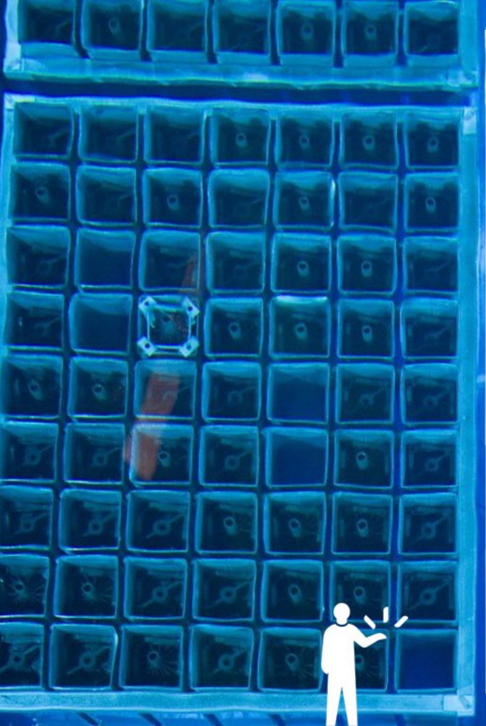


MICADO : Parallel Implementation of a 2D-1D Iterative Algorithm for the 3D Neutron Transport Problem in Prismatic Geometries

M&C 2013
6 may 2013

François Févotte and Bruno Lathuilière

EDF R&D - Sinetics, Clamart

CHANGER L'ENERGIE ENSEMBLE

Context

EDF develops a new simulation tool for the PWR based on SPn equations : COCAGNE.
For validation purposes EDF needs 3D Transport solver :

- DOMINO for cartesian mesh ;
- MICADO for unstructured and prismatic geometries ;

MICADO is a prototype to evaluate the properties of 2D-1D iterative algorithm[1] to treat a full 3D core without spatial homogenization.

[1] G. Lee and N. Cho. “2D/1D fusion method solutions of the three-dimensional transport OECD benchmark problem C5G7 MOX.” *Progress in Nuclear Energy*, **vol. 48(5)**, pp. 410 – 423 (2006)

Table of contents

1. 2D-1D Iterative algorithm

The transport equation

$$\varepsilon \frac{\partial \psi}{\partial x}(x,y,z) + \eta \frac{\partial \psi}{\partial y}(x,y,z) + \mu \frac{\partial \psi}{\partial z}(x,y,z) + \Sigma \psi(x,y,z) = Q(x,y,z),$$

where

- ψ is the angular flux ;
- ε, η, μ are the three components of direction $\vec{\Omega}$;
- Σ is the total macroscopic cross-section ;
- Q is the source term including fission and scattering.

Alternating direction

$$\varepsilon \frac{\partial \psi}{\partial x} \qquad + \eta \frac{\partial \psi}{\partial y} \qquad + \mu \frac{\partial \psi}{\partial z} \qquad + \Sigma \psi \qquad = Q$$

Alternating direction

$$\begin{cases} \varepsilon\frac{\partial\psi_R}{\partial x} & +\,\eta\frac{\partial\psi_R}{\partial y} & +\,\mu\frac{\partial\psi_R}{\partial z} & +\,\Sigma\psi_R & = Q \\ \varepsilon\frac{\partial\psi_Z}{\partial x} & +\,\eta\frac{\partial\psi_Z}{\partial y} & +\,\mu\frac{\partial\psi_Z}{\partial z} & +\,\Sigma\psi_Z & = Q \\ & & & \psi_Z & = \psi_R \end{cases}$$

Alternating direction

$$\begin{cases} \varepsilon\frac{\partial\psi_R}{\partial x} + \eta\frac{\partial\psi_R}{\partial y} + \mu\frac{\partial\psi_R}{\partial z} + \Sigma\psi_R = Q - \mu\frac{\partial\psi_Z}{\partial z} \\ \varepsilon\frac{\partial\psi_Z}{\partial x} + \eta\frac{\partial\psi_Z}{\partial y} + \mu\frac{\partial\psi_Z}{\partial z} + \Sigma\psi_Z = Q \\ \psi_Z = \psi_R \end{cases}$$

