

# 鱒の助

## 取扱説明書

第一版

北陸職業能力開発大学校

情報技術科 新浜 裕幸（開発当時：専門課程 2 年）

田中 信博（開発当時：専門課程 2 年）

# 目次

1 はじめに .....	4	3.9 2次元パラメトリック曲線グラフ表示.....	11
1.1 鱒の助とは .....	4	3.10 3次元パラメトリック曲線グラフ表示 .....	12
1.2 ライセンス・免責事項.....	4	3.11 3次元パラメトリック曲面グラフ表示 .....	13
1.3 利用環境.....	4	3.12 回帰分析グラフ表示.....	14
1.4 機能概要.....	4	3.13 積分グラフ表示.....	15
2 セットアップ .....	5	3.14 微分方程式のグラフ表示 .....	15
3 操作方法 .....	6	3.15 ベクトル場のグラフ表示.....	16
3.1 画面構成 .....	6	3.16 プロット、相関係数、折れ線.....	17
3.2 入力方法 .....	6	3.17 入力支援ウィンドウ.....	18
3.3 算術演算.....	7	3.18 ファイルの新規作成、保存、印刷 .....	18
3.4 組込定数.....	8	3.19 文字のフォント、色を変える.....	18
3.5 組込関数.....	9	3.20 初期入力(プロファイル)を設定する .....	19
3.6 関数定義.....	9	3.21 コメントを書く .....	19
3.7 $y=f(x)$ 形式のグラフ表示 .....	10	3.22 出力先をファイルに設定する.....	19
3.8 $y=f(z,x)$ 形式のグラフ表示.....	10	3.23 マクロを使用する .....	20
		3.24 プロンプトの変更 .....	20
		3.25 度数法、ラジアン法で計算を行う .....	20
		3.26 定義した関数、変数を削除する .....	20

3.2.7	グラフウィンドウ .....	21
3.2.8	軸、グリッドの表示・非表示の切り替え .....	21
3.2.9	2D、3Dモード切り替え .....	22
3.3.0	グラフの拡大、縮小表示 .....	22
3.3.1	スケール .....	22
3.3.2	グラフをいろいろな角度から見る .....	22
3.3.3	グラフの表示領域を変更する .....	23
3.3.4	視野の移動 .....	23
3.3.5	グラフの生成範囲を変える .....	24
3.3.6	滑らかな曲面、曲線のグラフを表示する .....	24
3.3.7	一度描いたグラフを消す .....	24
3.3.8	グラフの色を変更する .....	24
3.3.9	論理単位を変更する .....	25
3.4.0	グラフをファイルに出力する .....	25
3.4.1	グラフウィンドウの表示・非表示切り替え .....	25
3.4.2	魔法陣を表示する .....	25
3.4.3	パスカルの三角形を表示する .....	26
3.4.4	カレンダーを表示する .....	26

4	組込関数 .....	27
4.1	一般数学関数 .....	27
4.2	三角関数 .....	29
4.3	統計関数 .....	30
4.4	その他の関数 .....	30
5	感想・バグ報告等連絡先 .....	32
6	困った時は .....	32

## 1 はじめに

### 1.1 鱒の助とは

Java 環境で利用可能な数式処理ソフトです。数式を解析し、その計算結果やグラフを出力する事が出来ます。

### 1.2 ライセンス・免責事項

鱒の助の著作権は新浜裕幸・田中信博・相川政和が有しております。

鱒の助はフリーソフトです。誰でも無料で自由に使用できます。

鱒の助の利用によって仮に何らかの障害や損失が生じた場合、たとえそれが鱒の助のバグに起因するものであったとしても、著作権者はその責任を負いません。利用者の自己責任において利用して下さい。

### 1.3 利用環境

Java の実行環境があれば鱒の助を利用できます。

- ・ J2SE JRE (Java Runtime Environment)

または、 J2SE SDK (Software Development Kit)

