

Title	スーパーコンピューターSX4を用いる画像対話型動力学計算の試み
Author(s)	齋藤, 賢一; 稲葉, 武彦
Citation	大阪大学大型計算機センターニュース. 1999, 112, p. 138-146
Version Type	VoR
URL	https://hdl.handle.net/11094/66340
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

スーパーコンピュータ SX4 を用いる画像対話型動力学計算の試み

大阪大学大学院 工学研究科 機械物理工学専攻
齋藤 賢一・稲葉 武彦

saitou@mech.eng.osaka-u.ac.jp. inaba@mech.eng.osaka-u.ac.jp.

大阪大学工学部 機械工学科
Tan Soon Keong

1 本モニター活動の趣旨と目的

近年のコンピュータ環境(家庭用ゲームも含めて)におけるグラフィックの役割は増加の一途をたどっている。それに伴い、ハードウェア的にもソフトウェア的にも日進月歩の開発および供給がなされている。

これまでの数値シミュレーションでは、専門家がスーパーコンピュータなどを駆使して計算を行ない、得られた数値データを解析し理解し、余力があれば画像に変換することを行ない、さらに理解し、専門家間での議論に供してきた。だが、計算が大規模かつ高精度になるに従い、これらを行なうための数値データが膨大となり、そのハンドリングが厄介になってきた。そのために、リアルタイムで計算結果を表示すること、さらに進んでリアルタイムかつ対話的にシミュレーションを行なうことが必要となると今後は予測される。また、専門家がある程度予測を立てて行なったシミュレーションにはある意味で発見的な要素は乏しいといえないだろうか。実際のところ、専門家でも、あまり予測の立たない現象をシミュレーションしたい場合も多くあるわけで、このような対話的な実行が可能であれば、専門家の人、分野外の人、一般の人、の別を問わず、純粋に物理的(自然科学的)アイデアをとともに交えさすことが可能になるだろう。

このような大義名分／一般論はさておき、このモニター活動では、スーパーコンピュータ SX4 の画像処理性能を探ることに大きな目的がある。これはつぎの安易なユーザーの要求に基づく⁽¹⁾。

- 大型計算機センター(以下、大計センター)に行って、SX4 で計算する際に、画像でその状況や結果を見たい。
- 大計センターに行かなくてもオンラインで SX4 に接続して計算させると同時に結果を見たい。
- 【新しい要求】スタートレックのように¹、計算の途中に自分の作用を入れたい。

計算の対話性に関してのメリット、技術的なことなどは既に十分に論じられてきており⁽²⁾⁽³⁾、当センターでもスーパーコンピュータ(歴代の SX)を対象としたその試みはもう既に始まっている⁽⁴⁾と認識している。また、分子シミュレーションにおいても画像的に対話するシステムがすでにある⁽⁵⁾⁽⁶⁾。しかし、上の率直で安易な要求と絡めてスーパーコンピュータでの対話的計算の行ない方、可能性を検討してみたいということが本モニター活動の立場である。

2 対話型分子動力学法について

2.1 分子動力学計算における対話計算の有効性

分子動力学法(MD法)²では、物質を(古典的な場合)その基本構成要素である原子／分子の粒子集合体として捉え、それらの相互作用から質点系の運動方程式を立てて、各々の運動の時間発展を求めていく⁽⁷⁾。

¹そこでは、リクリエーションとして、コンピュータが作り出してくれる仮想的な現実に関心が没頭できたり、シミュレーション結果をその場で表示してくれたりする。バーチャルリアリティとかいう大げさなものにはする気はないが、思想的には多分それに近い考えだと思う。

²原子を扱っていても、"分子"動力学法と呼ぶことが多い。

すなわち、MD 法はその名の通り、動力学的を見ることにその本質的なメリットがある。統計力学の理論の支えを得て平衡系としての扱いがこれまで主であったが、外力に対する応答の過程などの非平衡的な場として捉え得る解析に威力を発揮する手法である。そして、人間が対話的に計算対象をいじる³ということは、非平衡な場を人間が意識的⁴に作り出していることになる。この性質をうまく使えば、非平衡系に現れるあらゆる状況を意のままに設定できるかも知れない。

