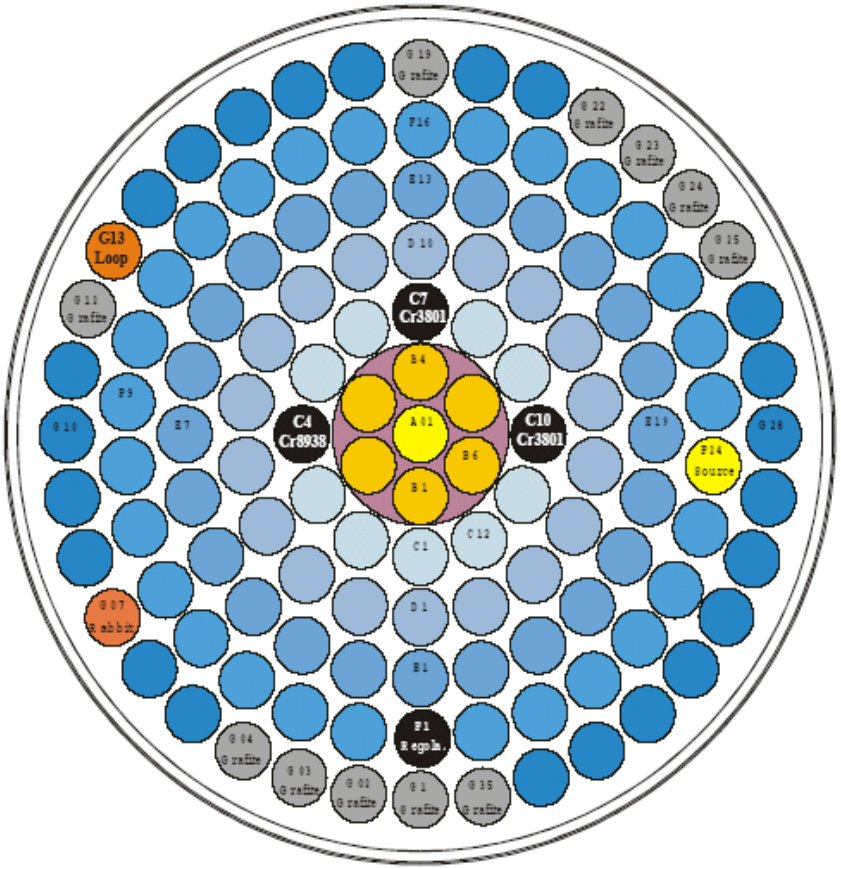
Some advantageous features of KAPROS/KANEXT

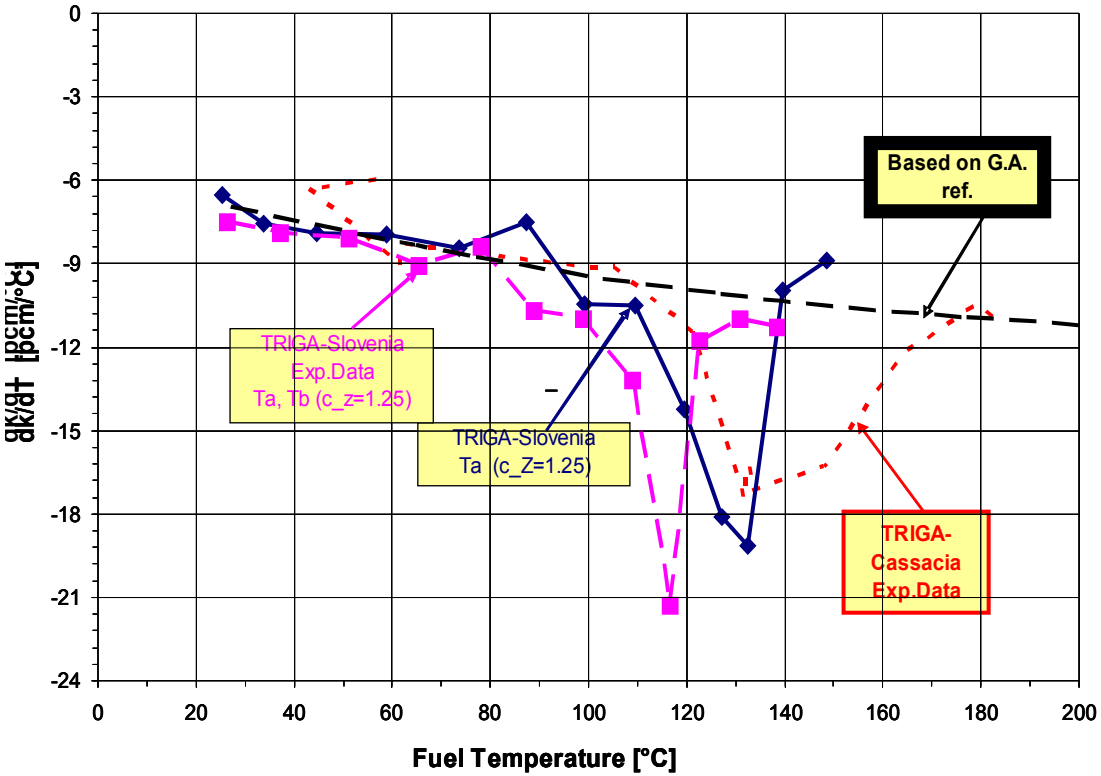
- Modularity and flexibility, especially in view of zone-wise cross section generation.
- The “secondary input” option for the cross section generation allows for sensitivity analyses of selected isotopes and reactor types from different resources.
- Suitable interface files for the cross section sets of most of the international neutron physic codes are available.
- A recently developed module enables burn up calculation together with fuel elements relocation for three-dimensional geometries
- Flexible coupling to thermal hydraulic codes proven

Reactivity phenomena of TRIGA fuel at 140 °C

The reactivity measurements were carried out in two TRIGA cores:

