# Parallel Multilevel Iterative Linear Solvers for Large-Scale Computation

Kengo Nakajima

Department of Computational Earth Sciences, Research Organization for Information Science and Technology (RIST), Tokyo, Japan (e-mail: nakajima@tokyo.rist.or.jp, phone: +81-3-3436-5271, fax: +81-3436-5274).

## Abstract

**Preconditioned iterative solver is one of the most powerful choice such as IC (Incomplete Cholesky) or ILU (Incomplete LU) factorization method for large-scale scientific computation. But in these methods, iteration number until convergence increases as the problem size becomes larger. In multigrid solvers, the rate of convergence is independent of problem size and the number of iterations remains fairly constant. Multigrid is also a good preconditioner for Krylov iterative solvers. In this study, multigrid preconditioned conjugate gradient iterative method (MGCG) on parallel computers has been developed and applied to the Poisson equation in the region between dual sphere surfaces on semi-unstructured prismatic grids generated adaptively. Moreover this procedure has been also applied to the grids with local refinement.**

## Introduction

In many large-scale scientific simulation codes, almost computation is spent for linear solvers. Preconditioned Krylov iterative solvers such as ICCG (Incomplete Cholesky Factorized Conjugate Gradient)[1] provides robust convergence for wide range of scientific applications. IC and ILU (Incomplete LU Factorization) include global dependent operations but they can be localized on parallel computations[2] and smooth convergence has been obtained. Problem is that ICCG type solver is not *scalable* which means that iteration number increases as the problem size becomes larger. This is very critical if we solve $>10^9$ DOF linear equations on $>10^4$ processors.

Multigrid method[3] is a very well-known scalable method. In multigrid solvers, the rate of convergence is independent of problem size and the number of iterations remains fairly constant. Multigrid is also a good preconditioner for Krylov iterative solvers. Multigrid solvers and preconditioners have been widely used in finite-difference methods with structured grids since mid 80's but they are not popular in finite-element methods with unstructured grids. Recently, various types of multigrid methods for unstructured grids have been developed[4][5][6][7][8] both for parallel and serial computers.

In this study, multigrid preconditioned conjugate gradient iterative method (MGCG) on parallel computers has been developed and applied to the Poisson equation in the region between dual sphere surfaces on semi-unstructured prismatic grids generated adaptively based on the method in [8]. Moreover this procedure has been also applied to the grids with local refinement.

## Multigrid Method : Overview

Multigrid is an example of scalable linear solver. Relaxation methods like Gauss-Seidel relaxation are used there. Gauss-Seidel relaxation methods can efficiently damp high-frequency error but low-frequency error is left. The multigrid idea is to recognize that this low-frequency error

can be accurately and efficiently solved for on a coarser grid. Recursive application of this idea to each consecutive system of coarse-grid equations leads to a multigrid V-cycle[3] (Fig.1). If the components of the V-cycle are defined properly, the result is a method that uniformly damps all error frequencies with a computational cost that depends only linearly on the problem size. In other words, multigrid algorithms are scalable.

Multigrid algorithms tend to be problem specific and less robust than preconditioned Krylov iterative methods. Fortunately it is easy to combine the best features of multigrid and Krylov iterative methods into one algorithm, multigrid preconditioned Krylov iterative methods. The resulting algorithm is robust, efficient and scalable

One of the most important issue in multigrid is how to construct coarse grids. There are two basic multigrid approaches: geometric[7][8] and algebraic[6]. In geometric multigrid, the geometry of the problem is used to define the various multigrid components. In contrast, algebraic multigrid methods use only the information available in the linear system of equations.

The convergence rate of standard multigrid methods degenerates on problems that have anisotropic discrete operators which appear in very thin meshes near the wall for Navier-Stokes computation. Error becomes smooth in the direction where the connection is strong but not in the direction where the connection is weak. One popular approach in order to deal with anisotropic operators is to use *semi-coarsening* where the multigrid coarsening is not applied uniformly to all of the coordinate directions. By selecting NOT coarsening the grid in a certain direction, the anisotropy can be reduced on the coarser grid. This makes it easier for the smoother to eliminate other components of the high frequency error on the coarse grid.

## Current Approach

### Applications

In this study, target application is 3D incompressible thermal convection in the region between dual sphere surfaces. Semi-implicit pressure-correction scheme[9] is applied. In this method, momentum and energy equations are solved explicitly and pressure-correction Poisson equation is solved for incompressible constraint. This Poisson equation part is the most expensive process in the entire computation and convergence acceleration of this process is very critical issue for the total speed of the method. In this study, the Poisson equation solver part is mainly considered.

Semi-unstrucred prismatic grids generated from triangles on sphere surface are used. Meshes start from icosahedron and are globally refined recursively as in Fig.2. The grid hierarchy due to recursive refinement can be utilized for the coarse grid formation. In the current application, velocity components and temperature are defined at cell corners and pressure and potential for pressure correction are defined at cell center.

