

JavaScriptを用いたインターネット上の科学技術 シミュレーション教材 I — モンテカルロ法 —

川 口 高 明*

Takaaki KAWAGUCHI
Monte Carlo Simulation Using Java Script

[キーワード：JavaScript、インターネット、科学技術シミュレーション教材、モンテカルロ法]

1. はじめに

近年のインターネットの急速な広がりには驚嘆すべきものがある。特に、世界中に存在する無数のWorld Wide Web(WWW)サイトから公開されているホームページは、時々刻々と多様性を増している。そのホームページ作成で必要となるのは、Hyper Text Markup Language (HTML)¹⁾と呼ばれる言語であるが、近年、これに加えてさらにその機能を強化できるJavaおよびJavaScript¹⁻⁴⁾と呼ばれるある種のプログラミング言語が開発された。これらを用いると、ホームページを動的かつよりInteractiveなものにすることが可能である。そのため、これら言語を用いてホームページ上に教材を作成することにより、ネットワークを介しての仮想教室の試みが、様々な教育機関で情報教育の一環として行われている。そこで、我々はそのような方向に向けた情報教育の在り方に着目し、さらにその中に科学技術教育を融合させることを目標に研究を精力的に進めている。特に近年、我々はJavaScript言語の学習容易さと高機能さに着目し、それを用いてモンテカルロ法に基づくシミュレーション教材の開発を試みてきた。モンテカルロ法とは、従来から非常に高度なシミュレーション技法として知られており、科学技術シミュレーション分野において必要不可欠であるが、その計算処理速度等の問題から、ホームページ上での科学技術教材としては不向きであると考えられなくもなかった。

しかし、本研究において、そのシミュレーションプログラムが非常に簡潔な構造で作成され、科学技術シミュレーション教材としての有用性を確認できたので、その結果およびその将来への展望についてここに報告する。

2. JavaScript言語とは

上述のJavaScriptとJavaの両プログラミング言語は、名称は類似しているが、異なる開発経緯を経てつくられた本質的に異なる言語であることが知られている¹⁻⁴⁾。ここで、JavaScriptはNetscape Communications社とSun Microsystems社が共同で開発したスクリプト言語で、HTML文章の中に直接書き込んで用い、かなり簡素化されたオブジェクト指向仕様となっている。そのプログラムは、JavaScript対応のWWWブラウザに内蔵されているインタプリタによって翻訳実行される。JavaはC++言語等をもとに開発された本格的オブジェクト指向言語であり、JavaScriptよりも先に開発は行われた。そのプログラムの実行には、あらかじめJavaコンパイラによってコンパイルして、中間コードを生成しておく必要がある。どちらの言語もHTMLと共に用いることによって、ホームページまたは広くはネットワーク上でHTMLを超えた機能を追加することができる。

*島根大学教育学部技術科教育研究室

3. モンテカルロ法とは

モンテカルロ法とは、確率的方法を用いて問題を解く方法である⁵⁻¹⁴⁾。その応用分野は極めて広範囲におよんでおり、例えば、数値多重積分のような科学技術計算から、さらには社会現象のシミュレーションに至る分野にまで用いられている。モンテカルロ法の特徴として、その確率的扱いには乱数^{6,7,12,15)}を用いることが挙げられる。この乱数は、通常は計算機によって生成し、一般に一様な分布を持つ乱数が基本となるが、シミュレーションの目的に応じて、ある確率分布に従うように工夫されることもある。

4. 本シミュレーション教材のプログラムの内容、特徴および動作

本研究で作成した教材の対象は、磁気薄膜の示す磁化の振る舞いである。この磁気薄膜は、計算機を利用している者にとって非常に身近な材料である。ハードディスク、フロッピーディスクおよび磁気テープのような計算機の周辺機器における磁気記憶媒体は、ほとんどこの磁気薄膜を用いて構成されている。この意味で、磁気薄膜の物理的振る舞いをシミュレーションして学習することは、情報機器の基礎や動作原理および磁気材料の物質的性質を理解する上で非常に重要である。

