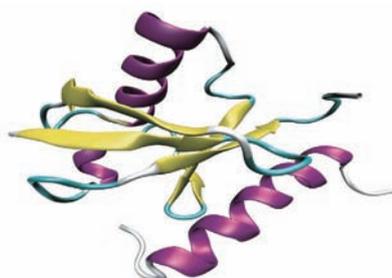


# TSUBAME ESJ.



フラグメント分子軌道法を用いた  
生体分子のNMR 化学シフトの計算

気液二相流シミュレーションの  
大規模 GPUコンピューティング

大規模並列計算システムを用いた  
進化的画像・映像符号化



# フラグメント分子軌道法を用いた 生体分子のNMR化学シフトの計算

Qi Gao<sup>\*/\*\*</sup> 横島智<sup>\*\*</sup> 中村振一郎<sup>\*\*</sup> 櫻井実<sup>\*</sup>

<sup>\*</sup>東京工業大学バイオ研究基盤支援総合センター <sup>\*\*</sup>三菱化学科学技術研究センター

フラグメント分子軌道法 (fragment molecular orbital method: FMO法) を利用した巨大分子に適用可能なNMR化学シフトの計算法を開発した。FMO法は分子系をいくつかのフラグメントに分割し、フラグメントごとの計算を並列に行うことで計算コストを劇的に減らすことができる計算法である。いくつかの系に適用したところ、FMO法を用いて計算された化学シフトは従来の化学シフトの計算法で得られたものと正確に一致することが示された。

われわれの研究は、タンパク質のような巨大分子であっても、TSUBAMEのような強力な計算資源を用い並列計算を行うことで、そのさまざまな性質を短時間かつ高精度で評価できる可能性が高いことを示した。

## 序論

# 1

NMRスペクトルにおいて、化学シフトというパラメーターは核のエネルギーレベルが分子中の電子的環境に依存することを表している。言い換えると、核の周囲の分子環境の違いにともなう共振周波数の違いを表している。それゆえ、化学シフトは分子のコンフォメーション、化学組成や周囲の溶媒環境の違いによって敏感に変化する。化学シフトの観測は、他のNMRパラメーター (J-coupling、緩和時間やNOE) の観測と組み合わせると、特に溶液中の分子の3次元構造に関し、極めて有用な情報を与える。

しかしながら、タンパク質分子のような巨大分子に対して、実験的なアプローチだけで化学シフトを対応する立体構造に帰属することは容易ではない。化学シフトの観測に基づいて正確な構造解析を行うためには、分子の電子波動関数がコンフォメーションや他の構造変化によってどのような摂動を受けるかについて情報を得る必要がある。それゆえ、NMRの枠組みの中で分子の構造を決定するためには、理論的なアプローチが非常に重要な役割を担うことになる。われわれは、量子力学に基づいてタンパク質の化学シフトを計算する方法を開発することを目指している。

巨大な分子系における化学シフトの計算はその複雑さから従来の量子化学計算では不可能であると考えられていた。最も大きな問題は化学シフトを計算する際に膨大な計算コストが必要なことである。信頼性のある化学シフトの計算を行うためには摂動論に基づいた高度な量子化学計算が必要である。残念ながら、これらの方法は計算コストが高く、小さな分子にしか適用することができなかった。

本研究では、すでに十分確立されてきた *ab initio* 法に基づく化学シフト計算法とFMO法を組み合わせる方法を開発した。FMO法では巨大分子を小さなフラグメントに分割し並列計算を適用することにより計算の高速化を実現できる。今回われわれはこの方法により、76残基からなる高度に保存されたタンパク質であるユビキチンの全ての化学シフトを予測した<sup>[1]</sup>。さらに、FMO法に基づくNMR計算法の精度を上げるために、カットオフ距離のパラメーターを用いることで適切なフラグメントの大きさを決定する方法も開発した<sup>[2,3]</sup>。

## NMR計算

# 2

化学シフトは、ハミルトニアンに外部磁場を表すベクトルポテンシャルを導入したシュレーディンガー方程式を解くことにより、理論的に評価することができる。ここで新たな問題が発生する。現実の計算では有限個の基底関数を使わなければならないため、計算された化学シフトの値はベクトルポテンシャルのゲージ原点の位置に依存する。ゲージ原点の依存性をなくし、精確に核磁気遮蔽テンソルの値を計算できる手法として、GIAO<sup>[4]</sup>とCSGT<sup>[5]</sup>という二つの方法がある。

GIAO法では、核磁気遮蔽テンソル $\sigma_{\alpha\beta}$ をエネルギー $E$ の外部磁場 $B$ と核磁気モーメント $\mu$ に関する二次微分から計算する。ゲージ依存性の問題は磁場に依存した原子軌道を用いることで取り除かれる。

$$\sigma_{\alpha\beta} = \frac{\partial^2 E}{\partial \mu_\alpha \partial B_\beta} \quad (1)$$

一方CSGT法では、テンソル $\sigma_{\alpha\beta}$ を外部磁場 $B$ に対する線形応答として式(2)を用いて計算する。ここで、 $\mathbf{r}_N$ は原子核の位置を表しており、 $J^{(i)}(\mathbf{r})$ は一次の誘起電流密度である。

$$\sigma_{\alpha\beta}(\mathbf{r}_N) = -\frac{\mu_0}{4\pi} \int d^3 r \left[ \frac{\partial J^{(i)}(\mathbf{r})}{\partial B_\beta} \times \frac{\mathbf{r}_N - \mathbf{r}}{|\mathbf{r}_N - \mathbf{r}|^3} \right]_\alpha \quad (2)$$

CSGT 法においてゲージ原点の問題は、ゲージ原点  $R_0$  を電流密度を評価したい位置  $r$  に依存するパラメーター関数として再定義することで対処されている。このパラメーター関数は誘起電流が発生する場所を意味している。

化学シフト  $\delta$  と磁気遮蔽テンソル  $\sigma_{\alpha\beta}$  は、基準物質の等方的な磁気遮蔽  $\sigma^0$  により以下の様に関係づけられる。

$$\delta = (\sigma^0 - \sigma^{\text{iso}}) / (1 - \sigma^0) \times 10^6 \approx (\sigma^0 - \sigma^{\text{iso}}) \times 10^6$$

ここで、 $\sigma^{\text{iso}}$  は以下のように表わされる。

$$\text{Tr}(\sigma_{\alpha\beta}) / 3$$

## FMO法

# 3

Restricted Hartree-Fock 法 (RHF) レベルの計算では、分子のエネルギー  $E$  は以下の方程式を解くことで計算される。

$$\tilde{\mathbf{F}}\mathbf{C} = \mathbf{S}\mathbf{C}\boldsymbol{\varepsilon} \quad (3)$$

$\tilde{\mathbf{F}}$  は Fock 行列、 $\mathbf{C}$  は分子軌道係数行列、 $\mathbf{S}$  は重なり行列である。

HF 法による方程式を解くための計算時間は系の大きさを  $N$  として  $O(N^3)$  で増加する。その結果、巨大な生体分子に従来の *ab initio* 計算法を用いると計算コストがかかりすぎてしまう。それゆえ、水中のタンパク質のような大きな系に対しては、短時間で系を計算できる FMO 法の利用が必要である。

FMO 法では、巨大分子は  $N$  個の小さなフラグメント (モノマー) に分けられる。すると全エネルギーは以下のように表わされる。

$$E = \sum_I E_I + \sum_{I>J} (E_{IJ} - E_I - E_J) \quad (4)$$

全エネルギーは分子中のモノマー  $E_I$  とダイマー  $E_{IJ}$  (2 モノマーのペア) の合計である。RHF レベルの計算において、モノマーやダイマーの全エネルギーへの寄与は以下のように計算される。

$$\tilde{\mathbf{F}}^x \mathbf{C}^x = \mathbf{S}^x \mathbf{C}^x \boldsymbol{\varepsilon}^x \quad (5)$$

$x$  はモノマー ( $x=I$ )、ダイマー ( $x=IJ$ ) を表している。ここで、すべてのモノマーやダイマーの RHF 計算は、系全体から生じる静電ポテンシャル (ESP) を考慮して行われる。ESP は、モノマーレベルでのエネルギーが収束するまで、すべてのモノマーの RHF 方程式を繰り返し解くことで求められる。そのあとダイマーレベルでのエネルギーは、モノマーの FMO 計算で求められた ESP を用いて、個々のダイマーの RHF 方

式を1回解くことで得られる。図1に FMO 法の計算手順を示した。これまでの生体分子系を対象とした FMO 法の研究<sup>[6-8]</sup>によって、計算コストと必要な計算精度のバランスを保つには、ダイマーレベルまでの計算が適切であることがわかっている。

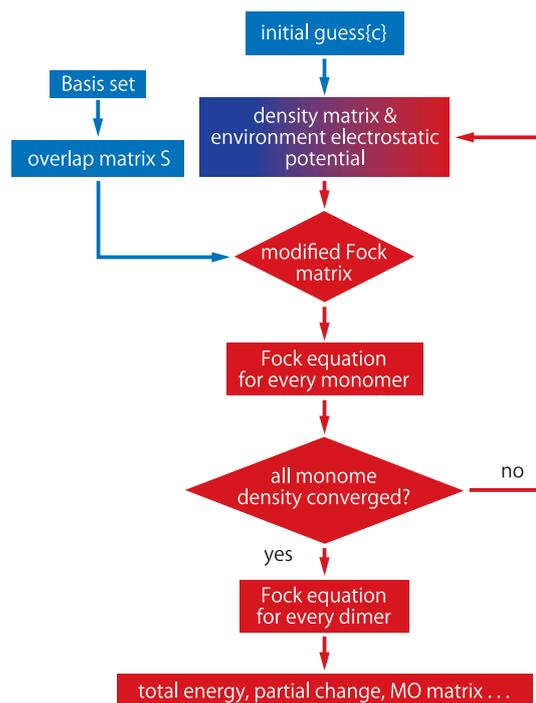


図1

標準的なFMO2の計算手順。個々のモノマーのFock行列は静電ポテンシャルを含むハミルトニアンを用いて求められる。すべてのモノマーのエネルギーが同時に収束するまで繰り返して計算する。