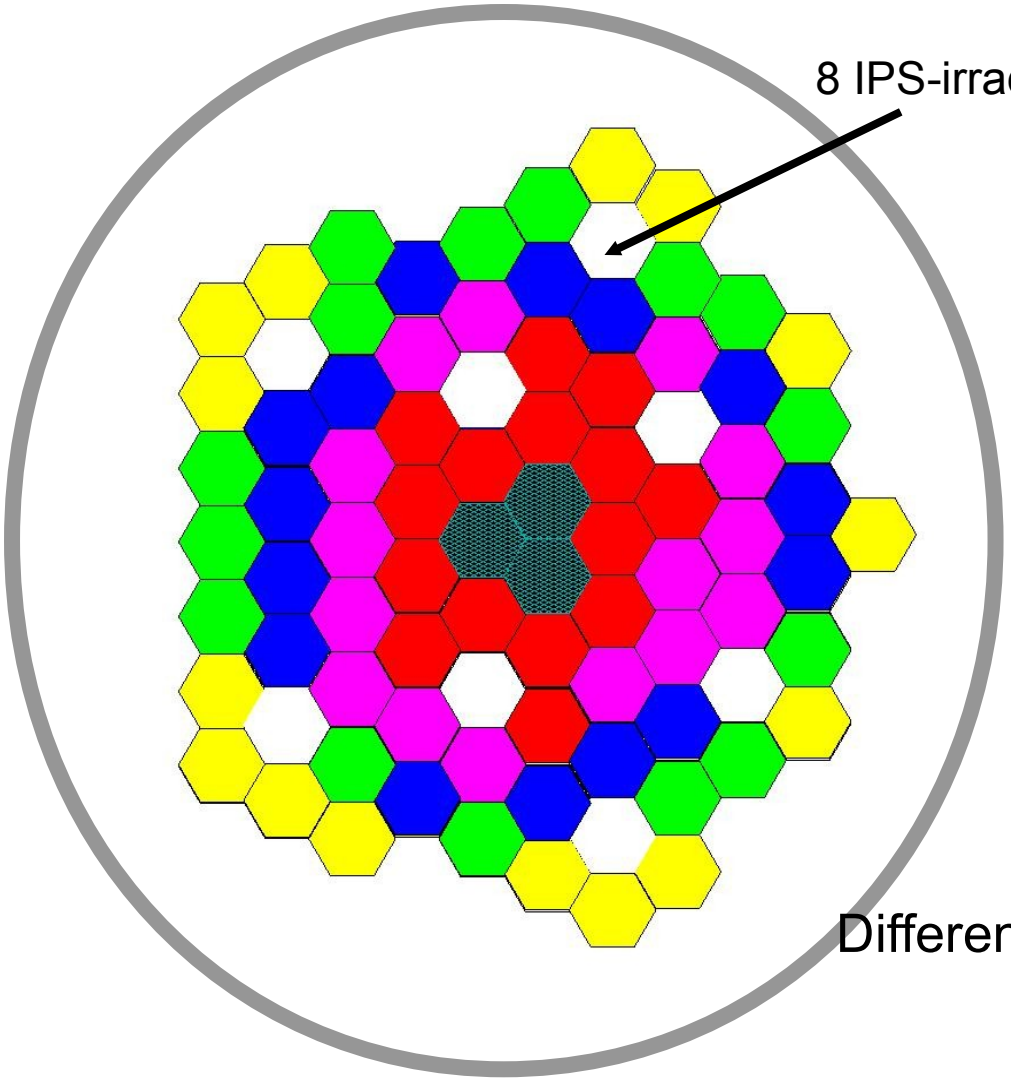
- TRIGA RC-1 Cassacia with the existing core configuration (burned-fuel)
- TRIGA Mark II Slovenia with completely fresh uniform core



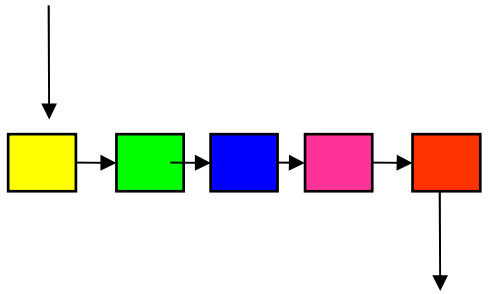


The reactivity feedback coefficients for two TRIGA cores experimental results (Cassacia and Slovenia) and the G.A. reference calculations

Optimized reshuffling of the fuel during operation



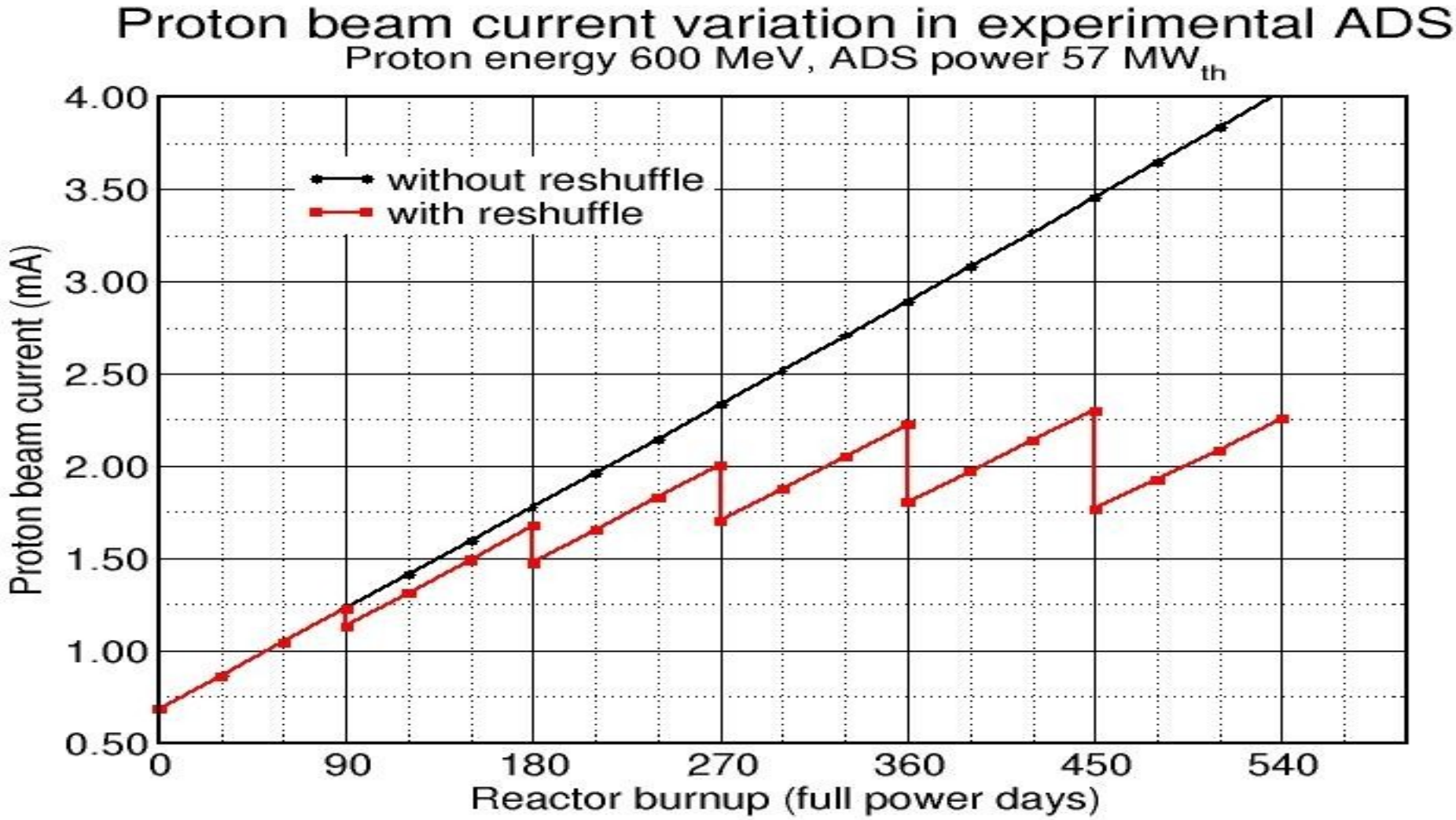
Insertion of Fresh fuel sub-assembly at the core periphery



