A parallel multigrid solver on a structured triangulation of a hexagonal domain

DDXXI

June 29, 2012

Kab Seok Kang `kskang@ipp.mpg.de`

High Level Support Team (HLST)
Max-Planck-Institut für Plasmaphysik, EURATOM Association,
Boltzmannstraße 2, D-85748 Garching, Germany

High Level Support Team (HLST)

– Main tasks: a support unit to ensure optimal exploitation of HPC-FF and HELIOS computers,
– Support for code development:
  - Parallelization & optimization of codes for massively parallel computers,
  - Improvement of the parallel scalability of existing codes
  - Implementation of algorithms and mathematical library routines to improve the efficiency of codes
  - Visualization of large data sets
– Members: Core team (6 persons, IPP, Garching, Germany) + 4 staff (50%) in other European Fusion institutes.

OUTLINE

- – Model Problem
    - – Parallelization
- – Multigrid Method
    - – PETSc vs own code
- – Domain Decomposition method
    - – BDDC
    - – FETI-DP
- – Scaling properties
- – Conclusions

Model problem

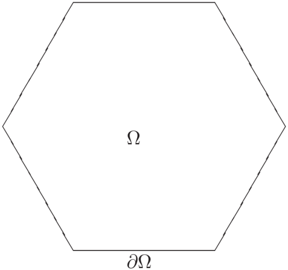
– Purpose: solve the 2nd order PDE in Plasma Physics simulation codes for Tokamak experiments

- Solution is sought at each time step $\rightarrow$ less than 0.1 sec

Tokamak experiments	ASDEX	JET	ITER	DEMO
DoF	2M	8M	32M	?

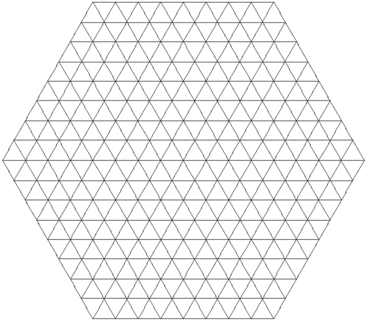
– The second order PDE (Helmholtz type) problem on a hexagonal domain with Dirichlet boundary condition

$$\begin{cases} (A - \nabla \cdot B\nabla)\, u = f, & \text{in } \Omega \\ u = 0, & \text{on } \partial\Omega \end{cases}$$

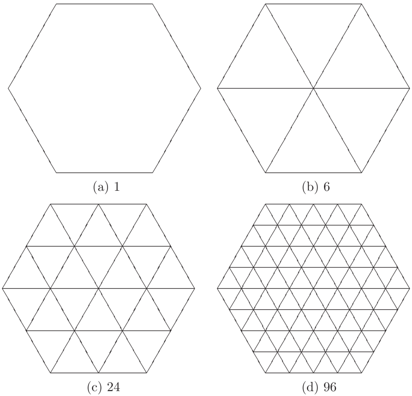


Discretization

– Linear Finite element method or Finite volume method
– Triangulation with regular triangles
– Boundary nodes: No degree of freedom. Ghost nodes.



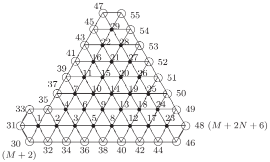
Parallelization



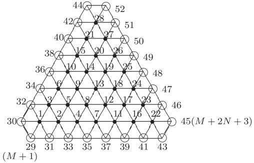
Sub-domains

- Divide a regular hexagonal domain with regular triangular sub-domains
- Limited number of cores: 1, 6, 24, 96, 384, ...
- Determine where the boundary nodes of the sub-domain are included.
- Use Fortran 90 and MPI library

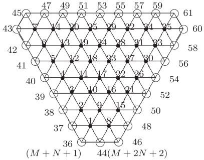
Parallelization



Type I: 0,6,9,12,15,18,21,24, ...



Type II: 1, 2, 3, 4, 5, 8, 11, ...



Type III: 7, 10, 13, 16, 19, 22, 25, ...

– Consisted by Real (•) and Ghost (◯) nodes.

– Classify three types of sub-domains.

– Need **five steps** for data communication for matrix-vector multiplication.