連続体近似の計算と異なり、MD 法のような粒子離散系での基本的な出力量は粒子の位置と速度と種類が少ないことから、計算対象の中で操作を与える範囲を限定することが比較的容易である。同じ理由から、局所的な変化を視覚的に捉えることも粒子離散系では容易であるので、対話方式には馴染みやすい。対話型シミュレーションとして、MD 法は良い実験材料である⁵といえる。また、MD 法が威力を発揮するのは比較的微小な部分の現象であるので、実生活の感覚による予測とは異なった状況になる可能性があり、対話的な手法はかなり前から要望されていたはずである。

2.2 SX4 でリアルタイム対話型計算をどうやるのか

昨年度のモニター活動の経験⁽¹⁾ から、SX4 における OpenGL (3次元グラフィックライブラリ) の利用にある程度の期待が持てたので、今回はそこで利用した OpenGL ライブラリを使う 3次元可視化ソフトウェア⁽⁸⁾を基として対話型計算プログラムへと発展させることにした。OpenGL⁽¹⁰⁾⁽¹¹⁾⁽¹²⁾ は根本から 3次元的な考え方で構成されているので、視点の回転、拡大、縮小、などのビューの変換がリアルタイムで行なえる。また、時間に伴って変化する画像に対しての処理 (つまりアニメーション) も考慮されている⁶ので、リアルタイム処理は難なくこなせる (と思っていた)。まづもって、業界標準の 3次元ライブラリであること⁷が強みである。

2.3 MD 計算の対話的可視化

『対話型シミュレーション』の環境は、D.C.Rapaport によると、次のように解釈される⁽⁹⁾。

"one in which the simulation is run interactively, with the experimenter being able to change the parameters of the simulation and the mode of visualization at will as the computation proceeds."

すなわち、シミュレーションの計算をしながらも、計算のパラメータや表示方法の変更が可能な状況であると考えれば良い。上記の定義に基づいて、対話的性質は次のような制御に分けることができる。まず、シミュレーションのパラメータ、例えば MD 法の場合、原子数、位置、温度などに変更を加えることができるような制御がある。一方で、出力している画像の表示の仕方、例えば、対象物の拡大・縮小や回転を可能にする制御が求められる。

今回は、基本思想として、グラフィック表示プログラムに MD 計算のステップを組み入れることにした⁸。具体的には、図 1 のようにする。最初に用意した MD 計算プログラム⁹はフォートラン言語で書かれていたが、グラフィック表示プログラムとの整合性を取るために、あえて前もって C 言語に変換しておいた。

2.4 対話型インターフェース

具体的な対話型インターフェースを考えるに当たり、以下に一般的に良いと考えられる対話型インターフェースの条件をを列挙しておく。

³系にとっては初期条件や境界条件が突如変更されるという事態になる。

⁴人間の意識が自然現象に入るべきかどうかの疑念が湧くが、よくわからない。

⁵逆に、アウトプットの種類の少なさは離散的な系で起きている現象の捉えがたさにつながり、MD 法の弱点になるとも言える。大量、複雑になった場合に、捉えどころが無くなる。