Extracting fuel elements from the core inner zone

Different Color = different fuel assembly group

Beam Current Reduction for XTADS

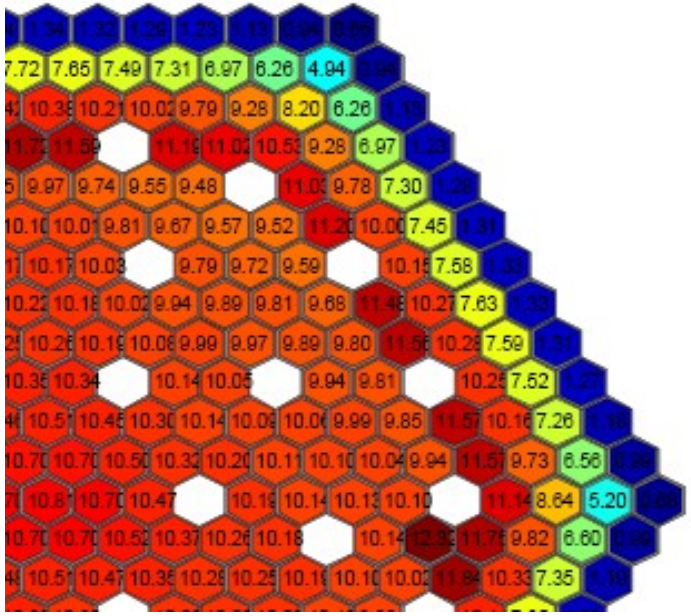


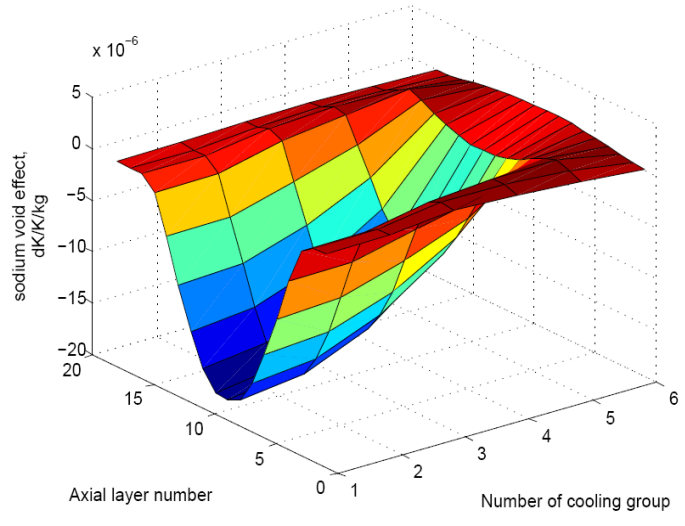
Activities on fast neutron sodium cooled reactor

Main goals of neutronic calculations:

getting a wide range of neutron physics parameter

- Keff for fresh/equilibrium core;
- power distribution and peaking factors;
- integral and local reactivity coefficients (Doppler reactivity, sodium density reactivity, clad and fuel material reactivity);
- sodium void effect;
- CRs worth;
- power and thermal core expansion reactivity effect;
- breeding ratio;





Coupling of TORT-TD and KANEXT

TORT-TD

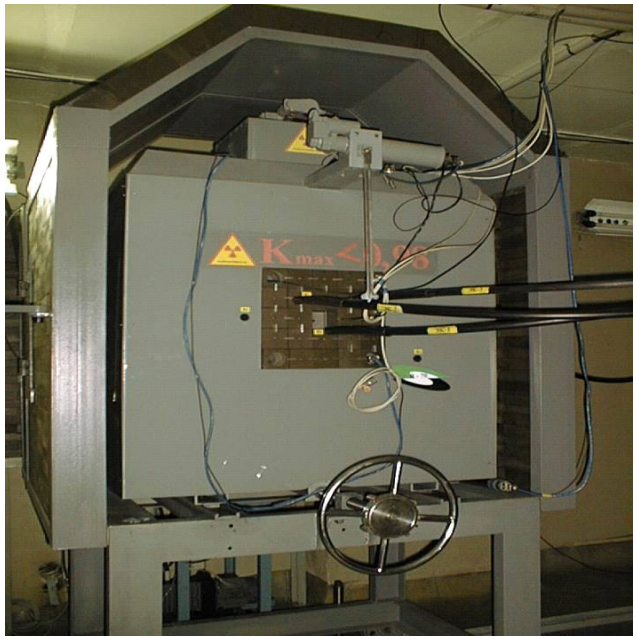
- GRS development based on DOORS/TORT discrete ordinate code from ORNL
- Solves the 3-D, steady-state and time-dependent multi-group Boltzmann equation directly
- Modified to account for time-dependent, anisotropic sources
- Suitable to simulate 3-D, spatial resolved flux evolution after a neutron pulse

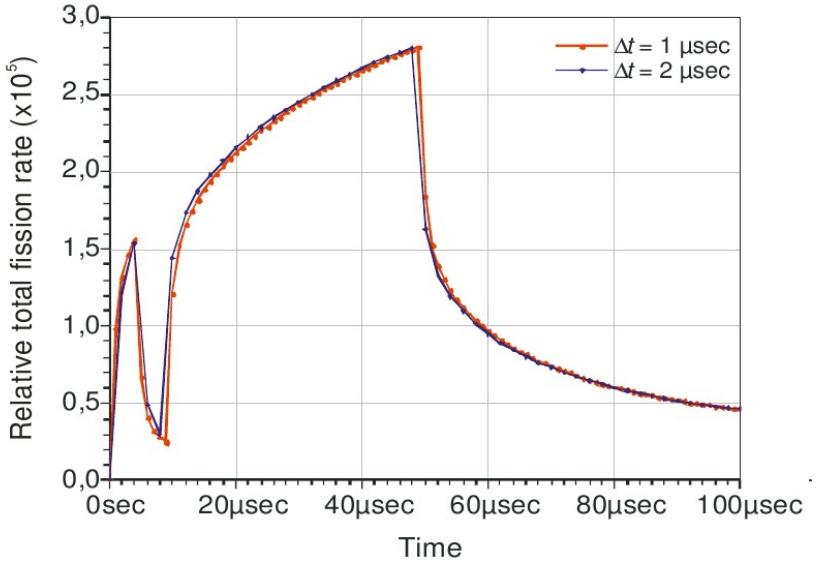
KANEXT

- Writes cross sections with interface module CRGIP in TORT readable format
- Provides prompt and delayed nuclear data averaged over pin cell with module CHICOR
  → dependent on isotopic content
  → prompt and 6-8 delayed fission spectra
  → prompt and 6-8 delayed fission yields
  → delayed time group constants