## FMO-NMR法

# 4

われわれは FMO 法を GIAO 法か CSGT 法と組み合わせ、巨大な生体分子系の化学シフトの計算をするための手法を開発した。その手法のオプションとして Model-I と Model-II という二つのモデルがある。Model-I は点電荷近似で一つのフラグメントの遮蔽テンソルを計算する方法である (つまり他のフラグメントからの ESP を点電荷として置き換える)。点電荷はすべてのモノマーの密度行列にから作成される。Model-II は点電荷近似を使わずに直接にフラグメントの ESP を使用する方法である。遮蔽テンソルを計算するために、われわれはまず隣接したモノマーのペア上に作られたダイマー Fock 行列を計算した。

$$\tilde{\mathbf{F}}^J = \tilde{\mathbf{H}}^J + \mathbf{G}^J \quad (J-I=1) \quad (6)$$

## フラグメント分子軌道法を用いた 生体分子のNMR化学シフトの計算

そして、得られた Fock 行列を用いて CPHF 計算を行うことでダイマーの遮蔽テンソルを計算した。GIAO法を Model-I に適用し、一方、CSGT法は Model-I と Model-II 両方に適用した。これら二つのモデルの概念を図2に示す。

われわれは巨大な生体分子の代表例としてユビキチンタンパク質を取り上げて化学シフトの計算をした。計算は6-31G(d)レベルで行った。 $^{13}\text{C}$ 、 $^{15}\text{N}$ や $^1\text{H}$ の化学シフトは図3に示しており、実験的に測定された値と比較してある。すべての原子種の化学シフトの平均誤差は実験データに近い結果となった。例えば、 $^{13}\text{C}_\alpha$ の平均誤差は1.20 ppm (Model I/GIAO)、1.59 ppm (Model I/CSGT)、1.68 ppm (Model II/CSGT)である。標準偏差はそれぞれ1.16 ppm (Model I/GIAO)、1.65 ppm (Model I/CSGT)、1.64 ppm (Model II/CSGT)である。しかし、最大誤差は特に $^{15}\text{N}$ や $^1\text{H}$ において期待されている精度を満たしていない。その原因はこれらの原子が $^{13}\text{C}$ 原子と比べ、周辺環境の影響に敏感であるためである。従って、われわれの求めている精度を達成するためには、原子の周りの電子雲を正しく表すような周辺環境の取り扱いが必要である。

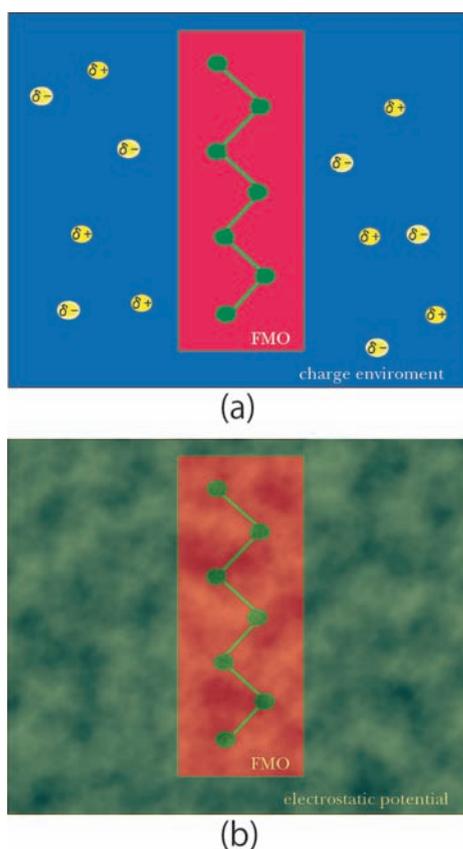


図2 FMO-NMR計算の概念図。周辺の環境の影響を点電荷 (Model-I(a))や電荷密度分布 (Model-II(b))を使って表すことで目的のフラグメント(赤いブロック)の化学シフトが求められる。

## FMO (merged)-NMR 法

# 5

FMO-NMR法で得られた化学シフトの値は、実験の測定値や従来の *ab initio* 計算値をよく再現することができた。一方でわれわれの目標として重原子は1ppmまで、水素原子は0.3ppmの範囲の誤差が理想的だが、全ての原子種において誤差はそれらの基準より大きくなってしまった。より正確な計算結果を得るために、われわれはカットオフ距離というパラメータを用いて、mergedフラグメントという概念を導入した。カットオフ距離パラメータはNMR計算に最適なmergedフラグメントの大きさを決定するためのものである。われわれはその方法をFMO (merged)-NMR法と呼ぶ。

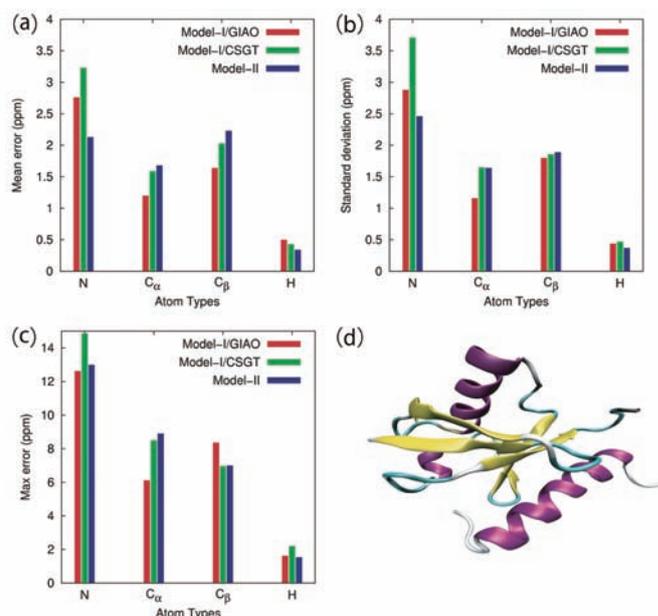


図3 FMO-NMR法を用いて計算したユビキチン(d)の化学シフトの結果と実験結果との比較(a-c)

FMO(merged)-NMR法において、それぞれのフラグメントの点電荷 (Model-I)や電荷密度 (Model-II)はFMOモノマー計算により求められる。そして、目的のモノマーの重心から設定されたカットオフ距離値内にあるすべてのモノマーを集め、mergedフラグメントとする。その後、周辺の化学的環境を含む形でmergedフラグメントのFock行列を求める。最後にそのFock行列を用いて、GIAO計算やCSGT計算を行うことで、目的モノマーの化学シフトを求める。これらの手順を用いることで、mergedフラグメント内における目的モノマーに対する周辺環境の影響を正確に再現することができる。図4にModel-I、Model-IIにおけるFMO(merged)-NMR法を示した。

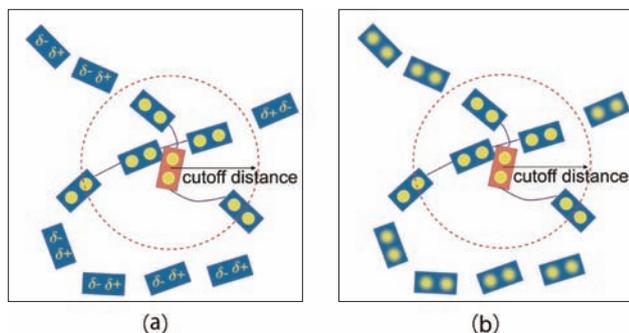


図4 FMO(merged)-NMR計算の概念図。Model-I(a)とModel-II(b)の両方において、目的モノマー(オレンジのブロック)の化学シフトを正しく求めるための最適なフラグメント(mergedフラグメント)の大きさはカットオフを用いて決定している。

FMO(merged)-NMR法の精度を評価するために緑色蛍光タンパク質中の198~229残基からなる一つの $\beta$ シートを用いてテスト計算を行った。カットオフ距離は10Åとした。この値は先行研究により周辺環境の影響を正確に含めるのに十分な範囲であることが示されている<sup>[9]</sup>。

図5において、化学シフトの誤差は従来の*ab initio*計算法による化学シフトからの差として表わされている。FMO(merged)-NMR法を用いて行った計算では全ての原子種の誤差がFMO-NMR法で得られたものより小さくなった。例えば炭素原子( $^{13}\text{C}_\alpha$ や $^{13}\text{C}_\beta$ )の化学シフトの場合、Model-IIにおいてFMO-NMR法によるisotropic遮蔽定数の誤差は、最大で3.88 ppm、平均誤差は0.89 ppmであった<sup>[2]</sup>。一方、FMO(merged)-NMR法では、炭素原子の化学シフトの最大平均誤差と平均誤差の絶対値を減らすことができた(それぞれ0.15 ppm以下と0.05 ppm以下となった)。これらの結果によって、FMO(merged)-NMR法はFMO-NMR法に比べてより正確に目的原子周辺の化学環境を再現できることが示された。

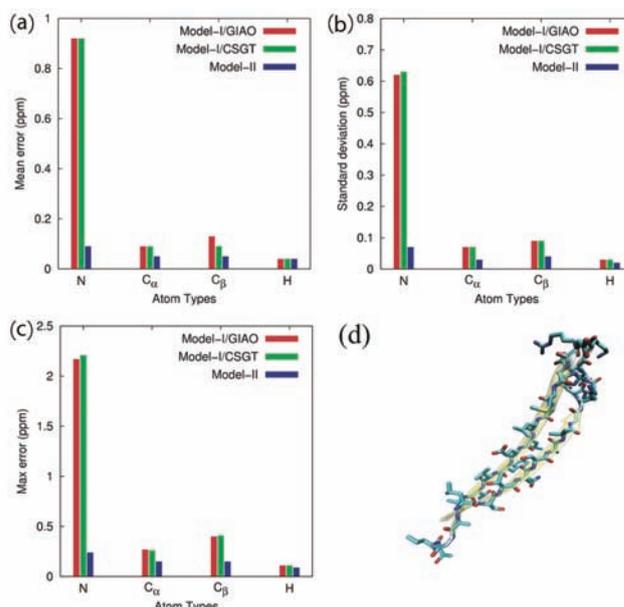


図5 FMO(merged)-NMR法(6-31G(d))を用いた緑色蛍光タンパク質中の一つの $\beta$ シート原子の化学シフトの結果と従来の*ab initio*計算の結果の比較。