Alternating direction

$$
\begin{cases}
\varepsilon \frac{\partial \psi_R}{\partial x} + \eta \frac{\partial \psi_R}{\partial y} + \cancel{\mu \frac{\partial \psi_R}{\partial z}} + \Sigma \psi_R = Q - \mu \frac{\partial \psi_Z}{\partial z} \\
\cancel{\varepsilon \frac{\partial \psi_Z}{\partial x}} + \cancel{\eta \frac{\partial \psi_Z}{\partial y}} + \mu \frac{\partial \psi_Z}{\partial z} + \Sigma \psi_Z = Q - \varepsilon \frac{\partial \psi_R}{\partial x} - \eta \frac{\partial \psi_R}{\partial y} \\
\psi_Z = \psi_R
\end{cases}
$$

Alternating direction

$$
\begin{cases}
\varepsilon \frac{\partial \psi_R}{\partial x} + \eta \frac{\partial \psi_R}{\partial y} + \cancel{\mu \frac{\partial \psi_R}{\partial z}} + \Sigma \psi_R = Q - \mu \frac{\partial \psi_Z}{\partial z} \\
\cancel{\varepsilon \frac{\partial \psi_Z}{\partial x}} + \cancel{\eta \frac{\partial \psi_Z}{\partial y}} + \mu \frac{\partial \psi_Z}{\partial z} + \Sigma \psi_Z = Q - \varepsilon \frac{\partial \psi_R}{\partial x} - \eta \frac{\partial \psi_R}{\partial y} \\
\psi_Z = \psi_R
\end{cases}
$$

Alternating direction

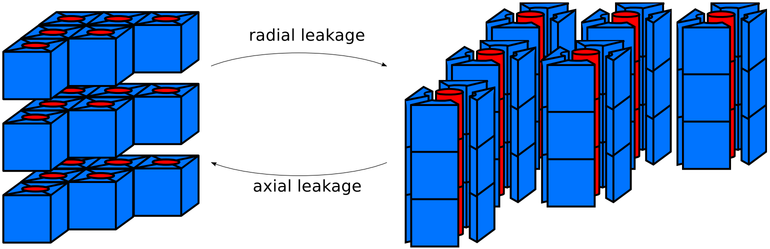
$$
\begin{cases}
\varepsilon \frac{\partial \psi_{R,n+1}}{\partial x} + \eta \frac{\partial \psi_{R,n+1}}{\partial y} + \cancel{\mu \frac{\partial \psi_R}{\partial z}} + \Sigma \psi_{R,n+1} = Q - \mu \frac{\partial \psi_{Z,n}}{\partial z} \\
\cancel{\varepsilon \frac{\partial \psi_Z}{\partial x}} + \cancel{\eta \frac{\partial \psi_Z}{\partial y}} + \mu \frac{\partial \psi_{Z,n+1}}{\partial z} + \Sigma \psi_{Z,n+1} = Q - \varepsilon \frac{\partial \psi_{R,n+1}}{\partial x} - \eta \frac{\partial \psi_{R,n+1}}{\partial y} \\
\psi_Z = \psi_R
\end{cases}
$$

Alternating direction

$$\begin{cases} \varepsilon \frac{\partial \psi_{R,n+1}}{\partial x} + \eta \frac{\partial \psi_{R,n+1}}{\partial y} + \cancel{\mu \frac{\partial \psi_R}{\partial z}} + \Sigma \psi_{R,n+1} = Q - \mu \frac{\partial \psi_{Z,n}}{\partial z} \\ \cancel{\varepsilon \frac{\partial \psi_Z}{\partial x}} + \cancel{\eta \frac{\partial \psi_Z}{\partial y}} + \mu \frac{\partial \psi_{Z,n+1}}{\partial z} + \Sigma \psi_{Z,n+1} = Q - \varepsilon \frac{\partial \psi_{R,n+1}}{\partial x} - \eta \frac{\partial \psi_{R,n+1}}{\partial y} \\ \psi_Z = \psi_R \end{cases}$$

Solve the 2D slices

Solve the 1D columns



- *Ni* : number of 2D regions
- *Nj* : number of slices

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R}\right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z}\right) dz$$

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z}\right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y}\right) dxdy$$

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

forall $j \in [\![1, N_j]\!]$ **do**

forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R}\right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z}\right) dz$$

Standard 2D MOC Solver

forall $i \in [\![1, N_i]\!]$ **do**

forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z}\right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y}\right) dxdy$$