ここでは、磁気薄膜のモデルとして、2次元の強磁性 ISINGモデル¹⁶⁻²⁴⁾を用いる。これは、薄膜を構成する2次元格子の各サイトで、一軸異方性を持ったスピン磁気モーメントが存在しているとする非常にシンプルなモデルで、各サイトでスピンの向きを上または下の2通りのみに定めることになる。そして、最隣接スピンドロウしが強磁性的に相互作用をしているとする。このことは、原子レベルでマイクロな磁化の極性が存在し、強磁性的に結びついている状態を考えると理解し易い。このモデルでは、ある有限温度で強磁性と常磁性の間の相転移が生じることが厳密に分かっているため、シミュレーションにおいても磁気構造の形成等が理解し易い。また、シミュレーション方法としては、モンテカルロ法が適している。このISINGモデルのモンテカルロ法的計算アルゴリズムは、すでに多くの文献^{6,8,10,11,13)}で解説されているので、本報告では省略する。ただし、本研究では、メトロポリス法に基づくモンテカルロ法を用いていることと、以下に示すシミュレーションの計算アルゴリズムに関しては、特に文献(8, 11, 13)を参考に行っていることを述べておく。ここでは、本研究においてJavaScriptを用いて作

成されたプログラム(図1)の内容と特徴および実行例について説明する。まず、プログラム1行目から4行目で、HTMLおよびJavaScriptの始まりの宣言をしている。5行目から17行目が、ブラウザ上の初期画面に縦横両方向に50サイト、合計2500サイトのスピンに対応した図形を描く関数である。ここで描画する図形はGIF形式のものを、上および下向きの各スピンに対応させて2種類用いている。18行目から26行目までで、モンテカルロ法で用いる数学的関数を定義した。27行目から101行目が、モンテカルロ法による計算を行うための関数である。この中で特に、47行目から70行目において、考慮するシステムのサイズが有限であることによる影響を避けるために、周期的境界条件を定めており、それに応じた場合分けが行われている。この境界条件は、システムの一つの端ともう一方の端が周期的に接続されていると仮定するもので、その有効性はシステムの有限形状を考慮する必要の無い問題では、すでに確立されている。さらに、ここで注意すべきは、JavaScriptは1次元配列しか定義できないので、この2次元的なスピンの並びの指定を1次元的に行わなければならないという点である。この制限は、もちろん境界条件を考慮する際にも技術的に問題となっている。各モンテカルロステップで2500個のスピンの向きを更新して再描画するが、この処理は85行目から94行目に記述されている。次に、ブラウザ上の文字と入力フォームの構成は、115行目から124行目で定められる。ここでブラウザ上に磁界強度、温度とモンテカルロステップ総数の数値入力を文字で促して、それらパラメータに対する入力フォームが用意されている。入力された数値は、Startボタンを押すことでモンテカルロ法の計算の関数に受け渡され、それらの数値をもとにモンテカルロシミュレーションが行われる。シミュレーションの間中、スピンの再描画を繰り返し行い、最後のモンテカルロステップを実行終了後に全体のスピンの向きを調べて、その状態の磁化を計算し、ブラウザ上の磁化のフォームにその値を出力する。これらの図形の描画機能およびフォームへの入出力機能により学習者は、ブラウザとの間にinteractiveな関係を築きながら体験的かつ視覚的に学習を進めることが可能となる。最後に、125行目から131行目で初期図形の読み込みと、JavaScriptとHTMLの終端を宣言している。ここで、計算上の注意点を挙げる。まず、1個のスピンについて、上向きの磁化を+1、下向きのものを-1と定義して磁化の計算を行っている。そして温度や磁界は、スピン間相互作用に関する物理量で規格化してあるために無次元量になっているので、フォームに入力する値の大小には、規格化に用いたスピン間

