Development of Solid Earth Simulation Platform

Sparse Approximate Inverse Preconditioner for Contact Problems using OpenMP

Project Representative
Kengo Nakajima Department of Earth and Planetary Science, The University of Tokyo

Author
Kengo Nakajima Department of Earth and Planetary Science, The University of Tokyo

The three-level hybrid parallel programming model consisting of MPI, OpenMP and vectorization with multicolor-based reordering methods provides optimum performance on SMP cluster architectures with vector processors such as the Earth Simulator (ES) for finite-element type applications. While the three-level hybrid and flat MPI parallel programming models offer similar performance, the hybrid programming model outperforms flat MPI in the problems with large numbers of SMP nodes. In the cases with many colors, fewer numbers of iterations are required for convergence, but the performance is worse due to the smaller loop length and greater overhead. On ES, hybrid parallel programming model is much more sensitive to color number. In this study, SAI (Sparse Approximate Inverse) preconditioning method has been implemented to a single SMP node of ES using OpenMP. Developed method has been tested for applications with contact condition, and demonstrated efficiency and robustness for a wide range of problem size.

Keywords: GeoFEM, Parallel FEM, Parallel Iterative Solvers, Hybrid Parallel Programming Models, Sparse Approximate Inverse (SAI)

1. Preconditioned Iterative Solvers on the Earth Simulator with Multicoloring

In order to achieve minimal parallelization overhead in symmetric multiprocessor (SMP) cluster, multi-level *hybrid* programming model with MPI and OpenMP, and *flat MPI* approach have been widely employed [1]. Efficiency depends on hardware performance (CPU speed, communication bandwidth, memory bandwidth, and their balance), features of applications, and problem size [2].

In the previous work [1], author developed an efficient parallel iterative solver for finite-element applications on the Earth Simulator (ES) [3]. The method employs three-level hybrid parallel programming model consisting of MPI, OpenMP and vectorization. Multicolor-based reordering methods have been applied in order to achieve optimum performance, in the ILU/IC type method. Developed method attained 3.8 TFLOPS with 176 SMP nodes of ES, corresponding to more than 33% of the peak performance (11.3 TFLOPS).

While the three-level hybrid and flat MPI parallel programming models offer similar performance, the hybrid programming model outperforms flat MPI in the problems with large numbers of SMP nodes [1].

In the cases with many colors, fewer number of iterations are required for convergence, but the performance is worse due to the smaller loop length and greater overhead. On ES, hybrid parallel programming model is much more sensitive to color number. Fig. 1 shows the effect of colors on performance of preconditioned iterative solvers for contact problem in geophysics [1, 4]. *Selective-blocking*, a special preconditioner for contact applications, has been employed to ICCG iterative solvers with multicolor ordering [1].

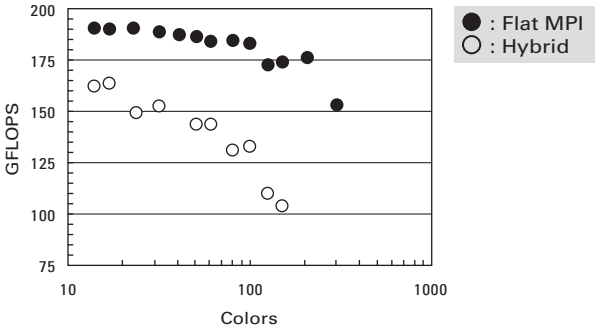


Fig. 1 Performance on 10 SMP nodes of ES (peak performance = 640 GFLOPS) using CG iterative method with *selective blocking* preconditioning [1] for the 3D elastic contact problem with MPC (linear multiple point constraint) condition ($\lambda = 10^6$) in Southwest Japan model, 23,301,006 DOF [1].