Multigrid Method

– Geometric multigrid method

– Well-known and well-analyzed fast solver

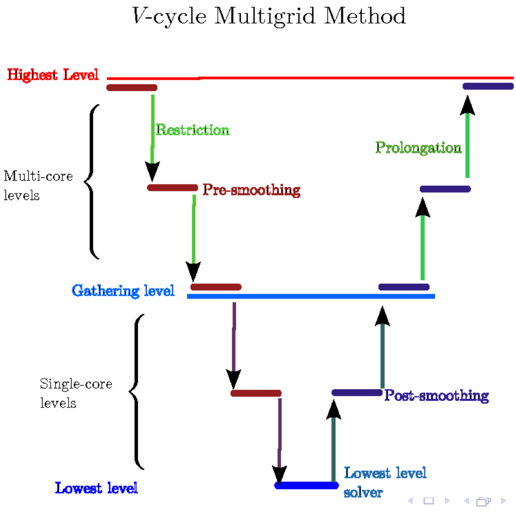
– The required number of iterations is fixed for many cases

– Lower levels: needs more data communication time in comparison to computing time

– Use V-cycle scheme as a solver and as a preconditioner

– Gather all data on each core for a certain lower level and solve by every core → Use `MPI_Allreduce`

Implementing Multigrid Method



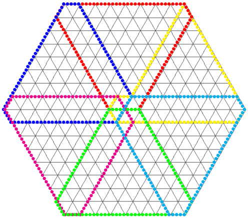
PETSc vs own code

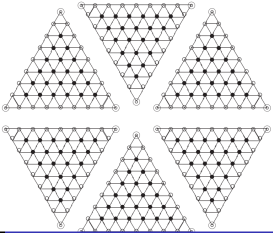
– PETSc: Well-developed scientific library to solve PDE based problems
  - good for beginners of parallel programming
  - large and slow (or need to be tuned)
  - hard to optimize with structured discretization property of some specific problems

– Own code: the code can be optimized according to the properties of the structured discretization
  - hard to implement $\rightarrow$ only expert get the benefit.

Domain Decomposition Method

– Divide into sub-domains and solve problems only on each sub-domain → naturally fit to distributed computers
– Overlapping (Schwarz) methods: minimally overlapping method (Block Jacobi) → slow
– Nonoverlapping methods: Enforce the conditions on the boundaries of the sub-domains
  - good for discontinuous or many part problems





Two-level DDM

The condition number does not depend on the number of sub-domains.

- Balanced domain decomposition with constraints (BDDC)
- FETI-DP

– Need to solve the coarse level problem

$\longrightarrow$ gather data on each core and solve on every core

– Solving local problem and coarse level problem on each core

$\longrightarrow$ Use Choleski factorization (LAPACK) for small number of DoF per core problem

$\longrightarrow$ Need more scalable solver for large number of DoF per core problem

$\longrightarrow$ Working now on multigrid method

BDDC: Preconditioner (Dohrmann (2003))

C: the constraints, enforced between subdomains, on **corners and/or edges**

$P_1 = R_I^T A_{II}^{-1} R_I$: Solve local Dirichlet BC problem

$P_2 = R^T D\Psi K_c^{-1} \Psi^T DR$ where $K_c = \sum \Psi^T A \Psi$ for $\begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \Psi \\ \Lambda \end{pmatrix} = \begin{pmatrix} 0 \\ R_c \end{pmatrix}$

$P_3 = R^T DQDR$ where $\begin{pmatrix} A & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} Qg \\ \mu \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$

BDDC1. Initial static condensation correction and residual update

$$u_0 = P_1 r, \quad r_1 = r - Au_0,$$

BDDC2. Coarse grid and substructure corrections and residual update

$$u_1 = P_2 r_1, \quad u_2 = P_3 r_1, \quad r_2 = r_1 - A(u_1 + u_2)$$

BDDC3. Final static condensation correction $u_3 = P_1 r_2$.

BDDC4. Preconditioned residual $Pr = u_0 + u_1 + u_2 + u_3$.