⁶OpenGL に限らずだが、ダブルバッファ的な機能が効果的である。

⁷この性質のために、プラットフォームによらずに様々なマシンで実行可能である。これは不精して研究室でも計算させたい向き (私) にはとても良い性質である

⁸これが唯一の方法ではないことを断っておく。ただし、OpenGL の画像対話的機能を使うにはこれが手っ取り早いと言える。

⁹分子動力学法では原子間ポテンシャルの設定が重要である。ここではアルゴンを表わすレナードジョーンズ型を用いた。

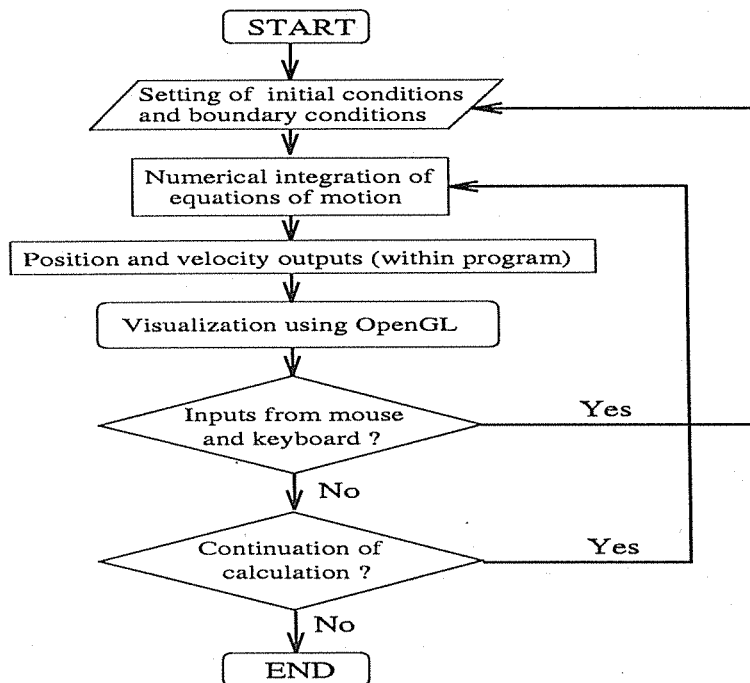


図 1: 対話型分子動力学シミュレーションに用いたフローチャート

1. 入力に対して反応が早いこと

計算は、一塊ではなくて、幾つかに分割して実行する。ユーザーの割り込み処理後、すぐに計算を再開することで反応は速くなる。また、割り込みができないときはその旨をユーザーに知らせるために何らかの表示があるのが好ましい。

2. 操作手順の学びやすさ及び簡単さ

反応が速くても操作しにくいと使いやすいとは言えない。従って、簡単な操作が必要である。コマンドを覚える必要のないメニュー方式やアイコンで操作を行うことが好ましい。また、マウスとキーボードへの機能の割り振りを適切にする。

3. 平穏に画像を表示すること

ディスプレイ・フラッシング (display flashing)¹⁰が起きないようにする。この問題はダブル・バッファリング (double buffering)¹¹によって解決できる。また、結果でも述べているが、計算と表示回数調節によって平穏な画像表示が実現できる。

この対話性に加えてリアルタイム性⁽¹³⁾があることで、対話型シミュレーションが実現する。曖昧な書き方ではあるが、ユーザーからの割り込み入力と応答との関係の透明さがリアルタイム性である。割り込み入力をしてから、画像表示の反応 (応答) が余りに遅いと対話的性質のメリットがなくなる。

今回は『見えるものを見るまま操作する』という発想に基づいて、ある直接画面操作型インターフェースを作った。このインターフェースの利点はユーザーが意味論 (セマンテックス) のレベルからデータを把握し、変更することを可能にした点にある。例えば、キーボード入力による加工では、原子識別番号がユーザーにとって必要となるが、本来原子は識別番号を持っていないわけで、どちらかというプログラムす

