

# ハイブリッドシミュレーションに適応した 可視化ソフトウェア Akira の開発

中村 貴英,<sup>1,2,\*</sup> 河野 貴久,<sup>3</sup> 小林 亮,<sup>1,2</sup> 尾形 修司<sup>1,2</sup>

<sup>1</sup>名古屋工業大学大学院 創成シミュレーション工学専攻, 〒466-8555 名古屋市昭和区御器所町

<sup>2</sup>科学技術振興機構 CREST, 〒332-0012 埼玉県川口市本町4-1-8

<sup>3</sup>名古屋大学 太陽地球環境研究所, 〒464-8601 名古屋市千種区不老町

\*Email: nkmrtkhd@gmail.com

(Received: January 13, 2011; Accepted for publication: June 22, 2011; Advance publication: August 20, 2011)

対象系全体を必要な物理精度に応じて部分系に分け、各部分系に必要な精度を持つ計算手法を適用する同時並列型ハイブリッドシミュレーション法が最近注目されている。我々は、原子(粒子)描像であったり連続体描像であったりするなどのハイブリッドシミュレーションコードが出力する多様かつ多数な物理変数の数値を統一的に扱い、さらにそれらの時間変化をアニメーションとして可視化するソフトウェア Akira を開発した。Akira を用いると、原子描像と連続体描像を混在させたり、多彩に表示した原子群のダイナミクスを原子毎の軌跡として表示させたりするなど様々な形式で、シミュレーションの結果を物理的な解釈がしやすくなるように表現することが出来る。簡易な操作で時間変化を、静止画のシークエンスとしても出力出来る。ソースコードは Java 言語で OpenGL を用いて作成しているため、プラットフォームの違いを気にしないで利用出来る。ユーザーが独自に改変することも、ソースコードと共に開発資料を公開しているため容易である。本論文では Akira の特徴的な機能を、いくつかの応用例を通じて概説する。

キーワード: Visualization, Hybrid simulation, Java, OpenGL, Volume rendering

## 1 はじめに

ナノメートルスケールの構造体や原子配列を人為的に作り出し、バルクでは得られない特性や機能を実現するナノテクノロジーが工学の主要なテーマとなって久しい。ナノテクノロジーの原理は、量子力学に従う電子をナノスケールの原子群が作り出す境界条件を通じて制御すること、また逆に電子状態の特性を利用してナノスケールの機械(構造体)を実現することといえよう。目に見えないスケールを扱う技術であり、無数の設定が考えられることから、経験や勘に頼らない研究開発手法が必要となっている。走査型トンネル顕微鏡や電子エネルギー損失分光法を併用した透過電子顕微鏡など、各種顕微鏡の開発により個々の原子を観察する事が可能になっているが、化学反応過程のダイナミクス等を直接観測することは依然として難しい。このため、近年の計算機の急速な性能拡大と相まって、計算機シミュレーションにより、実験では観測しにくい原子のダイナミクスを調べ

ることが盛んになっている。特に、原子系に対しては経験的な原子間ポテンシャルを用いる古典分子動力学(Molecular Dynamics, MD)法、電子状態については分子軌道法や密度汎関数(Density Functional Theory, DFT)法などが広く使われており、これらの計算手法を多角的に適用して解析することが通常である。

シミュレーションの結果は一般的に、膨大な数の多次元の数値として出力される。シミュレーション結果を物理的に解釈する際には、数値の可視化が不可欠である。実際、学術用途ではフリーに使える可視化ソフトウェアとして、電子状態の計算結果を表示することに適した VESTA [1] や XCrySDen [2], MD 計算結果の表示に適した AtomEye [3], ParaView [4], VMD [5] などがあり、いずれも運用するプラットフォーム(Macintosh, Linux, Windows など)に合った実行コードを用いるとほぼ同じ使い勝手で利用出来る。単一の計算手法を用いた通常のシミュレーションの結果については、これらのソフトウェアを用いれば実用上問題無く解析出来る。

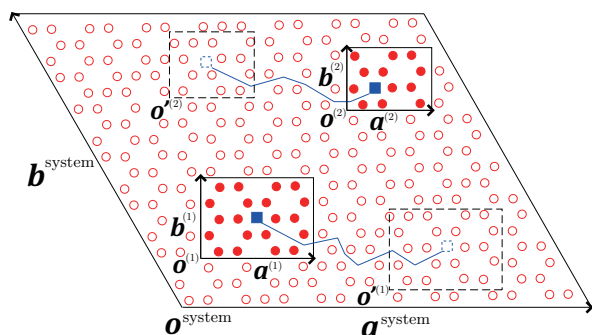


Figure 1. A schematic view of the hybrid QM-CL simulation. Two impurity atoms diffuse through crystalline host atoms. The circles are the host atoms, the squares are the impurity atoms. An open circle represents MD atom, while filled circles and squares represent DFT atoms. The vectors  $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ , and  $\mathbf{c}^{\text{system}}$  represent the total system box ( $\mathbf{c}^{\text{system}}$  is perpendicular to the sheet). There are two QM-regions that are treated by DFT. The origin and shape of each QM region are denoted by  $\mathbf{o}^{(\text{region})}$  and  $\mathbf{a}^{(\text{region})}$ ,  $\mathbf{b}^{(\text{region})}$ , and  $\mathbf{c}^{(\text{region})}$ . The QM regions move to trace the diffusion of the impurity atoms.