FETI-DP (Farhat et el (2001))

– Imposed the continuity on the inner boundary nodes (using Lagrange multipliers B) and corners u_c

$$\text{Solve} \begin{pmatrix} A_{rr} & A_{cr}^T & B^T \\ A_{cr} & A_{cc} & 0 \\ B & 0 & 0 \end{pmatrix} \begin{pmatrix} u_r \\ u_c \\ \lambda \end{pmatrix} = \begin{pmatrix} f_r \\ f_c \\ 0 \end{pmatrix}$$

Solve $F\lambda = d$ where

$$\begin{aligned} F &= BA_{rr}^{-1}B^T + BA_{rr}^{-1}A_{cr}^T S_{cc}^{-1} A_{cr} A_{rr}^{-1} B^T = B\tilde{A}^{-1}B^T \\ d &= BA_{rr}^{-1} f_r + BA_{rr}^{-1} A_{cr}^T S_{cc}^{-1}(f_c - A_{cr}A_{rr}^{-1} f_r) = B\tilde{A}^{-1}\tilde{f} \\ S_{cc} &= A_{cc} - A_{cr}A_{rr}^{-1}A_{cr}^T \qquad \text{(Coarser level)} \end{aligned}$$

– Conjugate gradient method with Dirichlet preconditioner

$$\begin{aligned} u_c &= S_{cc}^{-1}(f_c - A_{cr}A_{rr}^{-1} f_r + A_{cr}A_{rr}^{-1} B^T\lambda) \quad \text{(global)} \\ u_r &= A_{rr}^{-1}(f_r - A_{cr}^T u_c - B^T\lambda) \quad \text{(local)} \end{aligned}$$

BDDC and FETI-DP

K_c and S_{cc}: Globally defined coarse level problem

$\rightarrow$ K_c has large number of DoF than S_{cc},

– Four times if corners and edges used for the constraints

– The required mumber of iterations of FETI-DP and BDDC

($\leq 10^{-8}$)

Ratio = h/H

h: finer level mesh size, H: lower level mesh size

cores	24		96		384		1536	
Ratio	FDP	BD	FDP	BD	FDP	BD	FDP	BD
1/8	12	7	15	9	16	8	17	8
1/16	14	8	18	9	19	10	20	10
1/32	17	10	22	11	23	11	24	11
1/64	20	11	24	13	26	13	27	13

HELIOS

– IFERC: The International Fusion Energy Centre is located at Rokkasho, Japan
  - EU(F4E)–Japan Broader Approach collaboration

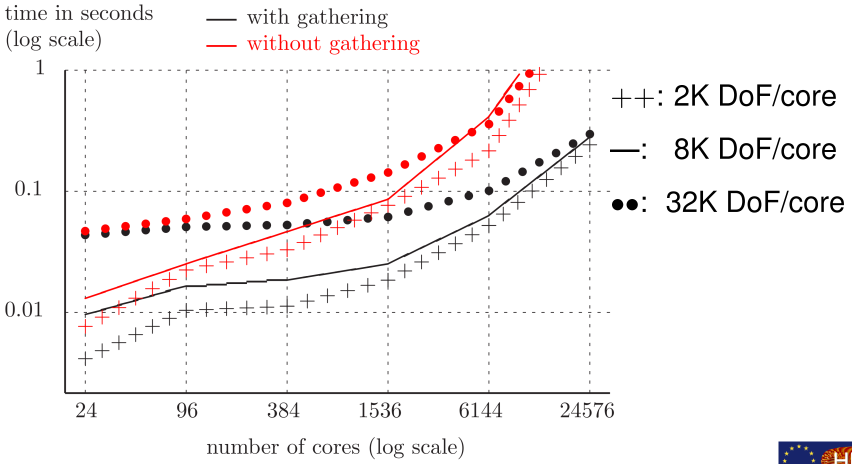
– The Computaional Simulation Centre (CSC): To exploit large-scale and high performance fusion simulations

– HELIOS: 4410 Bullx B510 Blades, 70,000 cores
  - 1.3 Pflops peak performance (No. 12 in Top500)
  - Two Intel Sandy-Bridge EP 2.7 GHz per node
  - Interconnection: Infiniband
  - Duration of project: April 1, 2012 – Dec. 31, 2016

