

# 自動チューニングインターフェース OpenATLib における 自動チューニング機能の評価

櫻井隆雄<sup>†</sup> 片桐孝洋<sup>††</sup> 直野健<sup>†</sup>  
黒田久泰<sup>‡,††</sup> 中島研吾<sup>††</sup> 猪貝光祥<sup>§</sup>  
大島聡史<sup>††</sup> 伊藤祥司<sup>††</sup>

科学技術計算等で利用される行列計算ライブラリは高い演算性能が得られるパラメータの選択や入力に多大な手間が必要なため、それを自動的に設定する方式が求められている。そこで、筆者らは自動チューニングインターフェース OpenATLib を開発している。本稿では OpenATLib の提供する機能の 1 つであるリスタート周期自動チューニング機能について述べる。本機能では残差履歴を用いて最適なリスタート周期を自動的に選択する。T2K オープンスーパーコンピュータを用いて 3 種の解法で本機能の効果を評価した結果、固定値と比較して最大で 38.5 倍の性能差があり、機能の有効性が確認できた。

## Evaluation of Auto-Tuning Function on OpenATLib

Takao Sakurai<sup>†</sup> Takahiro Katagiri<sup>††</sup> Ken Naono<sup>†</sup>  
Hisayasu Kuroda<sup>‡,††</sup> Kengo Nakajima<sup>††</sup> Mitsuyoshi Igai<sup>§</sup>  
Satoshi Ohshima<sup>††</sup> Shoji Itoh<sup>††</sup>

Matrix libraries have many parameters as inputs by the user. They include problem parameters what are difficult to set values and the approach of automatically setting them is needed. Then, we proposed Auto-tuning interface "OpenATLib." In this paper, we explain a runtime automatic tuning approach for deciding the size of projection matrix in Krylov subspace methods. This approach searches the best size of projection matrix with history of residual values at runtime. Performance evaluations of OpenATLib using 3 solvers on T2K Open Supercomputer indicates that the maximum speedup establishes 38.5x.

## 1. はじめに

科学技術計算などに利用される行列計算ライブラリでは、入力パラメータの値の選択、疎行列ベクトル積演算や直交化演算といった重要な演算のアルゴリズムの選択により全体の演算時間が大幅に変化する。特に一部のパラメータの選択を誤った場合、解が収束しなくなるなど、ライブラリの機能そのものに影響を及ぼすことがある。一方で、短い時間で演算できる良好なパラメータやアルゴリズムは入力行列や計算機環境によって大きく異なり、かつ事前予測が難しいものが存在する。そのため、行列計算ライブラリのユーザが良好な条件でライブラリを利用するのが困難となっていた。

そこで、これらの問題を解決するために、入力パラメータやアルゴリズム選択の支援をする自動チューニング方式 (AT 方式) が注目されている。また、近年はそれらの方式を備えた数値計算ライブラリが開発されてきた。例えば、PHiPAC[1], ATLAS[2], FFTW[3], I-LIB[4], ABCLib[5], OSKI[6]などである。これらの方式やライブラリを用いることで良好な条件での行列計算が容易になってきている。

しかし、既存のライブラリでは対応していない特殊な行列を処理する場合などにおいて、行列計算ライブラリを自作する場合もある。このような自作のライブラリに対し、その作成者が AT 方式を適用する場合、プログラム記述のためにコード量が増加する可能性がある。これらの問題のため、作成者が多数の AT 方式を自作ライブラリに適用するには非常に多くの労力を必要とする。そこで、これまで個別に提案されてきた自動チューニングの実装に関して再利用性、可搬性を高める手段が求められてきた。

この課題に対し、筆者らは AT 方式の共通部分を関数として提供する AT インターフェース OpenATLib を提案している。OpenATLib により少ない労力で AT 方式を自作ライブラリに組み込むことが可能となる。

本稿では OpenATLib の 2011 年版における主要機能の 1 つであるリスタート周期の自動チューニング方式について述べ、それと同時に T2K オープンスーパーコンピュータ上で性能を評価することでその有効性を検証する。

## 2. ATインターフェースOpenATLibの概要

筆者らは、自動チューニングの実装の再利用性を高めるため、Fortran で記述された

<sup>†</sup> 日立製作所 中央研究所  
Central Research Laboratory, Hitachi, Ltd.