2. Sparse Approximate Inverse (SAI) Preconditioner

Sparse approximate inverse (SAI) is based on the idea of least square approximation [5, 6]. A particular class of SAI is constructed based on the Frobenius norm minimization. Since we want M to be a good approximation of A^{-1},

it is ideal if *AM~I*, which leads to minimize the functional *f (M) = min ||AM-I||*. This is equivalent to minimizing the individual functions:

$$\left\| Am_k - e_k \right\|_2, k = 1, 2, ..., n \quad (1)$$

where, mk corresponds to each column of *M*, and *n* is number of unknowns. Therefore, m_k can be computed independently. In finite-element applications, matrix *A* is sparse, therefore (1) can be reduced to a small least square problem with dense rectangular matrix:

$$\left\| \tilde{A}\tilde{m}_k - \tilde{e}_k \right\|_2, k = 1, 2, ..., n \quad (2)$$

One approach to solve this least square problem is QR factorization as [5, 6]. Sparsity of *M* is defined according to parameter for *dropping tolerance* [5, 6].

Good feature of SAI is that this is a global preconditioning method [5, 6]. ILU/IC type method requires special partitioning for contact problems in parallel computing [1, 4]. But, no special treatments except overlapping among domains are required for SAI, because it is based on the global information of the entire matrix [5, 6] Usually, iterations for convergence of parallel ILU/IC type method increases according to domain number [1, 4], but SAI provides constant number of iterations for different number of domains (Table 1). Another feature is that SAI preconditioning process includes only matrix-vector product, therefore multicoloring is not required for optimum performance on vector processors. SAI seems to be a promising method as robust and efficient preconditioner for large-scale problems on ES.

Table 1 Iterations/computation time for convergence on AMD Opteron 1.8GHz cluster by preconditioned iterative solvers for the 3D elastic fault-zone contact problem in [1] (83,664 DOF) (SB-BIC(0): Block IC(0) with the selective blocking, GPBiCG: Generalized Product-type on BiCG[7]).

method	domains (PE's)	partitioner	iterations	set-up+solve (sec.)
SB-BIC(0)+CG	1	–	114	18.6
	8	general	3498	56.9
	8	special[1]	166	2.8
SAI+GPBiCG	1	–	221	25.4
drop tolerance=	8	general	221	3.7
0.25	8	special[1]	221	3.7

3. SAI on the Earth Simulator

In this study, SAI preconditioned iterative solver has been developed on a single SMP node of ES (8 PE's, 64 GFLOPS peak) using OpenMP. Performance is evaluated for linear-elastic applications with contact on simple block model (Fig. 2) [1].

As described in the previous section, SAI preconditioning process includes only matrix-vector products. There are no processes with dependency, such as forward/backward substitution in ILU/IC method. Therefore, no multicoloring is required, and it is rather easier to extract optimum performance of vector processors. Each element has been reordered by DJDS (Descending-order Jagged Diagonal Storage) manner for vector performance according to the number of off-diagonal components, as shown in Fig. 3 [1]. Cyclic reordering has been finally applied for load-balancing among threads.

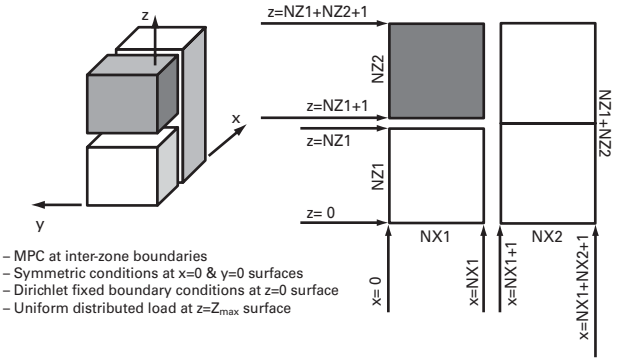


Fig. 2 Description of the Simple Block Model [1]