¹⁰ 画像が直接画面に映るときの画像の不連続再生のこと。

¹¹ この方式では、画像は計算直後に直接画面に映るのではなくて、前もってメモリの中で作られたものが映る。

るための人為構造，構文的データに過ぎないと言える。これよりも，見えている原子をマウスにより直接加工する方が自然であろうという考え方である。

2.5 イベント駆動の対話型 MD プログラム

対話型 MD プログラムでは，その一番上層で無限ループを回して，ユーザーからの作用（すなわち，キーボードやマウスからの入力）を待ち続ける。プログラム上では，auxMainLoop(ModelDraw)の部分である。

```
realdraw()          /* main engine of program */
{
    auxIdleFunc (ModelSpin);
    /* auxIdleFunc (ModelDraw); */ /* When spin isn't needed. */
    auxReshapeFunc (ReshapeWindow);
    /***** View control *****/
    auxMouseFunc(AUX_LEFTBUTTON,AUX_MOUSESDOWN,DragOn);
    auxMouseFunc(AUX_LEFTBUTTON,AUX_MOUSESUP,DragOff);
        ⋮                ⋮                ⋮                ⋮
    /***** Position Display *****/
    auxMouseFunc(AUX_MIDDLEBUTTON,AUX_MOUSESDOWN,AtomPosition);
        ⋮                ⋮                ⋮                ⋮
    /***** Computation control *****/
    Starting calculation *****/
    auxKeyFunc(AUX_g,CalcStart);
        ⋮                ⋮
    auxMainLoop(ModelDraw);
}
```

ユーザーからの割り込みがあればそれを解釈する関数が呼び出される。上記では auxKeyFunc() および auxMouseFunc() によって実現されている。

3 対話型分子動力学計算の検討結果

3.1 SX4 でのリアルタイム対話型分子動力学計算の実行

今回主に示すのは極めて少数の原子を扱った事例⁽¹⁴⁾である。分子動力学法の有効性を論じるのにはまだ隔たりはあるが，後程示すように，画像処理はシステムにかなりの負担がかかるために，現時点での効率的な対話型シミュレーションが実行できるように，そのように低負荷にした。

プログラムを実行しているときの状態を図 2 に示す。図 2(a) の方はシミュレーション状況の把握のために表示している端末ウィンドウの状態である。図 2(b) には OpenGL により生成されたグラフィックを示す。

3.2 シミュレーションの制御

シミュレーションの制御としては大きくわけて次の 3 種類を可能とした。括弧内は使用する装置である。

1. 計算の制御 (キー)
2. 視点の回転・遠近 (マウスのクリック+ドラッグ)
3. シェーディング表示の切り替え (キー)
4. 対話的に粒子位置を変える (マウスのクリック)

4 つめの要素に計算に対する対話的性質がとくに盛り込まれている。ここで，ユーザーはマウスの操作で，予め指定しておいた原子の位置を好きなところに変更することができる¹²。制御キーや表示の切り替え方法をまとめて，表 1 に示す。

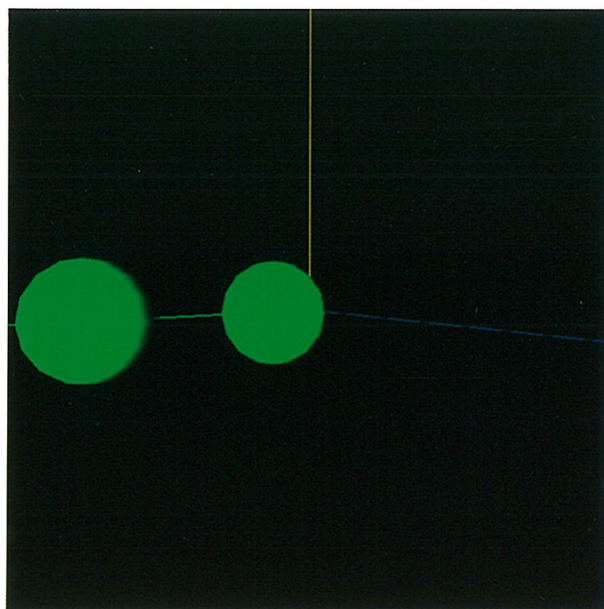
¹²計画では，OpenGL のセレクションモードの利用などをして，任意の粒子の位置を変更したかったが，現在のところ未実現である。

```

(Ns,Nstep) 194 200
194 -1.54512e-21 1.44778e-21
AtomSphere (ON)
(Ns,Nstep) 195 200
195 -1.49285e-21 1.39552e-21
AtomSphere (ON)
(Ns,Nstep) 196 200
196 -1.43539e-21 1.33805e-21
AtomSphere (ON)
(Ns,Nstep) 197 200
197 -1.37553e-21 1.27819e-21
AtomSphere (ON)
(Ns,Nstep) 198 200
198 -1.31523e-21 1.21789e-21
AtomSphere (ON)
(Ns,Nstep) 199 200
199 -1.25582e-21 1.15848e-21
AtomSphere (ON)
(Ns,Nstep) 200 200
200 -1.19816e-21 1.10082e-21
AtomSphere (ON)
CPU TIME [s]= 25.59
ELAPSED TIME[s]= 364.00
annie:/export/home/tan/mds/realmd1>