Preliminary results for source simulation

- Pulse simulation of YALINA reactor in Minsk, in cooperation with GRS.
- Time-dependent (d,t) source with rectangular shape of 5μs and 40μs duration seperated by 5μs
- The difference between the prompt neutron decay and delayed neutron decay is pronounced
- TORT-TD/KANEXT is the most advanced deterministic transport code capable of simulating a time-dependent source problem dependening on the cross section quality

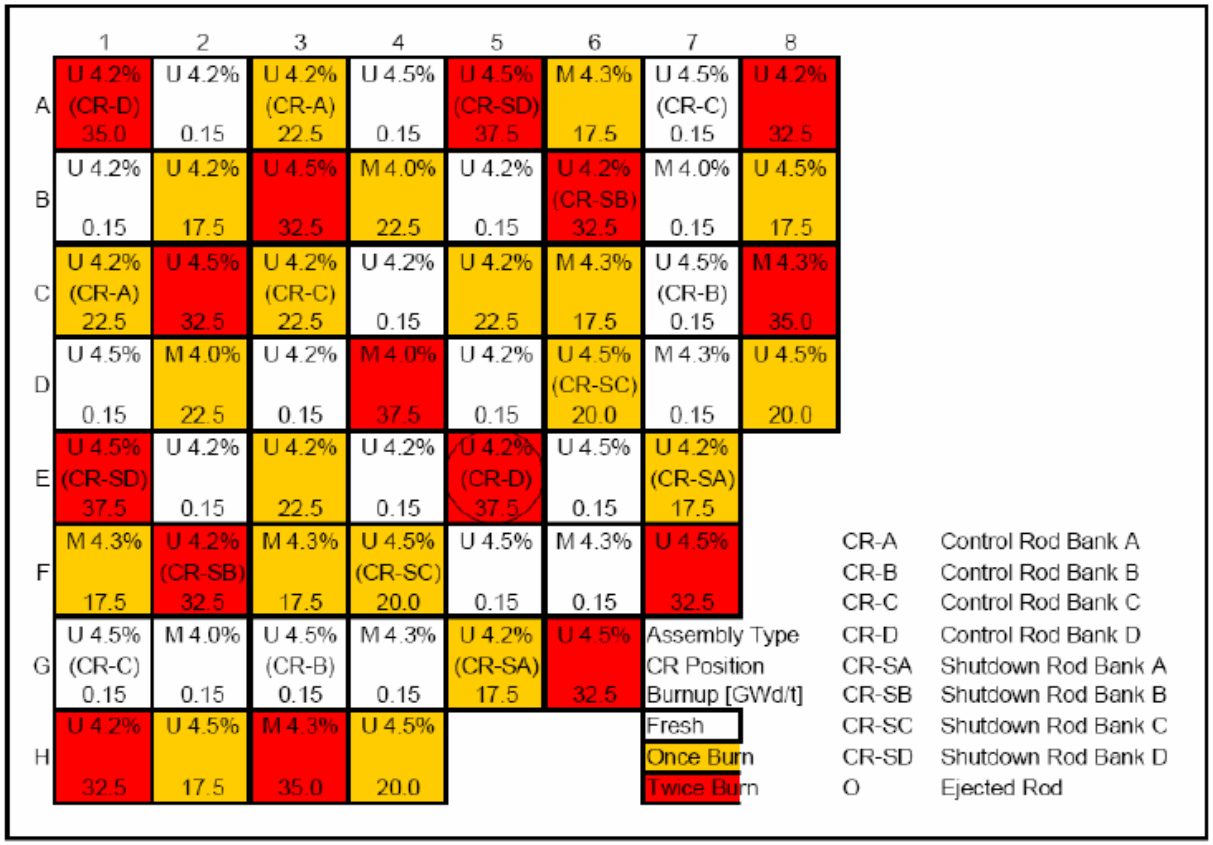




A. Seubert et al. (GRS), M&C 2009

PARCS – KAPROS coupling

- Example: Stationary part of OECD/NEA and U.S. NRC PWR MOX/UO_2 Core Transient Benchmark
- Coupling via KAPROS modules DANHOM and CRPAXS for stationary PARCS calculations