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R}\right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z}\right) dz$$

Standard 2D MOC Solver

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z}\right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y}\right) dxdy$$

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R} \right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z} \right) dz$$

Standard 2D MOC Solver

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z} \right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y} \right) dxdy$$

MOC 1D Solver

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R} \right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z} \right) dz$$

Standard 2D MOC Solver

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z} \right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y} \right) dxdy$$

MOC 1D Solver

Alternating direction algorithm (2/2)

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma \psi_{k,R} \right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z} \right) dz$$

Standard 2D MOC Solver

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma \psi_{k,Z} \right) dx dy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y} \right) dx dy$$

MOC 1D Solver

Alternating direction algorithm (2/2)

initialisation

while $\psi_R \neq \psi_Z$ **do**

  forall $j \in [\![1, N_j]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{z_j}^{z_{j+1}} \left(\varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} + \eta_k \frac{\partial \psi_{k,R}}{\partial y} + \Sigma\psi_{k,R}\right) dz = \int_{z_j}^{z_{j+1}} \left(Q - \mu_k \frac{\partial \psi_{k,Z}}{\partial z}\right) dz$$

Standard 2D MOC Solver

  forall $i \in [\![1, N_i]\!]$ **do**

    forall $\vec{\Omega}_k \in S_N$ **do**

$$\int_{\mathcal{C}_i} \left(\mu \frac{\partial \psi_{k,Z}}{\partial z} + \Sigma\psi_{k,Z}\right) dxdy = \int_{\mathcal{C}_i} \left(Q - \varepsilon_k \frac{\partial \psi_{k,R}}{\partial x} - \eta_k \frac{\partial \psi_{k,R}}{\partial y}\right) dxdy$$

MOC 1D Solver

Advantages and drawbacks

Advantages

- Solve the transport equation without introduction of lower order model (such as diffusion) ;
- Do not have to deal with huge 3D tracking ;

Remaining questions

- Does the iterative algorithm converge?
- What is the order of spatial convergence?
- Can the parallelism be efficient?
- Can the storage of $\mu_k \frac{\partial \psi_{k,z}}{\partial z}$ be reduced?
- Can the iterative algorithm be accelerated?

Advantages and drawbacks

Advantages

- Solve the transport equation without introduction of lower order model (such as diffusion) ;
- Do not have to deal with huge 3D tracking ;

Remaining questions

- Does the iterative algorithm converge?
- What is the order of spatial convergence?
- Can the parallelism be efficient?
- Can the storage of $\mu_k \frac{\partial \psi_{k,z}}{\partial z}$ be reduced?
- Can the iterative algorithm be accelerated?

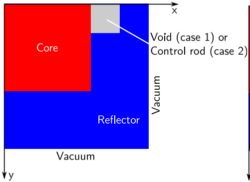
2. Numerical properties

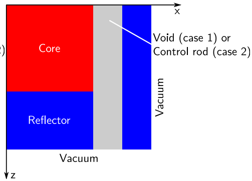
Theoretical results

- Under restrictive assumptions the convergence of the iterative algorithm can be proved ;
- The best spatial convergence order we can expect for the step characteristics and DD scheme is 1;
- If the iterative algorithm converge, it is to the transport solution.

Practical experimentation

Benchmark Takeda (case 2)

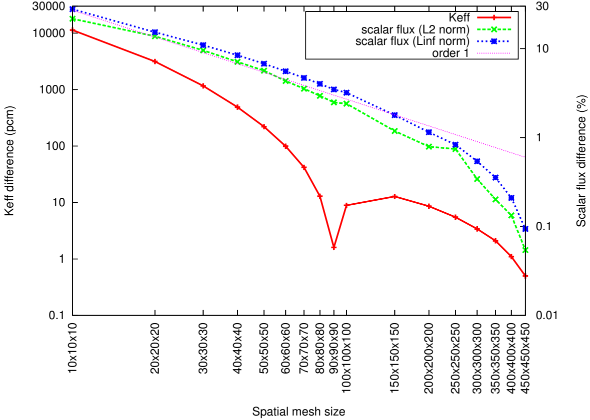




- angular discretization : Gauss Legendre 8×4
- 10 characteristics per side of square region
- stopping criterion $|\Delta k_{eff}| < 10^{-8}$