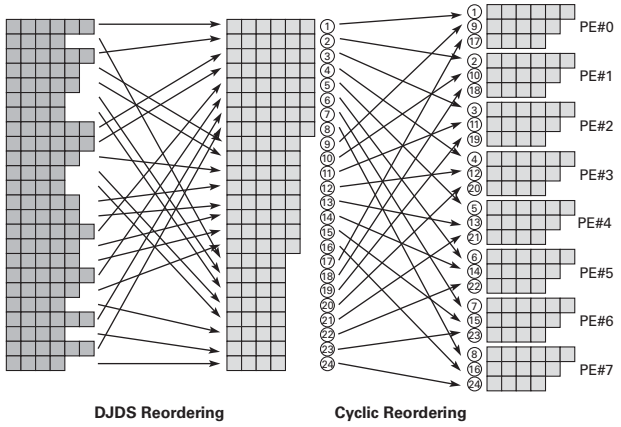
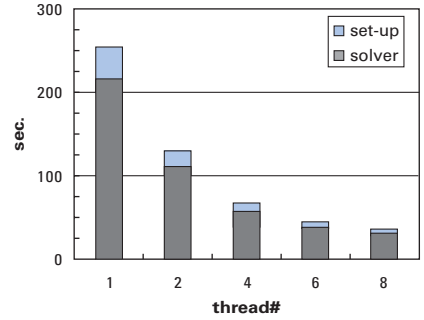


Fig. 3 DJDS and Cyclic Ordering for Vector/Parallel Performance on an SMP node by OpenMP.

Figure 4 (a) shows computation time for set-up (least-square minimization by QR factorization) and solver part of SAI/GPBiCG solver (Generalized Product-type on Bi-Conjugate Gradient) [7] for simple block model with 2,471,439 DOF using various configurations of thread number on a single SMP node of ES. Penalty number for contact constraint is with $\lambda = 10^6$ and dropping tolerance for SAI is 0.25. Figure 4 (b) shows speed-up effect according to thread number for set-up and solver part on both of small (0.35M DOF) and large (2.47 DOF) models. In both problem sets, scalability is very excellent. Especially, set-up part for least-squares provides almost perfect scalability, because solving equation (2) for each column by QR factorization can be done independently.

Figure 5 shows comparison with SB-BIC(0)/CG. Total computation time of SAI/GPBiCG is competitive with that of SB-BIC(0) CG. Performance of SAI/GPBiCG is not affected by problem size at all, and provides more than 35% of peak performance.

(a) Timing for Large Model



(b) Speed-up vs Thread#

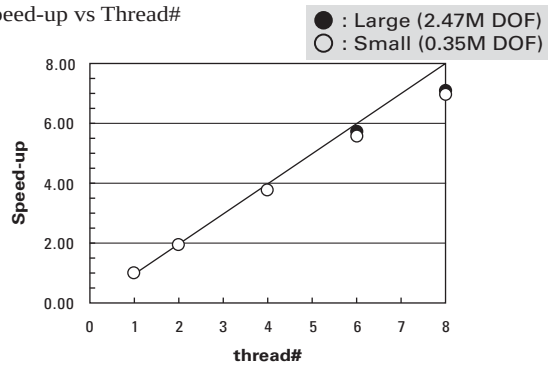


Fig. 4 Effect of thread number on computation time of set-up (least-square minimization by QR factorization) and solver part of SAI/GPBiCG for simple block model (single SMP node of ES, penalty number $\lambda = 10^6$, drop tolerance = 0.25).