一方で、興味がある物理過程がスケール階層性を持ち、単一の計算手法を対象系全体に適用することが非効率であることも多い。例えば、材料の破壊過程においては、クラック先端部は化学ボンドの切断などにより電子状態が大きく変化するため明示的に電子を取り扱うことがしばしば必要となるが、その周辺部は歪み応力特性が表現出来るなら古典原子あるいは連続体として取り扱うことでも良い。そこで、全系を必要な物理精度に応じて部分系に分け、それぞれに必要な精度を持った計算手法を適用して同時並列的に全系の時間発展を追うハイブリッドシミュレーション法が最近注目を集めている。この方法を適切に用いると、原理的には実際に近いサイズの対象系を高精度にシミュレート出来ることになる。我々も、これまでにDFT法とMD法を組み合わせたハイブリッド量子-古典法 [6]、固体の為に粗視化粒子法とMD法を組み合わせたハイブリッド原子-粗視化粒子法 [7]、粗視化粒子法と流体計算法を組み合わせたハイブリッド粗視化粒子-流体法 [8]などのシミュレーションコードを開発し、様々な現象の解析を行ってきた。前述の可視化ソフトウェアはDFT法やMD法を単独に使用することを想定して設計されているため、これらのハイブリッドシミュレーションの結果の可視化には適しているとは言えない。またそのコード改変も、ソースコードが非公開であったり、高機能故にコードが複雑すぎるために、困難である場合が多い。そこで我々は、ハイブリッドシミュ

レーションの結果を、そのダイナミクスを含めて統一的に可視化するソフトウェア Akira (顕微鏡の'顕'に由来する)を開発し、広く公開する。

## 2 Akiraの設計思想

ハイブリッドシミュレーションの特徴を示し、その結果を統一的に可視化する為の Akira の仕様を説明する。

### 2.1 ハイブリッドシミュレーションの特徴

ハイブリッドシミュレーション法では、全系を必要な物理精度に応じて部分系に分け、それぞれに必要な精度を持った計算手法を適用して同時並列的に全系の時間発展を追う。ハイブリッドシミュレーション法の特徴を理解する為に、ハイブリッド量子-古典法 [6]を考えてみよう。ハイブリッド量子-古典法では全系を部分系に分割し、特定の領域のみをDFT法で扱い、他の領域をMD法で扱うことで精度を保ちつつ計算負荷を下げる。よってハイブリッド量子-古典法では、DFT法で扱う領域を全領域(シミュレーションボックス)と一致させない場合が多い。さらにDFT法での扱いが必要な領域が複数個になる可能性もある。

このようなハイブリッド量子-古典法の適用例として、結晶中に2つの不純物原子が存在し、その不純物原子が拡散する様子を調べることを考える(Figure 1)。結晶原子間に働く適切な経験ポテンシャルは既に提案されていても、不純物原子と結晶原子間のポテンシャルは存在しない場合が多い。このように必要な経験ポテンシャルが完全ではない系をシミュレーションするには、電子状態計算に頼ると良い。しかし電子状態計算は一般に計算負荷が高いため、全系を電子状態計算することは効率的ではない。この例では、この不純物原子周りの領域のみをDFT法で、他の原子はMD法で扱い、両者を接続するにはBuffered-Cluster法 [9]を用いる。DFT計算により力をMD計算にフィードバックし、時間発展はMD法で行うのがBuffered-Cluster法の特徴である。Figure 1では、全系が $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ , および $\mathbf{c}^{\text{system}}$ で張られるシミュレーションボックス(ただし $\mathbf{c}^{\text{system}}$ は紙面垂直)に入っており、不純物原子を四角で、結晶原子を丸で表示してある。そしてDFT法で扱う原子は塗りつぶされた四角および丸で、MD法で扱う原子は白丸で表示してある。2つの不純物原子が存在しているので、DFT法で扱うべき領域も2つ(原点が $\mathbf{o}^{(1)}$ と $\mathbf{o}^{(2)}$ の領域)ある。また $\mathbf{o}^{(1)}$ から $\mathbf{o}^{(1)}$ ,  $\mathbf{o}^{(2)}$ から $\mathbf{o}^{(2)}$ への移動のように、DFT領域を不純物原子の拡散に追従させる。このようにDFT領域が時間と共に移動し、さらにその形状( $\mathbf{a}^{(1)}$ ,  $\mathbf{b}^{(1)}$ ,  $\mathbf{c}^{(1)}$ )および( $\mathbf{a}^{(2)}$ ,  $\mathbf{b}^{(2)}$ ,  $\mathbf{c}^{(2)}$ )がシミュレーションボックス( $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ )

と一致しない事がハイブリッド量子-古典法の特徴である。つまりハイブリッド量子-古典法は原子座標と、局所的でありつつ時間と共に移動するボリュームデータを同時に出力する。その為、シミュレーション結果をダイナミクスとして可視化の際は、各時間フレームにおいて  $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ , 原子座標, ボリュームデータに加え,  $\mathbf{o}^{(1)}$ ,  $\mathbf{a}^{(1)}$ ,  $\mathbf{b}^{(1)}$ ,  $\mathbf{c}^{(1)}$ ,  $\mathbf{o}^{(2)}$ ,  $\mathbf{a}^{(2)}$ ,  $\mathbf{b}^{(2)}$ ,  $\mathbf{c}^{(2)}$  が必須となる。単一手法を用いる通常のシミュレーションではこのような状況はあり得ないであろう。

他のハイブリッドシミュレーション法はもう少し単純である。ハイブリッド原子-粗視化粒子法 [7]では、原子と粗視化粒子という違いはあるものの基本的に粒子情報が出力されるのみで、ボリュームデータは存在しない。ハイブリッド粗視化粒子-流体法 [8]では、流体計算(結果はボリュームデータとして出力される)の一部分に粗視化粒子があるという描像になり、シミュレーションボックスに一樣分布したボリュームデータと併せて粗視化粒子情報が出力される。総じて、様々な手法を組み合わせるハイブリッドシミュレーション法は、性質の異なるデータが同時に出力されるという特徴を持つ。

## 2.2 Akiraの要件と仕様

Sec.2.1で述べたハイブリッドシミュレーションに適応した可視化ソフトウェアAkiraの要件を考えていこう。まず可視化ソフトウェアとしては、シミュレーション結果を簡単にアニメーションとして表示させたり、静止画であっても説得力のある多彩なカラーで表示出来る必要がある。さらに具体的に考えていこう。ダイナミクスを表示する為には、一定の間隔で出力される多数の時間フレーム情報が必要である。ハイブリッド量子-古典法の各時間フレームにはMD法による原子毎のデータとDFT法による局所的な電子密度のボリュームデータが混在している。性質の異なるデータが同時に存在していることがハイブリッドシミュレーションの特徴であり、これらのデータを統一的に扱いアニメーションを作成したい。さらに多数の時間フレーム情報は、ファイルサイズが大きくなる傾向があるが、可視化した際にはストレスなく操作したい。また静止画でも説得力のある多彩な表現を行うために、原子のカラーを原子毎の物理量(運動エネルギーやポテンシャルエネルギー、変位など)によって決定したい。つまり、様々な計算手法を組み合わせるハイブリッドシミュレーションの結果は様々な形式のデータの複合となるが、これらを統一的に扱い、任意の物理量に基づく多彩なカラーで可視化し、そのダイナミクスをアニメーションとして表示することが必要である。