Spatial convergence

- reference mesh : $500 \times 500 \times 500$

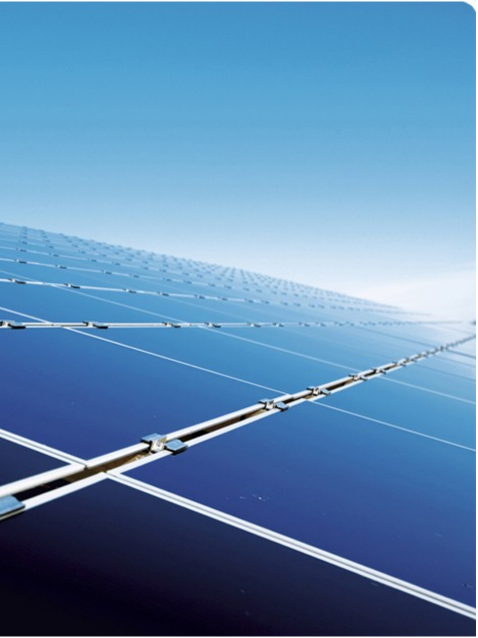


- empirically : the spatial order of convergence is 1

Comparison to the reference MCNP

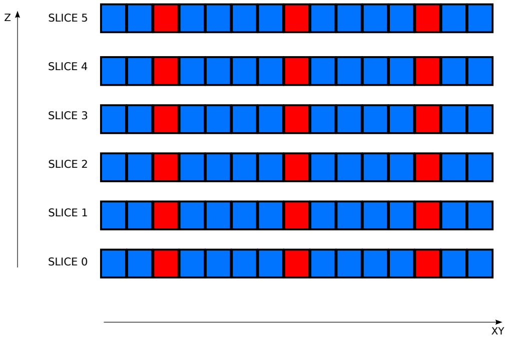
	Difference	MCNP Error bar
k_{eff}	1.3 pcm	60 pcm
flux core (grp 1 / grp 2)	-0.38 % / 0.12 %	0.13 % / 0.10 %
flux reflector (grp 1 / grp 2)	0.32 % / 0.29 %	0.23 % / 0.21 %
flux control rod (grp 1 / grp 2)	1.12 % / 0.04 %	0.72 % / 0.48 %

