情報処理 II 第 11 回

Mathematica 入門 (1)

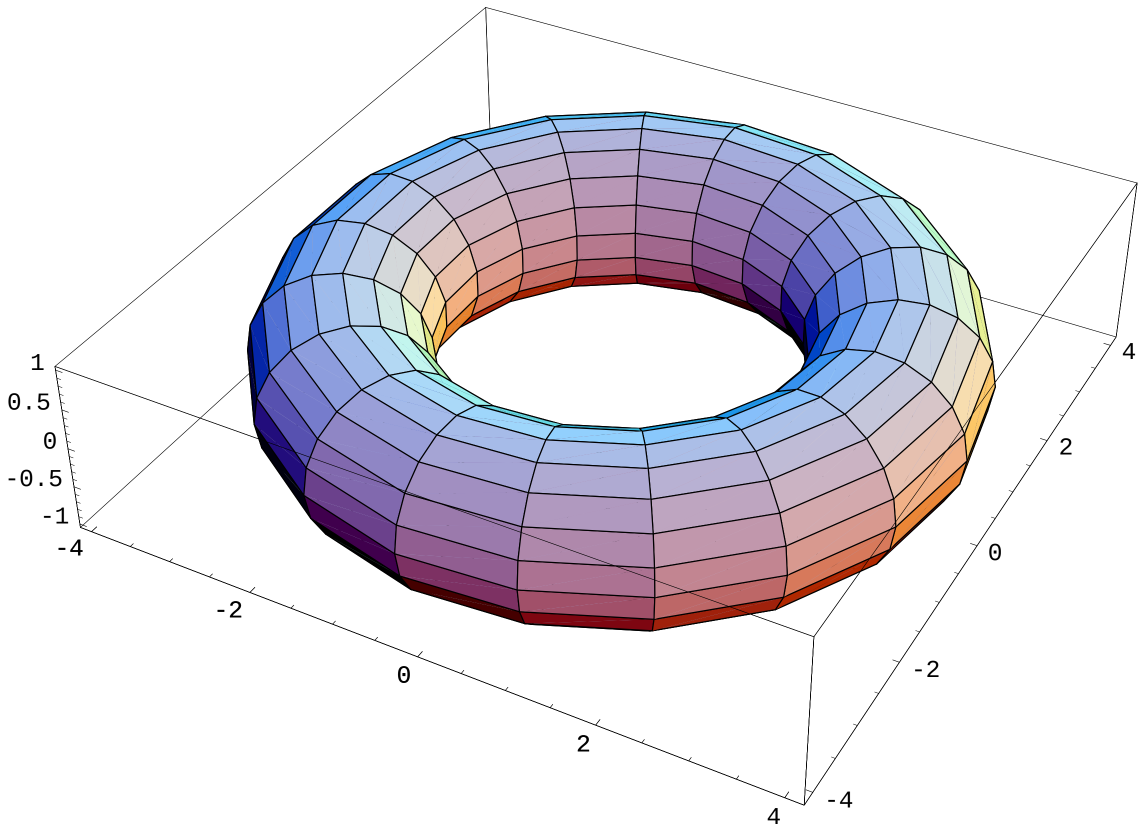
桂田 祐史（かつらだ まさし）

2001 年 7 月 5 日

代表的な数式処理系である Mathematica（マセマティカ）を体験しよう。数式処理でどういうことが出来るのか大体の雰囲気をつかんで、今後の学習・研究のヒントにしてもらうのがねらい。

1　Mathematica ってこんなもの

以下の例は、数学科の計算機である oyabun にログインして、 Mathematica を実行してみたものです。プログラムなどは書かずに、式を順次入力して計算結果を表示させています。



```
oyabun% math
Mathematica 4.0 for Solaris
Copyright 1988-1999 Wolfram Research, Inc.
 -- Motif graphics initialized --

In[1]:= 1/2 + 1/3                    ← 分数計算

        5
Out[1]= -                            → ちょっと見難いですけどね
        6

In[2]:= a={{0,1},{6,1}}              ← 行列の入力

Out[2]= {{0, 1}, {6, 1}}

In[3]:= Eigenvalues[a]               ← 行列の固有値の計算

Out[3]= {-2, 3}

In[4]:= Eigenvectors[a]              ← 行列の固有ベクトルの計算

            1        1
Out[4]= {{-(-), 1}, {-, 1}}
            2        3

In[5]:= Expand[(x+y)^6]              ← 式の展開

         6      5          4  2       3  3       2  4         5    6
Out[5]= x  + 6 x  y + 15 x  y  + 20 x  y  + 15 x  y  + 6 x y  + y

In[6]:= N[Pi,50]                     ← 円周率 50 桁

Out[6]= 3.1415926535897932384626433832795028841971693993751

In[7]:= Integrate[Log[x],x]          ← 不定積分

Out[7]= -x + x Log[x]

In[8]:= Plot3D[x^2 - y^2, {x,-1,1}, {y,-1,1}]  ← グラフ

Out[8]= -Graphics-                   → ここで画面に図が表示されます

In[9]:= Solve[x^3+2x==1]             ← 3 次方程式を解かせてみる

  結果は一見に価するけれど、紙を食うのでカットします。

In[10]:= ParametricPlot3D[{Cos[t](3+Cos[u]),Sin[t](3+Cos[u]),Sin[u]},
         {t,0,2Pi},{u,0,2Pi}]        ← トーラスを描かせる。

Out[10]:= -Graphics3D-

In[11]:= Quit                        ← 終了
oyabun%
```