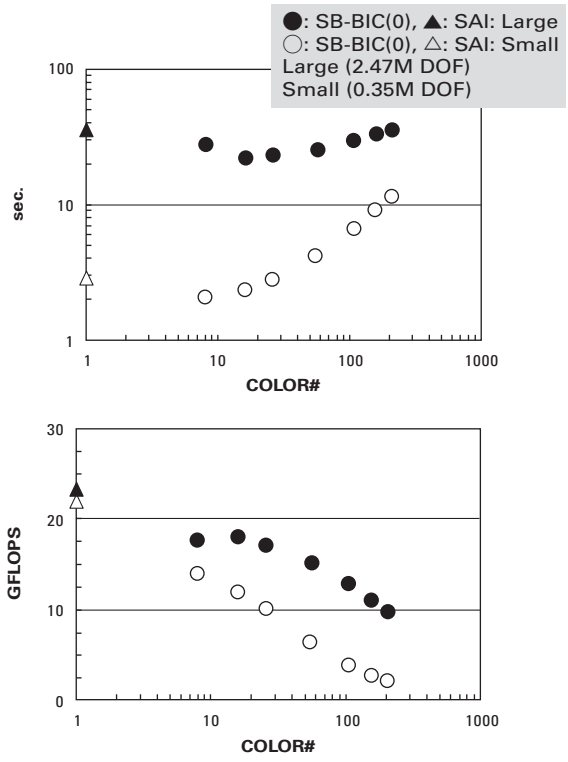


Fig. 5 Comparison between SB-BIC(0)/CG and SAI/GPBiCG for computation time (set-up + solver) (single SMP node of ES (peak: 64GFLOPS), penalty number $\lambda = 10^6$, drop tolerance = 0.25).

4. SAI on Other Architectures

In this section, SAI preconditioned iterative solver has been ported to a single SMP node of Hitachi SR8000/MPP at the University of Tokyo [8] and IBM SP-3 (Seaborg) at NERSC/Lawrence Berkeley National Laboratory [9]. Performance is evaluated for linear-elastic applications with contact on simple block model (Fig. 2). Table 2 summarizes the feature of a single SMP node of Hitachi SR8000/MPP, IBM SP-3 and the Earth Simulator.

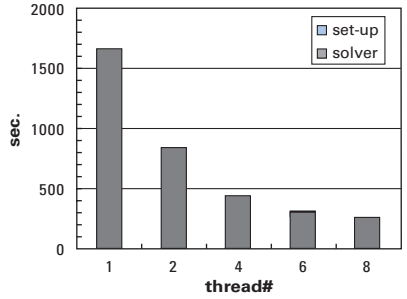
Each PE of Hitachi SR8000 and IBM SP-3 is a scalar processor. In Hitachi SR8000, codes for vector processors are optimized through its pseudo-vector capability [1, 4, 8]. Figures 6-9 show results on Hitachi SR8000 and IBM SP-3.

On the Earth Simulator, rate of set-up process in the total computation has been about 15% as shown in Fig. 4(a), but on Hitachi SR8000, this rate is less than 1% (Fig. 6(a)). Performance of set-up process on the Earth Simulator was not so good as solver part due to shorter inner-most loops. Scalability

Table 2 Node comparisons of Hitachi SR8000/MPP, IBM SP-3 and the Earth Simulator [3, 8, 9]

	Hitachi SR8000/MPP	IBM SP-3 NERSC/LBNL	Earth Simulator
PE#/node	8	16	8
Clock rate	450 MHz	375 MHz	500 MHz
Peak performance/PE	1.80 GFLOPS	1.50 GFLOPS	8.00 GFLOPS
Memory/node	16 GB	16 GB ~ 64 GB	16 GB
Memory-PE Bandwidth	32 GB/sec/node	16 GB/sec/node	256 GB/sec/node

(a) Timing for Large Model



(b) Speed-up vs Thread#

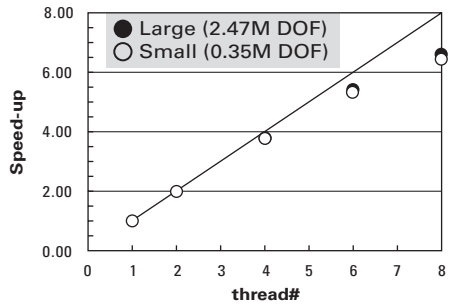


Fig. 6 Effect of thread number on computation time of set-up and solver part of SAI/GPBiCG for simple block model (single SMP node of Hitachi SR8000/MPP, penalty number $\lambda = 10^6$, drop tolerance = 0.25).

