

# 階層的独立固有時間刻み法によるグラフ可視化計算の高速化

松林 達史<sup>†</sup> 山田 武士<sup>†</sup>

本研究では、大規模なネットワークデータのための高速かつ効率的な可視化座標計算の手法を提案する。従来のネットワーク可視化手法の1つとして Fruchterman らによる力学モデルを用いる手法がよく知られている。彼らの手法はノード間やエッジに対し力学関数を与えることにより、系全体のエネルギーを定義し、加速度方向に各ノードの座標を更新することによって、系のエネルギーの極小状態を求める。この手法では座標の更新頻度は一様で、すべてのノードを毎回更新していたが、提案手法では階層的独立固有時間刻み法を用いて個々のノードに独立な更新時間を設定し、局所的に更新頻度を変えることにより計算の高速化を可能にした。この手法は、天体力学において用いられている局所的に密集した領域を精度良く計算する手法を、グラフ可視化手法に拡張したものである。また、提案手法は並列処理に適しており、粒子間相互作用専用並列計算機 MDGRAPE-3 PCI-X に実装することによって、計算速度の数百倍高速化が可能であることを示した。さらに、LGL (Large Graph Layout) 法を用いた Opte Project の可視化結果との比較を行い、提案手法により高精度な可視化が可能であることを示した。

## Hierarchical Individual Timestep Algorithm for Large Scale Graph-drawing

TATSUSHI MATSUBAYASHI<sup>†</sup> and TAKESHI YAMADA<sup>†</sup>

In this paper, we propose a fast and efficient method for drawing very large-scale graph data. The conventional force-directed method proposed by Fruchterman and Rheingold (FR method) is well-known. It defines repulsive forces between every pair of nodes and attractive forces between connected nodes on a edge and calculates corresponding potential energy. An optimal layout is obtained by iteratively updating node positions to minimize the potential energy. Here, the positions of the nodes are updated every global timestep at the same time. In the proposed method, each node has its own individual time and timestep, and nodes are updated at different frequencies depending on the local situation. The proposed method is inspired by the hierarchical individual timestep method used for the high accuracy calculations for dense particle fields such as star clusters in astrophysical dynamics. Experiments show that the proposed method outperforms the original FR method in both speed and accuracy. We implement the proposed method on the MDGRAPE-3 PCI-X special purpose parallel computer and realize a speed enhancement of several hundred times.

### 1. はじめに

#### 1.1 研究背景

複雑な関係データの表現方法として、ネットワークもしくはグラフを用い、これらを可視化するという手法は様々な研究分野で広く用いられている。重力や分子間力のように数学的に非常にシンプルに記述されるものや、遺伝子やタンパク質のように複雑に入り組んだもの、またインターネットのWEB ページや人間同士や社会的な存在同士の関係であるソーシャルネット

ワークもまた関係データであり、これら関係データは自然にネットワークとして表現することができる。

ネットワーク表現を用いる利点は、グラフ理論など数学的手法による構造解析が可能という点である。また、ネットワークを低次元空間に配置することにより、データの潜在的な構造を浮き彫りにすることができるという大きな利点がある。さらに、可視化することにより、大量なデータでも直感的理解が得やすいという点もあげられる。グラフ可視化においてはネットワークの特性を利用して、効果的な可視化手法を考える必要がある。

本研究では、Fruchterman ら<sup>1)</sup>によって提案された Force-directed method をベースとして、大規模なグラフデータのための高速かつ効率的な可視化手法を提

<sup>†</sup> 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所  
NTT Communication Science Laboratories, NTT Corporation

案する．Fruchterman らの手法では座標の更新頻度を一様とし，すべてのノードを毎回更新しているのに対し，提案法では個々のノードに独立な時間および更新時間を設定し，局所的に更新頻度を変えることにより効率的に座標更新を行う．この手法は，天体力学において用いられている“階層的独立固有時間刻み法”と呼ばれる手法を，グラフ可視化手法に拡張したものである．さらに我々は，提案法の並列性を利用し，粒子間相互作用専用並列計算機 MDGRAPE-3 PCI-X に実装することにより，百倍を超える計算速度の高速化を実現した．

以下本研究では，総ノード数を  $N$ ，総エッジ数を  $E$  とし，無向かつ重みのないグラフデータを対象に，数万ノードを超える大規模なグラフデータの計算に耐えうる可視化手法の開発を目指し，並列処理を利用した新手法を提案する．

## 2. 従来研究

グラフ可視化の研究分野では 1980 年代より様々な手法が考案されてきた．最も一般的な手法は，ノード間のグラフ距離や接続関係などに基づき，系に力学関係を与え，加速度方向に座標を更新することによって系の最小エネルギー状態を求める手法である．この手法は“Force-directed method”と呼ばれ，分子動力学に例えると，温度 0 の結晶構造を求めることに等しい．近年では様々な手法も考案がされており，以下主要な手法を紹介する．

### 2.1 Spring Embedder method

グラフ可視化において最もよく知られているのは“spring method”と呼ばれる手法であり，その原型は Eades<sup>2)</sup> により提案された．この手法は，リンクのあるエッジ間にバネのような力を与え，系全体のエネルギーが極小になるように加速度方向にノードの配置を更新する手法である．

### 2.2 Kamada and Kawai Method

Kamada ら<sup>3)</sup> は Eades の手法を基に，新たな spring method を提案した（以下 KK 法と略す）．すべてのノード間に仮想的なバネを仮定し，座標の更新手法を逐次的に 1 ノードずつ行う． $i$  番目と  $j$  番目のノード間のグラフ上での最短経路をバネの自然長  $l_{ij}$ ，ノード間のバネ定数を  $k_{ij}$ ，ノードの空間的な座標を  $\mathbf{x}_i$  とするとき， $i$  番目のノードに加わる加速度は以下のように表すことができる．

$$\mathbf{a}_{KK,i} = \sum_{j \neq i}^N k_{ij} \left( 1 - \frac{l_{ij}}{|\mathbf{x}_{ij}|} \right) \mathbf{x}_{ij}, \quad (1)$$

ここで  $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$  である．KK 法ではバネ定数を  $k_{ij} = k/l_{ij}^2$  とする．また  $k$  は無次元量の規格化定数である．

後述の Frucherman らの手法が全ノードの更新を等しい頻度で行うのに対し，KK 法ではエネルギー準位が高いものから逐次的に座標更新を行う．すなわちポテンシャル勾配の 2 次導関数を求め，Newton-Laphson 法を用いて座標更新を行う．しかしながら，ノード間の最短経路のデータをつねに保持する必要があり，ノード数  $N$  に対して  $O(N^2)$  のメモリ領域が必要になる．それゆえ，数 GB のメモリを持つ汎用計算機でも数万ノード程度の計算でメモリがオーバーフローを起こす．したがって，この手法は大規模な可視化計算には適していないといえる．