2　数式処理とは

プログラミング言語 (計算機言語) の中には、数値や文字だけでなく、

数式をデータとして扱うことの出来る「数式処理言語」

と呼ばれるものがあります。数式処理言語を使えるシステム (ソフトウェア) を数式処理系と呼びます。数式処理系としては Mathematica 以外に **REDUCE**(リデュース), **Maple**(メイプル) などが有名です。

C のようなプログラミング言語は、プログラムの中では「数式」を書けますが、関数 `scanf()` や `printf()` 等で入出力可能なデータは、数や文字列だけで、例えば `-2/5` のような分数式の入力は出来ません。またグラフを描くプログラムを作る場合に、範囲や、分割数の指定等は実行時に入力出来ても、グラフを描こうとしている関数自体は (普通の方法では) 入力できず、プログラムの中に自分で埋め込むしかなかったわけです。そういう意味では C は不自由な言語であると言えます[1]。

3　この講義で用いる Mathematica

C 言語等は国際規格があり、無償で利用できる処理系もありますが、 Mathematica は Wolfram Research という一企業の所有物で、処理系は同社が作成・販売しているものしかありません。予算の関係で、利用できる個数が限られています。

以前は情報科学センターもライセンスを持っていたのですが、現在は使えないようです (?)。

現在、数学科が持っているライセンスは以下の通り。

- version 1.2 を何プロセスでも利用可能であるライセンスのあるマシンが 1 台 (fiddle)
  コマンド名は `math1.2` です (現在何故かネットワーク越しにグラフが表示できません)。

- version 2.2 を 2 プロセス利用可能であるライセンスのあるマシンが 9 台
  (kobun00, kobun01, kobun02, kobun07, kobun08, kobun09, kashira, anesan, anego)
  コマンド名は `math2.2` です (現在何故かネットワーク越しにグラフが表示できません)。

- version 4.0 を 8 つ利用可能であるライセンス
  マシンはどれも可 — 上にあげたものでも OK で、それ以外に oyabun というマシンがあります。負荷を分散させるため、授業中は oyabun を使います。
  コマンド名は `math` または `mathematica` です。

`mathematica` コマンドは GUI インターフェイスですが、この講義では CUI インターフェイスの `math` コマンドだけを説明します。 `math1.2`, `math2.2` コマンドは基本的に `math` と同じです。

以上から分かるように、同時に $8+2\times 9+\alpha = 26+\alpha$ 個の Mathematica が利用可能です (α は fiddle で実行する分)。この講義時間中は、各ユーザーがどのマシンを利用するか、くじで決めます。 (講義時間外は自由にしてもらって構いません。 oyabun, fiddle, kobun00 で math コマンドあるいは mathematica コマンドを使うのが良いでしょう。)

[1] もちろん不自由さを補って余りある大きな利点があるから、現在でも盛んに使われているわけです。例えば、実際の処理系の数値計算の速さで比べると C が圧勝します。原理的には一つのプログラミング言語があれば、どんな計算でも出来るはずなのですが、実際的な意味で万能のプログラミング言語と呼べるものは存在せず、適材適所を心がけることが重要です。みなさんも、あまり一つの言語、一つのシステムにこだわらずに、機会があったら色々なものを勉強してみましょう。

4 基本的な使い方

情報科学センターから数学科のワークステーション上の Mathematica を利用する場合について説明します。

4.1 起動・終了

1. 最初に利用するマシン (以下ではそのホスト名を *hostname* と書く) を選びます。通常は前節で列挙したマシンのいずれか (oyabun, kobun00, fiddle が速い？)。講義中はくじ引きで当たったマシン。
2. 情報科学センターのマシン waltz*xy* からは xhost *hostname* に続いて rlogin *hostname* で *hostname* にログインします。万一ログイン出来なかった場合は、ゲスト・アカウントでログインします[2]。
3. *hostname* 上で setenv DISPLAY waltz*xy*:0.0 とコマンドを打ちます (xy は適当に)。
4. Mathematica の起動コマンドは、

   math CUI のコマンド

   mathematica GUI のコマンド

   math1.2 fiddle で、 version 1.2 の Mathematica を起動するためのコマンド。 CUI.

   math2.2 kobunxy ($xy = 00, 01, 02, 07, 08, 09$), kashira, anesan, anego で、 version 2.2 の Mathematica を起動するためのコマンド。 CUI.
5. 終了するには Quit コマンド (先頭が大文字であることに注意)。
6. *hostname* からログアウトするには logout とする。

空(す)いているときは、 GUI の mathematica コマンドか、 CUI の math コマンドを Emacs の shell buffer の中で使うのがお勧め。授業中はマシンへの負荷が心配なので遠慮して下さい[3]。

waltz11 から oyabun 上の Mathematica を使う

```
waltz11% xhost oyabun                  ← 図を表示するためのオマジナイ (1)
waltz11% rlogin oyabun
oyabun% setenv DISPLAY waltz11:0.0     ← 図を表示するためのオマジナイ (2)
oyabun% math                           ← Mathematica 4.0 を起動
Mathematica 4.0 for Solaris
Copyright 1988-1999 Wolfram Research, Inc.
-- Motif graphics initialized --
In[1]:=
....
In[n]:= Quit                           ← Mathematica を終了
oyabun% logout                         ← oyabun をログアウト
waltz11%
```

[2] rlogin *hostname* -l guest とする (l は L の小文字)。パスワードは尋ねてください。
[3] 現在 mule で M-x shell が動かない障害があります。 xemacs なら動くのですが...