Parallel 3D Neutron Transport Calculation
using Discontinuous Galerkin and Spherical Harmonics Methods

Kenneth Assogba[1,2], Lahbib Bourhrara[1] & Grégoire Allaire[2]

[1]Université Paris-Saclay, CEA, Service d'Études des Réacteurs et de Mathématiques Appliquées, 91191, Gif-sur-Yvette, France
[2]CMAP, École Polytechnique, Institut Polytechnique de Paris, 91128, Palaiseau, France

Abstract

Precise nuclear core and assembly calculation require solving the neutron transport equation. However, high-order discretization of this problem leads to linear systems with billions of unknowns. The target of this work is to take advantage of modern supercomputer architecture to provide fast and high-fidelity 3D neutron transport solutions. This goal is achieved using hybrid `MPI+OpenMP` programming model to build a parallel Krylov solver, here BiCGSTAB. The results obtained show that the method is able to handle complex 3D geometries to produce accurate solutions while achieving good scalability upon 10^3 cores.

Neutron Transport Problem

Determine the flux $u = (u^g(x,\omega))_{g=1,\cdots,G}$ and the associated eigenvalue λ solution of:

$$\omega \cdot \nabla u^g + \sigma^g u^g = \overset{\text{scattering}}{\boxed{H^g u}} + \frac{1}{\lambda} \underset{\text{fission}}{\boxed{F^g u}} + \overset{\text{source term}}{\boxed{q^g}} \quad \text{in } X = D \times \mathbf{S}^2$$

- with inflow boundary condition $u^g = f^g$ on $\Gamma_- = \partial D \times S^2_-$
- g is the energy group
- $x \in D$ spatial variable in 1, 2 or 3 dimensions and $\omega \in S^2$ is the angular variable

The variational formulation proposed in [2] is obtained using test function $v + \frac{1}{\sigma}(\omega \cdot \nabla v)$ and reads:

$$\text{finding } u \in W \text{ such that } a(u,v) = L(v), \quad \forall v \in W \tag{1}$$

with

$$a(u,v) = \int_X \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma uv \right) + \int_{\Gamma_+} uv(\omega \cdot n), \tag{2}$$

$$L(v) = \int_X \left(Hu + \frac{1}{\lambda} Fu + q \right) \left(v + \frac{1}{\sigma}(\omega \cdot \nabla v) \right) - \int_{\Gamma_-} fv(\omega \cdot n). \tag{3}$$

DG - $\mathrm{P_N}$ space-angular discretization

$\mathrm{P_N}$ method consists in performing a truncated development to order N of the angular flux on the real-valued spherical harmonics y_n^m

$$u(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega) \tag{4}$$

Then let D_h be a subdivision of D into distinct elements D_r. On each region, we assume the flux moment $u_n^m(x)$ is polynomial i.e. $u_n^m \in \mathcal{P}_k(D_r)$.

$$x \in D_r, \quad u(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{j=1}^{J} u_{n,j}^m \varphi_j(x) y_n^m(\omega) \tag{5}$$

The discrete problem consists in solving inside each region:

$$\text{find } u \in W_h \text{ such that } a(u,v) = L(v), \quad \forall v \in W_h \tag{6}$$

- $W_h = \mathrm{span}(\varphi_j y_n^m)$ is the local approximation space
- upwind: the incoming flux in a region is given either by f if we are on the domain boundary ∂D or by the outgoing flux of the adjacent region

Distributed Krylov subspace method

Moving to distributed memory requires three basic components: a mesh partitioner, distributed sparse matrix-vector multiplication (SpMV) and dot-products. Dot products are implemented with a reduce operation but represent the bottleneck of this approach due to global synchronization of all process.

Mesh partitioner Two partitioners are implemented: a simple based on mesh numbering and a geometric one based on space filling curves (SFC).

SpMV Each domain store the mesh elements in contact with others domains. At each product, incoming fluxes are exchanged using non-blocking `Isend` - `Irecv` instructions.

3D Assembly and Heterogeneous Core Calculculation

The numerical method described have been applied to a wide variety 3D reactor cores [1] and assembly calculation problems. Here, a reduced PWR system and the C5G7 3D core benchmark are presented. All calculations are performed on the `Rome` partition of the CEA cluster `Orcus`. Each `Rome` node is composed of two `AMD EPYC` 7352 2.3Ghz CPUs equipped with 24 cores each, and the calculations are done without hyperthreading.