## 結論

# 6

われわれの目的は巨大な生体分子系の化学シフトを短時間で正確に予測する*ab initio*計算法の開発であった。計算精度と計算速度は通常は両立しないものであるが、本研究では電子波動関数を得るのにFMO法を導入することによってこれを達成した。われわれは最初に開発した方法をFMO-NMR法と呼んでいるが、ここではフラグメント分割を通常のFMO計算のスキームに従って行っている。従って、基本的にはダイマーフラグメントを用いて、ターゲットアミノ酸の化学シフトを計算する。この方法により、ユビキチンの全原子の化学シフトを計算したところ実験値をよく再現した。しかしながら、 $^{15}\text{N}$ や $^1\text{H}$ 核については、われわれの要求する精度には達していなかった。これは、局所的環境の効果が正確に評価されていないためであると考えられた。そこで、この点を改良するため、FMO(merged)-NMR法を開発した。この方法では、フラグメントサイズをカットオフ距離で定義できるようになっており、ターゲットアミノ酸の化学シフトを精度高く計算するのに最適なサイズのフラグメントを選択できる。この方法をGFPの一部である一つの $\beta$ -シートペプチドに適用したところ、満足すべき結果が得られた。したがって、われわれとしては、今後巨大分子の化学シフト計算法としてはこの方法を推薦する。

量子化学計算は、多くの未知な現象やメカニズムを含むナノレベルの物理現象を解き明かにするのに役立つ。しかし、従来の量子化学計

## フラグメント分子軌道法を用いた 生体分子のNMR化学シフトの計算

算法は計算コストがかかりすぎるため、小さい分子にしか適用することができなかった。本研究では、並列計算に適合するような理論の定式化とTSUBAMEのような強力な計算資源の利用により、そのような欠点が克服できることを示した。今後は、巨大分子の様々な物理的性質が同様な手法により解き明かされるであろう。

### 参考文献

- [1] Gao, Q.; Yokojima, S.; Kohno, T.; Ishida, T.; Fedorov, D. G.; Kitaura, K.; Fujihira, M.; Nakamura, S. *Chem. Phys. Lett.* **2007**, 445, 331.
- [2] Yokojima, S.; Gao, Q.; Nakamura, S. *AIP Conf. Proc.* **2009**, 1102, 164.
- [3] Gao, Q.; Yokojima, S.; Fedorov, D. G.; Kitaura, K.; Sakurai, M.; Nakamura, S. *J. Chem. Theory Comput.* **2010**, 6, 1428.
- [4] Ditchfield, R. *Mol. Phys.* **1974**, 27, 789.
- [5] Keith, T. A.; Bawder, R. F. *Chem. Phys. Lett.* **1993**, 210, 223.
- [6] Fedorov, D. G.; Kitaura, K.; in: Starikov, E. B.; Lewis, J. P.; Tanaka, S. (Eds.), *Modern Methods for Theoretical Physical Chemistry of Biopolymers*, Elsevier, Amsterdam, **2006**.
- [7] Fedorov, D. G.; Kitaura, K. *J. Chem. Phys.* **2004**, 120, 6832.
- [8] Fedorov, D. G.; Kitaura, K. *J. Chem. Phys.* **2004**, 389, 129.
- [9] Sitkoff, D.; Case, D. A. *Prog. NMR Spectr.* **1998**, 32, 165.

# 気液二相流シミュレーションの大規模GPUコンピューティング

青木 尊之\* 杉原 健太\*\*

\*東京工業大学 学術国際情報センター \*\*東京工業大学大学院理工学研究科

水と空気が激しく入り混じる流れは日常的に目にするが、その数値計算は意外に難しく計算量も多い。気液二相流シミュレーションに最新の界面捕獲法、疎行列計算手法、高次移流計算スキームを導入し、高速演算が可能なGPUで実行できるようになった。従来は不可能であった高解像度の砕波計算ができるようになり、小さな気泡の巻き込みも再現できる。また、複数GPUでも十分良い効率で計算できることが示された。

## はじめに

# 1

最近のハリウッド映画では、水と空気が激しく入り混じるような流れのシーンを制作するために流体計算を行い、コンピュータグラフィクスで処理することで、実写できないような映像を作り出している。そこでは科学技術計算より大規模で高精細な計算が行われていることに驚かされる。GPUコンピューティングが当初は重力多体問題での粒子計算で成功したことが影響し、気液二相流などのGPU計算に対しても、SPH(Smoothed Particle Hydrodynamics)法などの粒子法が使われてきた。粒子法は計算したい粒子に着目し、カーネル半径という影響範囲内の全ての粒子と相互作用して及ぼされる力を求め、粒子を移動することで流体運動を計算している。3次元計算になると、低次精度にもかかわらずカーネル半径内の粒子数が増え、ランダムなメモリアクセス、演算量、計算精度の三つの観点から効率が悪くなる。特に半陰解法では圧力のPoisson方程式の非ゼロ要素数が多くなり、疎行列解法の収束性および分散メモリ環境での並列計算という点からも効率が低い。また、気液界面での非物理的振動の発生や単一粒子が界面から離れた場合の精度(スプラッシュに似ているが非なるもの)、強過ぎる数値粘性(渦を維持できない)等の問題も抱えている。

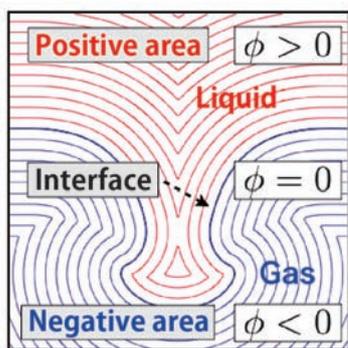


図1 Level Set 関数の等高線表示

\*\* 現・日本原子力研究開発機構

有限差分法、有限体積法、有限要素法などの格子法では、1格子点(要素)の計算が隣接格子点へのアクセスだけで済み、特に有限差分法では高次精度スキームの導入が容易であるため、高精度かつ高効率な計算を行うことができる。ハリウッドの映画制作においても、リアルな水のシーンの計算の殆どは粒子法から格子法に変わってきている。格子法で気液二相流を計算する代表的な計算手法として界面捕獲法がある。気体と液体を密度や粘性などの物性パラメータが異なるだけの同じ流体として統一的に扱い、気体と液体を区別するために識別関数を導入して界面を表現する手法である。また、界面で密度比が1000倍近く変化する流体に対するNavier-Stokes方程式を非圧縮性流体として解くため、圧力のPoisson方程式の係数行列がかなり悪条件になり、従来の反復解法では解きにくいという欠点がある。

本稿では、格子法による気液二相流計算の全ての部分をGPUコンピューティングで実装し、CPU計算では困難であった高解像度・大規模気液二相流の計算を実行する。

## 気液界面の捕獲法

# 2

気液界面は2次元面であるが、これを表現するために3次元空間の関数の断面を利用する手法がよく使われる。3次元の識別関数を用いると情報量としては冗長に思えるが、気泡の合体・分離などのトポロジー変化を容易に表現することができるという利点がある。代表的な識別関数としてLevel Set関数を使う方法<sup>[1]</sup>や気体・液体の流体の体積率分布を用いるVOF(Volume of Fluid)法<sup>[2]</sup>等が知られている。Level Set関数は界面からの符号付距離関数であり、例えば図1のように液体側の領域は界面からの距離に正の符号を付け、気体側の場合は距離に負の符号を付ける。こうすることにより界面はLevel Set関数のゼロの等値面として表現される。識別関数はなめらかなプロファイルになるため、曲率等を精度よく求めることができる。Level Set法(特にParticle Level Set法)は界面形状を表現する精度が高いが、本来保存されるべき気体・液体の体積保存が保証されず小さな気泡や液滴が消失する可能性がある。一方、VOF法は気体・液体のそれぞれの体積を保存するが、格子サイズに近い曲率半径を持つ気

液界面形状を精度よく表現することができない。この両者を十分満足させる界面捕獲手法はまだ開発されていない。ここでは、THINC<sup>[3]</sup> WLIC法<sup>[4]</sup>というVOF法をベースにし、気液界面の広がりを阻止するアンチ拡散性が入った手法で界面を捕獲し、表面張力や接触角を評価するためだけにLevel Set 法を導入している。

表面張力および壁との接触角についてはBrackbillのCSF (Continuous Surface Force) モデルにより有限幅に力を分散させて計算している。

## 非圧縮性 Navior-Stokes 方程式の計算

# 3

### 3-1 移流計算

Navior-Stokes方程式の移流項の計算とLevel Set 関数の再初期化計算には、5次のWENO スキームを用いる。単調性の確保が高波数成分をフィルタリングし、安定性の確保に貢献している。5次WENOスキームのステンシル・アクセスは図2のようになり、NVIDIA GPUのshared メモリをSoftware Managed Cacheとして使うことにより (FermiコアのGPUではL1キャッシュが導入されたため、このようにsharedメモリを使う必要性は低下した) ビデオ・メモリへのアクセスを低減できる。この部分だけなら1GPUで300 GFLOPSを超える高い実行性能が得られている。

### 3-2 単一気泡の上昇の検証

気液二相流計算において最も計算時間を要する部分がPoisson方程式の計算である。ここでは構造格子を用いているため、有限差分法で離散化した連立一次方程式の係数行列は非ゼロ要素の位置は規則的であるが、気液を统一的に解くために界面に急峻な密度変化が生じ、かなり悪条件の疎行列となる。そこで、クリロフ部分空間での反復法であるBiCGSTAB法をVサイクルのマルチグリッド法(図3)による前処理と組み合わせた収束性の高い疎行列解法をみずほ情報総研と共同研究し、GPU用のライブラリとして開発している。マルチグリッド法は大規模並列計算にも適用可能なアルゴリズムであり、Red & Black法を用いたILU (0) 法をスムーザとして用いることで安定した収束性を確保している。