### 2.3 Fruchterman and Reingold Method

Fruchterman ら<sup>1)</sup> もまた Eades の手法を基に，独自の手法を提案した（以下 FR 法と略す）．FR 法ではノードにかかる力は，エッジに働く力を引力  $\mathbf{a}_a$ ，ノード間に働く力を斥力  $\mathbf{a}_r$  として，2 つの力の和で与えられる． $\mathbf{a}_a$ ， $\mathbf{a}_r$  は以下で与えられる．

$$\mathbf{a}_{a,i} = \frac{1}{k} \sum_{j \neq i}^{E_i} |\mathbf{x}_{ij}| \mathbf{x}_{ij}, \quad (2)$$

$$\mathbf{a}_{r,i} = -k^2 \sum_{j \neq i}^N \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2}. \quad (3)$$

ここで  $E_i$  は  $i$  番目のノードに接続されているエッジの本数（次数）であり， $k$  は無次元量の距離でおおよそバネの自然長に相当する．したがって  $i$  番目のノードの加速度  $\mathbf{a}_i$  は，式 (2)，(3) を用いて

$$\mathbf{a}_i = \mathbf{a}_{a,i} + \mathbf{a}_{r,i} \quad (4)$$

と表すことができる．

FR 法では式 (4) で定義される加速度の方向に，全ノードの座標を同時更新する．この際冷却関数を用いて座標更新の最大更新幅を  $t_{cool}(t)$  で制限する．このとき更新式は以下で与えられる．

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{a}_i \times \min(1, t_{cool}(t)/|\mathbf{a}_i|). \quad (5)$$

ここで  $t_{cool}(t)$  は次式で与えられる冷却関数であり，

$$t_{cool}(t) = C \times (1 - t/T_{END}), \quad (6)$$

パラメータ  $C$  は更新幅の初期値である．しかしながら，この手法では式 (6) の冷却関数に対し，最適な  $T_{END}$  をあらかじめ設定する必要があり，大規模なネットワークの計算では特に問題となる．この問題に関しては 4.3 節において議論を行う．ここで，表 1 に FR 法の具体的な更新手順を示す．

表 1 FR の更新手法  
Table 1 Schematic of FR method.

(a)	初期配置 $x_i$ をランダムに与える
(b)	全ノードの加速度 $a_i$ を計算
(c)	冷却関数 $t_{cool}(t)$ を計算
(d)	全ノードの $x_i(t+1)$ を求める
(e)	時間を更新する, $t = t + 1$ .
(f)	(b)-(e) を, $t = T_{END}$ を満たすまで繰り返す

## 2.4 近年の研究

本節までに述べてきた可視化手法では、基本的に全ノード間の相互作用を計算する必要があるため、単純に考えるとノード数の 2 乗 ( $N^2$ ) の計算量が必要である。これに対しいくつかの計算の簡略化が試みられている。

Quigley ら<sup>4)</sup> では FADE と呼ばれる手法を提案し、遠距離力に Burns-Hut tree code [Burns ら<sup>5)</sup>] を用いることにより、 $O(N^2)$  の計算量を  $O(N \log N)$  に抑える手法を提案した。Tree code は天文学において重力計算のために開発され、近傍の天体からの重力は直接計算を行い、一方で遠方の集団からの力を重心計算として近似する手法である。

Adai ら<sup>6)</sup> は大規模なネットワークの描画手法として LGL (Large Graph Layout) 法を提案した。この手法はデータのリンク構造から最小全域木 (Minimum spanning-tree) を生成し、最小全域木に従い座標を決定する。近年では、この手法を用いた 500 万ノードを超えるネットワーク地図の可視化計算が Lyon<sup>7)</sup> によって行われている。これは “Opte Project” と呼ばれ、ノード数 500 万、エッジ数 5,000 万の大規模なインターネットの地図を 252.68 時間かけて作成したものである。しかしながら、この手法は最小全域木を生成することによってネットワーク構造を破壊してしまうため、可視化結果の品質には問題を残している。この可視化結果に関しては 5 章で再び議論を行う。

## 3. 提案手法

ノード間の加速度計算に関しては、これまでも様々な効率化手法が提案されているが、座標の更新手法にはいまだ改良の余地が残っている。従来の Force-directed 法はいずれも “Shared timestep method” と呼ばれる手法であり、座標更新はすべて同時に行われる。一方本研究では独立固有時間刻み法 (Individual timestep method) を用いて、各ノードに固有の更新時間を持たせる手法を提案する。この手法はもともと天体力学の高精度かつ高速な計算のために開発され<sup>8)-10)</sup>、天体の星団や銀河団のような密度の粗密の

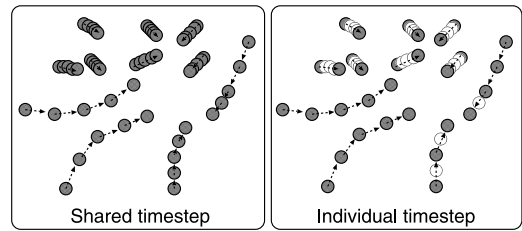


図 1 Shared timestep method と Individual timestep method の更新概念図。○ は予測子

Fig. 1 The left and right figures show a the schematic figure of time evolution for the shared and individual timestep scheme, respectively. For Individual timestep, longer timestep nodes are calculated as predictor ○.

差が大きな粒子計算を扱う計算において用いられ、特に局所的に密集した領域を精度良く計算を行う手法として知られている。

Shared timestep method では全ノードを毎回等しい頻度で更新を行うために、更新座標がほとんど変化しない、つまり加速度がゼロに近い場合でも毎回ノードを更新し続ける。一方、Individual timestep method では、全体を支配する大局的な時間とは別に、個々のノードに独立な更新時間を設定し、加速度の小さなノードは更新のタイミングを長くする。その結果、効率良く計算を高速化することができる。ここで Shared timestep method と Individual timestep method の、更新手法の概念図を図 1 に示す。

Individual timestep method では、個々に独立に固有な時間を持つノードは、加速度計算のために、つねに大局的な時間に従った近似的な座標である予測子を計算しなければならない。しかしながら、各ノードが完全に固有な時間を所有すると予測子の計算頻度が増し、莫大な計算コストがかかる。そのために固有時間を階層化することにより、複数のノードを時間同期させる。これは予測子の計算量の軽減だけでなく、並列性を高めることによって計算の効率化を図ることができる。