以下の環境で動作確認しました。

OS: Microsoft Windows 2000 Professional、Microsoft Windows XP Professional

Microsoft Windows XP HomeEdition

J2SE SDK v1.4.2

### 1.4 機能概要

鱒の助が提供する主な機能を表 1-1 に示します。

表 1-1 鱒の助の提供機能

#	機能
1	算術式の解析
2	変数、組込定数
3	組込関数（三角関数、他）
4	関数定義
5	総和（ ）、直積（ ）
6	微分、積分、微分方程式
7	回帰分析（最小 2 乗法）、相関係数
8	グラフ表示、グラフ操作
9	コマンド履歴
10	マクロ、プロファイル
11	入力支援
12	オプション設定
13	グラフ画像出力
14	編集（切取、複写、貼付）
15	ログ出力
16	その他各種コマンド

## 2 セットアップ

### 2.1 同梱ファイルリスト

同梱されているファイルは以下の通りです。

readme.txt

リリースノートです。最初にお読み下さい。

Trout.jar

鱒の助のプログラム本体です。

Trout.bat

鱒の助を起動するためのバッチファイルです。

lib/noah3d.jar

鱒の助がグラフ表示に利用しているライブラリ NOAH3D です。

doc/鱒の助取扱説明書

取扱説明書です。

doc/はやわかり鱒の助入門

初めて鱒の助を利用される方向けの入門書です。

### 2.2 セットアップ

インストーラはありません。

以下の手順でセットアップを行って下さい。

#### (1)Java の実行環境を準備

鱒の助は Java で開発されました。従って、Java の実行環境が必要です。

Java2 SDK SE 1.4.x を入手し、インストールして下さい。

SDK の代わりに JRE(Java Runtime Environment)でも結構です。

#### (2)NOAH3D をコピー

鱒の助はグラフ表示部に NOAH3D1.0(当校で開発)を利用しています。

同梱の noah3d.jar を次の場所にコピーしてください。

(J2SDK のインストール先)/jre/lib/ext

バージョンは 1.0 以降でないと動きません。

#### (3)鱒の助をコピー

HDD の適当な場所に Trout.jar と Trout.bat をコピーして下さい。

必要に応じて Trout.bat のショートカットを作成して下さい。

以上でセットアップ完了です。

バッチファイル Trout.bat をマウスでダブルクリックすると鱒の助が起動します。

### 3 操作方法

#### 3.1 画面構成

- ・ メインウィンドウ
- ・ 入力支援ウィンドウ
- ・ オプション設定ウィンドウ
- ・ グラフウィンドウ …etc

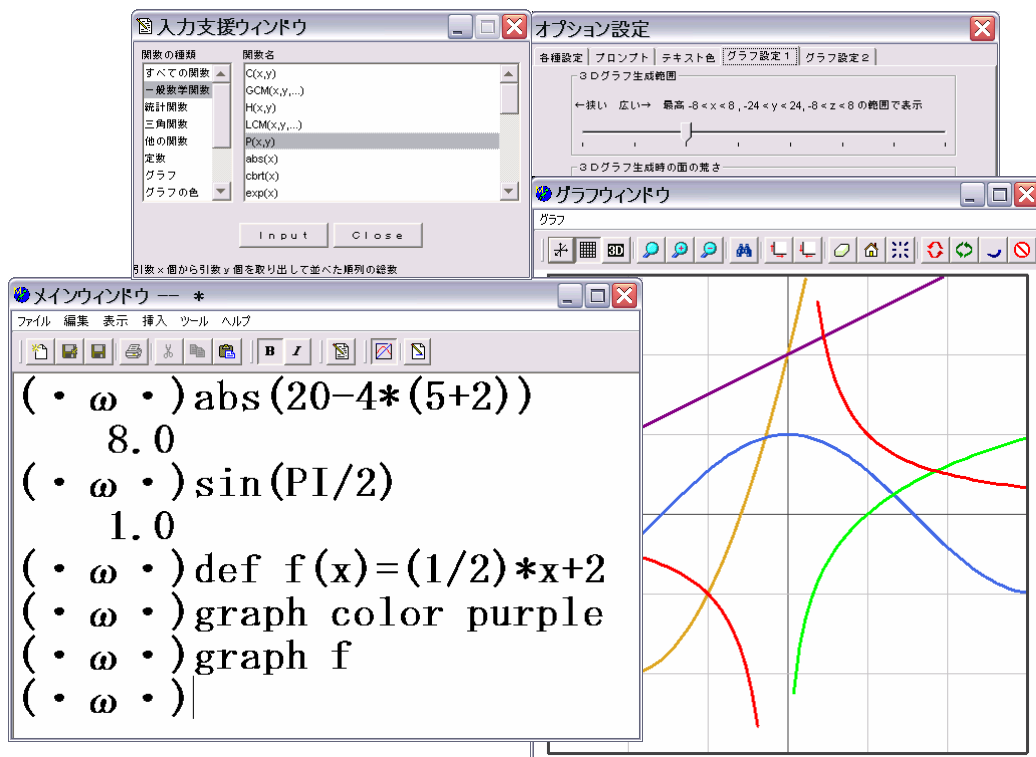


図 3-1 各種ウィンドウ

#### 3.2 入力方法



図 3-2 メインウィンドウ





##### ・入力

鰯の助を起動すると、テキスト画面にプロンプトとカーソルが表示されます。最新のプロンプトの隣が入力行であり、そこにコマンドを打ち込む事で結果が表示されます。文法は次のようになります。


(・・)コマンド

コマンドを記述して、それを実行するには、<Enter> キー（改行キー）を押します。すると、コマンドの結果が次の行に表示されます。次の行のプロンプトの隣が入力行に変わります。