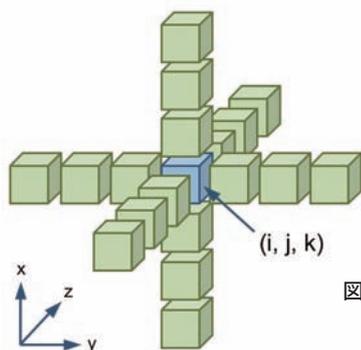


図2 5次WENOスキームのステンシル

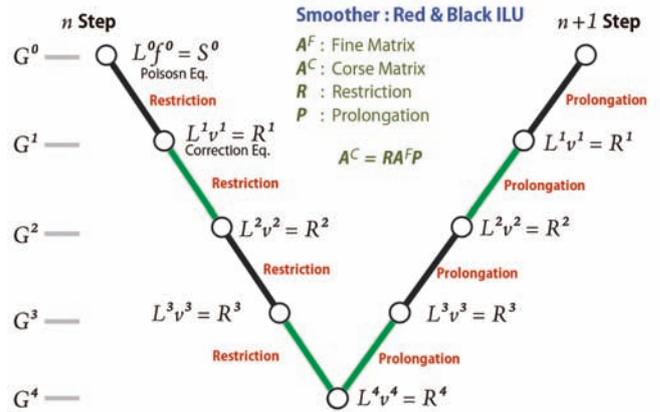


図3 マルチグリッド前処理のVサイクル

## 単一気泡の上昇の検証

# 4

気液二相流計算の基本的な検証として単一気泡の上昇を計算した。Graceダイアグラム<sup>[5]</sup>によると、単一気泡の上昇は無次元のEotvos数、Morton数、Reynolds数に応じて球型、楕円型、スカート型、くぼみ付き楕円型に分類される。その典型的なパラメータに対して得られた計算結果を図4に示す。定常の上昇速度になったときの気泡形状はGraceダイアグラムの分類と良く一致しており、さらに上昇速度は実験値<sup>[6]</sup>と非常によく一致している。

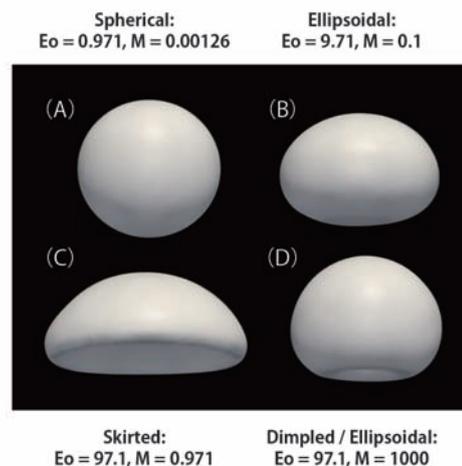


図4 無次元パラメータの違いによる上昇する単一気泡の形状変化

ミルククラウンの検証

5

静止している比較的浅い液面に液滴を落下すると、ミルククラウンが形成されることは良く知られている。フィンガーがでるメカニズムや本数などは未だに議論が続いているが、実際と同じ粘性、表面張力の下で、実験と同じ速度で液滴を落下させミルククラウンの形成を計算した。落下速度や液の厚さに応じてミルククラウンの形成の様子が大きく変化するが、図5に示すように実験と非常に良く一致する数値計算結果が得られた。



図5 ミルククラウンの形成過程

また、乾いた床にミルクを滴下すると、図6のように広がった液膜の先端が床から跳ね上がる様子も良く捉えられている。

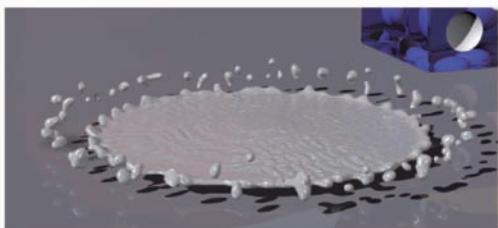


図6 ミルクの乾いた床への滴下

ダムブレイクの濡れた床への浸水

6

複雑な気液二相流としてダムブレイクの計算を行い、九州大学・応用力学研究所のグループとの共同研究で行った実験との比較も行った。通常はプレートで仕切り、溜めておいた水を乾いた床に浸水させ、その先端の速度などを実験と比較するが、ここでは濡れた床へ水を浸水させる計算と実験を行った。濡れた床の場合、浸水を塞ぎ止める効果が大きいため、後から押し寄せる水の速度が先端速度を上回り浸水直後から碎波が起こり先端は大きく乱れる。

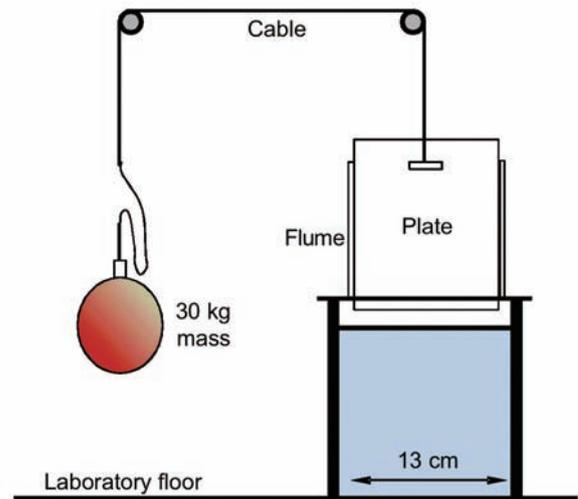
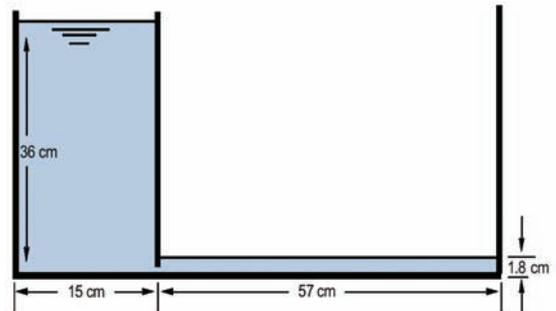


図7 ダムブレイク実験のセットアップ

計算条件は72cm×13cm×36cmの計算領域に1.8cmの浅い水面を設定し、初期に幅15cm、高さ36cmの水柱を置く。気相および液相の物性値はそれぞれ空気、水の値を用いた。576×96×288格子を用いて計算した結果を図8の最初の3つに示す。最後の1つが実験のスナップショットである。

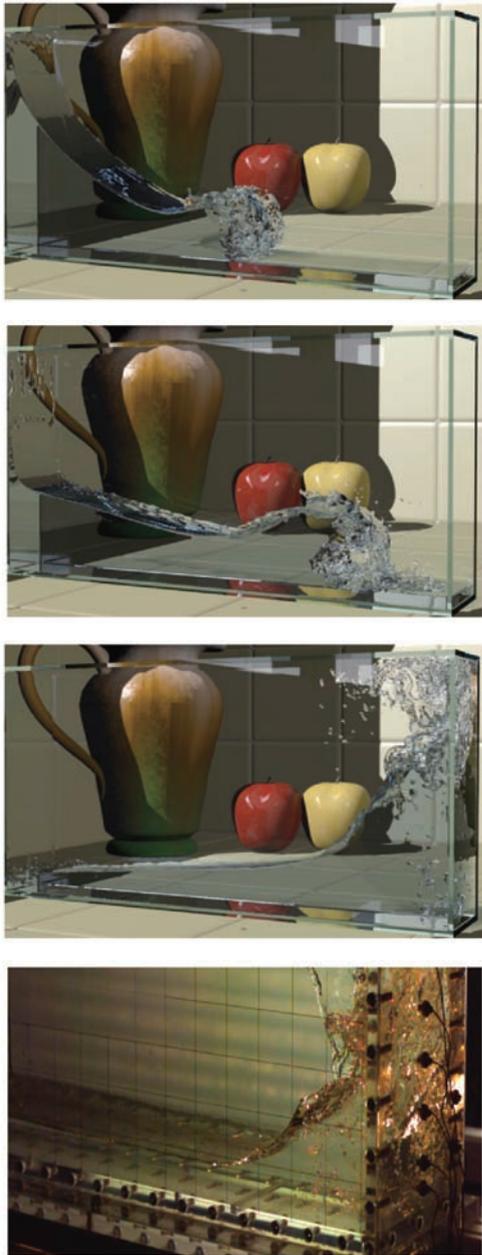


図8 ダムブレイク計算と実験の比較

計算と実験の比較は水槽のサイズ等が必ずしも同一条件ではないが、計算は碎波の過程を良く再現している。また、壁との衝突により小さな気泡が水中に巻き込まれている様子も良く捉えられている。碎波の水面形状は初期の水柱高さと濡れた床面の水深の比が大きいほど噴流の角度が浅く、進行方向に大きく巻き込む<sup>[7]</sup>。

実験では、壁への衝撃圧を高感度センサーで測定しており、今後、定量的な比較・検討を行う予定であり、空気の巻き込みが重要であることが示唆されている。遡上する津波が建物に及ぼす影響の評価にも利用でき、震災の復興支援にも貢献できるものと思われる。

## 気液二相流の複数 GPU による計算

# 7

本計算では、気液二相流計算のためのコンポーネントを全てGPU計算として実装し、速度、圧力、Level Set 関数、VOF関数といった従属変数をGPU上のメモリに置いている。CPU側からはGPU計算を実行するカーネル関数をcall(実行命令)するだけとなり、CPUとGPUの頻繁なデータ交換を排除した。これによりGPU本来の演算性能、メモリバンド幅を有効に利用して実行することができ、単一CPUコア(シングルCPUスレッド)に対して単一GPUを用いて数10倍の実行性能を達成している。

大規模計算に対しては、複数のGPUを用いて各GPUのビデオ・メモリのサイズで計算可能になるように領域分割を行い、各GPUは割り当てられた領域だけを計算する。CPU計算と同じように分割領域間のデータ通信が必要になり、図9のようにGPUからCPU上のメモリを介してMPIライブラリを使う3ステップからなるデータ通信を行う。GPUスパコンでは、しばしばノードの演算性能に対してノード間のインターコネクションの通信性能が不足するため、GPU数を増やすにつれて通信時間が大きなオーバーヘッドになり、計算と通信をオーバーラップすることにより通信時間を隠ぺいするような工夫が必須となる<sup>[8]</sup>。