シミュレーションを用いた研究では一般に、パラメー

ター値を変えて何度も実行を繰り返し、それらの結果を比較することが通常である。可視化ソフトウェアは、直感的に操作出来ることと同時に、その操作再現性が高いことも必要と考える。さらにGaussian [10]やVASP [11], Amber [12]などといった広く使われているシミュレーションパッケージを利用するユーザーがAkiraを使う可能性も考慮すべきである。以上から我々は要件として以下の6項目を挙げる: (i)複数の時間フレームデータを高速に処理出来ること, (ii)ダイナミクスをアニメーションとして簡単に見られること, (iii)原子に加え局在するボリュームデータを同時に扱えること, (iv)様々な物理量をその値に依存したカラーで表現出来ること, (v)任意のファイルフォーマットに対応出来る拡張性があること, (vi)簡易に定型処理が行えること。

上記(i)から(iv)を満たす為、コードの仕様を以下とする: 第一に、前処理として多数の時間フレーム情報から可視化に必要なデータのみを集めた専用バイナリファイルを作成し、ダイナミクス表示におけるファイル入出力を高速化すること。第二に、ユーザーが与えた物理量に依存してカラーを選択出来るように、可視化ソフトウェア内で原子座標に加え各原子に補助データセットを保持すること。第三に、ボリュームデータの場合も、ただ一つの補助データを持つ原子と見なし、原子と同様の扱いを可能とすること。これらの方針に合わせて、独自のデータフォーマット(Akiraフォーマット)を提案する。また(v)に関しては、前処理を行なうコードに読み込み関数を追加していくことで対応する。なお(vi)に関連しては、Java言語に付属のGUIコンポーネントを利用する事で、マウス操作のみで定型処理を行えるようにする。

## 2.3 開発と管理

シミュレーションを実行する研究者が独自に可視化ソフトウェアを開発する意義は、自分で可視化ソフトウェアを改変することで自分の要求に合わせた最適な表現が可能になることにある。開発および改変が容易に出来るように、(i)コンピューターグラフィックスについての情報が得やすいこと, (ii)ソースコードの構造が単純であること, (iii)系統的にソースコードを管理出来ることが重要と考える。特に、グラフィックスライブラリと開発言語の選定には注意を払う必要がある。

OpenGLはコンピューターグラフィックスの分野で最も一般的なライブラリであり、関連書籍やサンプルコードも多数あり、Web等を通じて情報が得やすい。そこで我々はコンピューターグラフィックスライブラリとしてOpenGL [13]を採用する。また開発言語としてはJava [14]を採用する。Javaは、オブジェクト指向であり、サブルーチンの独立性をある意味で強制する言語である



ため、ソースコード全体の構造が単純となる。Javaで作られたソースコードはプラットフォームに依存せずに実行でき、GUIコンポーネントも付属しているため操作性もプラットフォームに依存しない。さらに、Javaにはソースコード中のコメントを自動的に抽出し、サブルーチンの機能一覧とそれらの依存性をhtmlドキュメントで生成する機能も付属しており、開発者向けの文章を作る事が容易である。ソースコードのバージョン管理については、Mercurial [15]を用いることにする。

### 3 Akiraの概要

Akiraは、データの前処理を担当するAkiraConverterと、実際の可視化を担当するAkiraViewerで構成されている。AkiraConverterは、様々なフォーマットで出力されるシミュレーション結果を読み込み、可視化に必要なデータのみを集めた専用バイナリファイル(Akira ファイル)を作製する。つまり多様なデータフォーマットの違いをこのAkiraConverterで吸収する。AkiraViewerは、このAkiraファイルを読み込んで画像化する。以下ではAkiraConverterとAkiraViewerの詳細を説明する。

#### 3.1 AkiraConverter

Akiraのユーザーは、シミュレーションの結果を1つの時間フレームのデータが1つのファイルとなるように出力しておく必要がある。ファイル名は任意で良いが、out001, out002, ...の様に、必ず連続した自然数を含む名前とする。現在、各時間フレームのフォーマットは以下の4つに対応している。特に要望が多いフォーマットに関しては、正式に対応する予定である。

- **Akira フォーマット**: 新たに設計した出力形式
- **CHGCAR フォーマット**: VASP [11]の出力形式
- **Cube フォーマット**: Gaussian [10]の出力形式
- **XYZ フォーマット**: gnuplotで読み込める形式

ただし原子のカラーを物理量に基づいて決定したい場合は、Akira フォーマットを用いなければならない。

次にこのAkiraフォーマットの詳細について述べる。Akiraフォーマットでは第1行目に、全原子数と各原子が持つ補助データの数、ボリウムデータの領域数、ボリウムデータのデータ点総数を書く。次にシミュレーションボックスを表すベクトル $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ を書き、続いて、各原子の情報を原子数個分書く。この原子情報には原子種および原子座標に加え、補助データとして任意の物理量を与える。ただし制約として、原子種は1から100の整数(これ以外の整数はAkira内部で使用する特別な値である [16]), 原子座標はベクトル $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ で規格化し(つまり第*i*番目の原子の座標の

$\mu$ 成分は $a_{\mu}^{\text{system}} r_{\mu}(i)+b_{\mu}^{\text{system}} r_{\mu}(i)+c_{\mu}^{\text{system}} r_{\mu}(i)$ である), 各原子あたりの補助データは最大9個までとする。これらの制約はAkira内部のデータ構造に由来している。 $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ に用いた単位が、長さの単位となる。その後、ボリウムデータに関する情報をボリウムデータの領域数分書く。ボリウムデータに関する情報とは、ボリウムデータの $\mathbf{o}^{\text{system}}$ から測った原点 $\mathbf{o}^{(\text{region})}$ および形状 $\mathbf{a}^{(\text{region})}$ ,  $\mathbf{b}^{(\text{region})}$ ,  $\mathbf{c}^{(\text{region})}$ とボリウムデータである。なお、ボリウムデータの原点もベクトル $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ ,  $\mathbf{c}^{\text{system}}$ で規格化されている必要がある。これらを図で示したものをFigure 2に示す。またAkiraフォーマットでシミュレーション結果を出力するコード例をFigure 3に示す。Figure 3の様にコーディングすれば、出力ファイル名は自動的に連番となる。