##### ・編集

[編集]メニューの[切り取り]をクリックすると切り取りが行えます。ツールバーにある  ボタンをクリックしても同じです。切り取りと同様に、[コピー]をクリックもしくは  ボタンのクリックでコピー、[貼り付け]をクリックもしくは  ボタンは貼り付けです。また、[全て選択]または  ボタンでテキスト全てが選択されます。また、右クリックでも同様の作業ができます。

#### ・履歴


入力に失敗してしまった時、特に長い数式の一字のスペルミスでエラー表示が起きると入力し直すのは面倒です。一度入力されたコマンドは履歴として記憶されており、 ボタンの押下で表示させる事が出来ます。

前回の入力の履歴を読み込む必要がない場合は読み込まないように設定する事が出来ます。[ツール]メニューの[オプション]をクリックし、[各種設定]タブをクリックします。[コマンド履歴を保持(history.txt)]チェックボックスを外せば次回起動時からは読み込まれません。

### 3.3 算術演算

#### ・加減乗除

試しにプロンプトの隣の入力行に下のように打ってみてください。

 1+1 [改行] (Enter キーを押す)

すると、入力した式とその計算結果が出力されます。

罫の助で用いられる加減乗除の演算記号は一般に数学で使われるものと少し異なります。表 3-1 のようになります。

表 3-1 加減乗除記号対応表



演算記号	罫の助
+	+
-	-
×	*
÷	/

乗算はアスタリスク (\*)、除算はスラッシュ (/) を使って計算を行います。また、 $1 \div 3$  という式の答えは  $0.3333\cdots$  という循環小数になります。

0 で除算する数式の場合、エラーメッセージが表示されます。

#### ・代入文 (割り当て文)

変数を用いれば、計算の途中経過を残す事が出来ます。

 x=2  
2.0  
 x+2  
4.0

変数に使える変数名は、英字から始まる半角英数字もしくはアンダーバーを含めた 32 文字以内が使用可能です。もし、32 文字以上変数名を入れた場合は、最初の 32 文字を変数とみなし格納します。ただし、PI,E などの定数は変数名としては使用できません。また、英字に関しては、大文字と小文字を使い分けして下さい。

#### ・剰余

剰余とは、割り算で、割り切れずに残った部分 (余り) のことです。剰余を計算する時にはパーセント (%) を用います。また、一般数学関数の中の mod を使い、mod(5,2) と打っても同じ結果を返します。

#### ・累乗 (べき乗)

累乗とは、同じ数字を何度か掛け合わせる事です。累乗では山形ハット (^) を用います。 $2^3$  は  $2 \times 2 \times 2$  すなわち  $2^3$  を表しています。また、一般数学関数の中の pow を使い、pow(2,pow(2,3)) と打っても同じ結果を返します。

#### ・階乗

1 から  $n$  までの自然数の積を  $n$  の階乗といい、 $n!$  で表します。 $n!$  を数学的に解くなら、 $n! = n \cdot (n-1) \cdot (n-2) \dots 3 \cdot 2 \cdot 1$  となります。また、数学関数の中の `fact` を使っても同様の事ができます。補足として、小数や括弧で囲んだ負の整数を階乗すると、エラーになります。