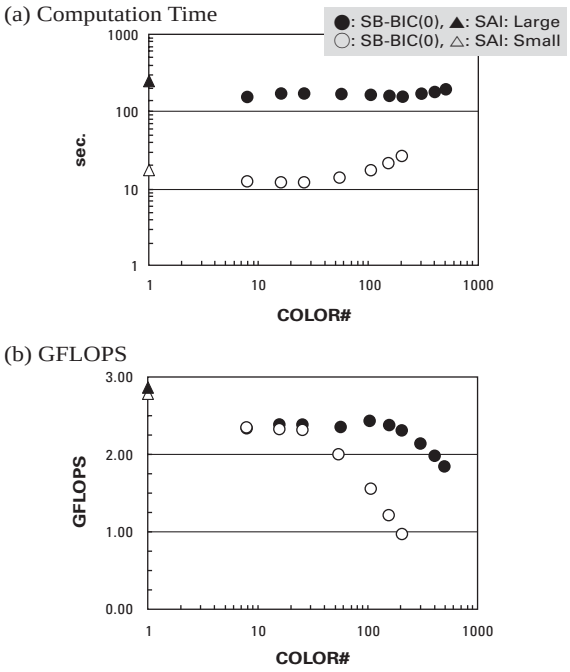


Fig. 7 Comparison between SB-BIC(0)/CG and SAI/GPBiCG for computation time (set-up + solver) (single SMP node of Hitachi SR8000/MPP (peak: 14.4 GFLOPS), $\lambda = 10^6$, drop tolerance = 0.25).

according to thread number is slightly worse than that of ES (Fig. 4(b) and Fig. 6(b)) due to smaller memory bandwidth of Hitachi SR8000 (Table 2). Performance of SAI/GPBiCG is not affected by problem size at all on Hitachi SR8000 (Fig. 7), and provides more than 24% of peak performance.

On IBM SP-3, speed-up ratio is less than 8 for 16 PE's on a SMP node due to rather smaller memory bandwidth (Fig. 8, Table 2). The rate of set-up process is less than 0.10%, because cache is utilized more efficiently for this process than for solver part. Performance of SAI/GPBiCG is around 5% of peak performance (Fig. 9).

5. Concluding Remarks

In this study, SAI preconditioning method has been implemented to a single SMP node of the Earth Simulator using OpenMP. Performance and robustness has been competitive with that of SB-BIC(0) for finite-element applications with contact condition. SAI/GPBiCG provides excellent performance for a wide range of problem size on ES, and is competitive with SB-BIC(0)/CG. Set-up procedure for least-square minimization by QR factorization is expensive on ES due to small vector length. Optimization of this procedure is a future work. The developed code has been ported to other systems, such as Hitachi SR8000/MPP and IBM SP-3. Feature of performance on Hitachi SR8000/MPP was similar to that of ES due to its pseudo-vector capability. But performance of SAI/GPBiCG on IBM SP-3 was not good.

Hybrid parallel programming model with MPI will be implemented for larger scale problems in the future works.