# of nodes	1 - 16
# of cores	24 - 768
# of MPI process per node	2
# of OpenMP threads	24

281-groups Pressurized Water Reactor assembly

A reduced PWR 3D 5×5 assembly with 9 axial layers is computed here. A basic mesh partition based on mesh numbering is used. A self-shedding calculation is performed first to obtain cross-sections. P_6 angular and linear spatial discretization yield to 63 million of DoFs.

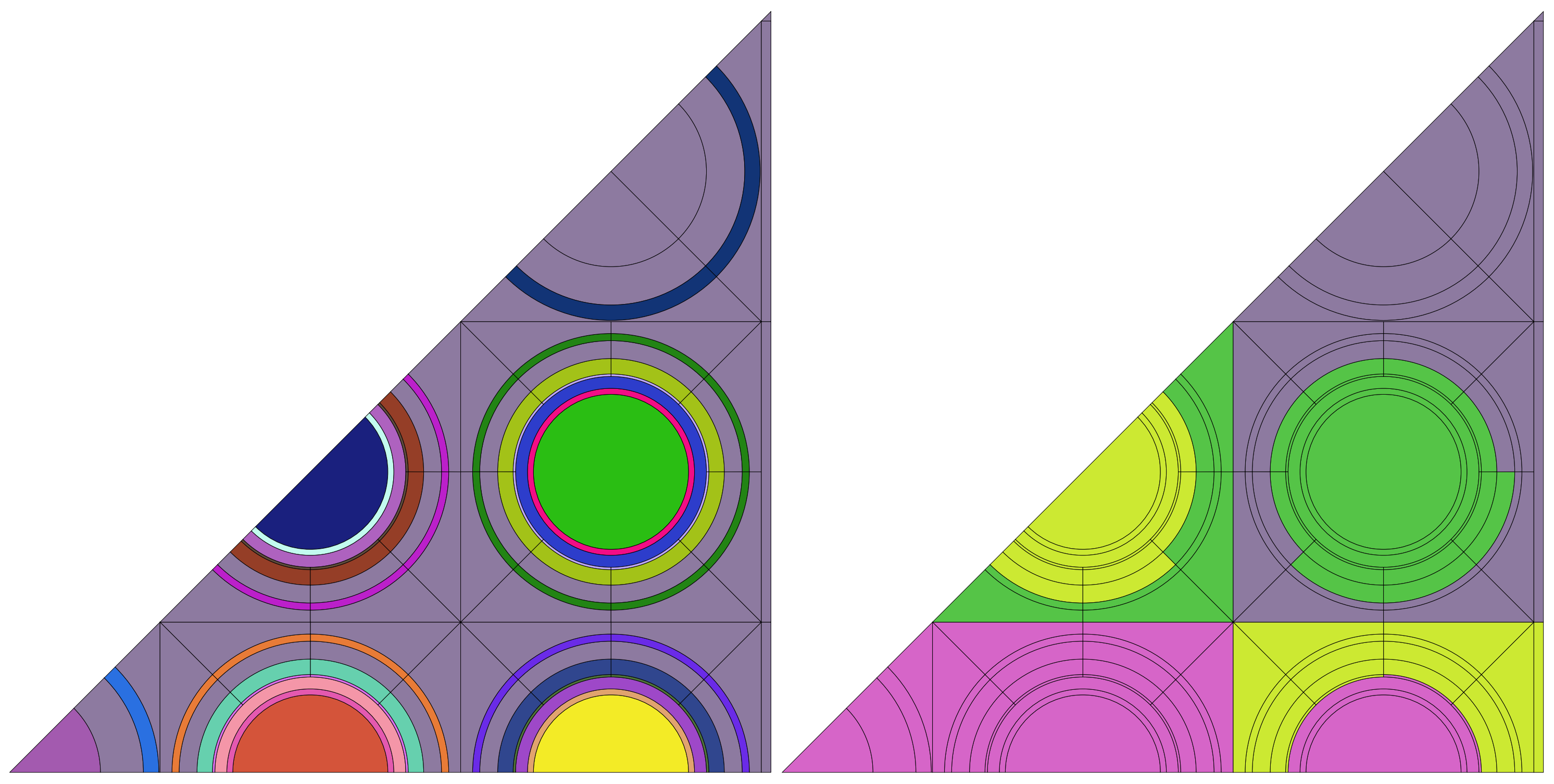

Figure 1: 5×5 PWR assembly (i) radial mesh and materials (ii) partition in 4 domain

Data		Results		Comparison w/ APOLLO3® `TDT` $\mathrm{DP_N}$	
Discretization	P6 & linear poly	Eigenvalue	0.909253	Δ_{keff}	-84 pcm
# of domains	4	Time(s)	1417	Speedup `NYMO`	2.5 × faster

3D C5G7 core strong scalability

The calculations previously done in shared memory [1] are now performed in distributed memory. In all calculations performed by varying the type of partitioning and the number of domains, the relative error with respect to the Monte-Carlo reference solution is less than 12 pcm. Compared to the shared memory solution, we reach a speedup of 18.7 on 768 cores reducing the calculation time from ½ hour to 100s.