### 3.1 階層的独立固有時間刻み法

従来、天体力学などに用いられる計算手法は、エネルギーを保存させるために高精度に微分方程式を解く必要があり、一方可視化における座標更新では、系のエネルギーを効率良く減少させる必要があり、天体力学で用いられている手法をそのまま直接利用することはできない。したがって本研究では、座標更新式において加速度勾配の低次成分のみを残し、固有時間刻み幅を加速度の大きさに反比例させて与える手法を提案する。以下、階層的独立固有時間刻み法 (Hierarchical

Individual timestep method) に基づく提案法の詳細を述べる．

座標更新は FR 法と同様に Force-directed 法を用いて、加速度方向に座標を更新する．このとき  $i$  番目のノードにはそれぞれ独立固有時間  $t_i$ ，独立固有時間刻み  $\Delta t_i$  を与える． $i$  番目のノードは時刻が  $t_i$  において、加速度方向へ  $\Delta t_i$  倍だけ更新する．このとき更新式は以下で与えられる．

$$\mathbf{x}_i(t_i + \Delta t_i) = \mathbf{x}_i(t_i) + \Delta t_i \times \mathbf{a}_i(t_i). \quad (7)$$

ここで、各ノードに完全に独立な固有時間を与えると、計算効率が低下するため、階層的に離散的な固有時間を与えて複数のノードを同期させる．そこで  $\Delta t_i$  は以下のように与える．

$$\Delta t_i = 2^k \quad \text{if } 2^k \leq \frac{\eta}{|\mathbf{a}_i(t_i)|} < 2^{k+1}, \quad (8)$$

ここで  $\eta$  は無次元定数で、数値計算の精度を決めるとともに、各階層での最大時間更新幅を与える．また  $k$  は整数である．上記のように固有時間を 2 のべき乗で与えることによって、数値計算上完全な時間同期を行うことができる．

次に、各ノードの更新タイミングに対し、制限を設ける．全体の大局的な時刻  $t$  に対して、以下の条件を満たしたノードのみが更新される、

$$t \bmod \Delta t_i = 0. \quad (9)$$

またこの制限は、ノードの並べ替えに要する数値計算負荷を軽減させるとともに、以下に述べるように固有時間刻み  $\Delta t_i$  の更新に対する制限を与える．

初めに  $i$  番目のノードに対して、 $\Delta \tilde{t}_i$  を以下のように与える．

$$\Delta \tilde{t}_i = \frac{\eta}{|\mathbf{a}_i(t)|}. \quad (10)$$

このとき  $\Delta \tilde{t}_i < 2^k$  であれば無条件に  $k \rightarrow k-1$  とする．しかし  $k \rightarrow k+1$  とする場合には同期させる時刻に注意しなければならない．なぜなら時刻  $\Delta t_i$  が式 (9) で制限されるために、 $2^{k+1}$  もまた式 (9) を満たす必要がある．したがって、

$$t \bmod 2^{k+1} = 0 \quad (11)$$

が固有時間刻み増加のタイミングの条件となる．以上より、 $i$  番目のノードに対して、新しい固有時間刻み、 $\Delta t_{\text{new},i} = 2^{k_{\text{new}}}$  は以下で与えられる．

$$k_{\text{new}} = \begin{cases} k-1 & \text{if } \Delta \tilde{t}_i < 2^k \\ k+1 & \text{if } \Delta \tilde{t}_i \geq 2^{k+1} \\ & \text{and } t \bmod 2^{k+1} = 0 \\ k & \text{otherwise.} \end{cases} \quad (12)$$

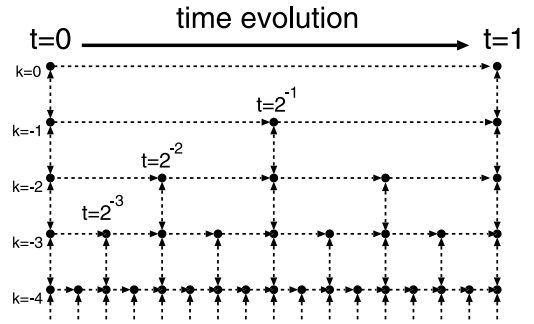


図 2 階層的独立固有時間刻み方の時間更新概念図  
Fig. 2 Time evolution scheme for the hierarchical individual timestep method.

ここで図 2 に更新概念図を示す．各階層に分けられたノード群は“●”上に固有時間を持ち、時刻  $t$  が左から右へ向かって進行し、 $t$  と同期したノードのみが固有時間を更新する．このとき  $\Delta t_i$  の変更は式 (8) ~ (12) の条件より、図の点線上の方向のみ許されることになる．

またさらに、計算の更新条件として時間刻みに上限  $\Delta t_{\text{max}}$  と下限  $\Delta t_{\text{min}}$  を設定する．上限を与えることにより、一定の間隔で定期的に全ノードの時間同期を行い、また下限を設定することにより時間刻みの幅に最小値を与える．下限値の設定は、初期配置を乱数で与えることによる加速度の発散を回避するためである．したがって、 $\Delta t_i = \Delta t_{\text{min}}$  となったときは、

$$\mathbf{x}_i(t_i + \Delta t_{\text{min}}) = \mathbf{x}_i(t_i) + \Delta \tilde{t}_i \times \mathbf{a}_i(t) \quad (13)$$

とする．なお本研究では  $\Delta t_{\text{max}} = 1$ ， $\Delta t_{\text{min}} = 2^{-10}$  とした．

### 3.2 更新手法

提案法では、固有時間が時刻  $t$  と同期したノード集合のみが更新される．ここで時刻  $t$  の更新は以下で与えられる．

$$t = \min (t_i + \Delta t_i). \quad (14)$$

このとき、同期しているノード集合  $S(t)$  は

$$S(t) = \{i | t_i + \Delta t_i = t\} \quad (15)$$

で定義され、そのノード数を  $n_s(t)$  とする．すなわち

$$n_s(t) = |S(t)|. \quad (16)$$

したがって、ノード間の相互作用の計算量は、時刻  $t$  においては 1 ステップあたり  $O(N^2)$  から  $O(n_s(t) \times N)$  に軽減できる．

次に階層的独立時間刻み法では、異なる固有時間を持つノードからの力の計算は、予測子を用いて近似的な座標からの相互作用として計算する．予測子は固有時間との差分だけ座標更新させて、以下のように与える．

$$\tilde{\mathbf{x}}_j(t) = \mathbf{x}_j(t_j) + (t - t_j) \times \mathbf{a}_j(t_j). \quad (17)$$

表 2 階層的独立固有時刻み方の更新手法  
Table 2 Schematic of our method.

