

## 視覚研究のための実験環境：Mathematica

大竹史郎

松下電器産業（株）研究本部

〒 619-02 京都府相楽郡精華町光台 3-4

### 1. はじめに

Mathematica は「a general software system for mathematical and other application (数学及びその他の応用のための汎用ソフトウェアシステム)」である<sup>1,2)</sup>。その演算のために準備された組み込み関数は豊富であり、演算結果を表示するグラフィクス機能は強力である。筆者は数値解析ツールとして愛用している。

Mathematica のグラフィクス機能のなかにアニメーション機能がある。これは少しずつ形状の異なる複数の画像を逐次提示することにより、「動画」をつくりだす機能である。本稿では、この機能を応用した視覚実験の例、および留意点を述べる。

### 2. Mathematica のプラットフォーム

Mathematica は、1988年に Wolfram Research, Inc. から発売された。最初は Macintosh 版のみであったが、1996年現在では、流通しているたいのコンピュータの機種で動作する。表 1 に、PC/AT 互換機+MS-Windows、および

Macintosh+System 7 についての動作環境を示す。他のコンピュータおよび OS での動作環境は少しずつ違う。その詳細は、興味があれば代理店等で確認していただきたい。

メモリサイズは大きいほうが使い勝手がよい。このことは、他のアプリケーションでもいえるが、Mathematica でも同じである。ノートブック (notebook) と呼ばれるグラフィカル・ユーザー・インターフェイスのフロントエンド (front end) だけでなく、計算を実行するカーネル (kernel) もまたメモリを占有するからである。特に、グラフィクス機能を使用している場合は、メモリ不足に留意しながら使う必要がある。

筆者は、PC/AT 互換機、Macintosh および WS (Sun SS10) で Mathematica を使った経験がある。本稿の趣旨は機能の紹介であるから、いずれかの機種に的を絞って説明したほうが内容を理解していただき易いであろう。このため、以下、便宜上、Macintosh 版についての例をもとに説明させていただく。どのプラットフォーム

表 1 Mathematicaの動作環境

	PC/AT 互換機	Macintosh
CPU	i80386以上	68020以上
OS	MS-Windows 3.1以降	System 7.0以降
メモリサイズ	最低 4 MB 推奨 6 MB 以上	最低 6 MB 推奨 10 MB 以上
ディスク容量	29 MB 以上を推奨 (インストール 13 MB) (スワップスペース 16 MB)	最低 7 MB

でも、おおすじは同じ筈であるから、他機種をお使いの読者にとっても不都合なことはないであらう。

### 3. アニメーション機能

Mathematica ディレクトリの中のサブディレクトリ sample のサンプルプログラムを例にアニメーション機能を説明する。たとえば、図1に示すような Drum.ma というノートブックを読み込んだとする。タイトルやコマンドなど表示はすべてブラケット (bracket) で囲まれたセル (cell) と呼ばれる単位で区切られている。このブラケットが層構造になっており、グループを構成している。Drum.ma で示されたグラフィックパターンはベッセル関数の数値解である。パターンをダブルクリック (もしくはクリックして、アップル・キー+y) すると、アニメーションが始まり、時間的変化を眺めることができる。もういちどクリックするとアニメーションは停止する。グラフィックパターンのある矢印のついたブラケットをダブルクリッ

クすると、図1右側に示すように、少しづつ形状が異なる複数枚のセル群が展開する。すなわちアニメーション機能とは、同じグループ内にあるグラフィックパターンのセルを順次、切り替えて提示する機能である。

セルのパターンは、Mathematica の高度な数学的演算機能、グラフィック機能のため、簡単な命令を組み合わせるだけで作成できる。セルを作成する段階では、フロントエンドとカーネルの双方で大きなメモリ容量が必要な場合が多い。その際、メモリ不足エラーを避けるには、(1) 少ない個数ずつセルを作って逐次保存し、フロントエンドのメモリ使用サイズを大きく設定した状態で読み出して結合する、(2) WS などメモリ容量の大きいプラットフォームでセル群をつくって保存し、所望のプラットフォームにそのファイルを転送する、などの方法がある。また、これは裏技に近いが、他のソフト (たとえば MacDraw Pro) で作成したグラフィックスを、カット&ペーストで Mathematica のセルとすることも可能である。このようにして、