<sup>††</sup> 東京大学情報基盤センター スーパーコンピューティング研究部門  
Supercomputing Research Division, Information Technology Center, The University of Tokyo

<sup>‡</sup> 愛媛大学 大学院理工学研究科  
Graduate School of Science and Engineering, Ehime University

<sup>§</sup> 日立超 LSI システムズ  
Hitachi ULSI Systems Co., Ltd.

疎行列反復法ライブラリ向けの汎用 AT インターフェース OpenATLib を提案している [7-9].

OpenATLib は自動チューニング機能を備えた疎行列ベクトル積や直交化演算のサブルーチンを API (Application Programming Interface) の形式で提供する. これらは主にライブラリ作成者が自作ライブラリに AT 方式を実装するために利用することを想定している.

主として対象とするライブラリは疎行列の反復解法であり, 実行前に予測が困難なパラメータやアルゴリズムの最適値を実行中に探索し, 1 回の演算で安定して高い演算性能を提供することが可能となる.

現在開発中の OpenATLib の 2011 年版が備える機能は以下の 4 つである. これらの機能の命名規則は次のとおりである. まず, “OpenATI\_” で機能が始まる. 続いて, “\_” 後の最初の 1 文字は演算精度 (S:単精度, D:倍精度) であり, 2 文字目と 3 文字目について, 補助機能の場合は “AF” とし, 演算機能の場合は, 2 文字目は行列形状 (S:対称, U:非対称), 3 文字目は行列の格納形式 (R:CRS, C:CCS) を表す. 4 文字目と 5 文字目以降は機能名となる.

#### ● リスタート周期最適化機能 : OpenATI\_DAFRT

Krylov 部分空間法において, 相対残差の履歴から算出した停滞を判断する指標を用いてリスタート周期の最適値を実行時に判断する.

#### ● 疎行列ベクトル積機能 : OpenATI\_D{S|U}RMV

複数用意された疎行列ベクトル積のアルゴリズムの中で, 与えられた行列と実行する計算機環境において最も高い性能が得られる方式を判断する. OpenATI\_DSRMV が対称行列向けの関数であり, OpenATI\_DURMV が非対称行列向けの関数である.

#### ● Gram-Schmidt 直交化機能 : OpenATI\_DAFGS

複数用意された Gram-Schmidt 直交化の実装方式の中で, 与えられた行列と実行する計算機環境において最も高い性能が得られる方式を判断する.

#### ● 数値計算ポリシーを反映するためのメタインターフェース :

##### OpenATI\_LINEAROLVE | OpenATI\_EIGENSOLVE

ユーザが要求する AT の方針 (ポリシー) に基づいてライブラリに演算を実行させるためのメタインターフェース.

本稿ではこれらの機能の中で, リスタート周期最適化機能に関して, 機能の詳細を述べるとともに, 性能評価を実施する.

### 3. リスタート周期最適化機能

#### 3.1 リスタート付き反復解法とその課題

疎行列解法として代表的な Krylov 部分空間法の中には, 計算機上で実行する場合, 使用するメモリを実行前に確定させるため Krylov 部分空間の次元数を一定数に制限する. それより大きくなった場合は適切な新しい初期値 (例えば得られた最新の近似解) を設定してリスタートさせ新たな Krylov 部分空間を生成する. このときの Krylov 部分空間の最大次元数をリスタート周期 (ここでは  $m$  とする) と呼び, これらのリスタートを行う反復解法をリスタート付き反復解法と呼ぶ. 代表的なリスタート付き反復解法として GMRES( $m$ )法, Lanczos 法等がある.

