数学科計算機入門 「Mathematica」 資料

Dec. 13, 1999, 奥乃*, okuno@is.noda.sut.ac.jp, http://csai03.is.noda.sut.ac.jp/~okuno/*

1 Mathematica とは何か

1. 数式処理 — 数式微分, 数式積分
2. 数値計算 — 無限精度での計算,
3. グラフィックス, サウンド, さまざまなパッケージ
4. 数学の分野 — 線形方程式, 微分方程式, 組合せ論, グラフ理論
5. 理科大はサイトライセンス (math)

2 簡単な使い方

2.1 数値計算

In[1] := `3.141 [+|-|*| |/|^] 2.718`
% は直前の入力の結果. %n あるいは `Out[n]` も使える.
定数： `Pi` $= \pi$, `E` $= e$, `I` $= i = \sqrt{-1}$,
`Infinity` $= \infty$, `Degree` $= \pi/180$
In[2] := `Sqrt[2.0]`
In[3] := `%^2`
In[4] := `FullForm[%]`
計算機内の内部表現を返す
In[5] := `?FullForm`
`?` はその後の関数名の説明を与える.
In[6] := `3/433` 厳密値
In[7] := `3/433 //N` 近似値 `N[3/433, 桁数]`
In[8] := `2^100` 厳密値 `2^100 //N` 近似値
In[9] := `Sqrt[2]` 厳密値 `Sqrt[2] //N` 近似値
In[10] := `N[Sqrt[2]]` 近似値 or `Sqrt[2]//N`
入力が ; で終ると, 出力表示なし.

2.2 数学関数

`Sqrt[`x`]` ($\sqrt{x}$), `Exp[`x`]` (e^x), `Log[`x`]` ($log_e x$), `Log[`b, x`]` ($Log_b x$), `Sin[`r`]` ($\sin(r)$), `Cos[`r`]` ($\cos(r)$), `Tan[`r`]` ($\tan(r)$), `ArcSin[`x`]` ($\arcsin(x)$), `ArcCos[`x`]` ($\arccos(x)$), `ArcTan[`x`]` ($\arctan(x)$), $n!$ ($n!$), `Abs[`x`]` ($|x|$), `Round[`x`]` (四捨五入), `Mod[`n, m`]` ($n \bmod m$), `Random[`x`]` (乱数), `Max[`$x, \cdots$`,]` (最大値), `Min[`$x, \cdots$`,]` (最小値), `FactorInteger[`x`]` (素因数分解)
`Re[`z`]` (実数部分), `Im[`z`]` (虚数部分), `Conjugate[`x`]` ($z^*, \overline{z}$), `Abs[`z`]` ($|z|$), `Arg[`z`]` ($\phi, z = |z|e^{i\phi}$)

2.3 変数・関数の定義・代入・消去

x `=` $value$ or x `=` y `=` $value$ 永久代入
x `= .` or `Clear[`x`]`
$expr$`/.` x `->` $value$ 式 $expr$ の x を $value$ で置換
$expr$`/.` { x `->` $xvalue$, y `->` $yvalue$}
式 $expr$ の x を $xvalue$ で, y を $yvalue$ で置換
f`[`x`_]` `:=` $expr$ 関数 $f(x)$ を定義.
`?`f 関数 f の定義式を表示
`Clear[`f`]` 関数 f のすべての定義を削除

2.4 関係演算子と論理演算子

`==, !=, >, <, >=, <=`
In[1] := $\{3 < 5 <= 6\}$
`!`p 論理否定 ($\neg p$),
p `&&` q `&&` $\cdots$ 論理積 ($p \wedge q \wedge \cdots$),
p `||` q `||` $\cdots$ 論理和 ($p \vee q \vee \cdots$),
`Xor[`$p, q, \cdots$`]` 排他的論理和
`If[`$p, then, else$`]` if 文
`LogicalExpand[`$expr$`]` 論理式を展開

2.5 リスト — データが並んだもの

$\{a, b, c\}$ リストの定義
In[1] := $\{8, 7, 12\}$
In[2] := $\{8, 7, 12\}$`^2 + 1`
加減算，乗除算, 数学関数が適用できる.
In[3] := `Sin[`$\{8, 7, 12\}$`]`
`Part[`$list, i$`]` or $list$`[[`i`]]` i 番目のリスト要素抽出
`Part[`$list, \{i, j, \cdots\}$`]` or $list$`[[`$\{i, j, \cdots\}$`]]`
$i, j, \cdots$ 番目のリスト要素抽出 $(i, j, \geq 1)$
`Part[`$list, i$`]` `=` $value$ or $list$`[[`i`]]` `=` $value$
i 番目のリスト要素を $value$ で置換