原子毎の補助データを記述するのがAkiraフォーマットの特徴であり、原子種あるいは補助データに基づいて各原子のカラーが決まる。Figure 2の例では、補助データとして各原子の運動エネルギーとポテンシャルエネルギーを出力するため、これらの物理量に応じたカラーでも原子を描画出来ることになる [17]。またボリウムデータを持つ領域が時間とともに移動することを考慮し、各時間フレームでのボリウムデータ領域の原点と形状の情報も記述している。

シミュレーションが完了し、全時間フレーム分のファイルセットをAkiraが動くコンピューターに転送した後、その格納ディレクトリ、時間フレームの総数、時間フレーム間の時間間隔 [18]を設定ファイルAkiraConverter.confに書き入れ、コマンドラインからAkiraConverterを以下の様に呼び出して前処理を開始する。

```
$ AkiraConverter.sh
```

Akiraファイルを一度作製すると、シミュレーション結果のデータに変更がなければ、再度の前処理は不要である。なお、時間フレームファイルがgzipまたはbzip2形式でアーカイブされた状態でも前処理は可能である。これは、アーカイブされたファイルを解凍する必要が無いため、ディスク容量の節約につながる。

#### 3.2 AkiraViewer

AkiraViewerは、以下のようにコマンドラインから専用のシェルスクリプトにより、AkiraConverterで生成したファイル名を付与して起動させる。

```
$ AkiraViewer.sh result.Akira
```