```

(a) 計算をチェックするための出力



(b) リアルタイムの画像表示 (球が原子を表わす)

図 2: 対話型分子動力学プログラムの結果 (実行風景)

表 1: 対話型分子動力学シミュレーションで可能な操作の詳細

utilities	action	purpose	explanation
key	g	MD step	starting simulation
	z	MD step	temporary stopping simulation
	x	MD step	continuation of simulation
	r	graphic	reseting view
	s	graphic	selection of shading-mode
	SHIFT + q	MD step	forced termination of calculation
mouse	right click + drag	graphic	enlargement and reduction of object's size
	left click + drag	graphic	rotation of view point
	middle click	interaction	pointing new position for a atom

3.3 対話的に変化を加える時の状況

対話的に変化を加える時の状況は、ある平面上に配置した9個の原子の系について検討する。対話する(変化を与える)前の状態およびその後の状態を図3に図示する。また、そのときの系の持つポテンシャルエネルギーと運動エネルギー及びそれらの総和の変化をグラフにして図4に示す。何もなければ設定条件として全エネルギー(ポテンシャルエネルギーと運動エネルギーの和)が保存されるが、対話があるたびにステップ状の変化を含むようになる。その後ある程度の時間(グラフの横軸)で、ポテンシャルエネルギーが小さくなる一方で運動エネルギーが増大するという緩和過程が再現されている。また、図5のように、計算の途中でビューの変換を行なうことも可能であり、これは原子間の位置関係を把握するのに有用である。