(a)	$t = 0$ とし, 全ノードの時刻を $t_i = 0$ に設定する
(b)	初期配置 $\mathbf{x}_i(0)$ をランダムに与える
(c)	全ノードの加速度 $\mathbf{a}_i$ を計算
(d)	全ノードの $\Delta t_i$ を計算
(e)	$\min(t_i + \Delta t_i)$ を求め, その値を $t$ とする
(f)	$t = t_i + \Delta t_i$ を満たすノード集合 $S(t)$ を求める
(g)	全ノードの予測子 $\tilde{\mathbf{x}}(t)$ を計算
(h)	$S(t)$ の加速度 $\mathbf{a}_i(t)$ を計算
(i)	$S(t)$ の座標 $\mathbf{x}_i$ を更新
(j)	$S(t)$ の固有時間 $t_i$ を, $t_i = t_i + \Delta t_i$ に更新
(k)	$S(t)$ の固有時間更新幅 $\Delta t_i$ を更新
(l)	(e)-(k) を, $t = t_{\text{END}}$ を満たすまで繰り返す

時刻  $t$  における加速度  $\mathbf{a}_i(t)$  は, 上記の予測子を用いて, FR 法の加速度公式 (4) より以下のように与える.

$$\mathbf{a}_i(t) = \frac{1}{k} \sum_{j \neq i}^{E_i} |\mathbf{x}_{ij}| \mathbf{x}_{ij} - k^2 \sum_{j \neq i}^N \frac{\mathbf{x}_{ij}}{(\mathbf{x}_{ij}^2 + \epsilon^2)}, \quad (18)$$

ここで

$$\mathbf{x}_{ij} = \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i, \quad (19)$$

であり, Softening parameter  $\epsilon$  は, 式 (18) において, 右辺第 2 項が  $|\mathbf{x}_{ij}| \rightarrow 0$  で発散してしまうのを防ぐための補正項である. ここで本研究では  $\epsilon = 0.01$ , また定数  $k = 1$  とした.

表 2 に提案法の更新方法を示す.

### 3.3 評価方法

#### 3.3.1 系のエネルギー $\Psi$

本研究では, ネットワークの可視化結果の定量的な評価手法として, 系のポテンシャルエネルギー  $\Psi$  を用いる. エッジおよびノード間に力学方程式を与えれば,  $\Psi$  と加速度との関係式はポテンシャル勾配より以下のように決定される,

$$\mathbf{a} = \nabla \Psi. \quad (20)$$

したがって, ポテンシャルエネルギー  $\Psi$  は式 (18) より以下のように与えられる.

$$\Psi = \frac{1}{3k} \sum_{i,j}^E |\mathbf{x}_{ij}|^3 - \frac{k^2}{2} \sum_i^N \sum_{i \neq j}^N \log |\mathbf{x}_{ij}^2 + \epsilon^2|^{1/2}. \quad (21)$$

以下, この  $\Psi$  を評価関数として議論を行う.

#### 3.3.2 平均ステップ数 $n_{ave}$

ここで計算量に対してのエネルギー収束性評価のために, FR 法の計算ステップ数  $T$  に対し, 提案法では式 (16) で定義される  $n_s(t)$  の時刻  $T$  までの積和を

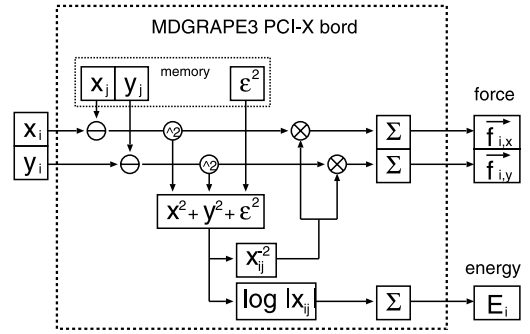


図 3 MDGRAPE3 の pipe line イメージ  
Fig. 3 Pipeline image of MDGRAPE3.

$N$  で割った, 以下の式で定義される  $n_{ave}$  を用いて評価した.

$$n_{ave} = \frac{1}{N} \sum_{t=0}^T n_s(t). \quad (22)$$

実際にはエッジに加わる加速度も計算負荷に関わり, またノードの並べ替えなどの計算負荷も考慮する必要があるが, 大規模ネットワークでは, 99%以上の計算時間をノード間の相互作用計算に費やす. それゆえ FR 法の計算量と比較を行う際, この  $n_{ave}$  を用いることは妥当であるといえる.

### 3.4 MDGRAPE-3 による高速化

加速度および, エネルギーの計算では式 (18), (21) に示されるように, ノード間の斥力計算は通常  $O(N^2)$  の計算量を要する. それゆえ, 大規模ネットワークの計算においてはノード間の計算時間が全体の 99%以上を占める. しかしながらノード間の相互作用計算は, 並列性利用し, パイプライン処理に基づく並列処理による高速化が可能である. そこで我々は, MDGRAPE-3 のパイプライン処理を用いてノード間の相互作用計算を行い, その他の計算はホストマシンで行う手法を用いた. MDGRAPE-3 は理化学研究所で開発された粒子間相互作用専用計算機であり, 本研究では PCI-X 版を用いて計算を行った.

図 3 に MDGRAPE-3 の Pipeline イメージを示す. MDGRAPE-3 はボード側のメモリに力を与える側のノードの座標情報を保持し, ホストマシンから力を受ける側のノードの座標を送り, それぞれのノード間の

MDGRAPE-3 は 2006 年の IEEE Gordon Bell Prize ではピーク性能部門 (Honorable Mention) を受賞<sup>11)</sup>. 同システムは, 3 次元座標まで計算が可能で, ベクトル関数およびスカラー関数は, 自由に設定することができる. 本研究で用いた PCI-X 版は, MDGRAPE-3 のチップを 2 基搭載し, ピークパフォーマンス 330 Gflops を誇る汎用機専用ボードである.

相互作用を流れ作業として一気に計算する。

実際に本研究では、FR法、提案法の両手法に対して、加速度およびエネルギーの計算に対してMDGRAPE-3を実装することによって、100倍以上計算の高速化を実現した。たとえばmodel A ( $N = 35,638$ )の計算では、Intel Xeon 3.2 GHzを搭載したホストマシンのみでの計算では5,000分を要する計算を、提案法では $n_{ave} = 10,000$ を約30分で計算することができた。このように、単純な粒子間相互作用計算には並列性を生かし、専用ボードを用いた演算処理が非常に有効であるといえる。

以下、本研究におけるネットワークデータの計算はMDGRAPE-3を用いて計算を行い、各モデルに対する計算時間の議論は、4.4節で行う。

### 4. 計算結果と考察

#### 4.1 モデルデータ

本研究において、可視化計算に用いたグラフデータは2.4節で紹介したOpte ProjectのIPネットワークのデータを用いた。このデータはClass-Cに分類されるネットワークアドレスをノードとし、tracerouteコマンドにより得られるネットワークの経路情報をもとに接続状態を調べ、接続されているIPアドレス間をエッジとしたネットワークデータである。同サイトからデータファイルをダウンロードすることが可能であり、本研究では表3に示すサイズの異なる4種のデータを用いた。