もしファイル名を付与せずに起動した場合には、ダイアログウィンドウが開くので、可視化したいAkiraファイルを選択すると良い。

AkiraViewerが持つ特筆すべき機能を以下に紹介する。

- **マウス操作による回転, 平行移動, 拡大縮小**: 系の回

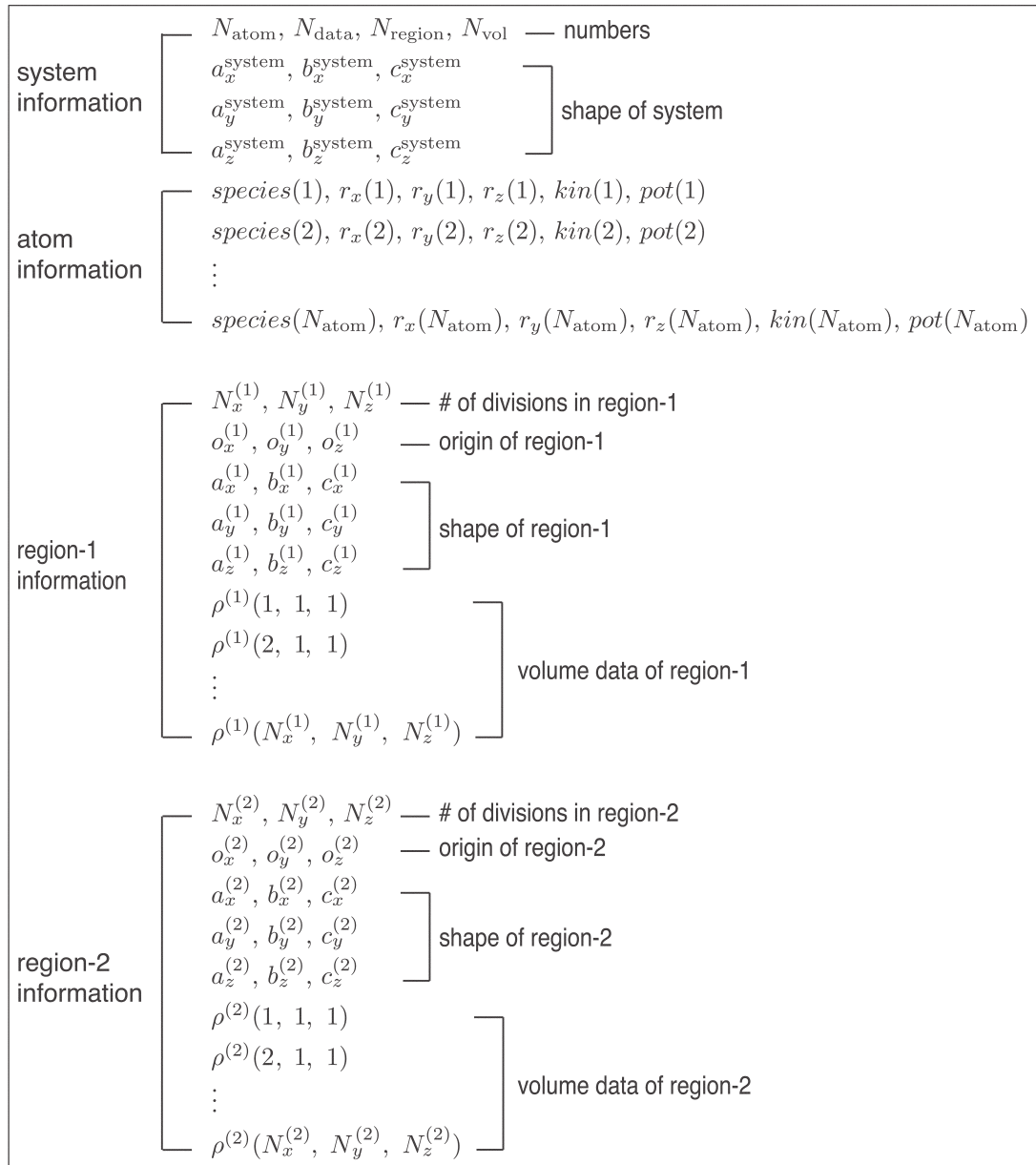


Figure 2. A schematic view of the Akira format. The  $N_{\text{atom}}$  is the total number of the atoms. The  $N_{\text{data}}$  is the number of the additional data for each atom. The  $N_{\text{data}}$  should be less than or equal to 9. The  $N_{\text{region}}$  is the total number of the volume-data regions. The  $N_{\text{vol}}$  is the total number of the volume data points, that is,  $N_{\text{vol}} = \sum_{\text{region}} N_x^{(\text{region})} \times N_y^{(\text{region})} \times N_z^{(\text{region})}$ . The  $\mathbf{a}^{\text{system}}$ ,  $\mathbf{b}^{\text{system}}$ , and  $\mathbf{c}^{\text{system}}$  are the vectors of the total system box. The  $species(i)$  is the species integer of the  $i$ -th atom. The  $\mathbf{r}(i)$  is the  $i$ -th atom's position normalized by the total system box. The  $kin(i)$  and  $pot(i)$  are the  $i$ -th atom's kinetic and potential energy as additional data. The  $\mathbf{o}^{(\text{region})}$  is the origin (i.e.,  $\rho^{(\text{region})}(1,1,1)$ ) of the volume data measured from the origin of the total system. The  $\mathbf{a}^{(\text{region})}$ ,  $\mathbf{b}^{(\text{region})}$ , and  $\mathbf{c}^{(\text{region})}$  are the vectors of the volume-data region. The  $\rho^{(\text{region})}(i,j,k)$  is the volume data. Note that the present data set is required for each time-frame file.

転はマウス左ボタンでのドラッグ, 平行移動はマウス右ボタンでのドラッグ, 拡大縮小はマウスのスクロールで

行う. なお, カーソルキーによる操作も可能である.

• **ダイナミクスの表示と動画の作成:** sキーを押すと

```

iframe_num=iframe_num+1           !increment frame-counter
filename='outXXX'                  !initialize output-file name
write(filename(4:6),' (i3.3)') iframe_num !set output-file name
open(10,file=filename)
write(10,*)n_atom,n_data,nv_region,nv_tot !numbers
write(10,*)a(1),b(1),c(1)             !shape of system
write(10,*)a(2),b(2),c(2)             !shape of system
write(10,*)a(3),b(3),c(3)             !shape of system

!output atom info.
do i=1,n_atom
  adata(1)=kin(i)                    !1st additional data: kinetic ene.
  adata(2)=pot(i)                    !2nd additional data: potential ene.
  write(10,*)isp(i),r(1:3,i),adata(1:n_data) !species, normalized pos., additional data
enddo

!output volume data
do iv=1,nv_region
  write(10,*)nv_x(iv),nv_y(iv),nv_z(iv) !# of division
  write(10,*)ov(1:3,iv)                !origin of volume data region
  write(10,*)av(1,iv),bv(1,iv),cv(1,iv) !shape of volume data region
  write(10,*)av(2,iv),bv(2,iv),cv(2,iv) !shape of volume data region
  write(10,*)av(3,iv),bv(3,iv),cv(3,iv) !shape of volume data region
  do iz=1,nv_z(iv)
    do iy=1,nv_y(iv)
      do ix=1,nv_x(iv)
        write(10,*)rho(ix,iy,iz,iv) !volume data (e.g., electron density)
      enddo
    enddo
  enddo
enddo
close(10)

```

Figure 3. A sample Fortran code that writes the hybrid simulation data at a time-frame in Akira format (ASCII). The `n_atom` is the total number of atoms. The `a(1:3)`, `b(1:3)`, and `c(1:3)` are the vectors of the total system box. The `isp(i)` is integer, that is the species of the *i*-th atom. The `r(1:3,i)` is the *i*-th atom's position that are normalized by `a(1:3)`, `b(1:3)`, `c(1:3)`. The `n_data` is the length of the additional data for each atom, that should be less than or equal to 9. The `adata(1:n_data)` contains additional data and users can set values of any physical quantities. The `nv_region` is the total number of the volume-data regions. The `ov(1:3,iv)` is the origin (i.e., `rho(1,1,1,iv)`) of the *iv*-th volume data measured from the origin of the total system. The `av(1:3,iv)`, `bv(1:3,iv)`, and `cv(1:3,iv)` are the vectors of the *iv*-th volume-data region. The `nv_tot` is the total number of the volume data points, that is,  $nv\_tot = \sum_{iv} nv\_x(iv) * nv\_y(iv) * nv\_z(iv)$ . The set of top four lines is useful to open many files repeatedly with a sequential number contained in each filename.

ダイナミクスがアニメーションとして表示される。ボリュームデータを含んでいると、その時間変化も同時に見ることが可能である。ある時間フレームにおいて、wキーを押すと描画面画がそのまま画像ファイルとして出力される。またshift+wキーで全時間フレームが自動的に画像ファイルのセットとして出力される。ダイナミクスを動画として保存するには、全時間フレームを画像ファイルとして出力し、出力された画像ファイルのセットを結合すれば良い [19]。

• **原子の多彩なカラーリング:** デフォルトでは、長さの単位がオングストロームであることを前提に、原子種

に基づいてカラー [20] と半径 [21] を与えている。なお、カラーと半径のデフォルト値はユーザーが変更出来る。また、Akiraフォーマットの特徴である原子毎の補助データに基づいて原子のカラーを決定することが可能であり、現バージョンではサーモグラフィ型、最大最小値強調型、虹色型、グレースケール型の4つの型から選択する。さらに、原子のラベルとして、原子種やシリアル番号などが表示出来る。

• **原子データと軌跡の表示:** 原子を選択すると、原子の持つ情報(原子種、座標、補助データ等)を表示する。原子を連続して選択することで、原子間の長さや結合

角、捻れ角を表示する。また選択した複数原子の個々の軌跡を描く事ができ、それらのカラーも変更出来る。

・**ボンドの描画:** アニメーションの表示を高速に行うためボンド情報はAkiraConverterで生成するが、AkiraViewerで動的に生成する事も可能である。ボンドは $O(N)$ で計算され、そのカラーは原子種やボンド長に基づいて決定される。

・**補助データのベクトルとしての描画:** Akiraフォーマットの特徴である原子毎の補助データから 任意に選んだ3つのデータを3次元ベクトルとして表示出来る。そのカラーは原子種やベクトル長に基づいて決定される。

・**任意平面でのスライス:** 任意の平面でスライスし、その断面を見ることが出来る。平面の指定は、法線ベクトルとその平面内の一点により行う。また、スライスした断面に含まれる各種データを 数値ファイルとして出力出来るので、別途にgnuplotなどのグラフソフトを用いて等高線等が描ける。