### Parallel MGCG solvers for Poisson Equations

Parallel MGCG (Multigrid preconditioned Conjugate Gradient) solver is developed where Gauss-Seidel relaxation is used as smoother. The feature of the method is as follows :

(1) V-cycle Multigrid Preconditioned MGCG solver, Gauss-Seidel smoothing.
(2) Semi-coarsening in lateral and normal-to-surface direction.
(3) Entire region is partitioned along radial (normal-to-surface) direction.
(4) Multilevel communication tables are defined at partition interfaces (Fig.3)
(5) Effect of local grid refinement can be considered.
(6) Written in Fortran 90 with MPI[10]

## Results

Fig.4 shows the CPU time on 2-16 Alpha Cluster connected through 100BT Ethernet. In these cases problem size per one PE is fixed as 320*2,048= 655,360 elements. Therefore $10^6$-$10^7$ size problems have been solved for 2-16 PEs. Solid line shows the results of MGCG cases. Computation time remains almost constant even for very large scale problems. This is because that iteration number is constant. On the contrary iteration number for ICCG solver increases along with PE number (= problem size).

## References


[1] J.J.Dongarra et. al. , 1994, *Templates for Iterative Methods*, SIAM.
[2] GeoFEM homepage http://geofem.tokyo.rist.or.jp/
[3] W.L.Briggs, 1987, *A Multigrid Tutorial*, SIAM.
[4] B.Smith et. al. , 1996, *Domain Decomposition*, Cambridge University Press.
[5] http://www.llnl.gov/CASC/linear_solvers/
[6] K.Stüben, 1999, GMD Report 53, GMD-Forschungstentrum Informationstechnik GmbH.
[7] M.F.Adams and J.W.Demmel, 1999, SC99 Proceedings, Portland, Oregon, USA.
[8] V.Parthasarathy, Y.Kallinderis and K.Nakajima, 1995, AIAA Paper 95-0670.
[9] Y.Kallinderis and K.Nakajima, 1994, AIAA Journal, Vol.32, No.8, pp.1617-1625.
[10] Gropp, W., Lusk, E. and Skjellum, A., 1994, *Using MPI*, MIT Press.

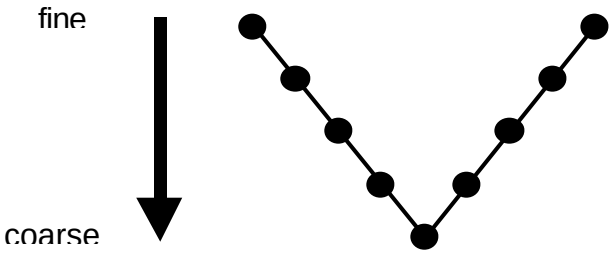
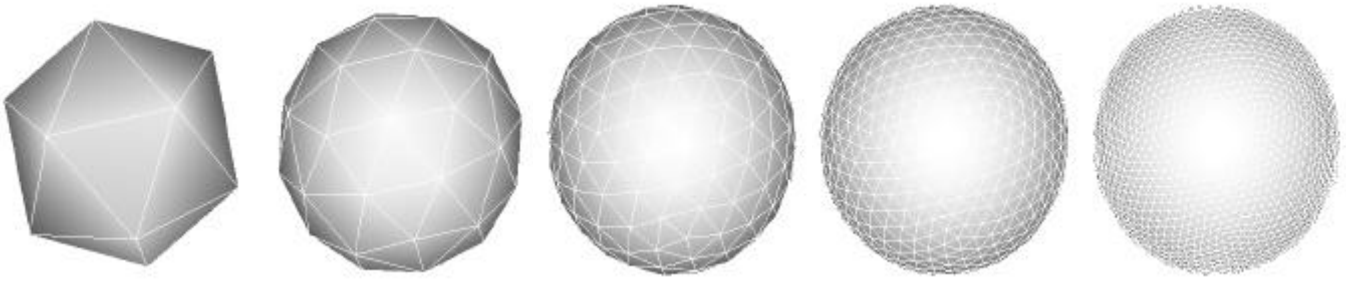




Fig.1 V-cycle for Multigrid Operation

*The down-cycle of a multigrid V-cycle uses smoothers to damp oscillatory error components at different grid scales. The up-cycle corrects the smooth error components remaining on each grid level by using the error approximations on coarser grids[4]*