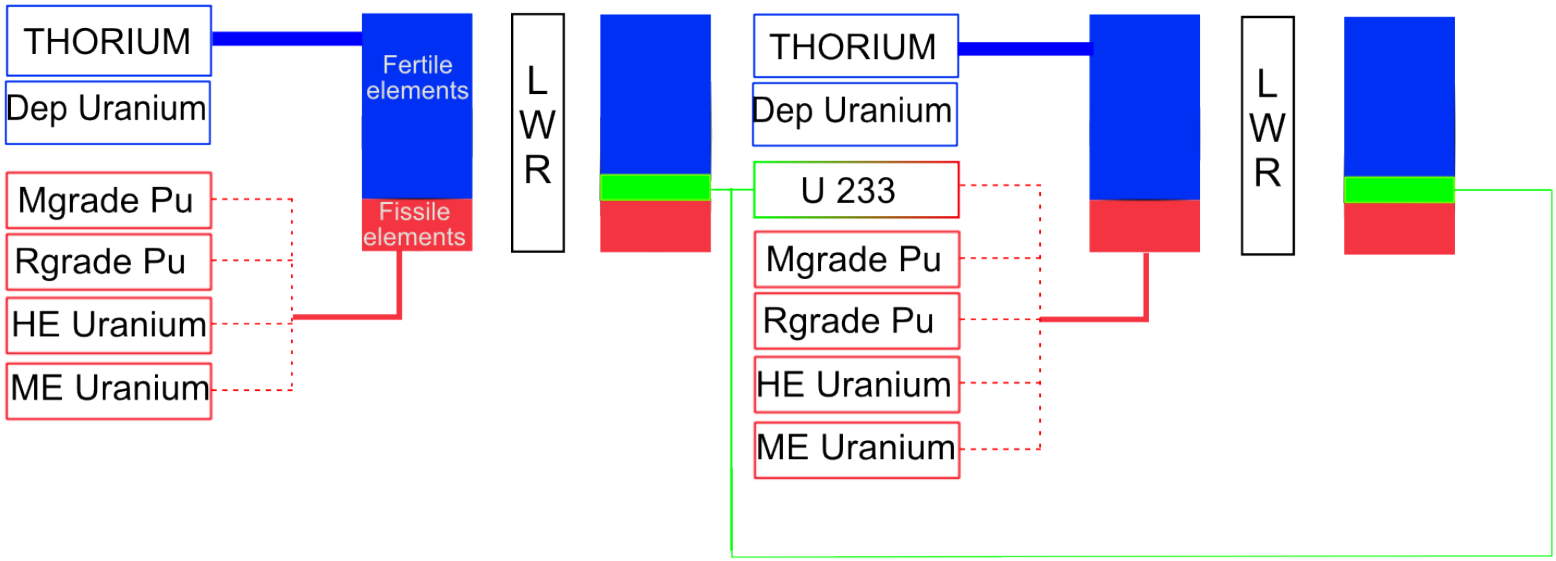
	1	2	3	4	5	6	7	8
A	U 4.2% (CR-D) 35.0	U 4.2% 0.15	U 4.2% (CR-A) 22.5	U 4.5% 0.15	U 4.5% (CR-SD) 37.5	M 4.3% 17.5	U 4.5% (CR-C) 0.15	U 4.2% 32.5
B	U 4.2% 0.15	U 4.2% 17.5	U 4.5% 32.5	M 4.0% 22.5	U 4.2% 0.15	U 4.2% (CR-SB) 32.5	M 4.0% 0.15	U 4.5% 17.5
C	U 4.2% (CR-A) 22.5	U 4.5% 32.5	U 4.2% (CR-C) 22.5	U 4.2% 0.15	U 4.2% 22.5	M 4.3% 17.5	U 4.5% (CR-B) 0.15	M 4.3% 35.0
D	U 4.5% 0.15	M 4.0% 22.5	U 4.2% 0.15	M 4.0% 37.5	U 4.2% 0.15	U 4.5% (CR-SC) 20.0	M 4.3% 0.15	U 4.5% 20.0
E	U 4.5% (CR-SD) 37.5	U 4.2% 0.15	U 4.2% 22.5	U 4.2% 0.15	U 4.2% (CR-D) 37.5	U 4.5% 0.15	U 4.2% (CR-SA) 17.5	
F	M 4.3% 17.5	U 4.2% (CR-SB) 32.5	M 4.3% 17.5	U 4.5% (CR-SC) 20.0	U 4.5% 0.15	M 4.3% 0.15	U 4.5% 32.5	
G	U 4.5% (CR-C) 0.15	M 4.0% 0.15	U 4.5% (CR-B) 0.15	M 4.3% 0.15	U 4.2% (CR-SA) 17.5	U 4.5% 32.5	Assembly Type CR Position Burnup [GWd/t]	
H	U 4.2% 32.5	U 4.5% 17.5	M 4.3% 35.0	U 4.5% 20.0			Fresh Once Burn Twice Burn	

CR-A	Control Rod Bank A
CR-B	Control Rod Bank B
CR-C	Control Rod Bank C
CR-D	Control Rod Bank D
CR-SA	Shutdown Rod Bank A
CR-SB	Shutdown Rod Bank B
CR-SC	Shutdown Rod Bank C
CR-SD	Shutdown Rod Bank D
O	Ejected Rod

- Agreement between KAPROS + PARCS (28 groups) and MCNP within 100 pcm for k_{eff}

- KAPROS/PARCS calculations by Ph. Oberle (INR)
- MCNP calculations by Kurchatov Inst. (Russia)

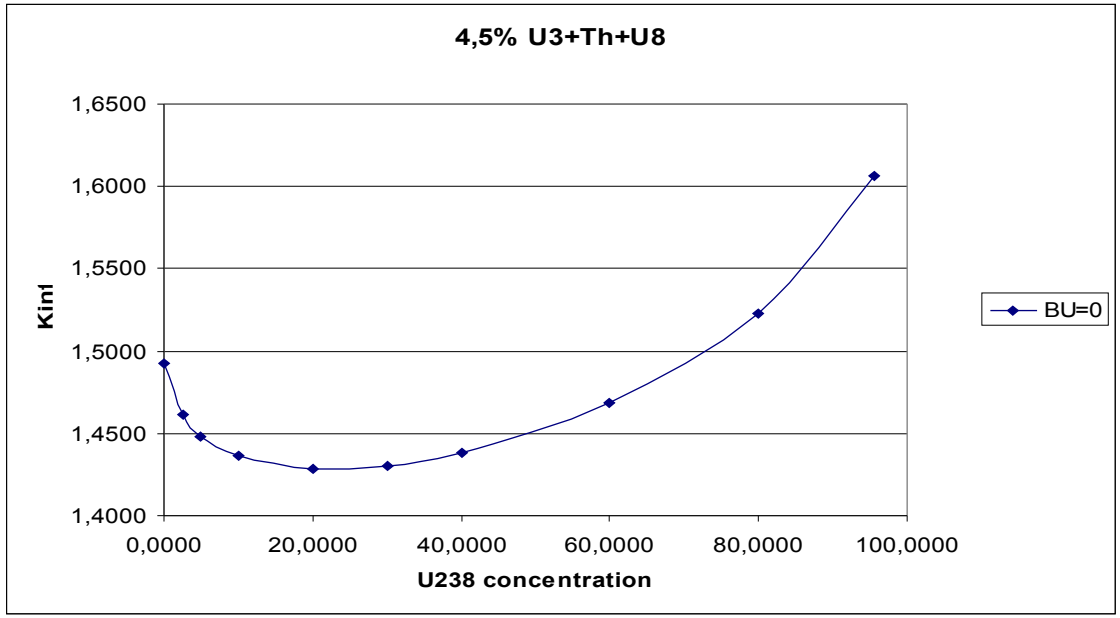
Multiple Recycling with Thorium fuel



Criticality

Burn-up calculations

Safety coefficients



Kinfinity as function of U238 fraction

Code structure

- It consists of kernel routines managing the flow of information between several modules assigned to a specific task.

- The code is written mainly in Fortran, with a thin layer of C-routines in the kernel. Open-source g95 and gcc compilers are used (+ Lahey fortran for further testing).