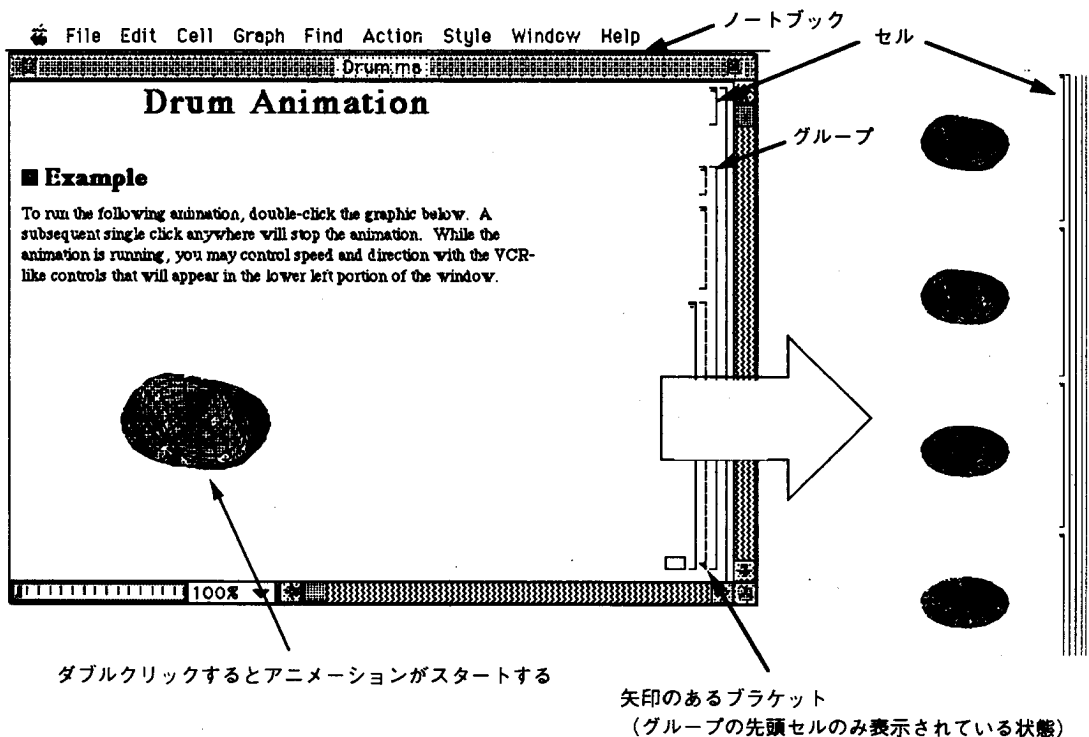


図1 サンプルノートブック Drum.ma

視覚刺激パターンのセルを複数枚あらかじめ作っておき、それらをアニメートすることにより、Mathematica をインストールしたコンピュータは簡単な視覚実験ツールとして使うことができる。以下、その応用例を紹介する。

#### 4. 視覚実験ツールとしての応用例

##### 4.1 最適周波数を求める実験

いま、格子状に配置した発光点群について、仮現運動がよく観測できるような配列、および点滅パターンおよび切り替わりの時間間隔を見出したい場合を例に説明する。付録2は、筆者のつくったサンプルプログラムである。これを実行すると、黒背景のなかの一部領域の格子点が白色に発光している2枚のセルができる(図2)。これらのセルのうち、いずれかをダブルクリックすればアニメーション機能がスタートし、二つのセルは交互に入れ替わって表示される。セル切り替わりの時間間隔は、ウィンドウの外枠にあるアイコンをクリックすることで変えることができる。ただし、毎秒60フレームが上限である。最も仮現運動が見えたと判断したところでアニメーションをストップし、アップル・メニュー Graph のサブメニュー

Animation Setting を開いて切り替え時間を見れば、ストップさせたときのセル切り替わり時間間隔が計測できる。また最初にこの設定を行ってからアニメーションをスタートさせて、見えを評定してもよい。

ここでは格子状の配置について説明したが、アルファ状の配置をはじめ様々な配置でも、簡単な命令群でセルをつくることができる。セルさえつくっておけば、あとの操作方法は同じである。これらの観測は、実際に表示素子を組み合わせることで実験装置を作ってもよいが、検討範囲の条件をすべて網羅する装置をすべて作成するには手間がかかる。しかし、このように Mathematica を使えば、数十行の命令でパターンをつくることができ、検討がすみやかにできる。

##### 4.2 知覚特性を評定する実験

この実験の好例は Cicerone et al. (1995) の dynamic color spreading 現象についての報告であろう<sup>3)</sup>。サンプルプログラムは文献の巻末付録に掲載されている。彼らは、色のついたランダムドットにおいて、一部の領域内のドットのみ異なる色とした刺激を Mathematica で作成した。それらのセルひとつひとつでは、二種類の