次に、各モデルデータのノードの次数分布を図4に示す。図からも分かるように、次数分布がべき則を示す典型的なスケールフリーネットワークであり、提案法の計算比較対象としては妥当といえる。

#### 4.2 パラメータ依存性

可視化座標計算に対するパラメータは、大きく分けて2つ存在する。1つは座標スケールに対しての定数 $k$ であり、空間の計量に相当する。ここで本研究では $k = 1$ とし、これはたとえばリンクの存在する2つのノードが1だけ離れることを意味する。

もう1つは時間のスケールである。FR法においては、時間更新幅は式(6)の $C$ で与えられ、提案法で

は式(10)の $\eta$ で与えられる。そこで以下、表3のmodel Aを用いてそれぞれのパラメータ依存性の予備実験を行った。

#### 4.2.1 パラメータ $C$ の効果

FR法では最大更新幅が $t_{cool}(t)$ によって与えられ、更新時間とともに減少しゼロに近づく。更新幅の大きさはパラメータ $C$ に依存し、 $C$ が大きいくほど、初期の更新幅が大きくなる。ここで $C$ を変えた計算結果を図5に示す。

図5は、FR法において $T_{END} = 10,000$ として、 $C = 1.0, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01$ とパラメータを変えて計算を行った。結果として、 $C$ が大きくなるに従い計算初期段階のエネルギー収束は速くなる一方、最終的なエネルギーの収束速度は $C$ に比例せず、最適な $C$ を設定することが困難であることが分かる。これは関数 $t_{cool}$ の収束速度にも関係している。 $t_{cool}$ は $T$ に線形比例して小さくなるために、 $C$ が十分に大きいと $T$ がある程度大きくなるまで $t_{cool}$ は小さくならない。逆に $C$ が小さすぎると大局的な構造が得られる前に $t_{cool}$ が小さくなりすぎてゼロに収束してしまう。以上、本研究で用いたモデルデータに対しては、 $C = 0.1$ が最適であると考えられ、以下 $C = 0.1$

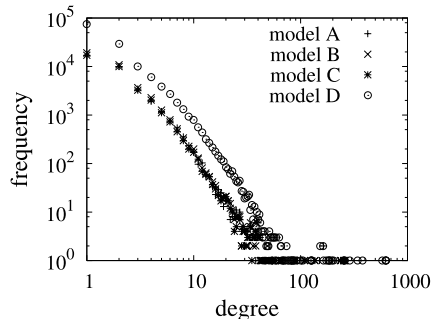


図4 データの次数分布  
Fig. 4 Degree distribution.

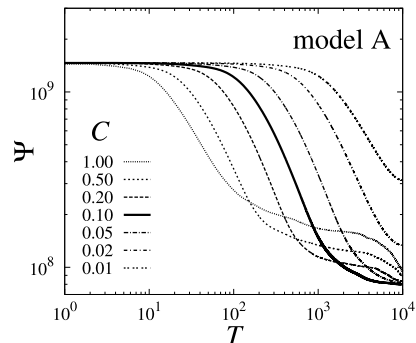


図5 FR法における $C$ の効果  
Fig. 5 Effect of  $C$  for FR method.

表3 ネットワークデータ  
Table 3 Network data.

model	ノード数 ( $N$ )	エッジ数 ( $E$ )
A	35,638	42,827
B	35,836	42,387
C	40,027	47,215
D	134,023	161,283

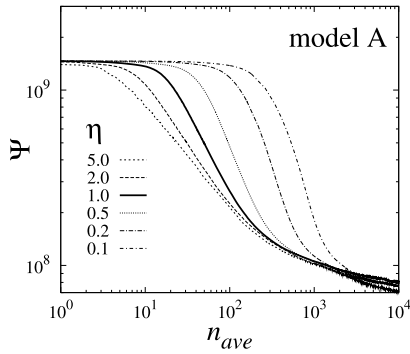


図 6 提案法における  $\eta$  の効果  
Fig. 6 Effect of  $\eta$  for proposed method.

を採用し、提案法との比較を行う。

#### 4.2.2 パラメータ $\eta$ の効果

提案法では最大更新幅が  $\eta$  によって決まるが、FR 法と異なり更新幅は更新時間に依存しない。ここで  $\eta$  を変えた計算結果を図 6 に示す。

図 5 と同様に、ネットワークの可視化計算を  $\eta = 5.0, 2.0, 1.0, 0.5, 0.2, 0.1$  と与えて計算を行い、 $n_{ave} = 10,000$  となるまで計算を行った。この結果より、 $\eta$  が大きくなるに従いエネルギー収束は速くなり、最終的なエネルギーの収束速度は  $\eta$  に比例するということが分かる。FR 法と異なり更新幅は時刻  $t$  にかかわらず  $\eta$  で決まる。 $\eta$  が十分小さければ局所的な構造まで精確に計算を行うことができるが、大局的な構造を求める場合には  $\eta = 1.0$  でも十分である。以下、本研究では数値計算において  $\eta = 1.0$  を使用する。

#### 4.3 計算結果

我々の手法と FR 法で大きく異なるのは  $T_{END}$  の設定である。FR 法では最大更新幅を抑えるために収束させる時間  $T_{END}$  をあらかじめ設定する必要がある。しかしながら大規模なネットワークの可視化座標計算では、 $T_{END}$  を予測するのは困難であり、十分にエネルギー  $\Psi$  が収束していない場合、再度計算を行う必要がある。そこで FR 法に対し、 $T_{END} = 100, 1,000, 5,000, 10,000$  と変えて計算を行った。結果を図 7 に示す。この図より  $T_{END} = 100, 1,000$  ではエネルギーがまだ十分収束せず、 $T_{END} \geq 5,000$  とする必要があることが分かる。対比のために提案法を実線で示した。提案法では FR 法よりも早期にエネルギーが収束していることが分かる。

可視化結果を図 8 と図 9 に示す。FR 法では  $T_{END} \geq 5,000$  としなければまったく見えてこなかった構造が、提案法では初期の段階から全体的な構造が見え始め、 $n_{ave} = 1,000$  で大局的な構造を見ることが出来る。これは提案法が階層的に構造を計算してゆくために、ク

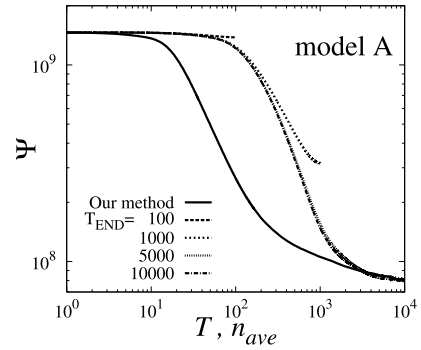


図 7 FR 法における  $T_{END}$  の効果と、提案法との比較  
Fig. 7 Effect of  $T_{END}$  for FR method, and comparison result with our method.