・**コピーによる拡大表示:** 周期境界条件を考慮して系を各方向に 無段階にコピー (シミュレーションボックス単位の整数倍ではない) して表示出来る。

・**視点の保存:** AkiraViewerで用いた視点を保存出来る。パラメーター値が異なるシミュレーション結果の比較の際に、特に有用となる。

・**簡易なバッチ処理:** 系を回転させながら画像出力するなどのバッチ操作を マウスを使って作成し、処理することが可能である。バッチファイルはテキストエディタでも編集出来る。

・**多様なフォントからの選択:** 利用したいフォントを、システムの利用可能なフォントリストから選べ、日本語の利用も可能である。文字カラーやサイズも変更出来る。

その他の機能や操作方法については、Akira付属のマニュアルに詳しく説明してある。

## 4 Akiraを用いたシミュレーション結果の可視化例

本セクションでは、様々なハイブリッドシミュレーションの結果をAkiraを用いて可視化した例を紹介し、前セクションで概説したAkiraの機能の有用性を示す。

### 4.1 ハイブリッド量子-古典シミュレーション

リチウムイオン二次電池は、既にPCや携帯電話などの電子機器を支える電池として利用されているが、これからは出力等を高めて自動車等の輸送機械に活用されることが期待されている。リチウムイオン二次電池の陰極

として利用されているグラファイト中で充電/放電時にLi密度が変化すると、ステージ構造と呼ばれる特異なLi分布を変遷すると共にLiの拡散係数が大きく変化することが実験で知られている。グラファイト中にLiが挿入すると炭素層の間隔が大きくなることから、我々は炭素層間隔とLi拡散の相関関係をハイブリッド量子-古典シミュレーションにより調べた [22]。このシミュレーションでは、Liの拡散に合わせてLiとその周辺の炭素原子群のみを動的に選択しながらDFT法で扱い、その他の炭素原子は経験的な原子間ポテンシャルで扱っている。

ある炭素層間隔に設定した場合のLi拡散の様子をFigure 4に図示している。Akiraの機能により、DFT法で扱っている領域について、Liを通る平面でスライスした電子分布をその密度に依存したカラーで表現している。Liの軌跡を表示することで、その拡散挙動が直感的に良く理解出来る。

### 4.2 ハイブリッド原子-粗視化粒子シミュレーション

材料の破壊過程は、内部応力場が亀裂先端から測った距離の0.5乗に逆比例する長距離場であるために、材料形状や外部応力に鋭敏に依存し、本質的にマルチスケールな過程であることが知られている。我々は、Ar系を良く再現するLennard-Jonesポテンシャルで表現された2次元固体の亀裂進展過程について、亀裂先端近傍はMD法で、それ以外の領域は統計熱力学を基礎に構築した粗視化粒子(Coarse-Grained Particle, CGP)法で扱うハイブリッド原子-粗視化粒子シミュレーションを行った。全系をMD法で扱わないために、大規模系にもかかわらず亀裂の応力場を適切に考慮したシミュレーションが出来た。

原子および粗視化粒子の歪みを補助データとして与えることで、ある時刻での歪み場の様子を、現バージョンのAkiraに組み込まれている4通りのカラー型毎にFigure 5 (a)-(d)に示している。これらの図から、MD領域とCGP領域の境界をまたいで、歪みがシームレスに変化していることが良くわかる。また、原子を粗視化しているCGP領域では、MD領域よりも特性長が長いために、その長さ未満の波長の波動は伝播できない。本ハイブリッドシミュレーションに導入したスケール間接続法 [7]が理想的に機能し、MD領域に存在する短波長の波動がMD-CGP境界で反射していないことも良くわかる。

### 4.3 ハイブリッド粗視化粒子-流体シミュレーション

近年、グラフェンのように方向に依存してナノスケールとメゾスケールを特性長に持つ構造体と流体との相互



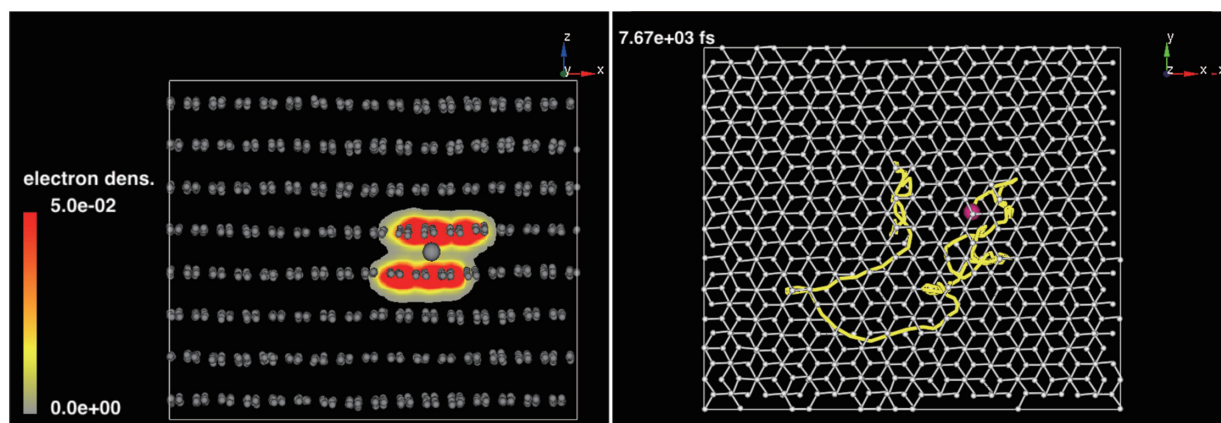


Figure 4. The hybrid QM-CL simulation results for a single Li in graphite. The QM region treated with the DFT method is selected adaptively around the Li [22]. (Left) All the atoms in the total system are shown. As for the QM region, the electron density ( $a_B^{-1}$  unit,  $a_B$  = Bohr radius) on the Li-plane parallel to the screen is depicted with colors. (Right) The trajectory of the Li.