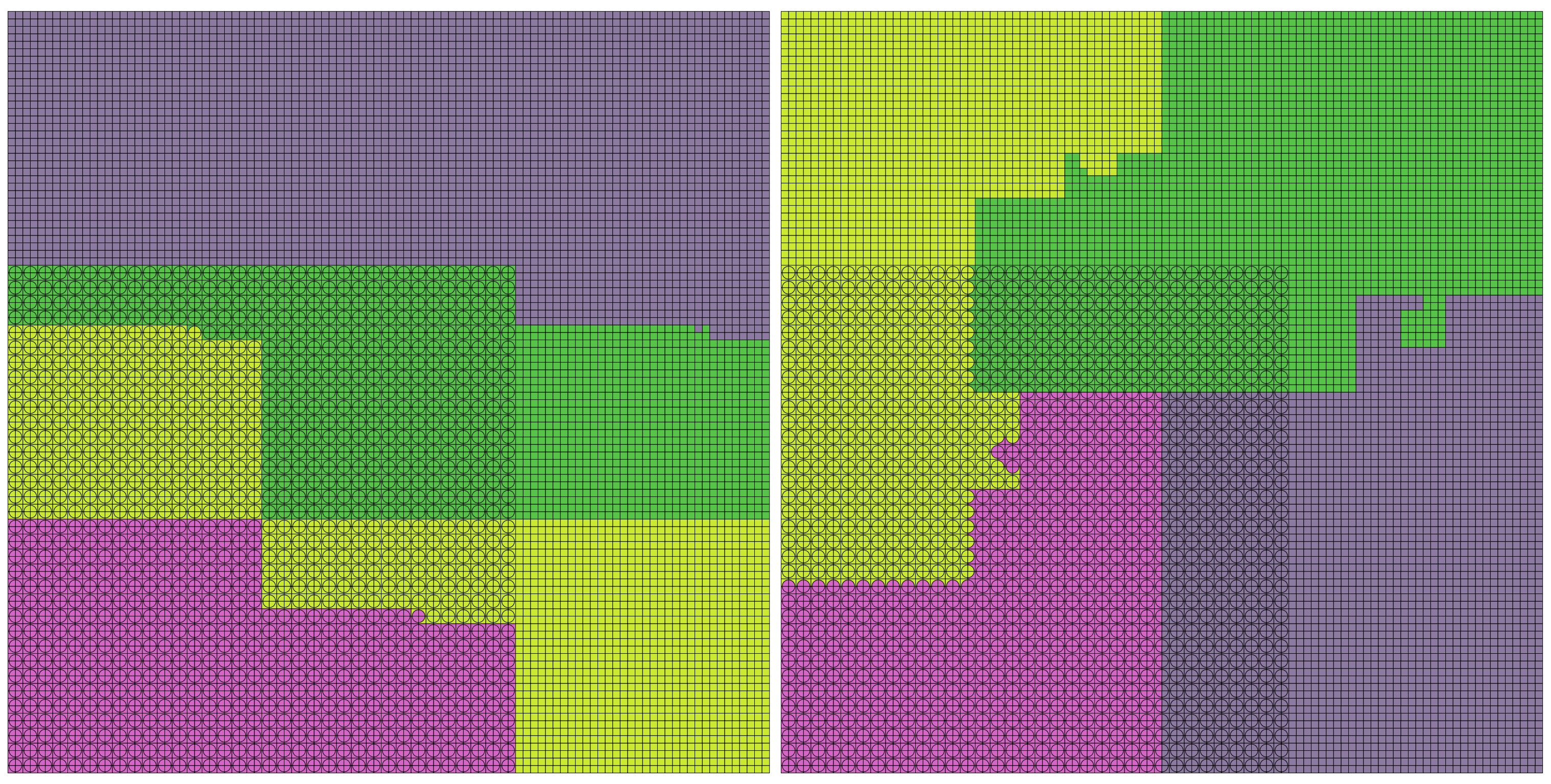

Figure 2: C5G7 core partioned in 4 domains (i) simple: based on mesh numbering (ii) geometric: based on SFC.