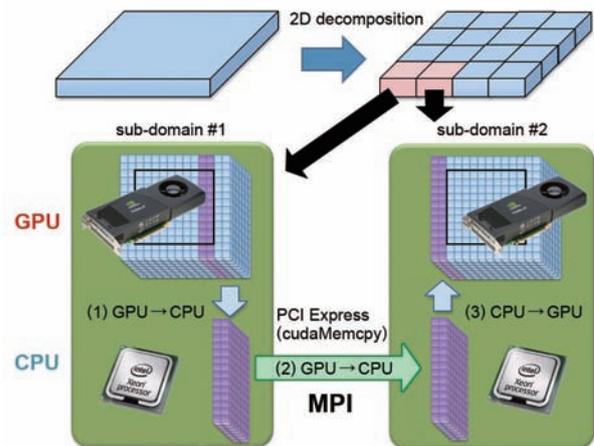


図9 GPU計算におけるノード間通信

ここではTesla S1070が搭載されていた時期のTSUBAME 1.2の60ノード(120GPU:2GPU/node)のうち1個~108個のGPUを用いて計算を行った。計算格子 $192^3, 384^3, 768^3$ のサイズの二相流計算に対し、複数GPUを用いた計算の実行性能を図10に示す。よい並列スケールングが示されていて、 $768^3$ 格子のケースでは108GPUを用いて4 TFLOPSの実行性能を達成している<sup>[9]</sup>。

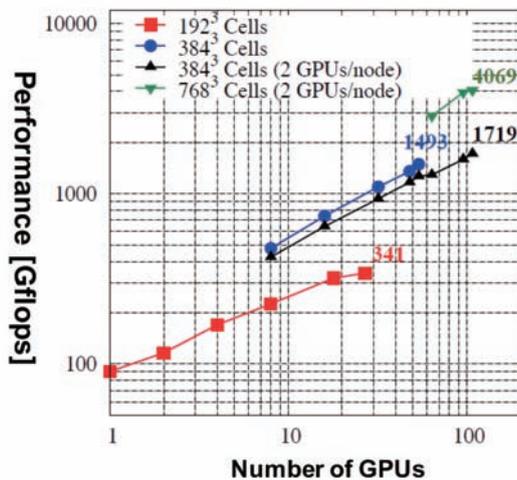


図10 二相流計算における複数GPUの実行性能

## おわりに

# 8

数値流体力学の中では、難しいとされる気液二相流計算がフルGPUで計算可能になったため、さらなる大規模計算が可能になった。今後は浮遊物体等との相互作用を含んだ計算に期待が寄せられている。また、いくら高解像度計算が可能になったとしても、乱流状態の二相流に対して、格子解像度では捉えきれない小さな気泡などに対してはLES (Large-Eddy Simulation) のモデル化を導入する必要がある。まだまだ課題は多い。

### 謝辞

本研究の一部は科学研究費補助金・基盤研究(B)課題番号23360046「GPUスパコンによる気液二相流と物体の相互作用の超大規模シミュレーション」、科学技術振興機構CREST「次世代テクノロジーのモデル化・最適化による低消費電力ハイパフォーマンス」および「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」、日本学術振興会(JSPS)グローバルCOEプログラム「計算世界観の深化と展開」(CompView)から支援を頂いた。記して謝意を表す。

### 参考文献

[1] M. Sussman, P. Smereka and S. Osher: A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, J. Comp. Phys., Vol. 114, pp.146-159 (1994)  
 [2] C. W. Hirt, et al.: SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries, Los Alamos Scientific Laboratory, pp.1-32 (1980)

[3] F. Xiao, Y. Honma and T. Kono: A simple algebraic interface capturing scheme using hyperbolic tangent function, Int. J. Numer. Method. Fluid., Vol. 48, pp.1023-1040 (2005)  
 [4] K. Yokoi: Efficient implementation of THINC scheme: A simple and practical smoothed VOF algorithm, J. Comp. Phys., Vol. 226, pp.1985-2002 (2007)  
 [5] J.R.Grace: Transactions of the Institution of Chemical Engineers, Vol. 51, pp.116-120 (1973)  
 [6] M. van Sint Annaland, N.G.Deen and J.A.M.Kuipers: Chemical Engineering Science, Vol. 60, pp.2999-3011 (2005)  
 [7] Takayuki Aoki and Kenta Sugihara: Two-Phase Flow Simulation on GPU cluster using an MG Preconditioned Sparse Matrix Solver, SIAM conference on Computational Science and Engineering 2011, Reno, Nevada (2011)  
 [8] T.Shimokawabe, T.Aoki, C.Muroi, J.Ishida, K.Kawano, T.Endo, A.Nukada, N.Maruyama, S.Matsuoka, "An80-fold speedup, 15.0 TFlops full GPU acceleration of non-hydrostatic weather model ASUCA production code" in Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC'10, IEEE Computer Society, New Orleans, LA, USA (2010)  
 [9] 杉原 健太, 青木 尊之: 大規模高次精度移流計算の複数GPUによる高速化と強スケーラビリティ, 日本計算工学会誌, Transactions of JSCES, Paper No.20100018 (2010)

# 大規模並列計算システムを用いた 進化的画像・映像符号化

高村誠之

日本電信電話株式会社 NTTサイバースペース研究所

概要: 遺伝的プログラミング (Genetic Programming, GP) に基づいた進化的手法による動的なアルゴリズム生成は、プラント制御、ロボット制御、株価予測など多くの分野に応用されている。

一方、JPEGやMPEG-2, MPEG-4, AVC/H.264等の従来の画像・映像符号化アルゴリズムは例外なく固定的な(動的でない)ものであった。この制約を突破することを目標とし、GPを画像ごとに個別に用い画素値予測アルゴリズムを生成する試みがなされ、一定の成功を収めている。しかしながらその課題の一つが演算量の膨大さであった。

本稿ではGPに基づく画像特化型符号化アルゴリズムを紹介し、その並列計算との整合性の高さを述べ、いくつかの高速化の試み、特にTSUBAME2.0やGPGPUを用いた大規模並列計算システムでの高速化について述べる。

## はじめに

# 1

JPEGやAVC/H.264のような画像・映像符号化方式において予測モードや量子化パラメータ、予測のための係数群、コンテキスト分類閾値等は適応的に選択され、符号化効率を最適化している。これら従来の画像・映像符号化方式においては、そうした「符号化パラメータ」は動的に最適化されるものの、「符号化アルゴリズム」自体は方式ごとに固定であった。換言すれば、新たな符号化の枠組みは人間が試行錯誤の末生み出し、かつ人間が実装するしかなかった。したがって、符号化コーデックの複雑さは人間が記述しうる程度を超えることはできなかった。さらに、画像・映像毎に最適なアルゴリズムを個別に開発することは現実的ではなかった。そこで、計算機により画像・映像毎に最適な符号化アルゴリズムを開発する方法を確立することを目的とした「進化的画像・映像符号化」が検討されている。本稿ではその検討内容をいくつか紹介する。

## 進化的手法と性能

# 2

遺伝的アルゴリズム (Genetic Algorithm, GA) は最適化問題を解く一手法である。GAは「解のパラメータ」を一次元配列(個体の遺伝子)とみなし、適者生存に基づく世代交代を繰り返すことによって実用解を得る探索手法であり、遺伝的プログラミング (Genetic Programming, GP)<sup>[1]</sup>は遺伝子を木構造にすることで「解の手順」の最適化を行えるよう拡張した手法である。GA, GPは「進化的手法」と呼ばれ、プラント制御やロボット制御、アナログ回路設計等広範に応用されている。

従来の進化的手法の画像・映像符号化応用例として、田中らは二値画像符号化におけるコンテキストテンプレートの最適化を行った<sup>[2]</sup>。高木ら<sup>[3]</sup>は映像符号化における分割領域を最適化した。これらはGAをパラメータ最適化に利用したものであり、符号化アルゴリズム自体は固定である。

### 2-1 アルゴリズム最適化手法

GPを利用することで動的なアルゴリズムを生成でき、プラント制御など様々な分野で応用されている。GPを用い画像ごとに画素予測手順を自動生成する方法は次のようになる<sup>[4,5]</sup>。

まず、予測アルゴリズムを木により表現する。木の葉には即値や周辺の復号済み画素値などが、木の枝には加減乗除やsin, cos等の算術関数や条件分岐が配置される。次いで母集団を、乱数で生成した木やMED(JPEG-LS<sup>[6]</sup>の用いる予測器)、GAP(CALIC<sup>[7]</sup>の用いる予測器)等から生成する。次いで複製選択(親の選択)、子の生成(交叉・突然変異)、生存選択により予測器を進化させる。世代交代モデルにはminimal generation gap<sup>[8]</sup>を用いた。

生存選択のための評価尺度として、二つの情報量の和(IT+IR [bits])を用いる。ITは木を表現するための情報量であり、全ノードの情報量の和となる。IRはその木が示す手順により画素値を予測した残差の情報量である。

## 2-2 予測性能

実験では4つの512x512, 8bit, グレースケール画像 (Lena, Baboon, Airplane, Peppers) を用いた。比較方式は前述のMEDやGAP、最小二乗予測器 (LS)、予測残差エントロピーが最小となるよう線形予測係数をオフラインで多次元探索した予測器 (LE) および本進化的手法による予測器 (EP) である。用いる復号済み画素値の個数は4,12とし、EP(4), LS(12) などのように表記する。予測残差はすべてCALICと同じコンテキスト分離<sup>[7]</sup>を施したのちエントロピーを計測した。結果を表1に示す。進化的予測器EPの予測性能の高さがわかる。特にEP(4), LS(12) を性能で上回っているのは、非線形予測が線形予測の限界を超える結果と言え、興味深い。

EP(12) の木サイズ(IT+IR) の平均は2062.2bitsであった。生成予測器の例としてPeppersのものを図1に示す。人手による生成は困難なほど多数の複雑な数式となっている。