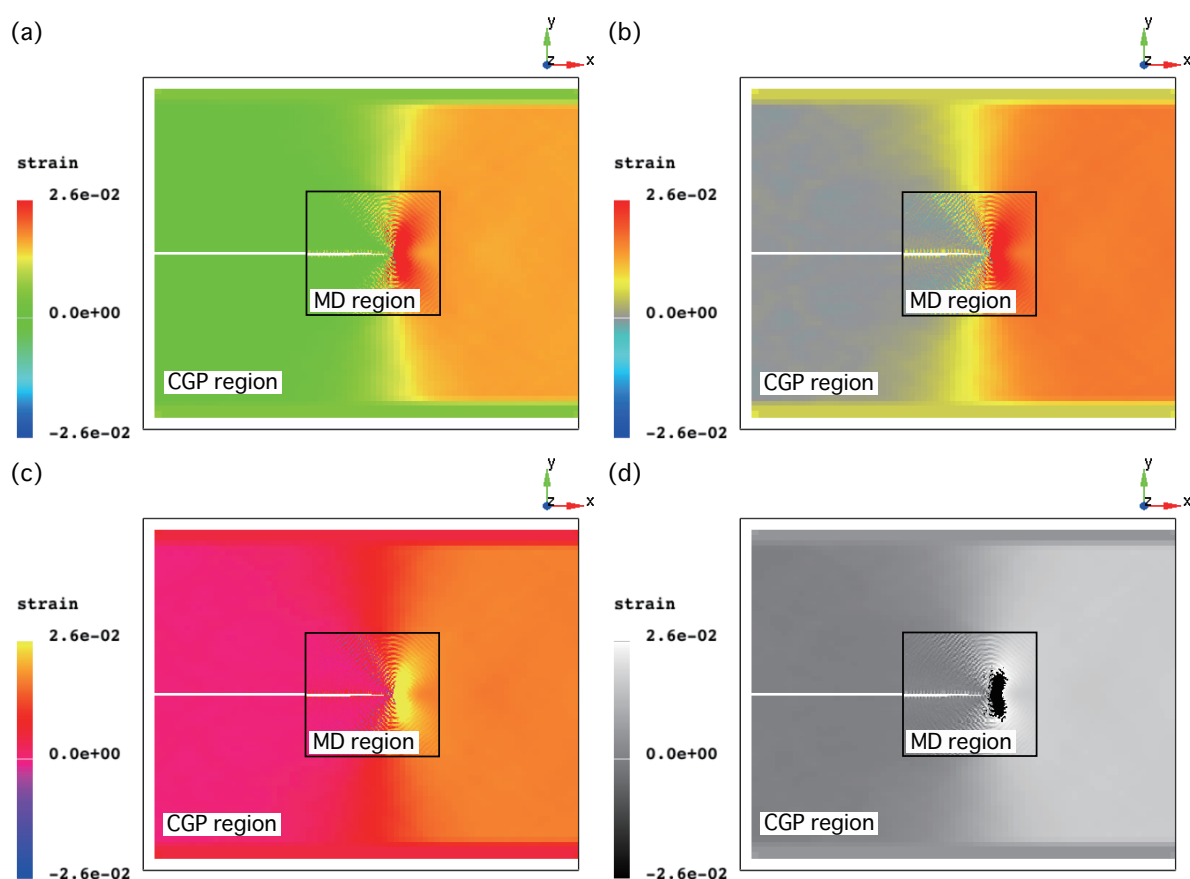


Figure 5. The hybrid atom-CGP simulation results for the crack propagation in the Ar system in 2D. A region around the crack tip is treated with the MD method. The strain field is drawn with various colorings: (a) the thermography type, (b) the max-min-enhanced type, (c) the rainbow type, (d) the grayscale type.



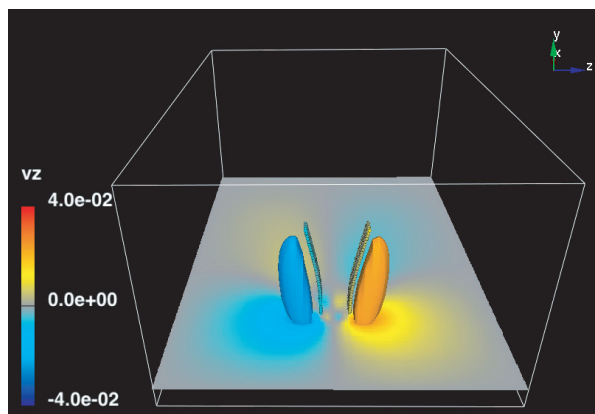


Figure 6. The hybrid CGP-fluid (LB method) simulation results for the two graphenes in air flow [8]. Both  $v_z$  distribution on a selected plane and isosurfaces of  $v_z$  (LB unit) that equals to  $\pm 2.4 \times 10^{-2}$  are shown with colors, to be compared with the local strains of the graphenes depicted also with colors.

作用が、ナノ流量計、ナノアクチュエーター等への応用から注目されている。この場合、流体の特性スケールはマイクロメートル超であり、構造体のスケールと大きく異なる。質量密度についても流体と構造体では数桁も異なる。時空間のスケールギャップが大きいため、原子を直接扱うMD法を適用して全系を解くことは現実的でない。他方、これまで通常行われてきたように、構造体を単純なバネビーズ系でモデル化し、さらにその質量やバネ定数を流体計算がしやすい値に設定することも適切でない。そこで我々は、実際の物理パラメーターを用いつつ、グラフェンを構成している炭素一層の原子系をCGP法により統計熱力学に基づいて第一原理的に粗視化し、格子ボルツマン(Lattice Boltzmann, LB)法で表現した流体計算と動的にハイブリッド化してシミュレートした。

例として示すFigure 6では、1マイクロメートル程度の大きさの矩形グラフェン2枚を同程度の距離だけ離して基盤に設置し、気体を流している。シミュレーションの結果、流速が小さい(レイノルズ数は40程度)にもかかわらず、グラフェン全体がその面垂直方向の柔らかさを反映して大きく振動すると共に、その周囲の流体には時間変化する渦が生じることが分かった [8]。流体のボリュームデータを用いて、 $0.1 L_y$  を通る  $y$  軸に垂直な平面内の  $v_z$  (流体速度の  $z$  成分) 分布、および  $v_z$  (LB unit)  $= \pm 2.4 \times 10^{-2}$  である等値面を表示し、グラフェンの変形は粗視化粒子毎のカラーで表示してある。 $v_z$  の等値面に沿ってグラフェンが変形している様子が良くわかる。