このリスタート付き反復解法の問題点として,  $m$  の設定が困難であることがある.  $m$  の値が大きい場合は収束が加速され反復回数が少なくなるが, 1 回の反復に必要な計算量が増加する. 一方で,  $m$  の値が小さい場合は 1 回の反復に必要な計算量が少なくなるが, 反復回数が増加し, 場合によっては指定した最大の反復回数では収束しないことがある. そのため,  $m$  が小さすぎても大きすぎても演算時間が増加することがある.

リスタート周期により演算時間が変化する例を図 1 に示す. 図 1 ではフロリダ大学の Sparse Matrix Collection[10]から取得した行列 `bmw7st_1` を筆者らが開発している Lanczos 法ライブラリである `Xablib_LANCZOS` を用いて 10 個の固有値を求める際に  $m$  の値を 12 から 2 刻みで 150 まで増加させた. 図 1 によると  $m$  の値が 12~14 のときは演算時間が 1 秒を超えているが, 16 になると 0.3 秒に低下し, それから  $m$  の値に従って単調増加する. よって, この場合は最適の  $m$  は 16 であると考えられる.

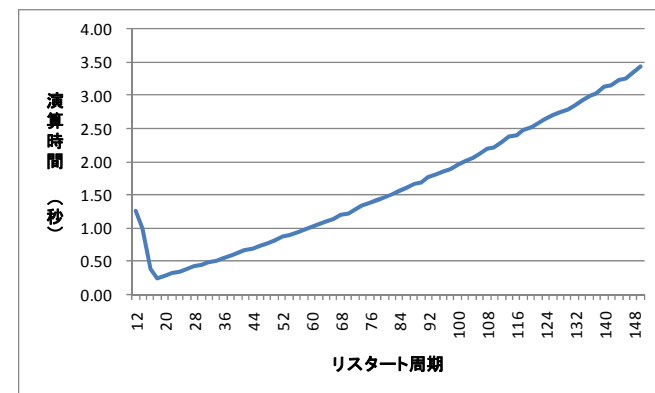


図 1 リスタート周期と演算時間の関係の一例

ここで、さらに問題となるのが最適の  $m$  の値が入力される行列や計算の条件（例えば求める固有値の個数）によって異なり、事前に予測できない点である。この問題のため、リスタート付き反復解法を使用して高い性能を得るには、ユーザが過去の経験から最適な  $m$  の値を予測して入力しなければならない。

そこで、ユーザの予測によらず高い性能が得られる  $m$  を探索し実行する仕組みが必要となる。

### 3.2 リスタート周期最適化機能OpenATI\_DAFRTの概要

OpenATI\_DAFRT はリスタート付き反復解法において高い性能が得られるリスタート周期  $m$  を探索するために、 $m$  を現在の値からより大きな値とするべきか判定する機能である。

最適なリスタート周期の推定は実行前には困難であるが、実行中であれば  $m$  の大きさを評価する指標が存在する。図 2 に示すように  $m$  が十分大きい場合は相対残差は反復ごとに大きく減少しているが、小さい場合は相対残差の減少が緩やかになる。よって、近似解の相対残差の履歴を監視し、相対残差の減少値を元とした指標を用いることで高い性能が得られる  $m$  の推定が可能となる。

相対残差の減少の緩やかさを示す指標として収束判定  $s$ - $t$  回目から  $s$  回目の残差の中での最大値を最小値で割った値を「残差 Max-Min 比」とし、以下では「MM 比」[11] と表記する。 $s$  回目の収束判定時の  $i$  番目の残差  $r_i$  に対する過去  $t$  回分の MM 比  $R_i(s,t)$  を式で表したものが以下である。

$$R_i(s,t) = \frac{\max_z \{r_i(z); z = s-t+1, \dots, s\}}{\min_z \{r_i(z); z = s-t+1, \dots, s\}}$$