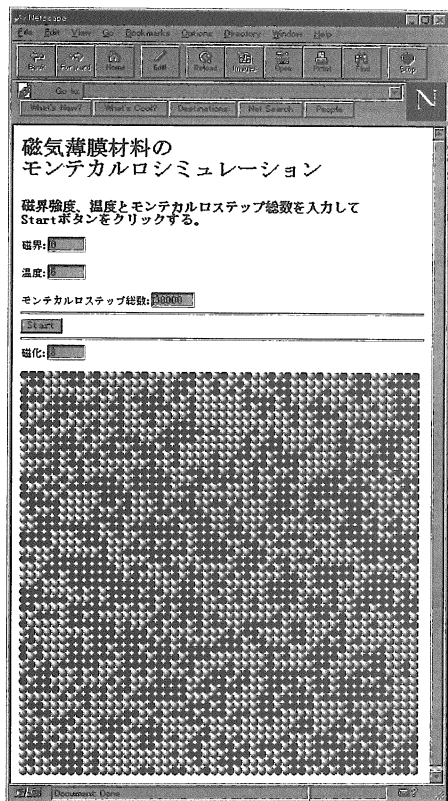
図1. プログラム

```

1 <HTML>
2 <HEAD>
3 <SCRIPT Language='JavaScript'>
4 <!--
5 function display 0
6 {
7   var sitex = 50;
8   var sitey = 50;
9   for (var j=0; j<sitey; j++)
10  {
11    for (var i=0; i<sitex; i++)
12    {
13      document.write('<IMG SRC="m.gif" BORDER="0" width="10" height="10">');
14    }
15    document.write('<BR>');
16  }
17 }
18 function rfact (scomp1, scomp2, jkt, b) {
19   if (scomp2==1) {
20     var rr=Math.exp (-2*(jkt*(2*scomp1-6)+b));
21   }
22   if (scomp2==2) {
23     var rr=Math.exp (2*(jkt*(2*scomp1-6)+b));
24   }
25   return rr
26 }
27 function montecarlo (konoForm) {
28   var lp=eval (konoForm.step.value);
29   var jkt=1/eval (konoForm.temp.value);
30   var b=eval (konoForm.field.value);
31   var m=50;
32   var m2=m*m;
33   var spin= new Array 0;
34   for (var i=1; i<=m2; i++) {
35     var sinit=1;
36     if (Math.random > 0.5) {
37       // sinit=1;
38       sinit=-1;
39     }
40     spin[i]=sinit;
41   }
42   for (var kloop=1; kloop<=lp; kloop++) {
43     for (i=1; i<=m2; i++) {
44       var iright=-1;
45       var ileft=1;
46       var iup=-m;
47       var idown=m;
48       if (i<m && i>1) {
49         iright=-1; ileft=1; iup=m*(m-1); idown=m;
50       }
51       if ((i-Math.floor (i/m)*m==1) && i!=1 && i!=m2-m+1) {
52         iright=m-1; ileft=1; iup=-m; idown=m;
53       }
54       if ((i-Math.floor (i/m)*m==0) && i!=m && i!=m2) {
55         iright=-1; ileft=-m-1; iup=-m; idown=m;
56       }
57       if (i>m2-m+1 && i<m2) {
58         iright=-1; ileft=1; iup=-m; idown=-m*(m-1);
59       }
60       if (i==1) {
61         iright=m-1; ileft=1; iup=m*(m-1); idown=m;
62       }
63       if (i==m) {
64         iright=-1; ileft=-m-1; iup=m*(m-1); idown=m;
65       }
66       if (i==m2-m+1) {
67         iright=m-1; ileft=1; iup=-m; idown=-m*(m-1);
68       }
69       if (i==m2) {
70         iright=-1; ileft=-m-1; iup=-m; idown=-m*(m-1);
71       }
72     }
73     var stot=spin[iright+i]+spin[ileft+i]+spin[iup+i]+spin[idown+i];
74     var scomp1=(stot+6)/2;
75 //
76     if (spin[i]>0) {
77       var scomp2=1;
78     }
79     else {
80       var scomp2=2;
81     }
82 //
83     var sflip=rfact (scomp1, scomp2, jkt, b);
84     var srand=Math.random 0;
85 //
86     if (srand < sflip) {
87       spin[i]=-spin[i];
88     }
89   }
90 //
91   for (x=1; x<=m2; x++)
92   {
93     var y=x-1;
94     if (spin[x]==-1) {
95       document.images[y].src=spining[0].src;
96     }
97     if (spin[x]==1) {
98       document.images[y].src=spining[1].src;
99     }
100   }
101 }
102 var ssum=0;
103 for (i=1; i<=m2; i++) {
104   ssum=ssum+spin[i];
105 }
106 konoForm.sum.value=ssum;
107 }
108 // -->
109 </SCRIPT>
110 </HEAD>
111 <BODY>
112 <SCRIPT Language='JavaScript'>
113 <!--
114 var spining=new Array (2);
115 spining[0]=new Image (10, 10);
116 spining[1]=new Image (10, 10);
117 spining[0].src="m.gif";
118 spining[1].src="p.gif";
119 // -->
120 </SCRIPT>
121 <H1>磁気薄膜材料の<BR>モンテカルロシミュレーション</H1>
122 <BR>
123 <H3>磁界強度、温度とモンテカルロステップ総数を入力して<BR>
124 Startボタンをクリックする。</H3>
125 <FORM>
126 磁界:<INPUT TYPE='text' NAME='field' SIZE=5><P>
127 温度:<INPUT TYPE='text' NAME='temp' SIZE=5><P>
128 モンテカルロステップ総数:<INPUT TYPE='text' NAME='step' SIZE=6>
129 <HR>
130 <INPUT TYPE='button' VALUE='Start' onClick='montecarlo (this.form) '>
131 <HR>
132 磁化:<INPUT TYPE='text' NAME='sum' SIZE=5><P>
133 </FORM>
134 <SCRIPT Language='JavaScript'>
135 <!--
136   display 0;
137 // -->
138 </SCRIPT>
139 </BODY>
140 </HTML>
141