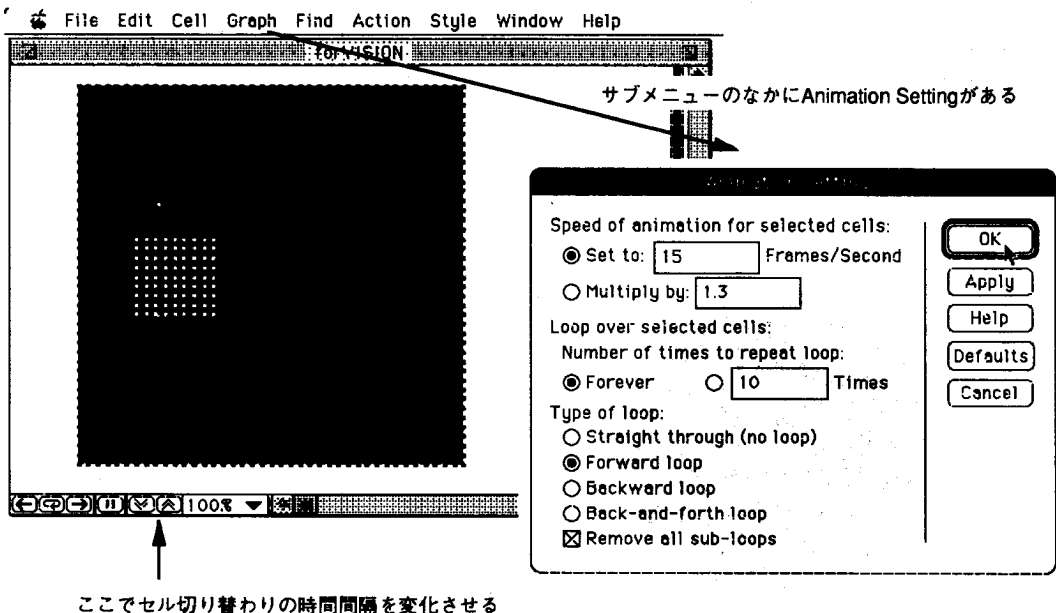


図2 セル切り替わりの時間を計測する手段

色のランダムドットがあるように見えるだけである。しかし、アニメーション機能を用いて、異なる色のドットとなる領域を動かすと、その領域内のドットだけでなく、その領域内だけ異なる色で照明されているかのように見える。彼らは、ドットの密度、領域の運動速度を変化させて、違いそのときの見えを評定している。(この論文の第二作者である Hoffman は Mathematica が好みである。筆者は彼の影響を受けて Mathematica をツールとして使うようになった。)

付録3のサンプルプログラムは、円運動している点光源で球体を斜め上から照明した場合のアニメーション例である(図3)。このようにして照明方向を変化させたとき、どういった陰影ができるかを観測することができる。このやり方では、対象物の立体の表面が数学的に記述できれば任意の形状について検討可能である。また、点光源は複数であってもよい。コマンドの詳細は、参考文献に挙げたマニュアルやハンドブックを参照していただきたい。

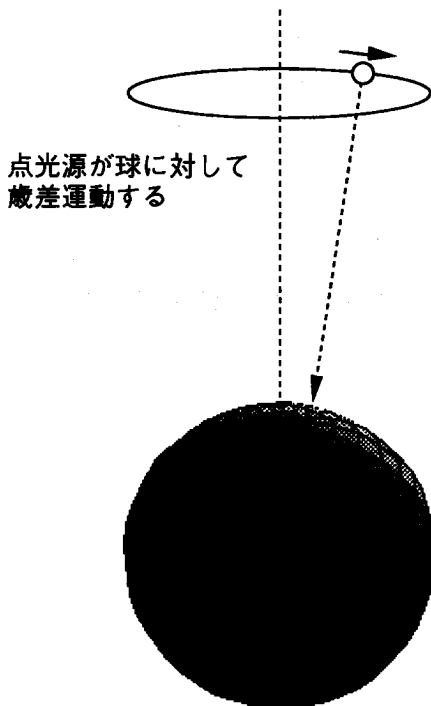


図3 球のモデリング

## 5. むすび

以上、述べてきたように、Mathematica のアニメーション機能は、数学的関数で記述できるパターンを繰り返し提示することは得意とするところである。とはいえ、その機能はどんな視覚実験にも適用できるわけではない。セルがつけられてはじめて、アニメーション機能を使用することができる。

たとえば、付録4に示すサンプルプログラムでは、中央部に固視点をもつ二分視野をつくることができる。これと順応視野とが交替するようなアニメーションをつくれれば、直接比較法による明るさマッチングが可能のように思える。しかしながら、左右の視野の明るさをマッチングさせるには、どちらが明るいかを判断した後、いったんアニメーションを止め、テスト視野の強度の異なるセルに置換してから再びアニメートさせる操作を繰り返す必要がある。すなわち、Mathematica のアニメーション機能は、調整法など、観測者の応答にしたがってダイナミックにセルを変化させる必要がある実験には不向きであるといえよう。

もしかすると、高速なプラットフォームにおいて MathLink (フロントエンドとカーネルの間のコミュニケーションスタンダード) をうまく利用して制御すれば可能かもしれないが、かえって手間であろう。むしろ、最初から Shell<sup>4)</sup> のようなソフトウェアを用いるほうが、実験環境としては良好であろう、と筆者は思う。