Predictor	Lena	Baboon	Airplane	Peppers	Avg.(incr.)
LS(4)	4.551	5.521	3.654	4.465	4.548(5.6%)
LS(12)	4.549	5.374	3.635	4.410	4.492(4.3%)
LE(4)	4.529	5.506	3.619	4.417	4.517(4.9%)
LE(12)	4.522	5.361	3.595	4.382	4.465(3.6%)
GAP(4)	4.539	5.556	3.568	4.468	4.533(5.2%)
MED(3)	4.692	5.592	3.644	4.646	4.643(7.9%)
EP4	4.481	5.462	3.521	4.352	4.454(3.4%)
EP12	4.385	5.175	3.411	4.262	4.308

表1 残差エントロピー (付加情報込)およびEP12からの増分("INCR"[%]). 予測器の次の数字は参照画素数

```

Peppers: if  $D - I_{10} \geq 0$  then  $I_{gap} \leftarrow \text{Pred0}$ 
else if  $-D - 31.1797 \geq 0$  then  $0.50196[\text{xor}(\min(\max(I_{03}, 0.578125(D + I_{00} + I_{02})), |I_{07} - I - \max(I_{00}, \text{if } x \geq 0 \text{ then } \rho \text{ else } \min(90.7501, I_{01})) + 2.5939|, I_{01}) + 0.50196I_{gap} \leftarrow \text{Pred1}$ 
else if  $-\sqrt{|I_{11} - I_{02}|} - \sqrt{|I_{10} - I|} + \min(|D| - I_{15}/2 - 75.91195|, I_{07}, I_{09}/2 + I_{00}/2) + \theta - \max(I_{03}, I_{00}, I_{07}) + 29.498 \geq 0$  then  $0.75147|0.28107|0.54697|0.24244I_{09} - 0.70967||I_{04} - I_{00} - I_{15} - I_{02} + I_{15}| + 0.18145||I_{07} - I_{04} + I_{00} - I_{05} + I_{03}| - 0.23255I_{04} + 1.59610| + 0.09227I_{06} - 0.617188I_{00} - 0.24171I_{03}| + 0.21722I_{03} + 0.25049I_{01} \leftarrow \text{Pred2}$ 
else or  $(\min(I_{gap}, \max(\min(I_{02}, |I + y + \min(I_{01}, I_{03}, |I + I_{15}| - I_{06}) - I_{08}, I_{10}^2), I_{09})), |I_{07} - |\tan.x| - I_{00} - I_{02} - I_{15}|/2) / 2 + I_{gap}/2 \leftarrow \text{Pred3}$ 

```

図1 画像Peppers専用に生成された予測器(EP12)と、その条件分岐構造。lxxは周辺の復号済み画素値 (詳細は<sup>[5]</sup>参照)

## 2-3 評価基準の調整

前節では、木の評価基準は「木サイズと残差情報量の和 (IT+IR)」であった。GPにより最小化すべきこの評価基準を調整することで、新たに好ましい効果を得ることができる。

- (a) 高速コーデックの生成評価基準に符号化時間(EncTime)と復号時間(DecTime)の項を導入する:

$$IT + IR + \lambda_1 \text{EncTime} + \lambda_2 \text{DecTime}$$

$\lambda_1$ と $\lambda_2$ を調整することで、ある程度の符号化効率を維持しながら高速な符号化器あるいは高速な復号器が生成できる<sup>[9]</sup>。

- (b) 木サイズへのペナルティ

上で述べた評価基準には木のサイズITも含まれているため、木の過度の肥大化(bloatと呼ばれる)はある程度抑制されている。しかしながら、時として大きなサイズの木が生存する可能性は残る。大きなサイズの木の変異の可能性は幅広いため、例えば符号化効率がほぼ等しい大小二つのサイズの木があった場合、小さな木よりも大きな木を進化させ効率化の方が難しい。

そこで評価基準をIT+IRでなく、ITへの重み(w)をつけることを考える<sup>[10]</sup>。

新たな評価基準は

$$w \text{ IT} + \text{IR}$$

となる。

結果を図2に示す。木サイズへのペナルティを与えることで、期待された通り進化の高速化およびよりコンパクトな木の生成が図れている。

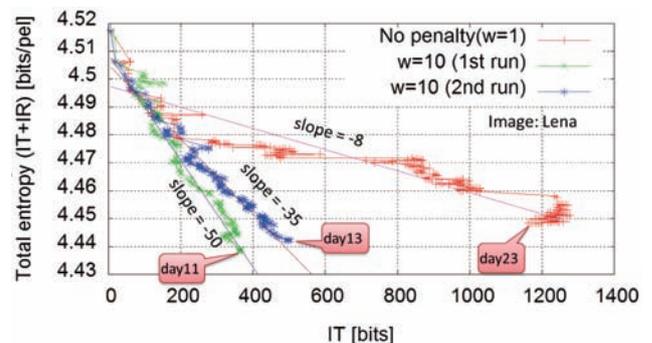


図2 木サイズへのペナルティの違いによる木の成長 (左上から右下へ)の違い

### 並列化による進化の高速化

# 3

進化計算をシングルプロセスで行う場合、高速化手段として(GPの)ツリー評価の能率化や高速プロセッサの利用などが考えられるが、期待できる高速化は高々2-3倍のオーダに留まる。

そこで並列動作向けに進化計算のプログラムフローを変更する。すなわち、共有ファイルに全体最良個体を記録し、母集団が一代代経る毎(10秒程度)に、そのタイムスタンプを確認し、変化があった場合(他プロセスによる記録更新があった場合。数十回に一回程度の頻度)、自己の最良個体へ取り込みあるいは共有ファイルの全体最良個体へ上書きする(サイズは数百バイト程度)。こうして複数プロセス間で全体最良個体が共有される。またファイルI/O負荷は無視できる程度である。

しかる後、図3に示すような複数プロセスによる進化計算を行う。ファイル共有によるプロセス間通信のため、ソースコードも実行環境も、OSに依存しない。またプロセス間は非同期に実行できるため、プロセスの途中離脱・途中追加が随時可能であり、かつスケラビリティが1~∞となる。またプロセス間の同期をとらないので各コアのloadは常に100%となる。さらに各プロセッサ/コアは均質である必要もない。

#### 3-1 大規模並列クラスタTSUBAMEによる高速化

東京工業大学学術国際情報センターのクラスタ型スーパーコンピュータTSUBAME<sup>[11]</sup>は、1ノードあたりAMD Dual-Core Opteron(2.4GHz)、8CPU、32GB MemoryのスペックのPCの集合体である。OSはSUSE Linux Enterprise Server 10(x86\_64) Patchlevel 2である。

この1万個以上のコアのうち約1,500コアを実験に用いた。また、2010年末にTSUBAME2.0に更新され、thinノードはIntel Xeon X5670 2.93GHz、各ノード12コア、OSはSUSE Linux Enterprise Server 11(x86\_64) SP1となった。この約1.5万コアのうち2,400コアを実験に用いた。

図4, 5は並列度による進化速度の違いを示したものである。並列化の効果は明白である。進化速度を定量化するために、総符号量が4.407bits/pelに到達するまでの所要時間を求め、表2に示す。右の列がシングルコアに対する速度向上割合である。コア数と速度向上割合の関係を図6に示す。両者の間にはほぼ線形な関係が観察され、並列化の効果が確認された。

#### 3-2 GPUによる高速化

近年のGPUハードウェアの進歩は著しく、並列化可能なアプリケーションの性能を著しく高めている。画素値予測は各ピクセルを独立に予測するため、GPUなどの並列化において理想的な応用先と考えられる。

我々は進化処理をCUDAプラットフォームに実装した<sup>[12]</sup>。デュアルXeon X5670 2.93 GHz(計12コア)での実装と、NVIDIA Tesla C2050(448 CUDAコア)を4個搭載するグラフィックカードにて実験を行った。OSはUbuntu Server Edition 10.04(64-bit)、コンパイラはGCC4.4である。1CPUコアと1Tesla C2050とで、木の評価速度の比較を行った。結果を表3に示す。用いた様々な予測器のサイズを通し、GPUにより約140倍の高速化が実現できた。

画像Lenaに対し、4.493 bits/pelに到達するまでの時間を比較するとCPUで18.86時間、GPUで217秒であった。この場合はGPUが312倍高速であったことになる。時間と総情報量推移の様子を図7に示す。

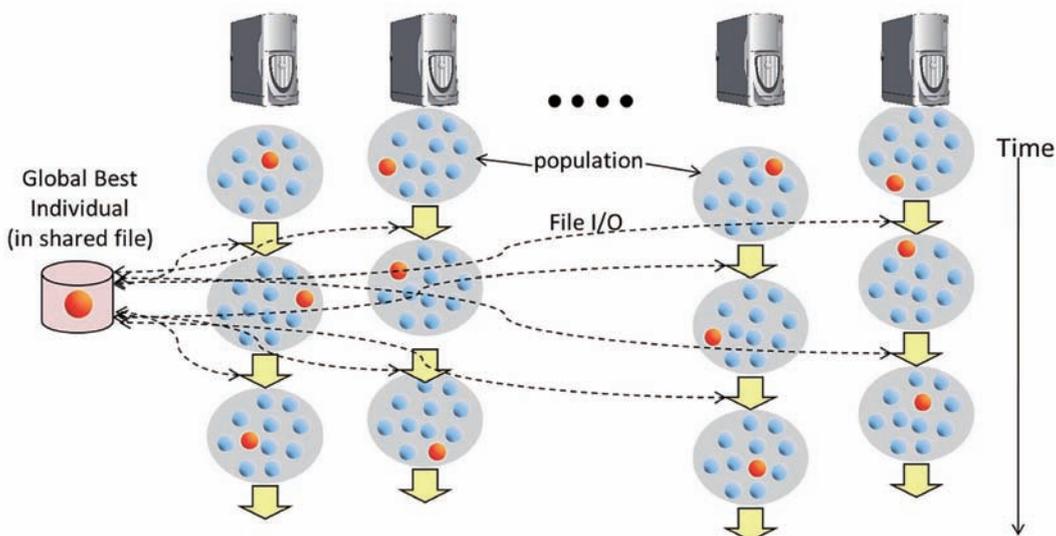


図3 非同期並列進化の実行イメージ(時間は上から下へ、楕円が母集団、球が個体、赤い球が最優良個体)