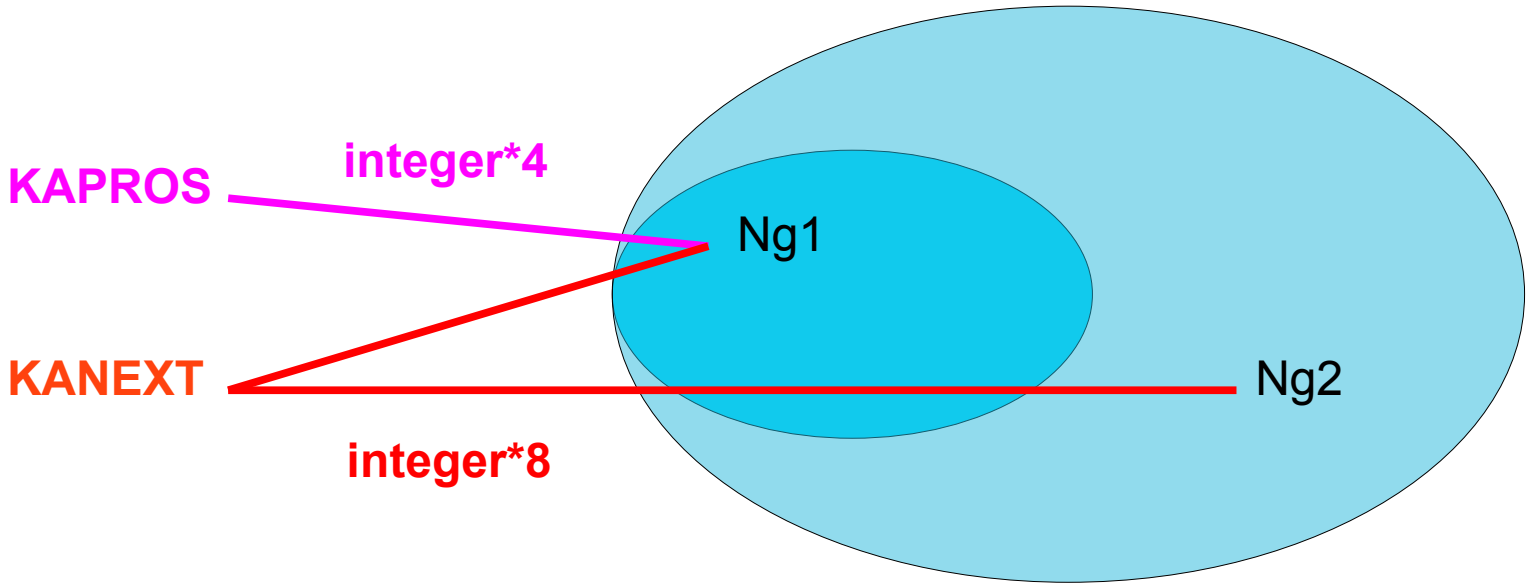
- Modules can call each other (next-level call), with transfer of shared memory. Recursivity possible.

- The coupling of external codes (e.g., VARIANT,...) is performed through interface files.

From KAPROS to KANEXT (1/2)

- KANEXT can operate on both 32- and 64-bit platforms.

- Adressable physical memory increased
- Data: number of energy groups (Ng1, Ng2), cross sections, ...
- Memory addresses moved from integer*4 (2^32) to integer*8 (2^64)



From KAPROS to KANEXT (2/2)

- Next-level calls (module calling another module) are now handled in a more portable manner.
- Shared memory management has been made more robust.
- Archiving can be done at any time in the execution.
- Nuclear data libraries provided in ASCII format, and translated into binary files once at installation (to ensure compatibility).

Current status of the upgrade

- All kernel routines have been updated, as well as most modules.
- The source files of the external coupled codes are being integrated within the KANEXT file repository.
- Test examples are available, both basic (to illustrate the modularity handling) and advanced (real-life studies).
- A professional software environment TRAC + Subversion is available.

TRAC & SVN demonstration

Next steps for KANEXT

- Update and testing of the remaining modules.
- Integration and testing of the remaining coupled codes.
- Internal release as full replacement for the current KAPROS.
- Further integration of user feedback.
- Improvement of the documentation.
- Mid-term aim: transfer to NEA databank for open distribution.

Possible (kan)extensions

Introduction of parallelization

- At the kernel level, enable the parallel execution of modules, e.g., for calculation of cross-section sets for many zones (~1000).

- Parallelization at the module / coupled code level

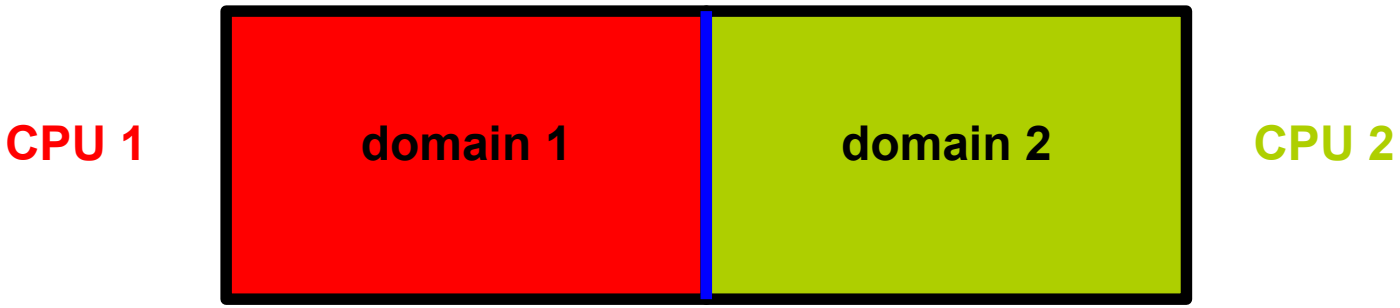
  - Today, among coupled codes, only PARTISN can run in parallel.

  - Ongoing work presented at M&C 2009:

    Parallel solver using domain decomposition methods.

Basic principle

To solve the Boltzmann transport equation, first split the reactor into several non-overlapping domains.



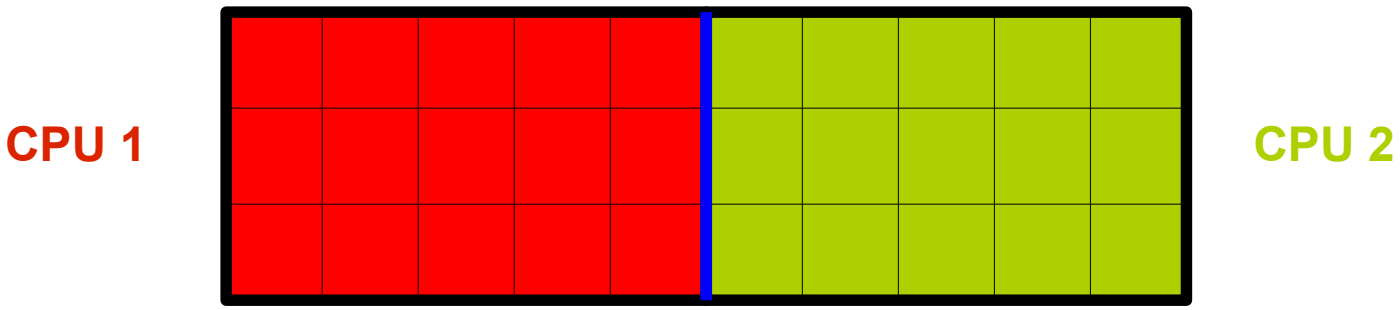
Then launch an iterative procedure:

- solve the Boltzmann equation in each domain („local solve“) can be done in parallel on different CPUs.
- use as boundary conditions for one domain, the interface values (+ derivatives) computed in the neighboring domains at the previous iteration.