リック性の高いサブネットワーク構造に対し、初期段階から優先的に計算が行われるためである。

提案法ではノードの更新を階層的に設定することにより、FR 法よりも早い段階でネットワーク全体の構造を与えることが示された。

次に model B-D の計算結果を図 10 に示す。FR 法に対してはすべて  $T_{END} = 10,000$  とした。この結果からも分かるように、すべてのモデルにおいて同様な計算結果が得られた。

#### 4.4 計算時間の比較

本研究では FR 法と対比させるために  $n_{ave}$  を用いて  $n_{ave} = 10,000$  まで計算を行ったが、提案法では  $n_{ave} = 1,000$  まで行えば十分なエネルギーの収束値が得られる。図 7, 10 より提案法では  $n_{ave} = 1,000$  ではほぼ収束値に近づいていることが分かり、大局的な全体構造を把握することができる。それぞれの計算時間を表 4 に示した。

提案法では少ない同期ノード数  $n_s(t)$  でも毎回 MDGRAPE-3 を用いて計算を行うために、FR 法で  $T_{END} = 10^4$  までの計算時間に比べて  $n_{ave} = 10^4$  まで行った場合は計算が遅くなる。これはノード間の計算が MDGRAPE-3 によって高速化されたために、MDGRAPE-3 とホストマシンの通信によるオーバーヘッドが相対的に高くなってしまったためである。しかしながら、たとえば model D のようなより大規模なネットワークの計算では両手法による計算のオーバーヘッドは 1 割未満となり無視できる。

提案法を用いることにより、数万程度の規模のネットワークを数分以内に可視化計算することが可能となる。このことは、大規模なネットワークをリアルタイムに可視化することが可能であることを意味し、この点でも提案手法は有用であるといえる。

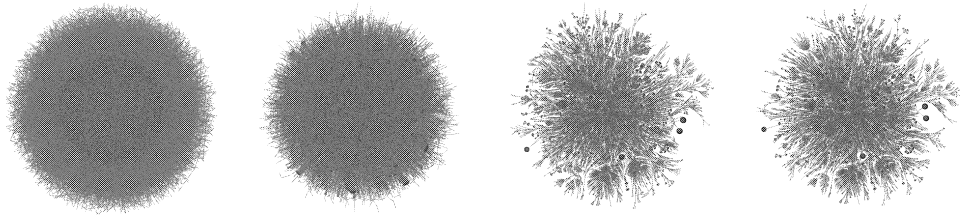


図 8 FR 法の可視化結果 . 左より  $T_{END} = 100, 1,000, 5,000, 10,000$   
 Fig.8 Graph-rayout results of FR method.

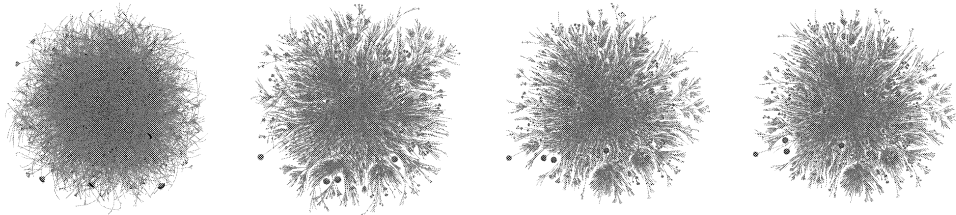


図 9 提案法の可視化結果 . 左より  $n_{ave} = 100, 1,000, 5,000, 10,000$   
 Fig.9 Graph-rayout results of our method.

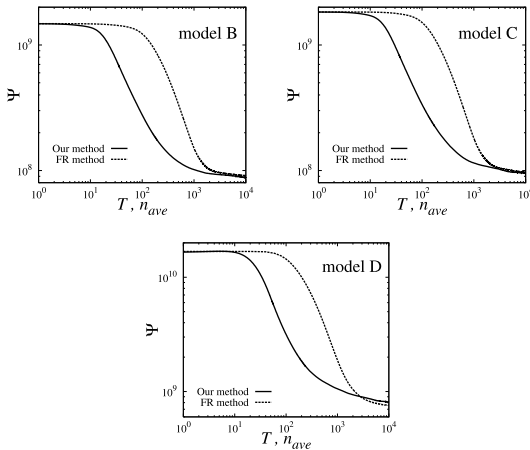


図 10 model B-D の FR 法と HI 法の計算結果  
 Fig.10 Results of FR method and our method for each models.

表 4 CPU time  
 Table 4 CPU time.

model	FR 法	提案法	提案法*	Host PC
A	25	3	29	5,000
B	25	3	29	5,100
C	31	4	36	6,300
D	324	35	348	70,000

FR 法:  $T_{END} = 10^4$  までの計算に要した時間 [分]  
 提案法:  $n_{ave} = 10^3$  までの計算に要した時間 [分]  
 提案法\*:  $n_{ave} = 10^4$  までの計算に要した時間 [分]  
 Host PC: 提案手法をホストマシンのみで行った場合に,  
 $n_{ave} = 10^4$  までの計算に要する時間 [分]  
 (ただし,  $n_{ave} = 100$  まで計算し 100 倍した値)

### 5. LGL 法との比較

前章までに述べてきたように,提案法と,MDGAPE-3を用いた並列演算処理を組み合わせることによって,従来では計算できなかった大規模なネットワークデータの比較的高精度な可視化が初めて可能となる.以下本章では,提案法による可視化結果と,最大規模のネットワークの可視化を行った,Opte Projectの可視化結果との比較検討を行う.

Opte Project は 2.4 節でも述べたように,可視化の座標計算には LGL 法を利用している.しかしながら LGL 法では,高速な計算を可能とするために,初期にネットワークのリンク構造を切り崩し,最小全域木 (Minimum spanning-tree) を作成する.そのため本来関連性の高いノードが離れた場所に配置されてしまう傾向がある.