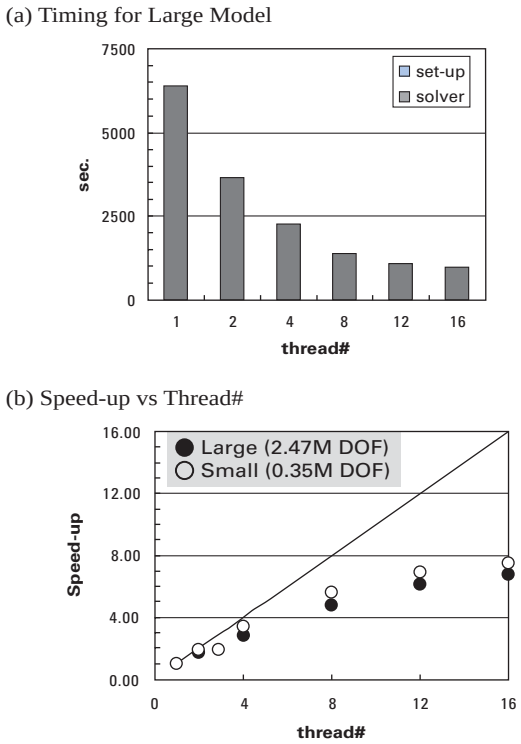


Fig. 8 Effect of thread number on computation time of set-up and solver part of SAI/GPBiCG for simple block model (single SMP node of IBM SP-3, penalty number $\lambda = 10^6$, drop tolerance = 0.25).

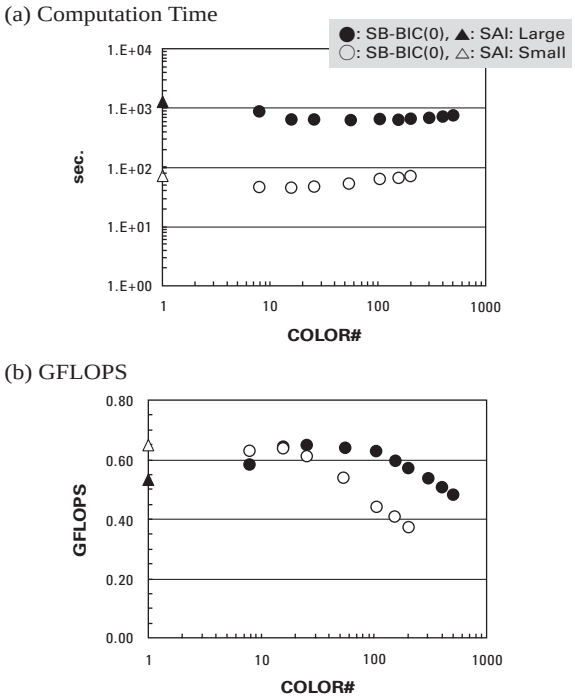


Fig. 9 Comparison between SB-BIC(0)/CG and SAI/GPBiCG for computation time (set-up + solver) (single SMP node of IBM SP-3 (8 of 16 PE's used,peak: 12 GFLOPS), $\lambda = 10^6$, drop tolerance = 0.25).

References

[1] Nakajima, K.: "Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator", SC2003 Technical Paper, Phoenix, AZ, USA, 2003.

[2] Rabenseifner, R.: "Communication Bandwidth of Parallel Programming Models on Hybrid Architectures", *Lecture Notes in Computer Science* 2327 (2002), pp.437-448.
[3] http://www.es.jamstec.go.jp/
[4] http://geofem.tokyo.rist.or.jp/
[5] Chow, E.: "A priori sparsity pattern for parallel sparse approximate inverse preconditioners", *SIAM J. Sci. Comp*., 21 (2000), pp.1804-1822.
[6] Wang, K. et al.: "Global and localized parallel preconditioning techniques for large scale solid earth simulation", *Future Generation Computer Systems*, 19 (2003), pp.443-456.
[7] Zhang, S.L.: "GPBi-CG: Generalized Product-type methods based on Bi-CG for solving non-symmetric linear systems", *SIAM J. Sci. Comp*., 18 (1997) pp.537-551.
[8] http://www.cc.u-tokyo.ac.jp/
[9] http://www.nersc.gov/