Within each domain...

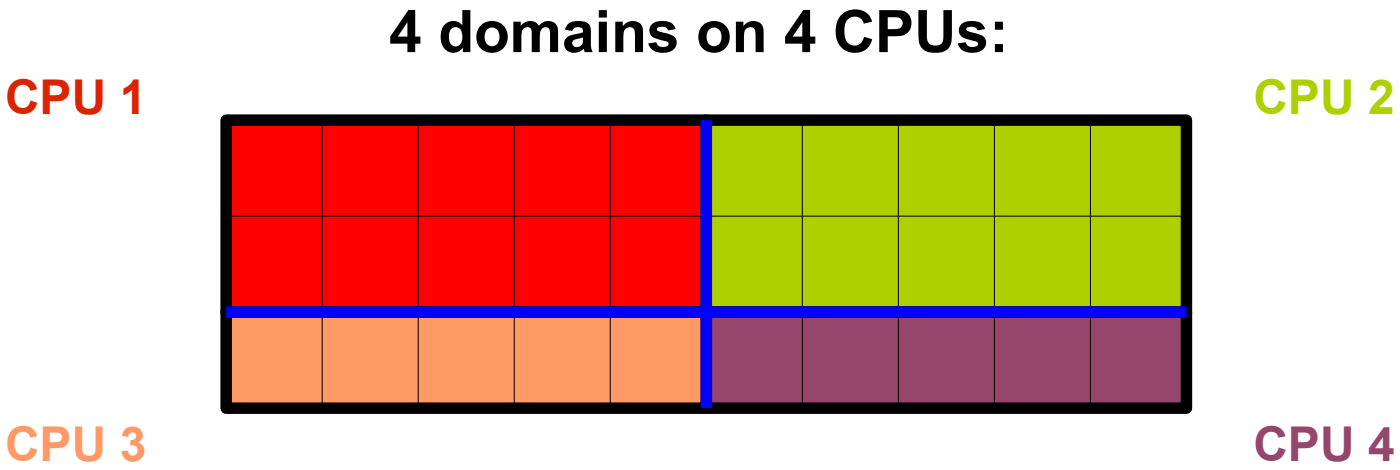
… the Boltzmann equation is solved using a Finite Element spatial discretization, and a Spherical Harmonic (Pn) angular discretization:

2 domains on 2 CPUs:



Interface finite element nodes are duplicated over CPUs.

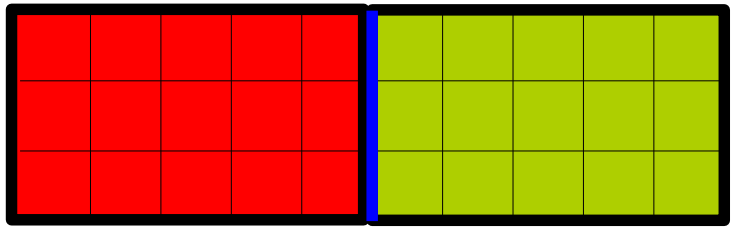
A 4-domain example



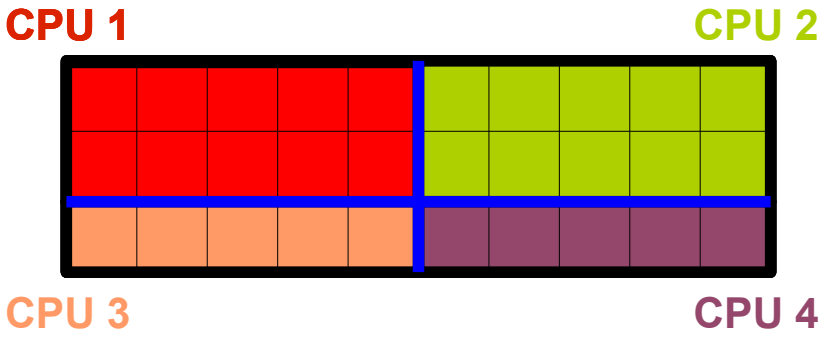
The problem with 4 domains is mathematically different from the problem with 2 domains (e.g., more duplicated unknowns).