- Convergence to the transport solution

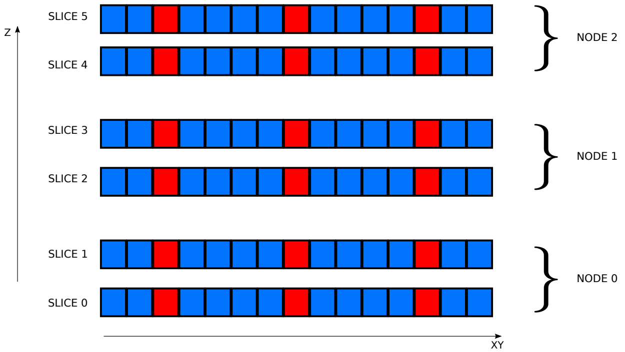


3. HPC properties

Parallel algorithm : data distribution



Parallel algorithm : data distribution

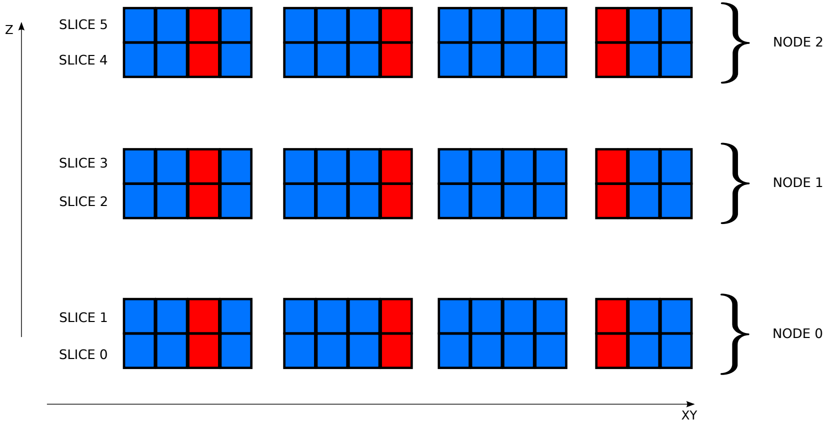


- Data distribution over slices

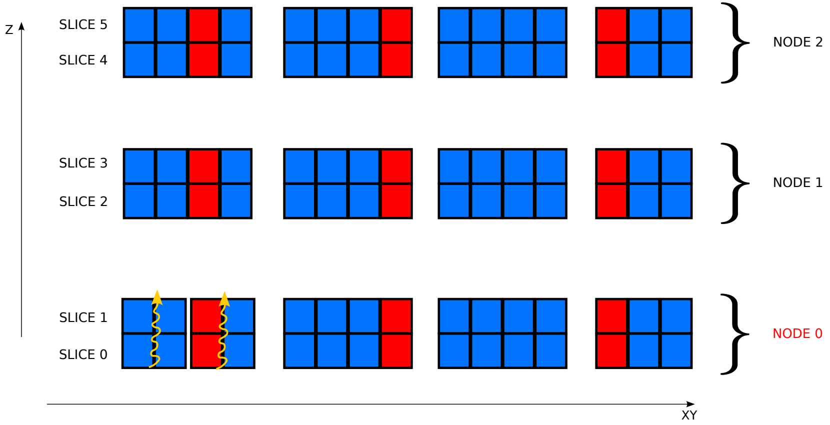
Parallelisation of solve2D

- Over Slices : MPI, TBB
- Over Angular direction : TBB

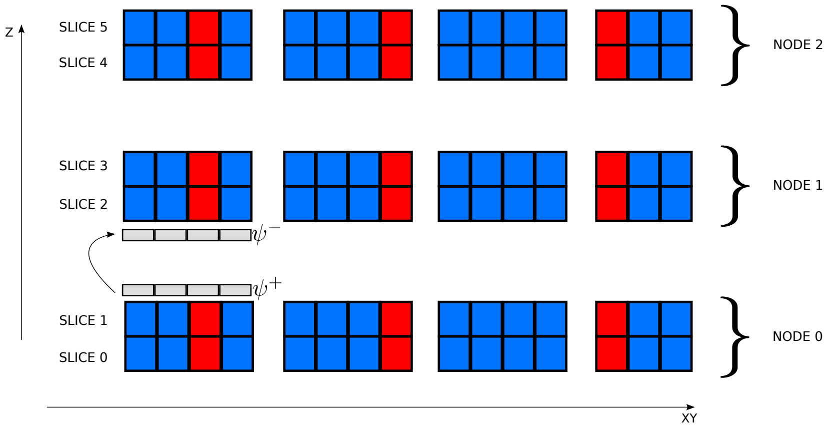
Solve1D : parallel sweep