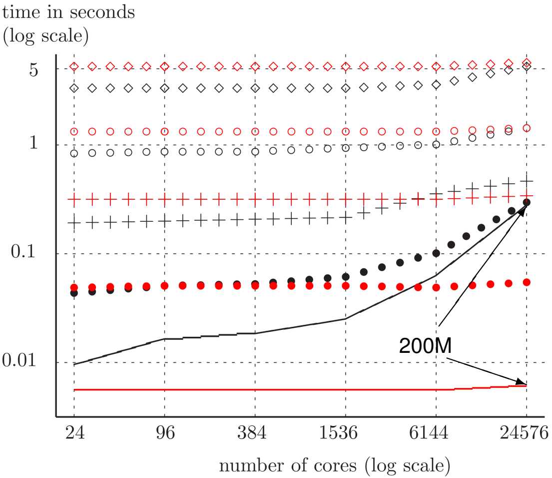
Highlights

- Multigrid method with gathering data is faster
- The matrix-vector multiplication: almost perfect weak-scaling
- For large DoF of local problem: Multigrid method has a very good (semi-)weak scaling property
- For small DoF per core problem, FETI-DP is the fastest
- Need scalable solver for FETI-DP and BDDC for large local (A_{II} or A_{rr}) and coarse (K_c or S_{cc}) problems

Multigrid with and without gathering



Multigrid as a solver for large problems



—: 8K DoF/core

●●: 32K DoF/core

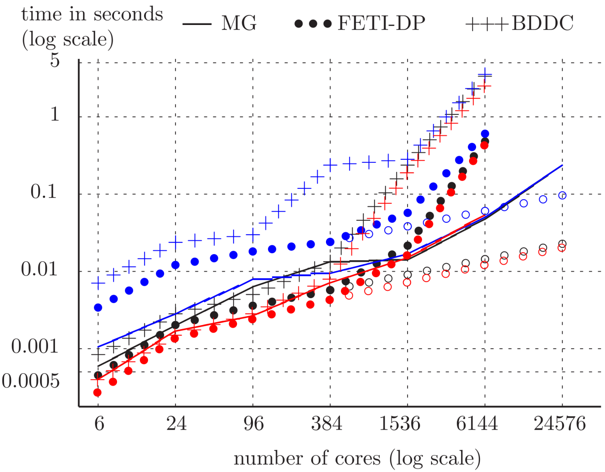
++: 128K DoF/core

○○: 512K DoF/core

◇◇: 2M DoF/core

Red: 100× Matrix-vector multiplication

Multigrid, FETI-DP, and BDDC



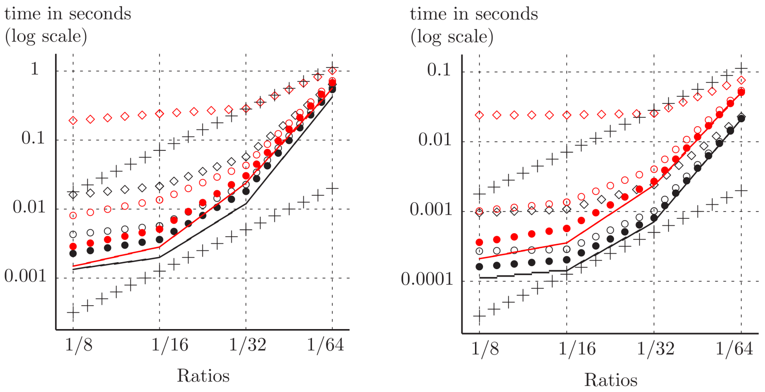
Red: 32 DoF/core

Black: 125 DoF/core

Blue: 500 DoF/core

○○: Expectation

FETI-DP and BDDC on fixed number of cores



Solution time

One iteration

Black: FETI-DP —: 24 cores ○○: 384 cores ++: references

Red: BDDC ●●: 96 cores ◇◇: 1536 cores

Conclusions

– For large DoF per core problems: Multigrid method has a very good (semi-)weak scaling property

– For small DoF per core problem: FETI-DP better than Multigrid and BDDC, but needs improvement of the scalability of the solver on the single core for large problems

→ Hybrid solver: Multigrid method with FETI-DP as a lowest solver, FETI-DP with multigrid method as local and global problem solver

Thanks

– Roman Hatzky (IPP, HLST Core team leader) and all core team members
– Bruce Scott (IPP, Project coordinator)
– David Tskhakaya (University of Innsbruck, Project coordinator)

Thank you for your attention!!