```

図2. ブラウザと実行画面の全体



相互作用の強度に対する相対的な大きさとしての意味がある。また、ここでは、乱数の発生にはJavaScriptのMathオブジェクトのメソッドを用いたが、精密で大規模なモンテカルロシミュレーションを行うには、Tausworthe法⁶⁾を用いて発生した長周期の乱数を用いるのが一般には良いとされている。このためにはビット演算が必要となるので、プログラムは多少複雑になる。だが、本研究対象のモデルのモンテカルロシミュレーションにおいては、以下に示すように、Mathオブジェクトのメソッドを用いても教材目的には十分実用になるようである。

次に、シミュレーションの実際を見てゆく。ここで使用した計算機にはPentium120MHzと48MBのメモリを搭載している。そして、用いるWWWのブラウザとしてはNetscape Communications社のNetscape Navigator ver. 3.0を採用した。現在、JavaScriptに対するサポート等からみて、このブラウザ上でJavaScriptプログラムが最も安定に動作すると考えられる。ブラウザも含めた教材全体の様子を図2に示す。ここで、ブラウザ上の下部に描かれた図形並びの中で、灰色および黒色の点は、それぞれが格子のサイトに存在する上向きまたは下向き

図3. 初期状態

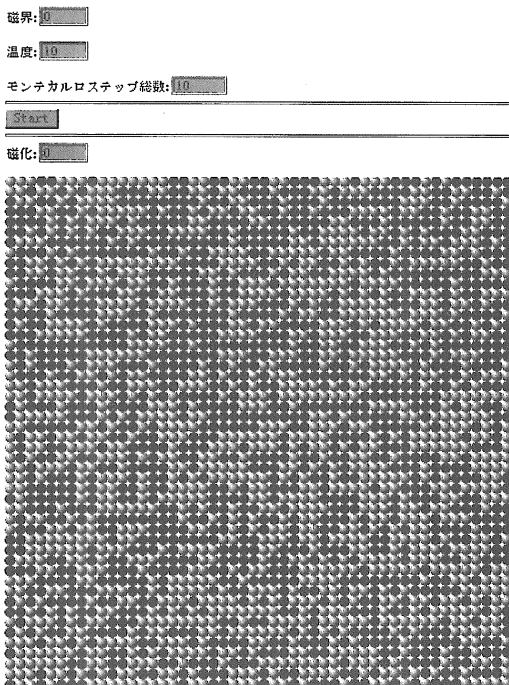
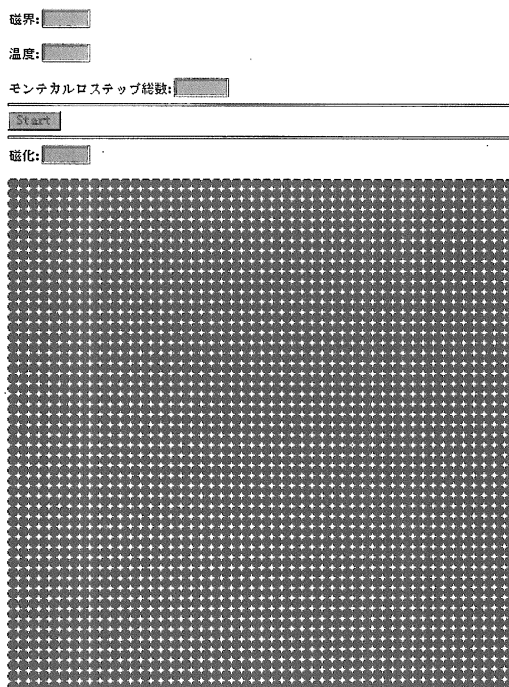


図4. 高温での実行結果

のスピンを表わす。示されている図形では、磁界強度、温度とモンテカルロステップ数を入力してシミュレーションを行って終了した状態になっている。終了時の全サイトのスピンの向きを表わす図形と、その状態での磁化の値が出力されている。次に、学習例をいくつかのプログラムの実行結果を見ながら説明する。この教材プログラムを起動した時の初期画面は、図3のようになる。ここでは、各サイトのスピンはすべて同一の向きに揃っている。ここから磁界強度と温度に応じて、図形がリアルタイムで変化して行く。まず、図4に非常に高温で温度=10の場合のシミュレーション実行結果を示す。各スピンはほとんどバラバラの向きを持ち、図4から分かるように、灰色と黒色の点は全く同じ数だけ存在する。このため磁化を計算するとゼロになる。ここでモンテカルロステップ総数は10にしてあるが、この場合、これ以上増やしてもほとんど同じ結果になる。次に少し温度を下げて、温度=3とすると(図5)、同じ向きのスピンのクラスタを形成して、図5から明らかに、上向きと下向きの小さな磁化クラスタが発生し始めていることが見て取れる。上向きと下向きの磁化領域が混在しているが、その面積をみると、下向きの方が少々大きい。このため全体の磁化は、-430となった。さらに低い温度0.2にすると(図6)、図5で見られた小さな磁化クラスタが成長して、