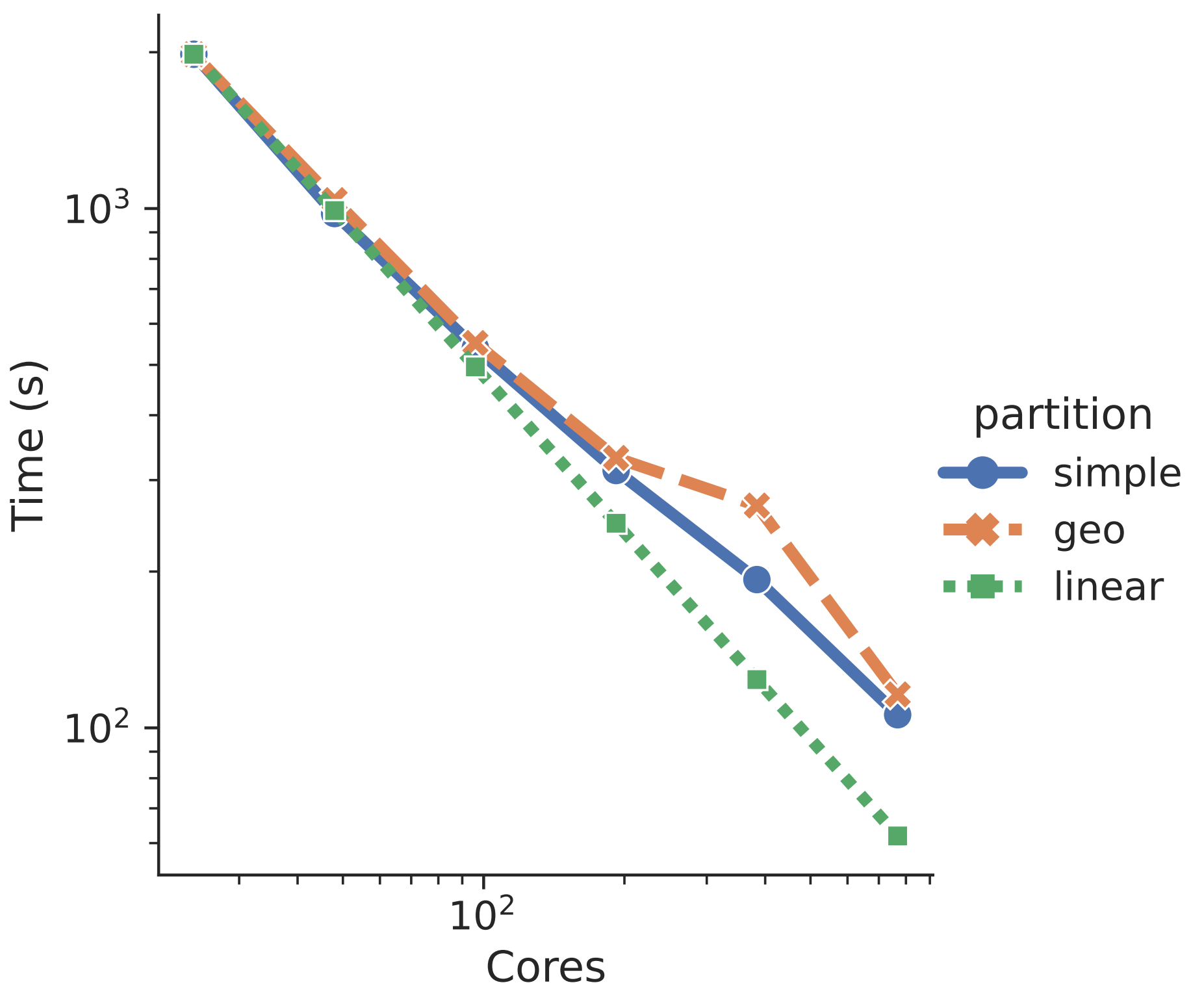

Figure 3: C5G7 strong scaling up to 10^3 cores with simple and geometric partitioning compared to linear speedup.

Conclusions

- Handle complex 2D / 3D meshes
- High-order accuracy and robust in parallel
- Good strong and weak scalability properties

Ongoing work

- DG - $\mathrm{P_N}$ scheme error estimate
- Parallel distributed one-step whole core calculation

References

[1] Kenneth Assogba, Lahbib Bourhrara, Igor Zmijarevic, and Grégoire Allaire. Precise 3D reactor core calculation using spherical harmonics and discontinuous galerkin finite element methods. *to appear in Proceeding of PHYSOR 2022, Pittsburgh, USA*, 2022.

[2] Lahbib Bourhrara. A new numerical method for solving the Boltzmann transport equation using the PN method and the discontinuous finite elements on unstructured and curved meshes. *Journal of Computational Physics*, 397, 2019.