リスタート周期が十分に大きいときは相対残差が大きく減少して MM 比は大きくなり、小さいときは相対残差の減少が緩やかになるため MM 比は小さくなる。

そこで、リスタート周期の大きさを自動的に最適化する場合は MM 比を監視し、MM 比が一定値を下回ったときにリスタート周期を大きくするようにすればよい。

OpenATI\_DAFRT は入力引数として残差履歴を受け取り、その MM 比を算出する。その結果、MM 比が閾値（デフォルト値は 100）以下であった場合は出力として 1 を返す。ライブラリ作成者は収束判定のための残差算出後に OpenATI\_DAFRT を呼び出し、その出力が 1 であったときにリスタート周期を大きくするように記述すれば反復解法の実行中に最適なリスタート周期を探索することが可能になる。

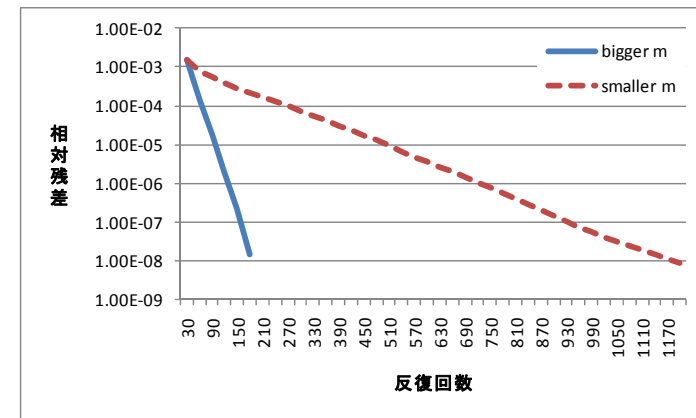


図 2  $m$  の大きさによる相対残差の減少値の差異

## 4. リスタート周期最適化機能の評価

### 4.1 評価環境とテスト行列

OpenATI\_DAFRT の有効性を確認するためにリスタート周期  $m$  を固定した場合との性能比較評価を実施した。本評価ではリスタート付き反復法ライブラリとして OpenATLib を用いて開発した Xabclib[10]を用いる。使用する解法は非対称行列向け連立一次方程式解法の GMRES(m)法[12]のソルバである Xabclib\_GMRES, 非対称行列向け固有値解法である Arnoldi 法[13]を実装した Xabclib\_ARNOLDI, 同じく対称行列向け固有値解法である Lanczos 法[14]を実装した Xabclib\_LANCZOS の 3 種である。

本評価での計算機は T2K オープンスパコン（東大版）1 ノードを利用した。T2K オープンスパコン（東大版）1 ノードの仕様およびコンパイル環境を表 1 に示した。

テスト行列はフロリダ大学の Sparse Matrix Collection[10]より対称行列と非対称行列をそれぞれ 10 個ずつ選出した。テスト用の対称行列を表 2, 非対称行列を表 3 に示した。表 2, 表 3 では分野ごとに行列をまとめ、次元数の少ない順に並べた。

解法の条件として、固有値解法において求める固有値の個数は 10, 連立一次方程式解法では解ベクトルは全要素が 1, 初期近似解ベクトルは全要素 0 とし、前処理として不完全 LU 分解を用いた。収束条件は共に  $1.0E-8$  とした。

OpenATI\_DAFRT の機能に関連するパラメータとして、MM 比の閾値を 100, 閾値を下回ったときに大きくする  $m$  の値を 5, リスタート周期の初期値を 10, 最大値を 150 とし高い性能が得られる  $m$  を探索させた。