また、二重階乗、素数階乗も使えます。二重階乗とは、1 (又は 2) から  $n$  まで、1 つ置きに自然数を掛け合わせることで、 $n!!$  と表します。

$n$  が奇数の時、 $n!! = 1 \times 3 \times 5 \times \dots \times n$

$n$  が偶数の時、 $n!! = 2 \times 4 \times 6 \times \dots \times n$

素数階乗とは、2 から  $n$  までの自然数の内、素数を掛け合わせることで、 $n\#$  と表します。また、一般数学関数の中の `primorial` を使っても同様の事ができます。

#### ・絶対値

絶対値は、`| |` で挟むと計算します。入力した実数が正の数または 0 ならば、入力した値を出力し、入力した実数が負の値ならば、入力した値の符号を取った値を出力します。また、数学関数の中の `abs` を使っても同様の計算ができます。

#### ・括弧

文字通り括弧 `()` を使えるようにした機能です。簡単にいうと優先度を高くして計算するためのものです。演算子の優先順位と結合規則は表 3-2 のようになっています。

表 3-2 演算子の優先順位と結合規則

優先度	演算子	結合規則
1	<code>()</code> <code>   </code>	
2	<code>!</code> <code>!!</code> <code>#</code>	
3	<code>^</code>	
4	<code>*</code> <code>/</code> <code>%</code>	
5	<code>+</code> <code>-</code>	
6	<code>=</code>	

この表の優先度の数字が小さいほど、優先度が高くなっています。

### 3.4 組込定数

あらかじめよく使われると思われる値を定数として用意します。定数は、表 3-3 のように設定してあります。定数に他の値を代入する事はできません。

表 3-3 定数表

文字	値	意味
PI	3.141592653589793	円周率
E	2.718281828459045	自然対数の底
G	6.67E-11	万有引力定数
g	9.81	重力加速度
C	299792458	真空中の光速
R	6.378E+6	地球の赤道半径



### 3.5 組込関数

罫の助は、関数を使う事が出来ます。文法は次のようになります。

```
( . . ) 関数名(引数)
```

引数には、求めたい値を入れて下さい。例えば、 $\sqrt{9}$  といった平方根の計算を行いたい時は、次のように入力します。

```
( . . ) sqrt(9)
3.0
```

因みに、sqrt は平方根を求める関数であり、引数を与える事で計算ができます。その他にも表 3-4 のような関数を使用することができます。

表 3-4 関数表

分類	関数名
三角関数	正弦、余弦、正接、割三角関数、逆三角関数、双曲線関数、頂点三角関数 等
一般数学関数	平方根、指数・対数、順列、組合せ、最大公倍数、最小公約数、素因数分解 等
統計関数	平均、中央値、最頻値、平均偏差、標準偏差、最大値、最小値、正規分布の乱数 等
その他の関数	微分、積分、微分方程式、総和、等差数列の和、等比数列の和 等

### 3.6 関数定義

関数を定義し、その関数を利用する事が出来ます。文法は次のようになります。

```
( . . ) def 関数名( 引数 ) = 関数式
```

def コマンドの後、関数名、括弧、引数、イコールの後、関数式を入力することで定義できます。関数式と引数を関連させることにより、関数値が求められます。例えば、 $y=3x+2$  の関数を定義する時は、次のように入力します。

```
( . . ) def f(x)=3*x+2
( . . ) f(5)
17.0
```

また、複数の引数をもつ関数を定義する場合の文法は次のようになります。

```
( . . ) def 関数名( 引数,引数,引数,... ) = 関数式
```

def コマンドの後、関数名、括弧、そして引数をコンマで区切ります。後はイコールの後に関数式を入力することで定義できます。引数の一つと同じように使う事が出来ます。例えば、 $y=x^2+z^2$  の関数を定義する時は、次のように入力します。

```
( . . ) def f(z,x)=x^2+z^2
( . . ) f(2,5)
29.0
```

関数定義の場合、同じ関数名であれば引数の数が異なっても前の関数は上書きされます。

### 3.7 $y=f(x)$ 形式のグラフ表示

$y=f(x)$ 形式のグラフ表示の文法は次のようになります。

```
( . . ) graph 関数名
```

関数名の中には、sin や cos などの組込関数や def コマンドで関数定義をした関数を使う事が出来ます。組込関数の中で表示できないグラフもあるのであらかじめお確かめ下さい。