図6. 低温での実行結果 2

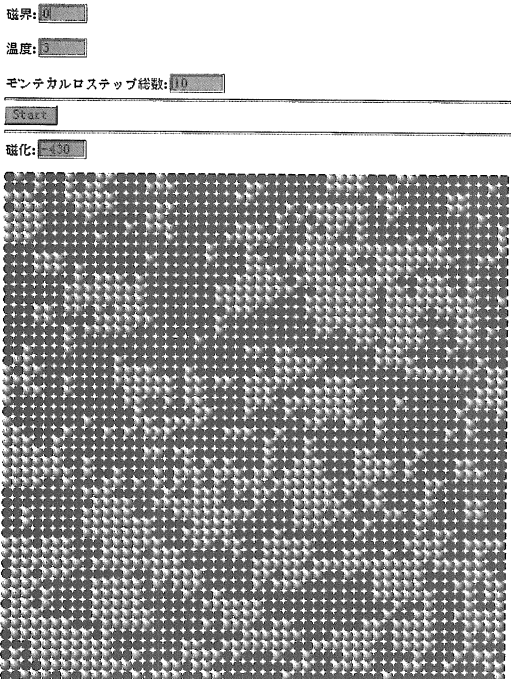
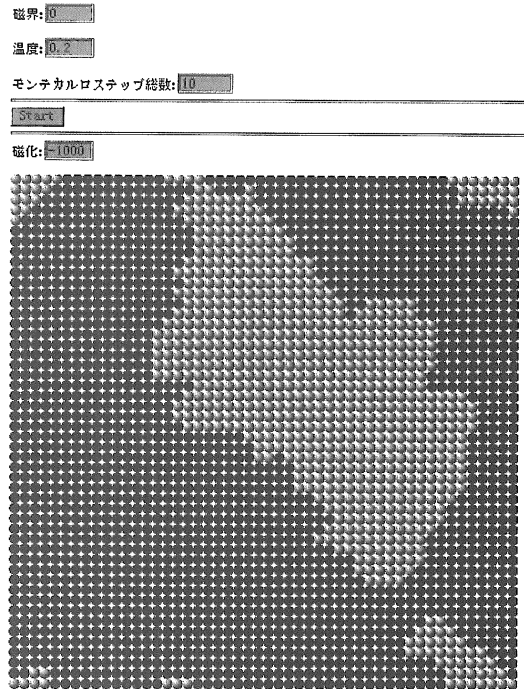