したがって、(1) 一定のコントラストをもつパターン群で構成された光刺激を用いた、(2) 繰り返し周波数の閾値実験か、もしくは一定の繰り返し周波数のもとでの知覚特性の評定実験であるとき、Mathematica は実用的な視覚実験ツールになるであろう。

最後に、Mathematica に限らず一般にいえることだが、視覚刺激を発生する表示デバイスの特性が実験の精度を決めることがしばしばある。すなわち、表示デバイスによって生じる輝度や色度ムラ、経時的变化を補正する手段を実験目的に応じて必要だけ持つことが重要であ

る。このようにソフトウェアシステムだけでなく、プラットフォーム及び周辺機器の特性をふまえたうえで、目的にかなった視覚刺激をつくることができれば、手持ちのコンピュータは立派な実験装置になる。

#### 文 献

1) S. Wolfram: Mathematica: a system for doing

Mathematics by computer, 2nd edition. Addison-Wesley Publishing Company, Inc., 1991.

2) M. L. Abell and J. P. Braselton: The Mathematica Handbook. AP Professional, 1992.

3) C. M. Cicerone, D. D. Hoffman, P. D. Gowdy and J. S. Kim: The perception of color from motion. *Perception and Psychophysics*, 57, 761-777, 1995.

4) 塩入 論: 視覚研究のための実験環境: Shell & MacGlib. *VISION*, 8, 31-36, 1996.

#### 付録 1

Mathematica の本業はあくまでも数学的処理である。その機能はたとえばデータ解析に威力を発揮する。そのためのフリーウェアも多くある。たとえば、Mathematica の開発元である Wolfram Research では、WWW により検索、取り込みが行えるようにしている。ポインタは下記のとおり。

<http://www.wri.com/mathsource/>

電子メールアドレス [mathsource@wri.com](mailto:mathsource@wri.com) (Help という 1 行コマンドを送ると、使用方法を記した電子メールを返信してくる) を利用して、このフリーウェアのデータベースを使ってもよい。ftp サーバーもあるが、これにアクセスできるかは、読者がアカウントをお持ちのドメインの管理体制に依存する。

また下記のポインタは、NTT 基礎研の西田さんに教えていただいたものだが、心理計測関数への最適近似を行なう Mathematica プログラム *Psychometrica.ma* などが載せられている。

<http://vision.arc.nasa.gov/mathematica/>

#### 付録 2

```
LL={};xo=-.25;yo=.5;ww=.10;dd=0.025;
nn=Floor[1./dd]+1;
xx=Table[dd*(i-nn*Floor[i/nn]),{i,0,nn*nn-1}];
yy=Table[dd*(Floor[i/nn]),{i,0,nn*nn-1}];
Do[
  LL={ {RGBColor[0,0,0],Rectangle[{0,0},{1,1]}},
        {RGBColor[0,0,0]},
        {RGBColor[1,1,1]}
        };
  xo=xo + 0.5;
  Do[
    If[({xx[[i]]>=xo-ww}&&(xx[[i]]<=xo+ww))
        &&((yy[[i]]>=yo-ww)&&(yy[[i]]<=yo+ww)),
      AppendTo[LL[[3]],Point[{xx[[i]],yy[[i]]}],
      AppendTo[LL[[2]],Point[{xx[[i]],yy[[i]]}],
    ],
  {i,1,nn*nn}];
```

```
Show[Graphics[{PointSize[0.01],LL}],
      AspectRatio->Automatic],
{j,1,2}];
```

### 付録3

```
Do[
  x=2*Cos[2*N[Pi]*i/20];
  z=2*Sin[2*N[Pi]*i/20];
  ParametricPlot3D[{Cos[t]Cos[u],Sin[t]Cos[u],Sin[u]},
                    {t,0,2N[Pi]},
                    {u,-N[Pi]/2,N[Pi]/2},
                    Axes->False,
                    Boxed->False,
                    PlotPoints->Automatic,
                    LightSources->{{{x,4,z},
                                      GrayLevel[1.0]}}},
  {i,1,20}];
```

### 付録4

```
ta=(-1./2.)*N[Pi];
tb=(+1./2.)*N[Pi];
tc=(+3./2.)*N[Pi];
Show[
  Graphics[{
    {RGBColor[0,0,0],Rectangle[{0,0},{1,1]}},
    {RGBColor[0.5,0.5,0.5],Point[{0.5,0.5]}},
    {RGBColor[0.5,0.5,0],
     Disk[{0.48,0.5},0.4,{tb,tc}]},
    {RGBColor[1,0,0],
     Disk[{0.52,0.5},0.4,{ta,tb}]},
  },
  AspectRatio->Automatic]
];
```