比較の対象となる固定したリスタート値は PETSsc[15]で標準とされる 30 と、10 から

150 の間で 2 刻みに測定したときに最も短い演算時間で収束した値 (best m) と最も長い演算時間を必要とした値 (worst m) とした。

表 1 T2K オープンスパコン (東大版) 1 ノードの仕様およびコンパイル環境

CPU	Quad-Core AMD Opteron(tm) Processor 8356 2.3GHz 16core/1node
L2 サイズ	2MByte/4core
主記憶サイズ	32GByte (8GByte/1Socket)
OS	Red Hat Enterprise Linux 5
コンパイラ	Intel Fortran Compiler Professional Version 11.0
コンパイルオプション	-O3 -m64 -openmp -mcmodel=medium

表 2 テスト用対称行列

Matrix	N	NNZ	Field
vibrobox	12328	177578	acoustics
Lin	256000	1011200	chemistry
cf1	70656	949510	fluid
cf2	123440	1605669	dynamics
Si5H12	19896	379247	structural
SiO	33401	675528	
H2O	67024	1141880	
F2	71505	2682895	
bmw7st_1	141347	3740507	
SiO2	155331	5719417	

表 3 テスト用非対称行列

Matrix	N	NNZ	Field
chem_master1	40401	201201	2D/3D
torso2	115967	1033473	
torso3	259156	4429042	
chipcool0	20082	281150	model reduction
memplus	17758	126150	electric circuit
poisson3Da	13514	352762	fluid
airfoil_2d	14214	259688	dynamics
wang4	26068	177196	semiconductor
ecl32	51993	380415	device
viscoplastic2	32769	381326	materials

#### 4.2 リスタート周期最適化機能の評価結果

リスタート周期を固定した場合と OpenATI\_DAFRT を用いて自動でチューニングした場合 (AT) を比較評価した結果を表 4~6 に示した。表 4 が Xabclib\_GMRES (GMRES(m) 法), 表 5 が Xabclib\_ARNOLDI(Arnoldi 法), 表 6 が Xabclib\_LANCZOS(Lanczos 法)での結果である。表中の“Not Conv”はその条件では収束しなかった場合を示す。

表 4~6 の結果から以下が読み取れる。OpenATI\_DAFRT を使用してリスタート周期自動設定した場合、最も長い演算時間を必要とした値 (worst m) と比較すると、GMRES(m)法, Arnoldi 法, Lanczos 法それぞれで解けなかった行列 2 種, 1 種, 5 種が解けるようになり、それぞれ最大で 93%, 96%, 97%, 平均して 68%, 74%, 82% の演算時間が削減できていた。また、リスタート周期を 30 に固定した場合と比べると、GMRES(m)法で解けなかった行列 1 種が解けるようになり、GMRES(m)法, Lanczos 法それぞれで最大 43%, 78%, 平均して 12%, 30% の演算時間が削減できた。以上から、OpenATI\_DAFRT を用いることで、高い性能が得られる m が選択できていると言える。

その一方で、Arnoldi 法で、リスタート周期を 30 に固定した場合と比べると、解けなかった行列 1 種が解けるようになったものの、最悪の場合で 441%, 平均して 68% 演算時間が増加していた。また、最も短い演算時間で収束したリスタート周期 (best m) と比較したとき、GMRES(m)法の memplus では 3% 演算時間が削減できていたが、GMRES(m)法, Arnoldi 法, Lanczos 法それぞれで最悪の場合 211%, 689%, 149%, 平均して 69%, 144%, 46% 演算時間が増加していた。

この原因として、OpenATI\_DAFRT 自体のパラメータである MM 比の閾値と、閾値を下回ったときに大きくする m の値が適切でないことが考えられる。この問題の原因

の調査と、これらのパラメータを調整する仕組みを追加することや、例えば、大きすぎる  $m$  の値を小さくするようにするなど方式そのものを改良することが今後の課題となる。