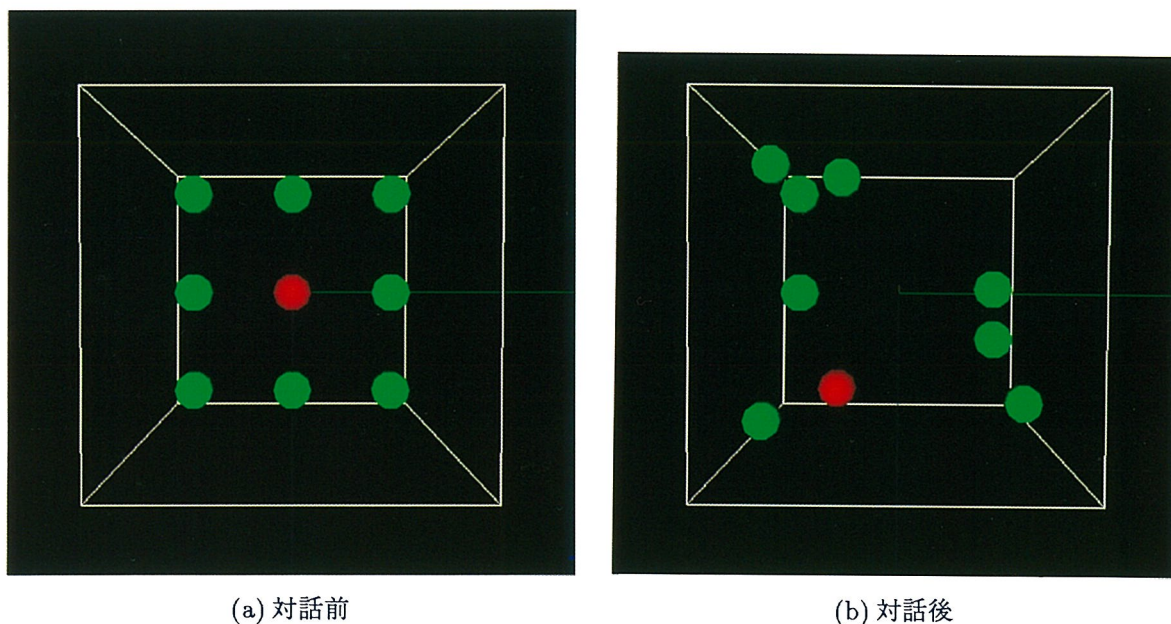


図 3: 対話の前後の原子配置の比較

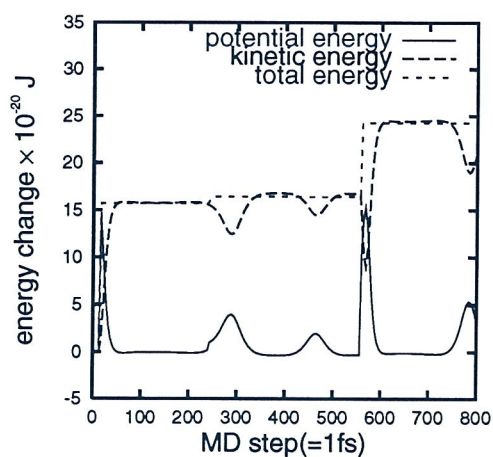


図 4: 対話が生じているときの各エネルギーの時間変化

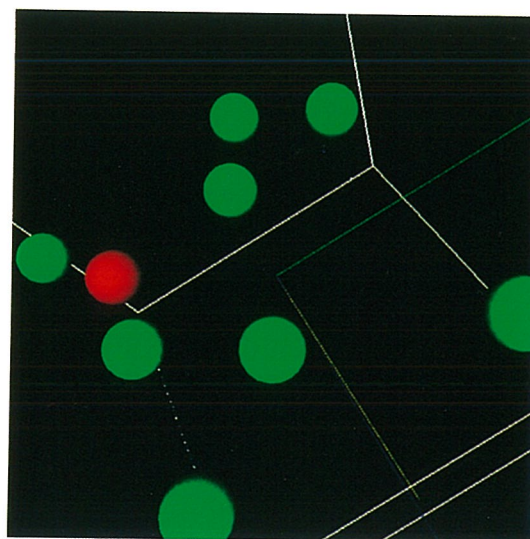


図 5: 対話計算途中のビューの変換

3.4 考察 1: 利用形態によるパフォーマンス比較

SX4で計算を実行するにしても、実際に画像をどこで表示するかに至っては様々な選択肢がある。そして計算機によって差が感じられるので、できるだけ定量的に検討したいと思った。強力な画像処理専用ワークステーションは大計センターでは使えるが、研究室では使えることは稀であるので、かなり旧型のマシンなどについても試すことにした。

計算を実行する側を『サーバー』と呼び、画像表示する側を『端末』と呼ぶことにする。比較に用いた計算機の仕様を次の表2にまとめる。全部で800ステップ、4ステップ毎に画像を一つ出力する、対話的割り込みは計算開始直後だけ一回だけ行ないその際の原子の移動量はおおよそ一定にする、というほぼ一定の条件のもとでプログラムを実行してシミュレーション全体の実行時間を計測する。結果を図3にまとめる。一部のデータは技術上の原因で得られていない (na で示す)¹³。