例えば、関数  $y=\sin 12x \cdot \sin x$  のグラフを表示する時は、次のように入力します。

```
( . . ) def f(x)=sin(12*x)+sin(x)
```

```
( . . ) graph f
```

この結果をグラフウィンドウに図 3-3 のようにグラフが表示されます。

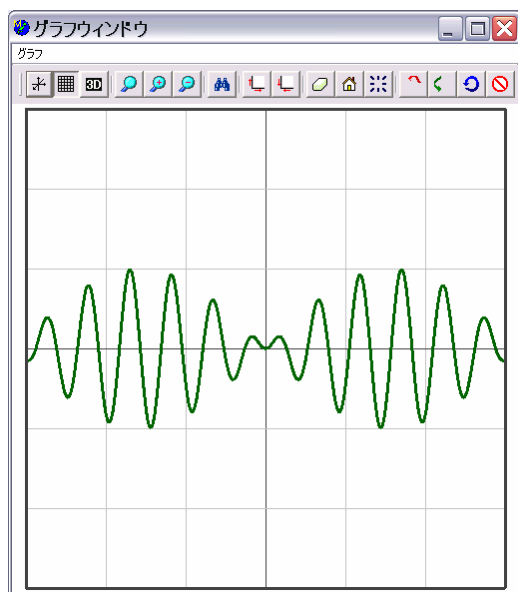


図 3-3  $y=\sin 12x \cdot \sin x$  のグラフ

### 3.8 $y=f(z,x)$ 形式のグラフ表示

$y=f(z,x)$ 形式のグラフ表示は、 $y=f(x)$ 形式のグラフ表示の場合と同様です。

例えば、関数  $y=x^2+z^2$  のグラフを表示する時は、次のように入力します。

```
( . . ) def f(z,x)=x^2+z^2
```

```
( . . ) graph f
```

この結果をグラフウィンドウに図 3-4 のようにグラフが表示されます。

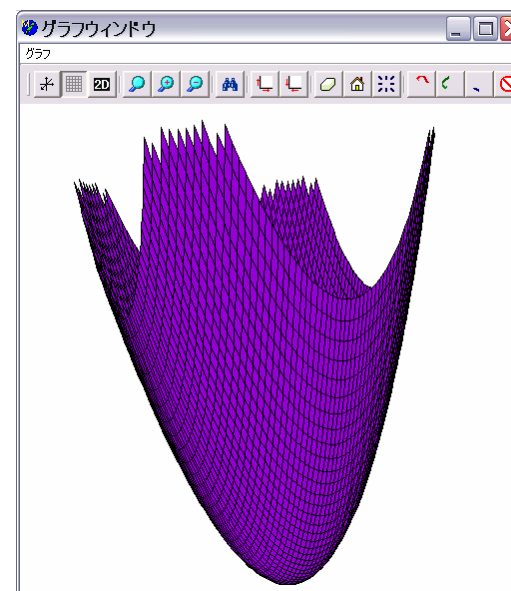


図 3-4  $y=x^2+z^2$  のグラフ

### 3.9 2次元パラメトリック曲線グラフ表示

$$\begin{cases} x = f(t) \\ y = g(t) \\ a \leq t \leq b \end{cases}$$

このような数式から図 3-5 のような自由度の高い曲線を描くことができます。  
この図からグラフウィンドウの表示領域のみ表示させます。

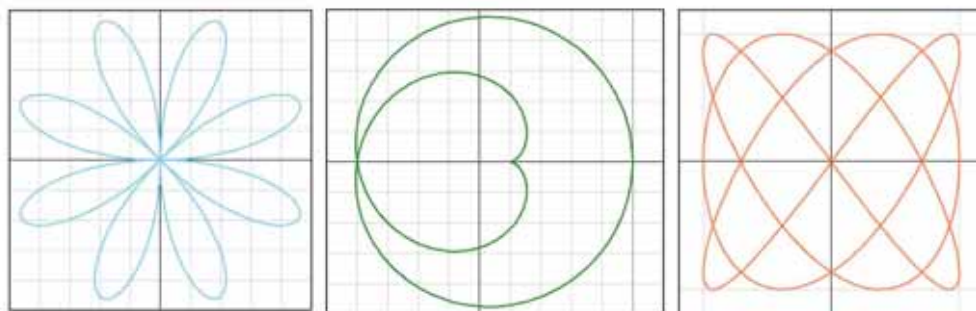


図 3-5 2Dパラメトリック曲線のグラフ例

2Dパラメトリック曲線グラフを表示の文法は次のようになります。

```
(. .) graph 関数名 : 関数名[媒介変数の範囲]
```

関数名をコロンで区切り、大カッコの中に、媒介変数の範囲を入力します。媒介変数の範囲には、始点と終点を入力します。また、その後に刻み幅を指定することも出来ます。刻み幅を細かくする事により、曲線が滑らかになります。