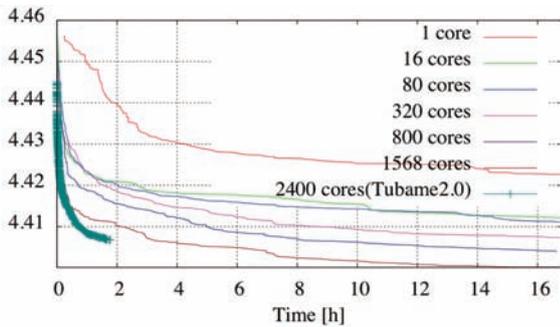


図4 並列度ごとの符号量の時間変化の比較

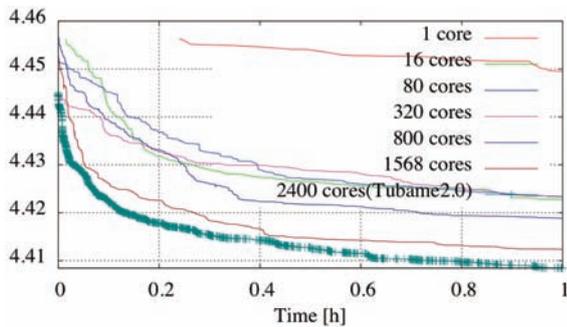


図5 並列度ごとの符号量の時間変化の比較

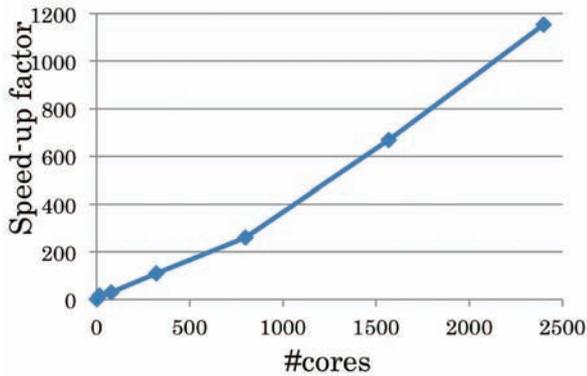


図6 コア数と速度向上度の関係

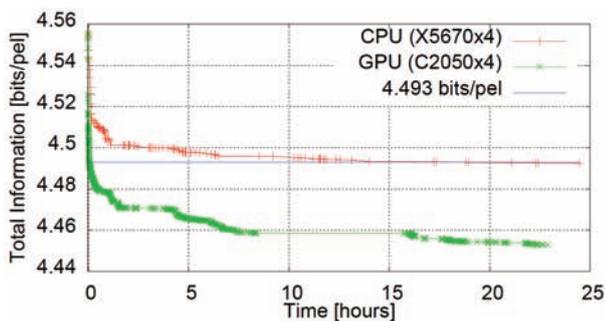


図7 Lena用予測器の性能の時間変化。赤がCPU、緑がGPU。

#cores	Time (hours)	Speed-up factor (vs. one thread)
1	2075.3	1
16	129.7	16.0
80	67.78	30.6
320	19.02	109.1
800	7.98	260.1
1,568	3.10	669.4
2,400	1.80	1152.9

表2 4.407bits/pelに到達するまでの所要計算時間と速度向上度

predictor size [nodes] / info. (IT) [bits]	CPU [msec]	GPU [msec]	speed-up factor (vs. CPU)
240 / 1631	1195	8.0277	148.8x
345 / 2251	1882	13.429	140.2x
422 / 2719	2311	16.564	139.5x
469 / 3079	2692	18.724	143.8x

表3 画像LENAに対する画素値予測時間とシングルコア比速度向上度

### 3-3 クラス分類による高速化

進化的手法により得られた予測器には条件分岐が含まれることがある。例えば Peppers において生成された予測器は、以下のような構造をもっていた (図1 参照) :

```

for each pixel
if (Condition0) Pred0
else if (Condition1) Pred1
else if (Condition2) Pred2
else Pred3

```

含まれている条件に従い、予測器 (Pred0...3) は画素ごとに使い分けられている。予測器の分布を図8に示す。性能を高めるため、エッジの方向・強度により予測器を使い分けるよう進化していることがわかる。

この進化挙動にヒントを得て、復号済み画素からエッジ方向を推定し9通りに画素を分類し (図9 参照) する。画像 Peppers の分類結果を図10に示す。この分類結果に対応して9個の予測器を独立に進化させた<sup>[13]</sup>。特定のビットレートに至るまでの時間を表4に示す。従来比べ25.4-344.0倍(平均180倍)ほど進化が高速化した。図11にPeppersの進化高速化の様子を示す。

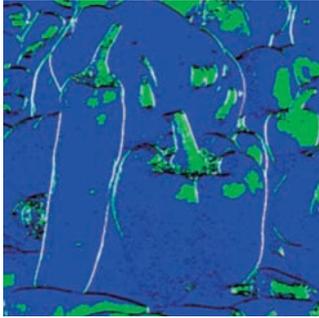


図8 Peppersの予測器分布。白,青,緑,黒の各色が4予測器 (Pred0...Pred3) のいずれかに対応

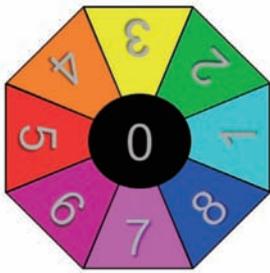


図9 エッジ方向に応じたクラス分類(0は平坦部)

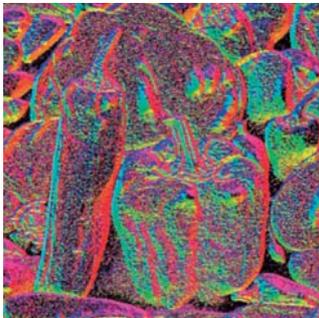


図10 Peppersの各画素を分類した結果 (色は図9に対応)

Image	Target bit rate [bits/pel]	Processing time [mins]		Speed-up factor	Predictor size (IT) [bits]	
		Conventional	Proposed		Conventional	Proposed
Lena	4.435	3638.5	21.3	170.8x	691.6	807.9
Peppers	4.263	4093.9	11.9	344.0x	1628.7	1094.7
Baboon	5.200	953.6	37.5	25.4x	549.4	1452.6
Avg.				180.1x	956.6	1118.4

表4 特定のビットレートに至るまでの時間と、得られた予測器規模の比較

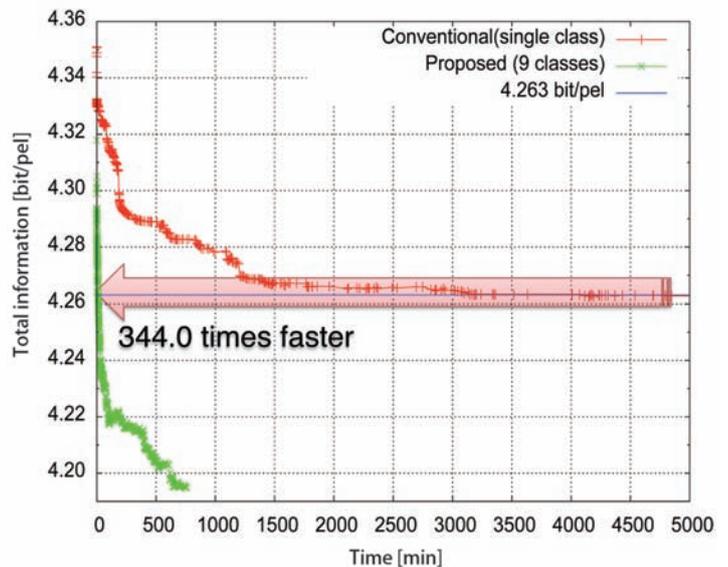


図11 進化予測器の性能遷移。赤: 従来進化方法、緑: 画素分類を用いた進化方法 (画像はPeppers)

## 今後の展望

# 4

### 4-1 究極の圧縮率

図2において、木の規模と符号化性能が直線的な関係にあるという実験事実は非常に興味深い。この傾向は他の画像についての試行でも同様に見られた。そこで

$$IT + IR = rate + k IT$$

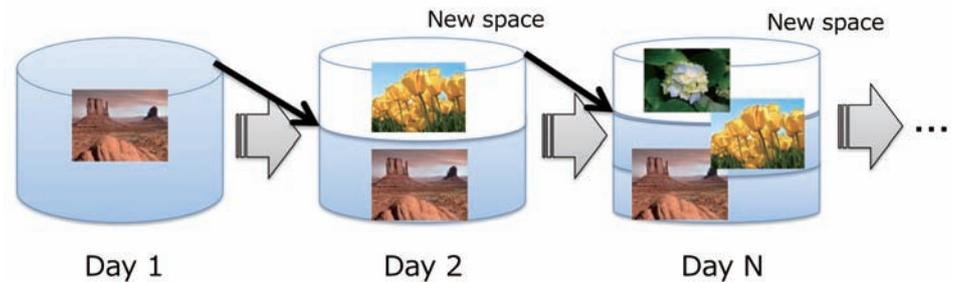
但し  $k = -50 \dots -35$  (図2の傾きに対応)、という予想が立てられる。8bit 画像の典型的な可逆符号化では  $rate \doteq 4$  bits/pel (原画像サイズの約半分) である。

IT, IR は非負であるので、上記の関係式から、 $IR=0$  のとき (予測が完全に当たり、予測残差が全て0となる、つまり予測器そのものが画像情報を含むと考えられる状態のとき) の圧縮率が究極となり、それは次の値となる:

$$IT + IR (= IT) = rate / (1 - k).$$

可逆符号化においてこの符号量 (原画像サイズの1%程度) はあまりに低い。この理由は画像に元々含まれている雑音 (元来圧縮できない信号) の影響を考慮していないためである。それであっても、雑音の情報量を若干超える程度までは圧縮ができるであろうこと (その際は予測残差が無相関雑音となる) が予想できる。