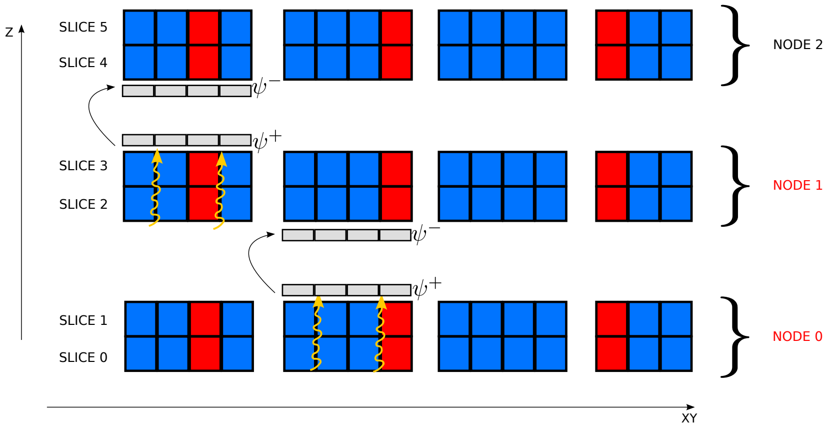
Solve1D : parallel sweep



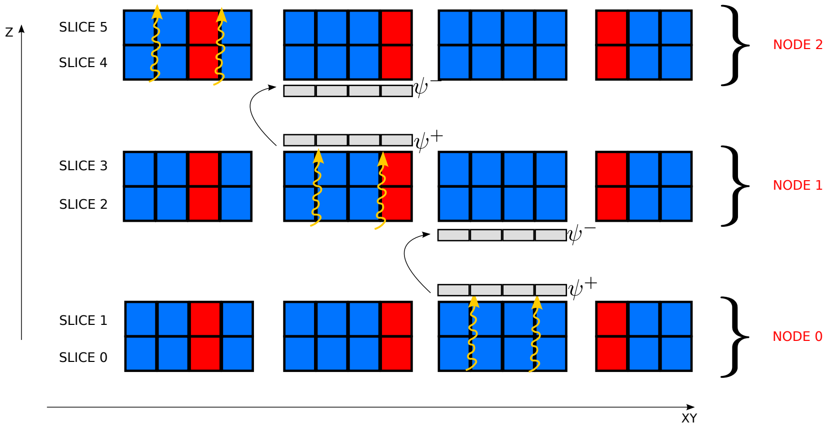
Solve1D : parallel sweep



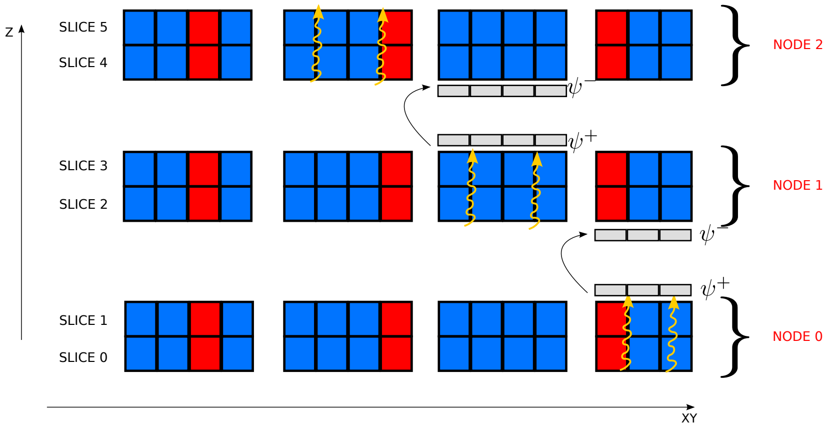
Solve1D : parallel sweep



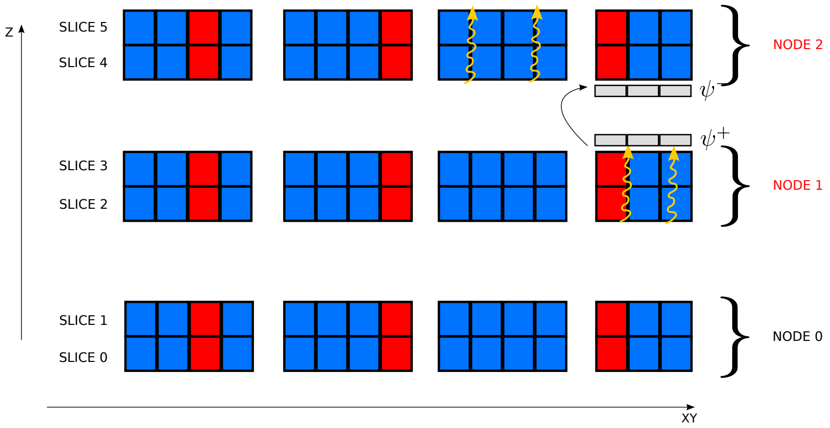
Solve1D : parallel sweep



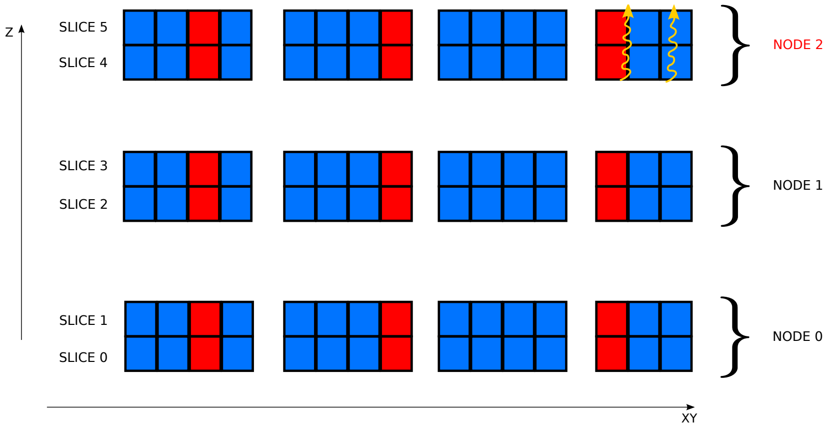
Solve1D : parallel sweep



Solve1D : parallel sweep



Solve1D : parallel sweep



- choice of mpi buffer size;
- feed the pipeline with direction with same sign of μ_k;