**Level 0**
12 nodes
20 tri's

**Level 1**
42 nodes
80 tri's

**Level 2**
162 nodes
320 tri's

**Level 3**
642 nodes
1,280 tri's

**Level 4**
2,562 nodes
5.120 tri's

Fig.2 Surface triangle meshes generated from icosahedron

*4 children generated from 1 parent triangle*

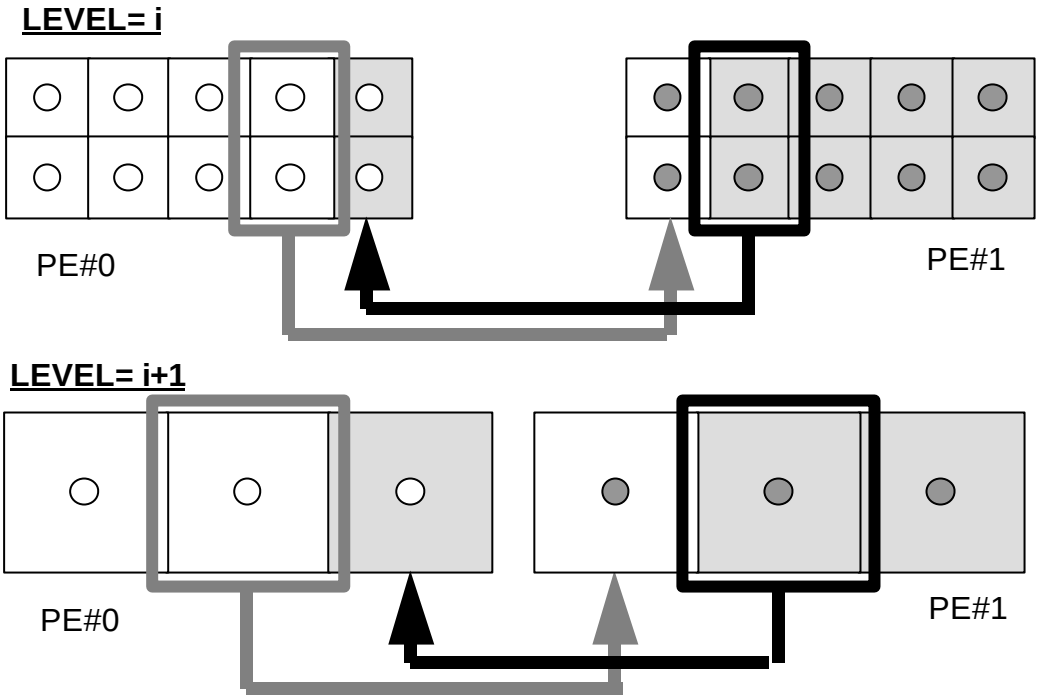



Fig.3 Multilevel Communication Table

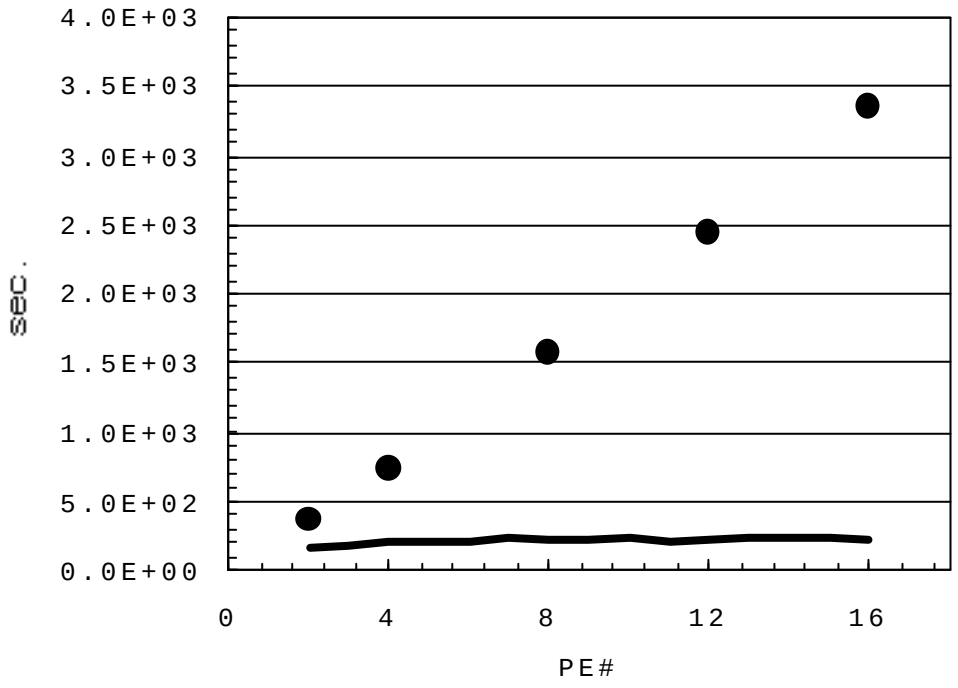



Fig.4 CPU time until Convergence : Parallel ICCG/MGCG Solvers
(solid line : MGCG, circle: ICCG)
*Problem size per PE is fixed as 320*2048= 655,360*
*10,485,760 elements for 16 PE case*
*CPU time for MGCG remains constant*