表4 Xabclib\_GMRES における OpenATI\_DAFRT の効果

行列名	①m=30	②best m	③worst m	④AT	④/①	④/②	④/③
chem_master1	2.01	0.83	2.84	1.18	0.59	1.42	0.42
torso2	0.60	0.21	4.40	0.29	0.48	1.38	0.07
torso3	5.46	2.98	15.55	4.75	0.87	1.59	0.31
chipcool0	0.55	0.50	1.71	0.81	1.47	1.62	0.47
memplus	0.71	0.69	2.13	0.67	0.94	0.97	0.31
poisson3Da	0.70	0.50	1.67	0.79	1.13	1.58	0.47
airfoil_2d	Not. Conv	1.22	Not. Conv	3.79	-	3.11	-
wang4	0.21	0.21	1.27	0.23	1.10	1.10	0.18
ecl32	9.88	1.97	Not. Conv	5.61	0.57	2.85	-
viscoplastic2	7.82	4.60	15.90	5.69	0.73	1.24	0.36
平均					0.88	1.69	0.32

表5 Xabclib\_ARNOLDI における OpenATI\_DAFRT の効果

行列名	①m=30	②best m	③worst m	④AT	④/①	④/②	④/③
chem_master1	3.66	2.51	46.14	19.80	5.41	7.89	0.43
torso2	39.49	7.51	45.82	11.46	0.29	1.53	0.25
torso3	41.91	14.71	272.43	18.21	0.43	1.24	0.07
chipcool0	0.93	0.79	4.21	1.53	1.65	1.94	0.36
memplus	117.81	8.92	411.83	18.21	0.15	2.04	0.04
poisson3Da	7.39	6.75	21.30	8.54	1.16	1.27	0.40
airfoil_2d	Not. CONV	2.75	Not. CONV	3.55	-	1.29	-
wang4	1.46	0.98	25.86	2.66	1.82	2.71	0.10
ecl32	0.97	0.79	9.67	1.01	1.04	1.28	0.10
viscoplastic2	7.41	7.41	42.95	23.65	3.19	3.19	0.55
平均					1.68	2.44	0.26

表6 Xabclib\_LANCZOS における OpenATI\_DAFRT の効果

行列名	①m=30	②best m	③worst m	④AT	④/①	④/②	④/③
vibrobox	0.72	0.41	4.96	0.62	0.86	1.51	0.13
Lin	468.71	108.85	Not. CONV	158.90	0.34	1.46	-
cfid1	9.31	5.89	Not. CONV	8.63	0.93	1.47	-
cfid2	9.71	9.38	20.05	9.41	0.97	1.00	0.47
Si5H12	5.61	3.62	25.02	3.80	0.68	1.05	0.15
SiO	9.03	4.78	Not. Conv	5.05	0.56	1.06	-
H2O	97.90	19.47	Not. Conv	27.81	0.28	1.43	-
F2	5.60	2.18	210.96	5.43	0.97	2.49	0.03
bmw7st_1	0.44	0.25	3.43	0.51	1.16	2.04	0.15
SiO2	268.54	52.91	Not. Conv	58.27	0.22	1.10	-
平均					0.70	1.46	0.18

## 5. おわりに

本稿では、自動チューニングインターフェース OpenATLib の 1 機能である、リスタート周期自動チューニング機能 OpenATI\_DAFRT について概要と性能評価を実施した。OpenATI\_DAFRT は近似解の相対残差の減少幅から、高い性能が得られるリスタート周期を自動的に選択し、実行することを特徴とする。また、GMRES(m)法、Arnoldi 法、Lanczos 法に本機能を適用し、フロリダ大の Sparse Matrix Collection から得た 20 個の行列を用いて性能評価を実施した。

評価の結果、OpenATI\_DAFRT により GMRES(m)法、Lanczos 法では標準のリスタート周期の値で演算した場合よりも平均して 12~30%程度高い性能が得られることが確認できた。また、標準の値では収束しなかったものが収束するようになったケースが 2 例あった。以上の結果により提案法の有効性が示された。