図5. 低温での実行結果 1

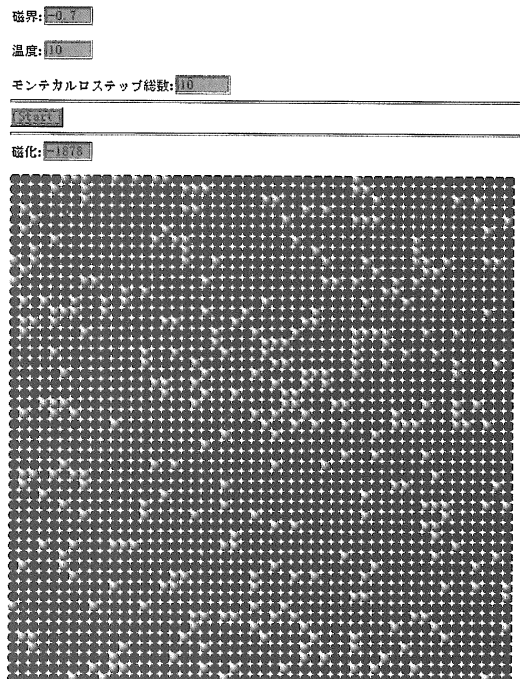


図7. 磁界印加時の高温での実行結果

システムの大きさに相当するほどの大きな磁化クラスタが発生する。この図6より、大きな上向き磁化クラスタが2個と全システムを覆うほどの下向き磁化クラスタが1個存在していることが分かる。この時、全体の磁化はこのクラスタ構造を反映して、大きな負の値、 -1000 を示す。なお、ここでこの上向き磁化クラスタは、モンテカルロステップをさらに大きくして観測して行くと、だんだん消滅してゆく傾向を持つことが分かる。つまり、このような低温の平衡状態では、どちらか一方の向きの磁化のみが存在することが認識できる。このように、温度変化をシミュレーション教材に入力してやることで、学習者は、磁気薄膜の磁化が高温では不安定であるが、低温にすると安定で様な構造を持つことが認識できる。次に、磁界の効果を図7において示す。ここで温度とモンテカルロステップ数は図4と同じに選んだ。磁界強度のみゼロから有限の値に変更している。先ほどの場合には磁化がゼロであったのが、磁界印加により大多数のスピンの向きが揃えられ、大きな磁化が生じたことが図形パターンから視覚的に容易に理解できる。これは磁界が存在するために、それに応じて全体のうちのいくつかの部分で、ミクロな磁化の極性の向きが反転したとみなせば理解し易い。これらの実行例から、本教材では、普段利用しているフロッピーディスクや磁気テープに対して、非常に高温または強い磁界の環境下にはさらさないという注意のもとになっている磁気薄膜の基本的性質を、シミュレーションを通して学習できることが明らかになった。さらに、これらの図には示すことはできないが、シミュレーション途中の図形パターンの変化から、モンテカルロステップが増えてゆくにつれて、リアルタイムで磁化のパターンが平衡状態へと近づいて行く様子も理解可能になっている。JavaScriptの乱数を発生させるMathオブジェクトのメソッドは、呼び出す度に異なる乱数列を発生するため、それに対応して、このブラウザ上のStartボタンを押してシミュレーションを行う度に、異なる磁化パターン変化を経て平衡状態にたどり着く。このため、磁化のパターン変化途中での再現性は無いが、色々なパターンを次々と確認できるという点では便利である。なお、本研究で用いた計算機上で、1回のモンテカルロステップでの計算と全体の図形の再描画は数秒とかからず実行される。これはJavaScriptの言語仕様レベルと用いた計算機の性能から考えると、かなり高速である。

5. これからの展望