図 11 は,左右それぞれ LGL 法と提案法による model D の可視化結果である.要した計算時間はそれぞれ LGL 法が 300 分,提案法では 348 分であった.色の濃淡は各エッジを元データのドメイン地域に応じて与えている.さらにドメイン地域ごとに濃淡を強くしたものが図 12 であり,それぞれ “Asia Pacific”, “North America”, “Europe/Middle East/Central Asia/Africa” を示す.LGL 法による配置では各地域ドメインのノードが全体的に分散して配置され,局所的な構造は見えるものの全体構造は不鮮明である.これは本来近接ノードであるものが,LGL 法の前処理によってネットワークが破壊され,遠方に配置された



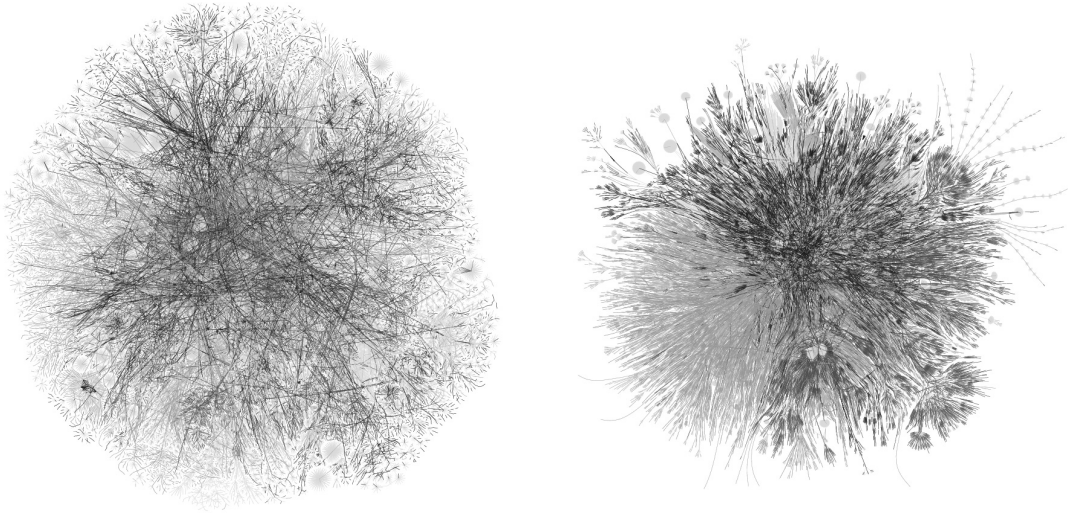


図 11 model D の可視化結果．左は LGL 法，右は提案法によるもの  
 Fig. 11 Left and right are the layout results of model D with the LGL method and proposed method, respectively.

ためと考えられる．一方提案法では，各ドメイン地域のノードがまとまって配置されていることが分かり，局所的にも，大局的にも各地域の IP 接続の特徴を見ることができる．

次に，LGL 法および提案法による可視化結果の定量的評価を行うために，計算によって得られた座標によるエッジの長さの分布を求めた．表 5 はそれぞれの結果の最大座標  $L_{\text{box}}$  (図 11 のそれぞれの横幅，もしくは縦幅の長い方)，全エッジの長さ  $|x_{ij}|$  の平均値，

$$\langle x_{ij} \rangle \equiv \frac{1}{E} \sum_{i,j} |x_{ij}|, \quad (23)$$

およびその分散  $\sigma$  である．比較のために  $\langle x_{ij} \rangle$ ， $\sigma$  を，それぞれ  $L_{\text{box}}$ ， $\langle x_{ij} \rangle$  で規格化をした．この結果により，両手法による平均的なエッジの長さは全体のスケールに対して 1%程度であることが分かる．また，LGL 法による結果の方が，提案法よりもエッジの平均長に比べて分散が大きいことが分かる．

図 13 は，全エッジ距離  $|x_{ij}|$  の分布を示したものである．ここで横軸は，各エッジの長さ  $|x_{ij}|$  を  $L_{\text{box}}$  で規格化し，長さ  $1/250$  の値域に区切った分布である．この図よりも，LGL 法による可視化結果は提案法に比べ全体的に長いエッジが多数存在することが分かる．これは初期の全域木処理による影響であると考えられる．提案法では最も長いエッジは全体の大きさの 10%未満であるのに対して，LGL 法では全体の大

きさの 10%以上の長さを持つエッジの数は全体の約 1%を占め，さらに最も長いエッジは全体の大きさの約 30%にもなる．

以上より，提案法のように全エッジの影響を考慮に入れた計算では，関係性のあるノードは近傍に配置されるのに対し，LGL 法を用いた場合には本来近接するべきノードが遠方に配置されてしまうという問題が生じることが分かった．これは minimum spring tree に基づく手法の限界と考えられる．

## 6. 結論と考察

提案法と FR 法では，大きな違いは 2 点ある．1 点は更新ノードに独立性を持たせ，FR 法がすべてのノードが同時に更新し続ける手法に対し，提案法は階層的にノードが更新される点である．そのため提案法ではノードの更新を効率化させることにより計算の高速化に成功した．もう 1 点は  $T_{\text{END}}$  の設定である．FR 法では図 8 から分かるように可視化結果が  $T_{\text{END}}$  の値に依存し，十分にエネルギー  $\Psi$  が収束していない場合，再度計算を行う必要がある．一方提案法では，あらかじめ計算の終了時刻を設定する必要はなく，また更新幅は計算時間に依存しないため再計算の必要性がない．

従来 FR 法ではこのような大規模なネットワークの可視化は想定されておらず，中規模の  $N \approx 100$  程度のネットワークの可視化を想定し， $T_{\text{END}}$  も 100 程度とされてきた．しかしながら，近年の計算機環境の向

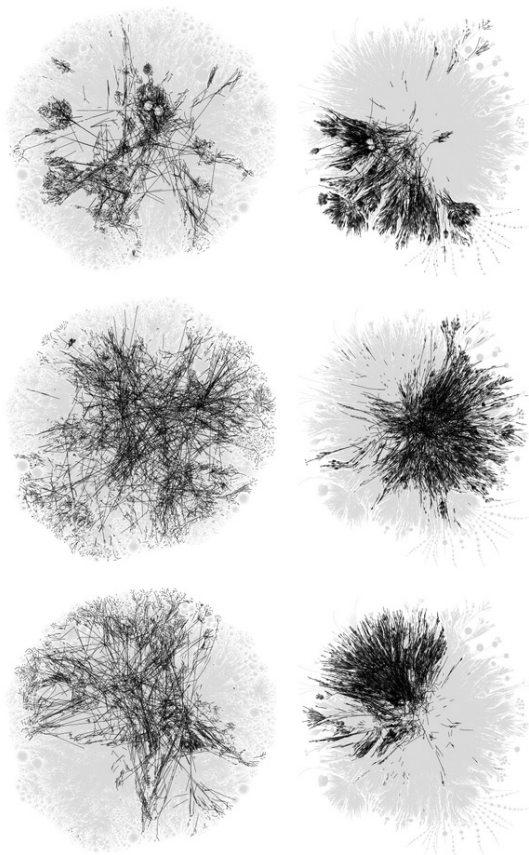


図 12 model D の可視化結果 . 左は LGL 法 , 右は提案法によるもの . 上からそれぞれ “Asia Pacific” , “North America” , “Europe/Middle East/Central Asia/Africa” , を抽出したもの

Fig. 12 Layout results of model D with the LGL method and the proposed method, left and right, respectively. Top, Middle, and Bottom indicate “Asia Pacific”, “North America”, “Europe/Middle East/Central Asia/Africa”, respectively.