`{a, b, c}` ベクトル (a, b, c)
`{{a, b}, {c, d}}` 行列 $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

2.6 オンラインヘルプ・パッケージ

`?Name` $Name$ に関する情報表示
`??Name` $Name$ に関する詳細情報表示
`?Azz*` Azz で始まるすべての事項に関する情報表示
`<<package` $package$ を読み込む
`Remove["name"]` 定義した関数 $name$ を除去

3 グラフィックス

`Plot[`f, $\{x, xmin, xmax\}$`]`
$f(x)$ の区間 $xmin \sim xmax$ のグラフを描く
`Plot[`$\{f1, f2, \cdots\}$, $\{x, xmin, xmax\}$`]` 複数のグラフ
In[1] := `Plot[Sin[x], x, 0, 2Pi]`
In[2] := `Plot[{BesselJ[1,x],BesselJ[2,x],`
`BesselJ[3,x],BesselJ[4,x]}, {x, 0, 10}]`
In[3] := `Plot[Evaluate[Table[BesselJ[n,x],{n,4}]]],`
`{x, 0, 10}]`
`Show[`$plot, \cdots$`]` 再表示
`Plot3D[`f, $\{x, xmin, xmax\}$`]` 3 次元プロット
`Plot[`$\{f1, f2, \cdots\}$, $\{x, xmin, xmax\}$`]` 複数のグラフ
`ContourPlot[`f, $\{x, xmin, xmax\}$, $\{y, ymin, ymax\}$`]`
$xmin \sim xmax$, $ymin \sim ymax$ での等高線の表示
`DensityPlot[`f, $\{x, xmin, xmax\}$, $\{y, ymin, ymax\}$`]`
$xmin \sim xmax$, $ymin \sim ymax$ での密度グラフの表示

In[4] := `ContourPlot[Sin[x] Sin[y],{x,-2,2},{y,-2,2}]`
In[5] := `Plot3D[Sin[x y], {x,0,3}, {y,0,3}]`
In[6] := `Show[%, ViewPoint -> {0, -2, 0}]`
In[7] := `Plot3D[10 Sin[x+Sin[y]], {x,-10,10},`
`{y,-10,10}, PlotPoints->50]`
In[8] := `Plot3D[BesselJ[0,Sqrt[x^2+y^2]],{x,-15,15},`
`{y,-15,15}, PlotPoints -> 50]`
`Display["ファイル名", expr, "EPS"]` ファイル出力

4 代数計算

`Expand[expr]` 展開し, 積和形に変形.
`ExpandAll[expr]` 完全展開
`Factor[expr]` 最小因数の積の形に変形.
In[1] := `Expand[ (1 + x + 3 y) ^ 4]`
In[2] := `Factor[%]`
`Simplify[expr]` 標準的な代数変形で簡略化
`FullSimplify[expr]` 標準外の代数変形も使い簡略化
In[3] := `Integrate[1/(x^4-1),x]` 不定積分
In[4] := `D[%,x]` 微分
In[5] := `Simplify[%]`
In[6] := `Simplify[Gamma[x] Gamma[1-x]]`
In[7] := `FullSimplify[%]` は $\pi \csc(\pi x)$ となる
`Together[expr]` 通分
`Apart[expr]` 部分分数に展開

5 記号数学

`D[`f, x`]` 偏導関数 $\frac{\delta f}{\delta x}$
`D[`$f, x, y, \cdots$`]` 偏導関数 $\frac{\delta}{\delta x}\frac{\delta}{\delta y}\cdots f$
`D[`f`,{`x, n`}]` n 階の偏導関数 $\frac{\delta^n}{\delta x} f$
`Integrate[`f, x`]` 不定積分 $\int f dx$
`Integrate[`f, x, y`]` 多重不定積分 $\int dxdyf$
`Integrate[`f`, {`$x, xmin, xmax$`}]` 定積分 $\int_{xmin}^{xmax} f dx$
`Sum[`f`, {`$i, imin, imax$`}]` 和 $\sum_{i=imin}^{imax} f dx$
`Sum[`f`, {`$i, imin, imax, step$`}]`
$i = imax$ から $imax$ まで $step$ 毎の和
`Product[`f`, {`$i, imin, imax$`}]` 積 $\Pi_{i=imin}^{imax} f$
`Product[`f`, {`$i, imin, imax, step$`}]`
$i = imax$ から $imax$ まで $step$ 毎の積

6 方程式を解く

6.1 代数方程式

`Solve[`lhs `==` rhs, x`]` 方程式 $lhs = rhs$ を x で解く
In[1]:= `Solve[3 x^3 -2 x^2 + 4x -3 ==0 , x]`
In[2]:= `N[%]`
In[3]:= `x /. %`
In[4]:= `x^3 + 5 x^2 + 4 x /. %%`
`Solve[{`$equ_1, eqn_2, \cdots$`}, {`$x, y, \cdots$`}]`
連立方程式 $eqn_1, eqn_2, \cdots$ を $x, y, \cdots$ で解く
`Eliminate[{`$eqn_1, eqn_2, \cdots$`}, {`$x, \cdots$`}]`
連立方程式で変数 $x, \cdots$ を消去する
`Reduce[{`$eqn_1, eqn_2, \cdots$`}, {`$x, \cdots$`}]`
連立方程式を変形・簡略化し, 条件付き解まで探す.