Speed-up calculations

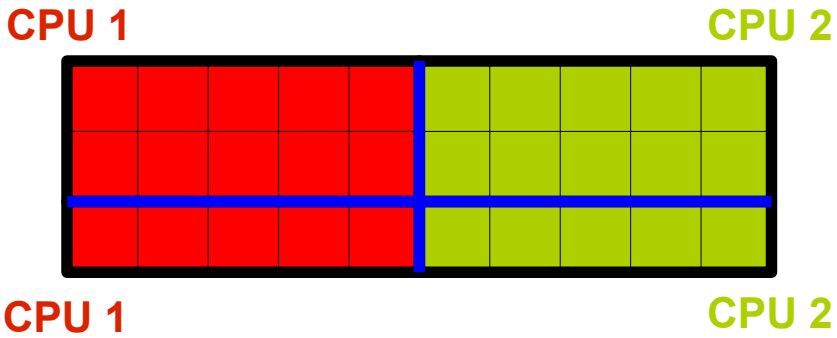
2 domains on 2 CPUs



4 domains on 4 CPUs



4 domains on 2 CPUs



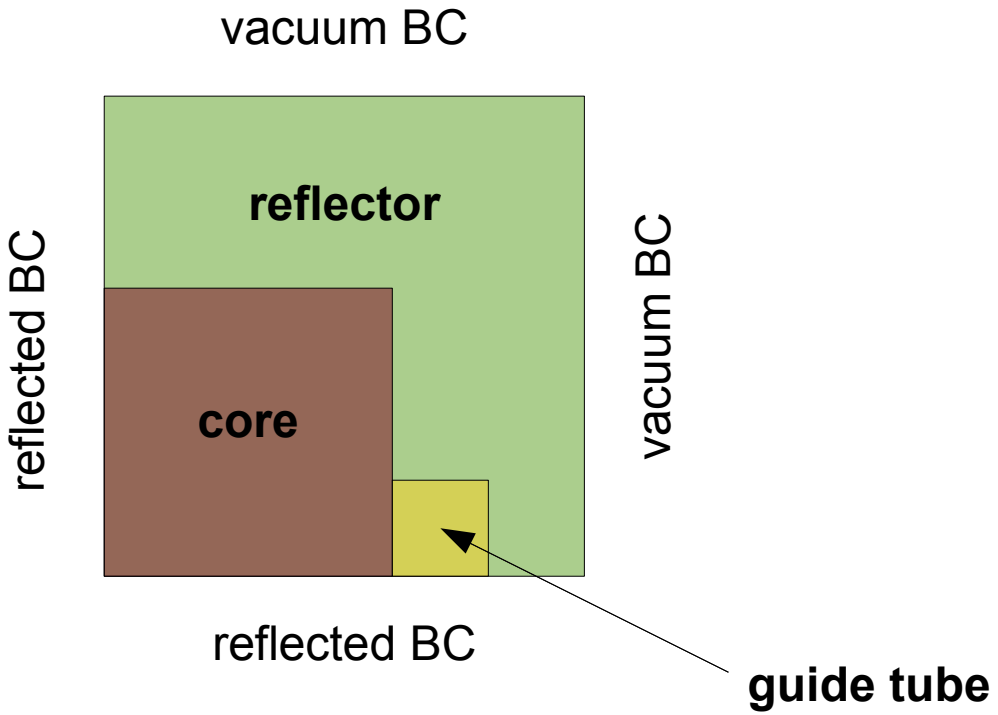
proper speed-up

This is only possible when enabling one processor to handle more than one domain (= main originality of this work).

The 3-D Takeda 1 benchmark

Cube representing a quarter reactor core.

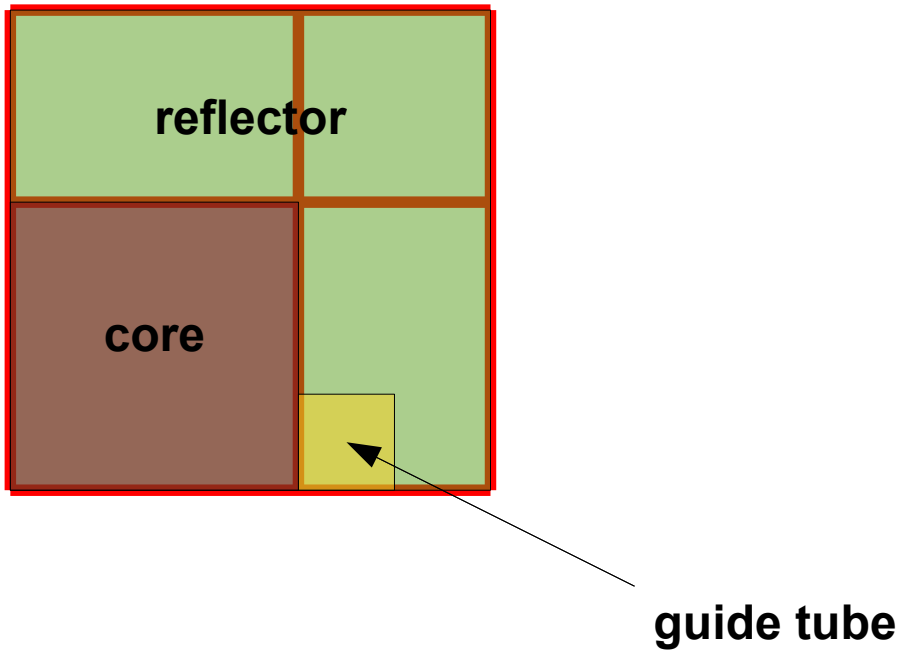
Simplified 2-D view:



Possible decompositions

2-by-2 domains

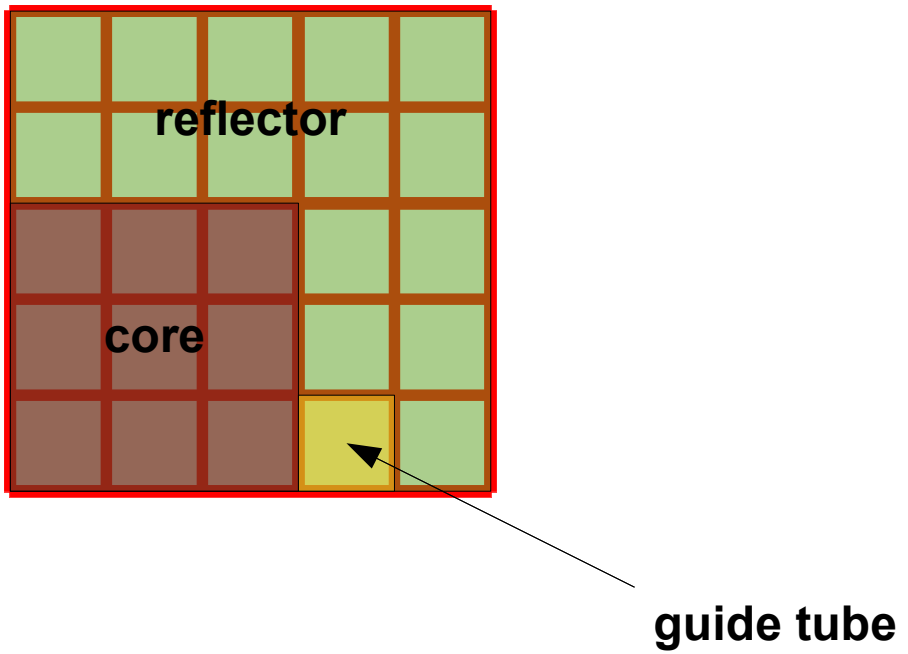
- different sizes
- inhomogeneous



Possible decompositions

5-by-5 domains

- same sizes
- homogeneous



Numerical results (3-D)

number of CPU = number of domains	Wall-clock time (s.)
2 x 2 x 2 = 8	2713
5 x 5 x 5 = 125	24

Factor 16

Factor 113

No proper speed-up !

Speed-up evaluations (5x5x5 decomposition)

Number of CPUs	Time (s.)	Speed-up	Efficiency
1	1222	1	1
5	267	4.58	.92
25	65	18.8	.75
50	39	31.3	.63
75	35	34.9	.47
125	24	50.9	.41

Using a better-quality decomposition yields faster solution with less CPUs.
Achievable only when allowing 1 CPU to handle more than 1 domain.

Conclusion from numerical tests

Enabling one CPU to handle more than one domain

- yields proper evaluations of the speed-up,
- allows the use of a better-quality decomposition (better load balancing) without requiring a higher number of processors.

Solver status

These results were obtained coupling:

- a stand-alone Finite Element - Spherical Harmonics (Pn) solver.
- a domain decomposition platform developed at the French Petroleum Institute (F.Nataf, P.Have), and adapted to finite element handling.

Also,

- written in object-oriented C++ and MPI (Message Passing Interface)
- run on FZK campusgrid.

This is long-time research.

Questions ?