表 2: 使用したコンピュータの性能の詳細

specification (place)	PC A (our laboratory)	ccindigo01 (computation center)	vis05 (computation center)
type	pc	workstation	workstation
cpu	Alpha21164(533MHz)	R4400(250MHz)	C200(200MHz)
main memory (MB)	256	—	256
video card	Oxygen202	—	Visualize FX-6
video ram (MB)	16	—	16
operating sys.	Windows NT4	IRIX Release 6.2	HP-UX (B.10.20)
specification (place)	PC B (our laboratory)	PC C (our laboratory)	ccsx4 (computation center)
type	pc	pc	super computer
cpu	Pentium(133MHz)	Pentium(120MHz)	—
main memory (MB)	80	48	—
video card	S3 Trio64V+	—	—
video ram (MB)	2	1	—
operating sys.	Linux+Accelerated X	FreeBSD+XFree86	SUPER-UX(8.1 SX-4)

表 3: 異なった端末/サーバーを用いた場合の対話型分子動力学シミュレーションの経過時間の比較

terminal	server			
	ccsx4	ccindigo01	PC C	PC A
ccindigo01	na / 43	na / 8	130/527	—/—
vis05	na / 45	na / 8	130/500	—/—
PC A	na / 67	na / 41	86/650	—/6~7
PC B	na /100	na / 64	56/130	—/—
PC C	—/—	—/—	33/ 64	—/—

Values in the table represent (CPU time)/(elapse time) in seconds.

サーバーに接続し、計算と表示を異なる計算機で実行する状況では、通信する時間も必要となる。とくに、複雑なネットワーク経由¹⁴であると確実に遅くなる。予測されたことではあるが、SX4は計算力はあ

¹³PC Cでは、GL拡張されていないXFree86というウィンドウシステムを用いているのでOpenGLのサーバーとしては使えない。代わりに同様の機能をもつ MesaGLを用いる。

¹⁴ODINS 経由であっても、ヴァーチャル LAN の利用や研究室内のタコ足配線などを理由に遅くなったようである。

るが今回のように MD 計算の 1 ステップにかかる負荷が極めて小さいときには SGI 社のワークステーションのような画像処理に特化された計算機をサーバーとするとときと比較して、遅くなってしまふ。これは、計算が比較的速い Alpha チップ搭載の PC(PC A:WindowsNT マシン) に対しても言えることである (これは加えて高速なグラフィックボードを積んでいることにも起因する)。参考のためにそのマシンの中だけでコンパイル¹⁵および実行する場合には SX4 に比べてはもちろんのこと、全体的に見ても極めて高いパフォーマンスを示す。

3.5 考察 2:ステップ数と表示回数

今回のコンピュータ利用環境およびグラフィックス表示プログラムでは、1 画像の生成に多くの時間が費やされる。そのために、画像表示と MD 計算ステップの兼ね合いが対話型シミュレーション全体の効率に影響を及ぼす。画像更新までに行なう MD 計算ステップ数を R とし、これを変化させて定性的に考察した。毎ステップ、すなわち $R=1$ とする場合をはじめ、 R が小さい値であると、画像生成が処理の負担になり無駄が多い。逆に、 R の値を大きく設定してしまうと、粒子の運動が不連続になること、対話をする機を逃してしまうこと、などの対話型シミュレーションに不可欠なリアルタイム性が低下する。従って、現状では計算機の機能を考慮しつつ、見た目には連続的な画像が出力されることを目安にこの数値を設定することになる。もちろん、適切な R の値は、シミュレーションの目的や対象とする現象によって大きく異なる¹⁶。

4 結論

SX4 にアプリケーション (ライブラリ) として搭載されている OpenGL を用いる場合のグラフィック表示性能について検討した。とくに、前回のように一回限りの表示では、一旦画像を生成すればそこその表示性能が出ていたが、今回のようにひっきり無しに画像の生成が行なわれる場合、ネットワークでの接続がかなりネックになる。表示回数をうまく少なく押えることで、リアルタイム性を損なわずに対話型シミュレーションが行なえることがわかった。