TAKEDA : performance measurement

Discretization

- Spatial scheme : step characteristics and diamond difference
- Spatial Discretization : 200 × 200 × 200
- Angular Discretization : Gauss Legendre 8 × 4
- Tracking : 10 characteristics per side of square region

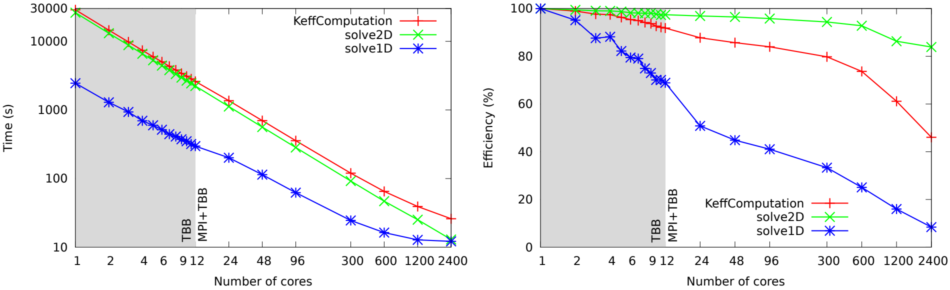
Iteration configuration

- Fixed Number of power iterations : 100
- Fixed Number of upscattering iterations : 1
- Fixed Number of free scattering iterations : 1

Cluster EDF : Ivanoe

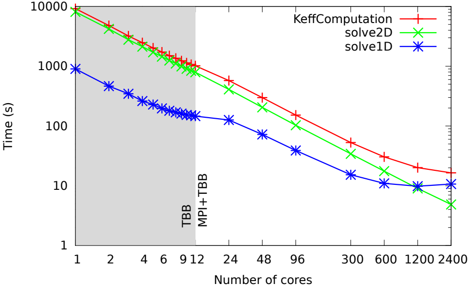
- Standard node properties : 2 processors Intel Xeon X5670 Westmere (12 cores per node)

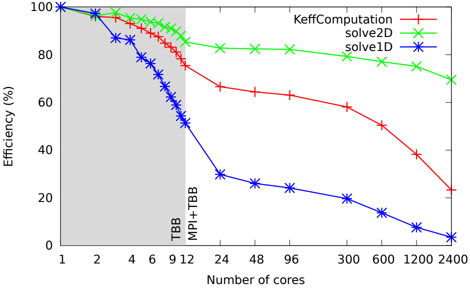
Parallel efficiency with Step Characteristic Scheme



- Solve2D is more expensive than Solve1D (factor 10.6) ;
- Parallel efficiency of Solve2D is always good (over 80%) ;
- Except for 2400 cores the global efficiency is good.

Parallel efficiency with Diamond Difference Scheme





- 3 times faster than step characteristic scheme ;
- Solve2D is still more expensive than Solve1D (factor 9) ;
- Parallel efficiency of Solve2D decreases in shared-memory due to lower arithmetic intensity;
- Until 300 – 600 cores the global efficiency is good.

4. Conclusions and perspectives

Conclusions

MICADO

Implementation of a new 3D transport solver :

- validated against Takeda Benchmark ;
- scalable until 1000 cores ;

2D-1D algorithm

The first preliminary tests of the iterative algorithm show that :

- the order-1 of spatial convergence is empirically reached ;
- its parallelism can be exploited.

MICADO perspectives

- Parallelization optimization :
  - parallelism of direction in distributed memory ;
  - the use of SIMD instructions (SSE2/ AVX) ;
  - a new level of shared memory parallelism in Solve 2D (in the characteristics sweep) ;
- Memory footprint reduction :
  - implementation of symmetry boundary condition ;
  - optimization of the tracking storage ;
  - reduce the storage of the leakage term ;
- Iterative algorithm improvement :
  - study acceleration at each level ;
  - improve the stopping and convergence criteria.