固体地球シミュレーションプラットフォームの開発

プロジェクト責任者
中島　研吾　　東京大学大学院 理学系研究科 地球惑星科学専攻
著者
中島　研吾　　東京大学大学院 理学系研究科 地球惑星科学専攻

1. プロジェクトの目的

固体地球シミュレーション・プラットフォーム「GeoFEM」において、プラグイン形式による解析システム結合、大規模線形システムの高速並列解法、大規模並列可視化、その他の汎用的解析支援機能に関する研究を行う。具体的な項目は以下の通り：(1)並列反復法による高速ソルバー最適化、性能評価(2)並列固体地球シミュレーション、連成解析カップラー、並列可視化からなる統合化プラットフォーム開発(3)西南日本地震発生サイクル解析、地球外核電磁熱流動解析へのプラットフォームの適用(4)地下水流動・物質拡散解析、津波シミュレーション、実機工学問題解析の実施(5)構造格子系、離散粒子系へのプラットフォーム拡張

2. 本年度成果

① SAI(Sparse Approximate Inverse)法による前処理手法を開発し、最適化を実施し、ハイブリッド型並列プログラミングモデルにおいて従来手法より高い性能を達成した。
② AMRによるメッシュ分割ツールの開発、境界要素法型解法向けプラットフォーム拡張、粒子型解法向け可視化手法の開発を実施した。
③ グループへのツールの提供、最適化支援を実施した
- 古村グループ(固体地球)　並列可視化機能
- 平原グループ(固体地球)　前処理付き並列反復法、AMRツール(図1)
- 松浦グループ(固体地球)　境界要素法型解法向けプラットフォーム
- 阪口グループ(先進創出:DEM)　ポアソン方程式ソルバー、粒子法可視化機能(図2)
- 奥田グループ(先進創出:原子力)　ポアソン方程式ソルバー、大規模メッシュ生成機能、並列可視化機能

④ これまで「地球シミュレータ」上で開発された「GeoFEM」ツールを使用して、有限要素法による第一原理計算コードを開発し、シミュレーションを実施した(図3)。

キーワード:GeoFEM，並列有限要素法，並列反復解法，ハイブリッド並列プログラミングモデル，
Sparse Approximate Inverse (SAI)

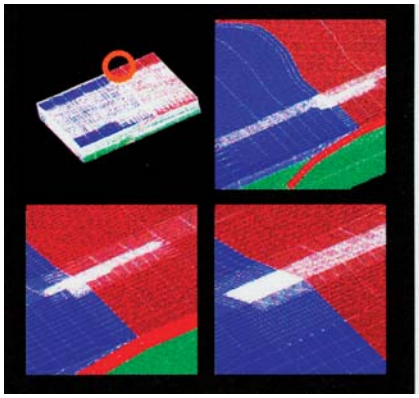


図1　複雑断層面上でのすべり応答関数算出のための局所的細分化

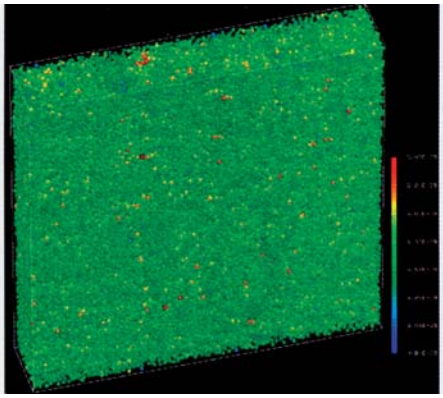


図2　粒子法のための可視化例

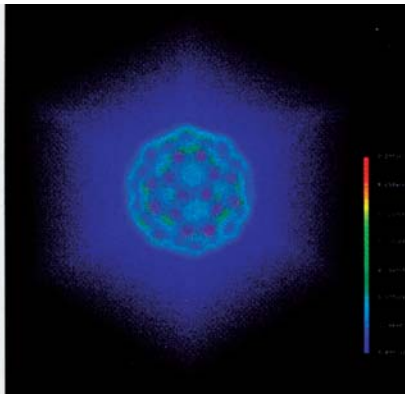


図3　実空間第一原理計算 C60フラーレン電子密度