6.2 微分方程式

`DSolve[`$eqns, y[x], x$`]`
独立変数 x で, $y[x]$ について微分方程式を解く
In[1]:= `DSolve[y'[x] == a y[x] + 1, y[x] , x]`
$Out[1]$:=$\{\{y(x) \to -\left(\frac{1}{a}\right) + e^{a\,x}\,\mathrm{C}(1)\}\}$
In[2]:= `DSolve[{y'[x]==a y[x]+1,y[0]==0},y[x],x]`
$Out[2]$:=$\{\{y(x) \to \frac{-1+e\hat{}a\,x}{a}\}\}$
`DSolve[`$eqns, y, x$`]`
$y[x]$ の解を純粋な x の値として返す

6.3 ベキ級数・極限

`Series[`$expr$`, { `$x, x0, n$`}]` $expr$ を $x = x0$ の近傍で n 次以下の精度でベキ級数展開
In[1]:= `Series[(1 - x)^n, x, 0, 3]`
$1 - n\,x + \frac{(-1+n)\,n\,x^2}{2} - \frac{(-2+n)\,(-1+n)\,n\,x^3}{6} + \mathrm{O}(x)^4$
`Normal[`$series$`]` ベキ級数から剰余項を除去
`Limit[`$expr$`, `x `->` $x0$`]` $lim_{x \to x0} expr$

7 数値計算

`NIntegrate[`f`, {`$x, xmin, xmax$`}]` 数値積分 $\int_{xmin}^{xmax} f dx$
`NIntegrate[`f`, {`$x, xmin, xmax$`}, {`$y, ymin, ymax$`}]`
数値多重積分 $\int_{xmin}^{xmax} dx \int_{ymin}^{ymax} dy f$
`NSum[`f`, {`$i, imin$`, Infinity}]` 数値総和 $\sum_{xmin}^{\infty} f dx$
`NProduct[`f`, {`$i, imin$`, Infinity }]` 数値累積 $\Pi_{i=imin}^{\infty} f$
`NSolve[`lhs `==` rhs, x`]` 方程式 $lhs = rhs$ の数値解
`NSolve[{`$equ_1, eqn_2, \cdots$`}, {`$x, y, \cdots$`}]`
連立多項方程式 $eqn_1, eqn_2, \cdots$ を数値解析的に解く
`FindRoot[`lhs `==` rhs`, {`$x, x0$`}]`
$x = x0$ を出発点として方程式の数値解を求める
`NDSolve[`$eqns, y$`, {`$x, xmin, xmax$`}]` $x_{xmin} \le x \le x_{xmax}$ で微分方程式を数値解析的に解く
`InterpolatingFunction[`$data$`][`x`]` 点 x で補間式

`ConstrainedMin[`$f, ineqs$`, {`$x, y, \cdots$`}]` 線形な制約条件 $ineqs$ の下で線形関数 f を最小にする $x, y, \cdots$ を探す
`FindMinimum[`f`, {`$x, x0$`}]`
$x = x0$ 近傍で f を最小にする x を探す
`FindMinimum[`f`, {`$x, x0$`}, {`$y, y0$`}]` 極小値の探索

`Fit[{`$y1, y2, \cdots$`}, {`$f1, f2, \cdots$`}, x]`
$y_1, y_2, \cdots$ にフィットする線形式 $c_1 f_1(x) + c_2 f_2(x) + \cdots$ の係数 $c_1, c_2, \cdots$ を求める.
`Fit[{{`$x1, y1$`}, {`$x2, y2$`}, ⋯}, {`$f1, f2, \cdots$`}, x]`
2 次元の観測値にフィットする f_i の係数 c_i を返す
`Fourier[`$data$`]` 数値フーリエ変換
`InverseFourier[`$data$`]` 逆フーリエ変換

8 反復処理

`Do[`$expr$`, {`$i, imax$`}]` for i = 1 to imax do expr
`Do[`$expr$`, {`$i, imin, imax, di$`}]`
`Print[`$expr$`]` $expr$ を表示
`Table[`$expr$`, {`$i, imax$`}]` Do の結果をリストにする
`TableForm[`$list$`]` $list$ を表形式で表示