しかし、Arnoldi 法においては標準の値で演算した場合と比べ性能が低下していた。また、全てのリスタート周期の中で最も高い性能が得られる値と比べると提案法の性能は劣っており、より高い精度で適切なリスタート周期を決定する仕組みが必要と分かった。その考案が今後の課題となる。

なお、OpenATLib とそれを用いて開発された疎行列反復法ライブラリ Xabclib のソースコードは、PC クラスタコンソーシアム経由で配布されている。

**謝辞** 本研究は文部科学省「e-サイエンス実現のためのシステム統合・連携ソフト

ウェアの研究開発」, シームレス高生産・高性能プログラミング環境, の支援を受けている.

## 参考文献

- 1) Bilmes, J., Asanovic, K., Chin, C.-W., and Demmel, J.W.: Optimizing Matrix Multiply Using PHiPAC: a Portable, High-Performance, ANSI C Coding Methodology, Proceedings of International Conference on Supercomputing 97, pp.340-347 (1997).
- 2) Whaley, R.C., Petitet, A., and Dongarra, J.J.: Automated Empirical Optimizations of Software and the ATLAS Project, Parallel Computing, 27, pp.3-35 (2001).
- 3) Frigo, M.: A Fast Fourier Transform Compiler, Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation, Atlanta, Georgia, pp.169-180 (1999).
- 4) 片桐孝洋, 黒田久泰, 大澤清, 工藤誠, 金田康正: 自動チューニング機構が並列数値計算ソフトウェアに及ぼす効果, 情報処理学会論文誌:ハイパフォーマンスコンピューティングシステム, 42, SIG 12 (HPS 4), pp.60-76 (2001).
- 5) Katagiri, T., Kise, K., Honda, H., and Yuba, T. : ABCLib\_DRSSD: A Parallel Eigensolver with an Auto-tuning Facility, Parallel Computing, 32, 3, pp.231-250 (2006).
- 6) Vuduc, R., Demmel, J.W., and Yelick, K. : OSKI:A Library of Automatically Tuned Sparse Matrix Kernels, Proceedings of SciDAC 2005, Journal of Physics: Conference Series (2005).
- 7) 片桐孝洋, 櫻井隆雄, 黒田久泰, 直野健, 中島研吾: OpenATLib:汎用的な自動チューニングインターフェースの設計と実装, 情報処理学会研究報告:ハイパフォーマンスコンピューティング, 2009-HPC-121(3) (2009).
- 8) 櫻井隆雄, 直野健, 片桐孝洋, 中島研吾, 黒田久泰: OpenATLib を利用した疎行列ライブラリの開発と評価, 情報処理学会研究報告:ハイパフォーマンスコンピューティング, 2009-HPC-121(17) (2009).
- 9) Xabclib プロジェクトホームページ, <http://www.abc-lib.org/Xabclib/index-j.html>
- 10) Davis, T. A. : Sparse Matrix collection, <http://www.cise.ufl.edu/research/sparse/matrices/>
- 11) 櫻井隆雄, 直野健, 恵木正史, 猪貝光祥, 木立啓之:リスタート付ランチョス法における実行時パラメータ自動チューニング方式の提案, 情報処理学会研究報告:ハイパフォーマンスコンピューティング, 2007-HPC-111(30), pp.173-178 (2007).
- 12) Saad, Y. : Iterative Methods for Sparse Linear Systems (2nd ed.), SIAM, Philadelphia (2003).
- 13) Lehoucq, R. B. : Analysis and implementation of an implicitly restarted Arnoldi iteration, TR95-13 (Rice Univ.)(1995).
- 14) Hernandez, V. , Roman, J. E. , and Tomas, A. : Evaluation of Several Variants of Explicitly Restarted Lanczos Eigensolvers and Their Parallel Implementations, High Performance Computing for Computational Science - VECPAR 2006, pp.403-416 (2007).
- 15) PETSc: Home Page, <http://www.mcs.anl.gov/petsc/petsc-as/>