## 5 まとめ

我々は、同時並列型ハイブリッドシミュレーションの可視化を念頭に、シミュレーション結果の解析に利用しやすいようダイナミクスを表示する可視化ソフトウェア Akira を、Java 言語で OpenGL を用いて開発した。現バージョンでは、ユーザーが与えた物理量(補助データ)に依存したカラーで原子を表現でき、マウスを用いた簡単な操作でダイナミクスの表示およびその画像出力が行える。シンプルなソースコードを心がけており、各種ドキュメントも整えてあるため、他のソフトに比べユーザーは自分の要求に合わせた改変が行いやすいと思われる。原子のカラーを物理量に基づいて決定したい場合は Akira フォーマットを用いなければならないが、現バージョンでは、CHGCAR フォーマット、Cube フォーマット、XYZ フォーマットの読み込みも可能である。もしこれら以外のフォーマットを利用しているユーザーは、対応しているフォーマットに変換するか、新たに AkiraConverter に読み込み関数を組み込むことで Akira を利用して欲しい。

今後の課題には、MPI (Message Passing Interface) やマルチコア CPU を用いたデータ処理の高速化、近年発展著しい GPU (Graphics Processor Unit) を使った描画の高速化、リモートサーバーで Akira を起動させ描画面のみをローカル PC で表示する遠隔可視化などがある。これらの課題と共にユーザーの要望をふまえた改良を進める予定である。

Akira は、無保証、著作権表示の保持、同一ライセンスの適用の下で複製・改変・再頒布を認めるという GNU General Public License (GPL) の下で、以下のサイトで公開する：<http://code.google.com/p/project-akira/>

## References

- [1] K. Momma, F. Izumi, *J. Appl. Crystallogr.*, **41**, 653 (2008). [[CrossRef](#)]
- [2] A. Kokalj, *Comp. Mater. Sci.*, **28**, 155 (2003). [[CrossRef](#)]
- [3] J. Li, *Model. Simul. Mater. Sci. Eng.*, **11**, 173 (2003). [[CrossRef](#)]
- [4] A. Henderson, P. Guide, A Parallel Visualization Application. Kitware Inc., 2007.
- [5] W. Humphrey, A. Dalke, K. Schulten, *J. Mole. Graphics*, **14**, 33 (1996). [[CrossRef](#)]
- [6] S. Ogata, Y. Abe, N. Ohba, R. Kobayashi, *J. Appl. Phys.*, **108**, 064313 (2010). [[CrossRef](#)]
- [7] R. Kobayashi, T. Nakamura, S. Ogata, *Int. J. Num. Meth. Eng.*, **83**, 249 (2010).

- [8] Y. Inoue, R. Kobayashi, S. Ogata, T. Gotoh, *Comp. Model. Eng. & Sci.*, **63**, 137 (2010).
- [9] S. Ogata, *Phys. Rev. B* **72**, 045348–1–17 (2005).
- [10] M. J. Frisch, et al., *Gaussian 03* (Gaussian, Inc., Wallingford, CT, USA, 2003).
- [11] G. Kresse, J. Hafner, *Phys. Rev. B*, **47**, 558 (1993). [[CrossRef](#)]
- [12] D. A. Case, et al., (2010) AMBER 11, University of California, San Francisco.
- [13] <http://www.opengl.org> Java からOpenGL を利用する場合はJava OpenGL(JOGL, <http://jogamp.org>) というライブラリを使う。
- [14] <http://www.java.com>
- [15] <http://mercurial.selenic.com>
- [16] 例えば, Akira内部ではボリウムデータを原子種が101の原子として扱っている. ボリウムデータは, AkiraConverterがその座標を生成し, ただ一つの補助データを持つ原子として処理しているに過ぎない. よって, ユーザーがボリウムデータの座標を計算し, 補助データにボリウムデータを与えた原子種101の原子とすることでもボリウムデータの可視化が可能である.
- [17] Buffered-Cluster法では, 時間発展をCL計算で行うため, 全原子に運動エネルギー, ポテンシャルエネルギーを定義する事が可能である.
- [18] 通常, シミュレーション結果をダイナミクスとして表示する際, シミュレーション結果は一定の間隔で出力される. よって, Akiraでは時間フレームの間隔を設定ファイルAkiraConverter.confに書くのみで, 各時間フレームのファイルには書かない.
- [19] 画像を結合して動画を作るには, 例えば ImageMagick(<http://www.imagemagick.org>)というコマンドラインツールを用い, `$convert -delay 20 *.png case1.mpg` とすれば良い. ImageMagick 以外にも画像を結合して動画を作るソフトは多数にあるので, 自分の好みに合ったソフトが利用出来る.
- [20] 化学分野でよく使われる, Corey,PaulingおよびKuntunによって作られたCPK テーブル.
- [21] E. Clementi, D. L. Raimondi, W. P. Reinhardt, *J. Chem. Phys.*, **38**, 2686 (1963). [[CrossRef](#)]
- [22] N. Ohba, S. Ogata, R. Kobayashi, S. Yamakawa, S. Hyodo, in *Proceedings of Multiscale Modeling of Materials 2010*, Eds. P. Gumbsch and E.v.d. Guessen, (Fraunhofer Inst., Germany, 2010), p. 110–113.

## The Visualization Package "Akira" for Analyses of Various Hybrid Simulation Results

Takahide NAKAMURA,<sup>1,2\*</sup> Takahisa KOUNO,<sup>3</sup> Ryo KOBAYASHI,<sup>1,2</sup> and Shuji OGATA<sup>1,2</sup>

<sup>1</sup> Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan

<sup>2</sup> CREST, Japan Sci.&Tech. Agency, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan

<sup>3</sup> Solar-Terrestrial Env. Lab., Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan

\*Email: [nkmrtkhd@gmail.com](mailto:nkmrtkhd@gmail.com)

The concurrent-type hybrid simulation scheme, in which a total system is divided into sub-systems and a proper numerical method is applied to each of them, has attracted much attention in recent years. We have developed a visualization package "Akira" to be applicable directly to analyses of various hybrid simulation results composed usually of both atomic (particle) and volume data. Akira can display the simulation results in impressive manners, for example, by showing both atomic and volume data with colors relating to the values of physical quantities, and by showing the traces of selected atoms. A user can easily create sequential image-files for the time-evolution. Akira is written in Java with OpenGL; hence, it is executable on various platforms. The source code of Akira is open for the public along with its developing documents. The characteristic functions of Akira are explained with demonstrations.

**Keywords:** Visualization, Hybrid simulation, Java, OpenGL, Volume rendering