図13 自己圧縮アーカイブのイメージ



#### 4-2 画像のリバースエンジニアリング

上記の議論の敷衍から、究極の予測器は画像に隠れているモデルを示唆しているかもしれないという考えに至る。簡単な例を用い、GPがモデルを推定する能力を持つかどうかを確認する実験を行った。まず図12に示すように、単純な数学モデルを三種用意し、対応する画像を生成した。

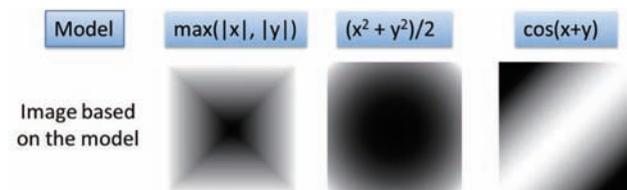


図12 単純なモデルと対応する画像

次いで、画像のみを進化予測生成器に与え(モデルは与えない)、予測器を生成させた。結果はそれぞれ次のようになった:

$$\max(|y|, |x|), |(x^2+y^2)/2|, \cos(x+y)$$

これらはすべて初期モデルと数学的に等価である。言い換えれば、進化的手法により画像のリバースエンジニアリングができる可能性が示されたことになる。

#### 4-3 自己圧縮アーカイブ

進化的画像・映像符号化の応用先の一つとして、画像・映像アーカイブが挙げられる。クラウド等からの余剰計算資源を利用し、アーカイブ自身が符号化アルゴリズムを日夜更新し続ける。画像・映像ファイルのサイズは減り続け、新たな画像・映像をアーカイブする余地が作られていく(図13)。

#### 4-4 カテゴリ特化型符号化

ここまで、常に画像に特化した符号化アルゴリズムを議論してきたが、あるカテゴリの画像に適し、適度に良好な性能を持つ「カテゴリ特化型」符号化アルゴリズムもあると便利である。

3種類のカテゴリ(雲、街の夜景、自然)の画像(各10枚、サイズ128x128, 8bit グレyscale)計30枚を用意し、アンカーとして全

30枚の符号量を最小化する符号化器を生成した。同様に、カテゴリ毎および画像毎に符号量を最小化する符号化器も生成した<sup>[14]</sup>。

category	anchor	for category	for image	JPEG2000
clouds	2.925	2.884(-1.4%)	2.887(-1.3%)	3.142(6.9%)
city lights	5.507	5.496(-0.2%)	5.499(-0.2%)	6.029(8.7%)
nature	6.406	6.387(-0.3%)	6.389(-0.3%)	6.846(6.4%)

表5 符号量およびアンカー比符号量増分

アンカーおよびカテゴリ特化型は、アルゴリズムを復号器でも既知とできるため伝送しなくてよい。表5に実験結果を示す。カテゴリ特化型、画像特化型、アンカー、JPEG2000可逆符号化の順で符号化性能が高いという結果が得られ、カテゴリ特化型の有効性が確認できた。またいずれもJPEG2000可逆符号化よりも6-8%程度符号量が少なかった。

#### 4-5 非可逆符号化への応用

進化的画像・映像符号化技術は、ここまでみてきた可逆符号化以外にも応用ができる。GPにより画像特化型非線形画像復元フィルタを生成することで、フィルタ情報量を加味しても、現在標準化策定中の次世代映像符号化方式であるHEVCの高効率モード比で0.9%ほど符号量を削減できることが報告されている<sup>[15]</sup>。

## おわりに

# 5

本稿では、進化的画像・映像符号化の概念、特性、ポテンシャルおよび将来展望について概説した。特筆すべきは並列計算との親和性の高さであり、TSUBAME/TSUBAME2.0等のHigh-Performance Computingプラットフォームの利用は、この実用性を大いに高めるものと期待できる。

参考文献

- [1] J. Koza, "Genetic Programming II, Automatic Discovery of Reusable Programs," The MIT Press, (1998)
- [2] M. Tanaka, H. Sakanashi, M. Mizoguchi, T. Higuchi, "Bi-level image coding for digital printing using genetic algorithm," Electronics and Communications in Japan part III: Fundamental Electronic Science, vol. 84, no. 9, pp. 1-10, (Apr. 2001)
- [3] K. Takagi, A. Koike, S. Matsumoto and H. Yamamoto, "Motion Picture Coding Based on Region Segmentation Using Genetic Algorithm," Systems and Computers in Japan, vol. 33, no. 5, pp. 41-50, (Mar. 2002)
- [4] S. Takamura, M. Matsumura and Y. Yashima, "Automatic Pixel Predictor Construction Using an Evolutionary Method," Proc. PCS2009, pp. 1-4, (May 2009)
- [5] S. Takamura, M. Matsumura and Y. Yashima, "A Study on an Evolutionary Pixel Predictor and Its Properties," Proc. ICIP2009, (Nov. 2009)
- [6] "Lossless and Near-Lossless Compression of Continuous Tone Still Images," ISO/IEC 14495-1, (2000)
- [7] X. Wu and N. Memon, "Context-Based, Adaptive, Lossless Image Coding," IEEE Trans. Commun. vol. 45, no. 4, pp. 437-444, (Apr. 1997)
- [8] H. Satoh, M. Yamamura and S. Kobayashi, "Minimal Generation Gap Model for GAs Considering both Exploration and Exploitation," Proc. 4th Int. Conf. Soft Computing, pp. 494-497, (Oct. 1996)
- [9] M. Matsumura, S. Takamura and H. Jozawa, "Evolutive Image Coding Based on Automatic Optimization for Coding Tools Combination," Proc. IWAIT2010, (Jan. 2010)
- [10] S. Takamura, "Evolutive Video Coding ~From Generic Algorithm towards Content-Specific Algorithm~, " PCS2010 tutorial talk (T1), (Dec. 2010)
- [11] <http://www.gsic.titech.ac.jp/en>
- [12] M. McCawley, S. Takamura and H. Jozawa, "GPU-Assisted Evolutive Image Predictor Generation," IEICE Tech. Rep., vol. 110, no. 275, IE2010-88, pp. 25-28, (Nov. 2010)
- [13] S. Takamura, M. Matsumura and H. Jozawa, "Accelerating Pixel Predictor Evolution Using Edge-Based Class Separation," Proc. PCS2010, P1-22, pp. 106-109, (Dec. 2010)
- [14] M. Matsumura, S. Takamura and H. Jozawa, "Generating Subject Oriented Codec by Evolutionary Approach," Proc. PCS2010, P3-13, pp. 374-377, (Dec. 2010)
- [15] S. Takamura and H. Jozawa, "A Basic Study on Automatic Construction of Nonlinear Image Denoising Filter", Proc. PCSJ2011, (Oct. 2011) [to appear]

● TSUBAME e-Science Journal No.4

2011年10月31日 東京工業大学 学術国際情報センター発行 ©  
ISSN 2185-6028

デザイン・レイアウト：キックアンドパンチ

編集：TSUBAME e-Science Journal 編集室

青木尊之 ピパットポンサー・ティラボン

渡邊寿雄 佐々木淳 仲川愛理

住所：〒152-8550 東京都目黒区大岡山 2-12-1-E2-1

電話：03-5734-2087 FAX：03-5734-3198

E-mail: [tsubame\\_j@sim.gsic.titech.ac.jp](mailto:tsubame_j@sim.gsic.titech.ac.jp)

URL: <http://www.gsic.titech.ac.jp/>

# TSUBAME

## TSUBAME 共同利用サービス

『みんなのスパコン』TSUBAMEは、当初は主に東工大学内の研究・教育のために利用されておりましたが、平成21年7月よりTSUBAME 共同利用サービスを開始し、学術・産業・社会へと広く貢献しております。

### 課題公募する利用区分とカテゴリ

共同利用サービスには、「学術利用」、「産業利用」、「社会貢献利用」の3つの利用区分があり、さらに「成果公開」と「成果非公開」のカテゴリがあります。現在は随時申請を受け付けており、申請課題は厳正な審査の下、採択の可否を決定します。採択課題の利用期間は当該年度末までです。

#### TSUBAME 共同利用とは…

- 東工大学内のみならず、より多くの方にTSUBAMEサービスを提供
- 他大学や公的研究機関の研究者の **学術利用** [有償利用]
- 民間企業の方の **産業利用** [有償・無償利用]
- その他の組織による社会的貢献のための **社会貢献利用** [有償利用]

### 共同利用にて提供する計算資源

共同利用サービスの利用区分・カテゴリ別の利用課金表を下に示しました。TSUBAME 2.0における計算機資源の割振りは口数を単位としており、1口は標準1ノード(12CPUコア, 3GPU, 55.82GBメモリ搭載)の3000時間分(≒約4ヵ月)相当の計算機資源です。この計算機資源は、1000CPUコアを1日半とか、100GPUを3.75日といった利用も可能です。

利用区分	利用者	制度や利用規定等	カテゴリ	利用課金
学術利用	他大学または研究機関等	共同利用の利用規定に基づく	成果公開	1口: 100,000円
産業利用	民間企業を中心としたグループ	「先端研究施設共用促進事業」に基づく	成果公開	トライアルユース(無償利用) 1口: 100,000円
			成果非公開	1口: 400,000円
社会貢献利用	非営利団体、公共団体等	共同利用の利用規定に基づく	成果公開	1口: 100,000円
			成果非公開	1口: 400,000円

### 産業利用トライアルユース制度 (先端研究施設共用促進事業)

共同利用サービスの「産業利用」は、東京工業大学学術国際情報センターが実施する文部科学省先端研究施設共用促進補助事業を兼ねております。その中のトライアルユース制度では、初めてTSUBAMEを利用する民間企業の方に限り、無償での利用(1利用期間は最長1年間、2回まで)が可能です。この制度でスパコンTSUBAMEの敷居を下げることで、より多くの方にスパコンの魅力を体験していただいております。

## お問い合わせ

- 東京工業大学 学術国際情報センター 共同利用推進室
  - e-mail [kyodo@gsic.titech.ac.jp](mailto:kyodo@gsic.titech.ac.jp) Tel. 03-5734-2085 Fax. 03-5734-3198
- 詳しくは <http://www.gsic.titech.ac.jp/tsubame/> をご覧ください。