今後、本教材のプログラムに改良を加えることによって、磁化の時間変化をリアルタイムでグラフ表示させて磁化過程を見せたり、さらに、モンテカルロステップでの統計平均を取って磁化率の温度変化等も計算してグラフに表示することが可能である。これにより、さらに高い教育効果が期待できる。

また、本研究では磁気薄膜の磁化過程のシミュレーション教材化をモンテカルロ法という技法にもとづいて行ったが、他に重要な代表的技法として、分子動力学法²⁹⁾がある。これは基本的に運動方程式の時間発展を調べる方法であり、すでに様々な科学技術分野で用いられている実績がある。最近では、材料工学および機械工学分野等における摩擦²⁶⁻²⁸⁾、亀裂破壊²⁹⁾や熱流体³⁰⁾の物性研究などへの応用がめざましくなされている。筆者個人的には、さらに異なる方法として、行列演算による電子物性の計算³¹⁾、有限要素法や境界要素法^{32,33)}、最適化問題^{34,35)}、そして半導体デバイス中のキャリアの運動のシミュレーション^{36,37)}等も教材テーマとして興味深いと考えている。なお、本研究ではMathオブジェクトのメソッドで済ませたが、乱数を発生させるシミュレーション自身が興味深い数学教育教材となりうる。ここで、これらの多くでは計算量が多く、困難が予測されるので、その場合、JavaScriptではなくJavaを用いてプログラムを記述し、比較的単純な問題を対象とするのが適当であろうと思われる。Javaはある意味でコンパイル言語であり、実行処理速度はインタプリタ言語であるJavaScriptに比べてほとんどの場合において高速であることが予想される。その際、高性能なJavaコンパイラの選択が重要になる。しかし、ここでは特に、モンテカルロシミュレーションがJavaScriptによって驚くほど高速で実行されたことを認識することが、JavaScriptそのものの可能性を考える上で重要である。

本研究で作成されたJavaScriptを用いたシミュレーション教材のプログラムは、中学校の情報基礎領域および高等学校の工業等の授業の中において十分習得可能な内容となっている。このことに加えて、JavaScriptは計算機の用いるOSの種類にほとんど依存せずに動作すること、そしてWWWのブラウザさえあれば教材学習が即座に可能なことを考えると、本研究で作成されたシミュレーション教材は、様々な教育現場において即座に容易に活用可能である。本研究で示したようなWWW上でのシミュレーション教材は、これからのインターネットの進歩や広がりを見ると、情報教育および科学技術教育の発展のための大きな可能性を有していると考えられる。また、もちろんネットワークに接続されていないスタンドアロ