表 5 エッジの平均距離と分散

Table 5 Average and dispersion of the edge length.

method	$L_{box}$	$\langle x_{ij} \rangle / L_{box}$	$\sigma / \langle x_{ij} \rangle$
LGL	188.64	0.0103	1.551
FR-HI	340.48	0.0120	0.575

上とネットワークデータの肥大化にともない、より大規模なネットワークの計算が必要である。

Opte Project に用いられる LGL 法は、大規模ネットワークの可視化手法としても利用され、効率的かつ高速な手法であるが、その反面、計算の効率化のために本来近くに配置されるべきノードが遠方に配置されてしまうという問題が生じる。したがって、LGL 法を用いたネットワークの可視化は慎重に行わなければならないといえる。

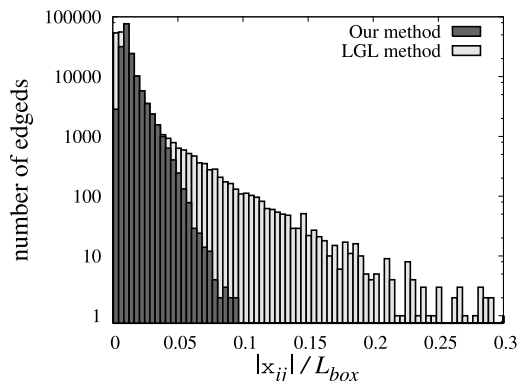


図 13 提案法および、LGL 法におけるエッジの分布  
Fig. 13 Histogram.

近年では CPU のマルチコア化が進み、容易に並列計算を行える状況が整っている。さらに GPU を利用した N 体計算<sup>12)</sup> も研究が進み、今後は提案法のように、MDGRAPE-3 などの並列処理手法を用いた、より高速かつ効率的な、大規模ネットワークの計算に耐えうる手法が必要である。

### 7. おわりに

本研究では更新粒子の選択効率を追求し評価するために、力を与える側は全ノードからの加速度を計算した。加速度計算の高速化手法として、FR のオリジナル手法では遠方からの力を完全に無視する方法が提案されているが、この手法は計算を高速化させる一方、可視化結果の品質については議論の余地が残る。しかしながら近年の手法では、FADE によって、Burns-Hut Tree code の重心近似を行うことにより計算の高速化を実現している。この手法は提案法に対しても実装可能であり、より高速かつ効果的な手法を構築することが可能である。我々は今後の課題として、提案法に対し Tree code などの近似手法を実装することを検討したい。

また、本研究で用いたネットワークデータは、LGL 法との対比のために Opte Project で用いられた IP 接続に基づくネットワークデータを利用した。今後はソーシャルネットワークなどを対象とし、ネットワークデータの特徴を考慮した、より効果的な可視化手法も検討したい。

### 参 考 文 献

1) Fruchterman, T.M.J. and Reingold, E.M.: Graph Drawing by Force-directed Placement, *Software — Practice and Experience*, Vol.11, No.21, pp.1129–1164 (1991).

- 2) Eades, P.: A heuristic for graph drawing, *Congressus Numerantium*, Vol.42, pp.149–160 (1984).
- 3) Kamada, T. and Kawai, S.: An algorithm for drawing general undirected graphs, *Information Processing Letters*, Vol.12, No.31, pp.7–15 (1989).
- 4) Quigley, A. and Eades, P.: FADE: Graph Drawing, Clustering, and Visual Abstractcion, *Proc. Graph Drawing 2000*, Lecture Notes in Computer Science, No.1984, pp.183–196 (2001).
- 5) Barnes, J. and Hut, P.: A hierarchical  $O(N \log N)$  force-calculation algorithm, *Nature*, Vol.4, No.324, pp.446–449 (1986).
- 6) Adai, A.T., Date, S.V., Wieland, S. and Marcotte, E.M.: LGL: Creating a map of protein function with an algorithm for visualizing very large biological networks, *Journal of Molecular Biology*, Vol.340, No.1, pp.179–190 (June 2004).
- 7) Lyon, B.: The Opte Project (2005). <http://www.opte.org/>
- 8) Ahmad, A. and Cohen, L.: A numerical integration scheme for the N-body gravitational problem, *Journal of Computational Physics*, Vol.12, No.389 (1973).
- 9) McMillan, S.L.W.: The Vectorization of Small-N Integrators, *Lecture Notes in Physics*, Vol.267, No.156 (1986).
- 10) Makino, J.: A Modified Aarseth Code for GRAPE and Vector Processors, *Publications of the Astronomical Society of Japan*, Vol.43, pp.859–876 (1991).
- 11) Narumi, T., Ohno, Y., Okimoto, N., Koishi, T., Suenaga, A., Futatsugi, N., Yanai, R., Himeno, R., Fujikawa, S., Ikei, M. and Taiji, M.: A 55 TFLOPS Simulation of Amyloid-

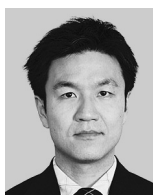
forming Peptides from Yeast Prion Sup35 with the Specialpurpose Computer System MDGRAPE-3, *Proc. SC06 (High Performance Computing, Networking, Storage and Analysis)*, CDROM, Tampa, USA (Nov. 2006).

- 12) Hamada, T. and Iitaka, T.: The Chamomile Scheme: An Optimized Algorithm for N-body simulations on Programmable Graphics Processing Units, *astro-ph/0703100* (2007).

(平成 19 年 2 月 2 日受付)

(平成 19 年 3 月 23 日再受付)

(平成 19 年 4 月 21 日採録)



松林 達史

昭和 50 年生。平成 12 年京都大学理学部物理学科卒業。平成 14 年 10 月より 2 年半、理化学研究所非常勤研究員。平成 17 年東京工業大学院理工学研究科地球惑星科学専攻博士課程修了。同年 NTT 入社。現在、NTT コミュニケーション科学基礎研究所研究員。主として可視化の研究開発に従事。理学博士。



山田 武士 (正会員)

昭和 39 年生。昭和 63 年 3 月東京大学理学部数学科卒業。同年 NTT 入社。平成 8 年より 1 年間英国コベントリー大学客員研究員。現在、NTT コミュニケーション科学基礎研究所創発環境研究グループリーダー。主としてネットワーク分析、機械学習、組合せ最適化等の研究に従事。博士(情報学)。電子情報通信学会、ACM、IEEE 各会員。