おわりに

画像生成も SX4 に任せて高パフォーマンスを出そうという思想は、以前の SXview¹⁷ から存在していたと思うが、近年のパソコンをはじめとする廉価なグラフィック機器の表示性能の増加には目覚ましいものがあり、それを利用しない手は無くなった。ただ、分子動力学計算のように大量の数値データを吐き出してそれを解析する場合、データ転送の時間やデータの保管場所 (つまり、研究室内のハードディスクの容量) にも気を取られる。端末ウィンドウに表示されている画像をそのまま、ビデオテープまたはコンピュータ上の動画ファイルなどに落せば、スーパーコンピュータを手軽な数値実験の装置としても捉えるというシミュレーション屋の夢に近づくことができる。本モニター活動で行なったような対話型のシミュレーションの需要は増えることはあっても減ることは無いと確信している。最後に、活発な議論を与えて下さった、大阪大学大学院工学研究科 機械物理工学専攻 稲葉研究室のメンバーと、大阪大学レーザー核融合研究センターの福田優子様、御協力頂きました大阪大学大型計算機センターの皆様にお礼申し上げます。

¹⁵ Visual C++5.0 を使用した。

¹⁶ 一つ一つの原子が瞬間移動してもいいので全体の変化を速回しで見たいという場合もあり得る。

¹⁷ SXview/IMG は SX4 で画像生成を行なった結果を超高速度画像表示装置 (UltraNet フレームバッファ) に出力することにより、リアルタイム・シミュレーションを可能にする。SXview/GWS は SX4 と GWS とのネットワーク環境のもとで利用可能な分散処理指向の対話型システムである。しかし、現在ではサポートされなくなつたと聞く。

参考文献

- (1) 齋藤, 大阪大学大型計算機センターニュース, 28-1(1998-5),61-71.
- (2) 例えば, 中嶋・川合, グラフィクスとマンマシンシステム, (1995), 2, 岩波書店.
- (3) 上村・内田 他, 並列計算機と汎用可視化ツールを用いた仮想空間内実験—インタラクティブシミュレーション—, 日本機械学会 第76期全国大会講演会講演論文集, No.98-3 II(1998), 41.
- (4) 福田優子, 私信. (1994年にODINS披露の企画としてSX3で行なわれていました.)
- (5) "Sculpt"という分子シミュレーションソフトウェアがある. (<http://www.intsim.com/>)
- (6) Leech, J. et al., SMD: Visual Steering of Molecular Dynamics for Protein Design, *IEEE Computational Science and Engineering*, 3-4(1996), 38-45.
- (7) 北川・北村 他, 初心者のための分子動力学法入門, (1997), 養賢堂.
- (8) 駒谷・齋藤 他, CGを用いたMD結果のプレゼンテーション手法の研究, 日本機械学会 関西学生会卒業研究発表講演会予稿集,(1998-3),250.
- (9) Rapaport, D.C., Interactive Molecular Dynamics, *Physica, A*, 240(1997), 246-254.
- (10) 川西, OpenGL入門 [第1回]3次元グラフィックス機能のインターフェース OpenGL, 日経 *Computer Graphics*, 1995年1月号(1995), 203-209.
- (11) 相川, OpenGLプログラミング・ガイドブック, (1996), 1, 技術評論社.
- (12) Kilgard, M.J., OpenGL Programming for the X Window System, (1997), 462, アジソン・ウエスレイ・パブリッシャーズ・ジャパン.
- (13) 可視化情報学会 編, 流れのコンピュータグラフィックス, (1996), 164, 朝倉書店.
- (14) TAN・齋藤 他, 対話型分子動力学法に関する検討, 日本機械学会 関西学生会卒業研究発表講演会予稿集,(1999-3),8-22.