ーンの計算機環境においても、ブラウザを用いた教材としての利用が可能であり、非常に本研究テーマは応用範囲が広い。

文献

- (1) Java技術研究会編：HTMLとJavaScript、工学図書株式会社、1996.
- (2) 上田学：JavaScript、オーム社、1997.
- (3) 松尾忠則、古籬一浩：JavaScript、インプレス、1997.
- (4) 大津真：JavaScript、BNN出版、1997.
- (5) J.M.Hammersley and D.C.Handscomb: Monte Carlo Methods, Methuen, 1964.
- (6) 津田孝夫：モンテカルロ法とシミュレーション、培風館、1997.
- (7) 宮武修、脇本和昌：乱数とモンテカルロ法、森北出版、1978.
- (8) K.Binder (editor):Monte Carlo Method, Springer-Verlag, 1987.
- (9) 津田孝夫：数値処理プログラミング、岩波書店、1988、p. 307~328.
- (10) R.W.Hockney and J.W.Eastwood: Computer Simulation Using Particles, Adam Hilger, 1988.
- (11) 上田顕：コンピュータシミュレーション、朝倉書店、1990.
- (12) H.Niederreiter: Random Number Generation and Quasi-Monte Carlo Methods, SIAM, 1992.
- (13) 宮下精二：熱・統計力学、培風館、1993.
- (14) 伏見正則：確率的方法とシミュレーション、岩波書店、1994、p. 1~54.
- (15) D.E.Knuth:The Art of Computer Programming, Vol. 12, Addison Wesley, 1981.
- (16) 太田恵造：磁気工学の基礎I、共立出版、1973.
- (17) 近角聡信他編：磁性体ハンドブック、朝倉書店、1975、p. 70~74.
- (18) 渡辺浩、高橋実：磁化機構・磁性化合物・磁性合金、丸善、1982、p. 246~251.
- (19) 宮原将平、野呂純子訳：磁性の理論1、講談社、1982、p. 58.
- (20) 日野太郎、森川鋭一、串田正人：電気電子材料、森北出版、1991、p. 187~222.
- (21) 内山晋編著：アドバンストマグネティクス、培風館、1994、p. 1~98.
- (22) 岩本光正：電気電子物性、オーム社、1995、p. 121~138.
- (23) 安達健五：化合物磁性、裳華房、1996、p. 275~283.
- (24) 一ノ瀬昇編著：電気電子機能材料、オーム社、1996、p. 105~138.
- (25) D.Frenkel and B.Smit:Understanding Molecular Simulation, Academic Press, 1996.
- (26) 川口高明、松川宏：物性研究、67(1997)No. 5、p. 613~617.
- (27) T.Kawaguchi and H.Matsukawa:Physical Review, B56 (1997) No. 7、p. 4261~4265.
- (28) T.Kawaguchi and H.Matsukawa:Physical Review, B56 (1997) No. 21、p. 13932~13943.
- (29) 北川浩：日本機械学会誌、97-907 (1994) p. 459~463.
- (30) 小竹進：分子熱流体、丸善、1990、p. 107~127.
- (31) T.Kawaguchi and M.Saitoh:Journal of Physics, Condens. Matter 3 (1991)、p. 9371~9380.
- (32) 矢川元基、吉村忍：有限要素法、培風館、1990.
- (33) 川井忠彦、神谷紀生、竹内則雄：計算力学入門、森北出版、1993、p. 83~174.
- (34) S.Kirkpatrick, C.D.Gelatt and M.P.Vecchi: Optimization by simulated annealing, Science, 220(1983) p. 671~680.
- (35) 茨木俊秀：離散最適化法とアルゴリズム、岩波書店、1993.
- (36) C.Jacoboni and P.Lugli:The Monte Carlo Method for Semiconductor Device Simulation, Springer-Verlag, 1989、p. 104~261.
- (37) C.Mogilestue:Monte Carlo Simulation of Semiconductor Devices, CHAPMAN&HALL, 1993、p. 115~298.