例えば、図 3-6 に示す単位円のグラフは以下の数式で表す事が出来ます。

$$\begin{cases} x = \cos(t) \\ y = \sin(t) \\ 0 \leq t \leq 2 \end{cases}$$

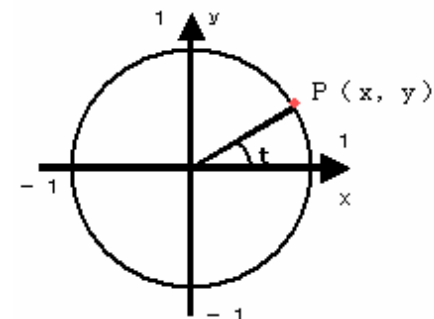


図 3-6 単位円

この数式を基にグラフを表示する時は、次のように入力します。

```
(. .) def f(t)=cos(t)
(. .) def g(t)=sin(t)
(. .) graph f:g[0,2*PI]
```

この結果をグラフウィンドウに図 3-7 のようにグラフが表示されます。

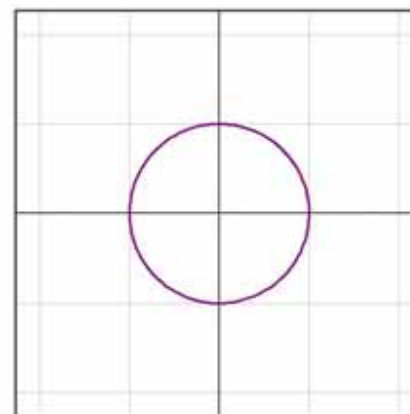


図 3-7 単位円のグラフ

### 3.10 3次元パラメトリック曲線グラフ表示

$$\begin{cases} x = f(t) \\ y = g(t) \\ z = h(t) \\ a \leq t \leq b \end{cases}$$

2Dパラメトリック曲線の式に、与える関数を一つ増やすことで、三次元空間内に図3-8のような3Dパラメトリック曲線が描けます。

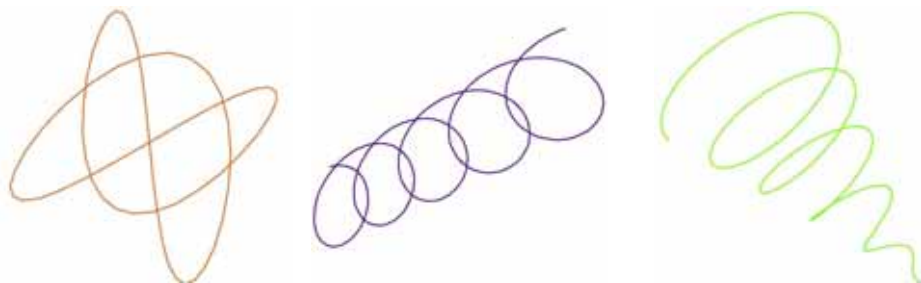


図3-8 3Dパラメトリック曲線のグラフ例

3Dパラメトリック曲線グラフを表示の文法は次のようになります。

```
(. .) graph 関数名 : 関数名 : 関数名 [媒介変数の範囲]
```

2Dパラメトリック曲線と同様に関数名をコロンで区切り、大カッコの中に、媒介変数の範囲を入力します。媒介変数の範囲には、始点と終点を入力します。また、その後刻み幅を指定することも出来ます。刻み幅を細かくする事により、曲線が滑らかになります。

例えば、図3-9に示すばねのグラフは以下の数式で表す事が出来ます。

$$\begin{cases} x = \sin(t) \\ y = t / -2.5 \\ z = \cos(t) \\ 0 \leq t \leq 10 \end{cases}$$

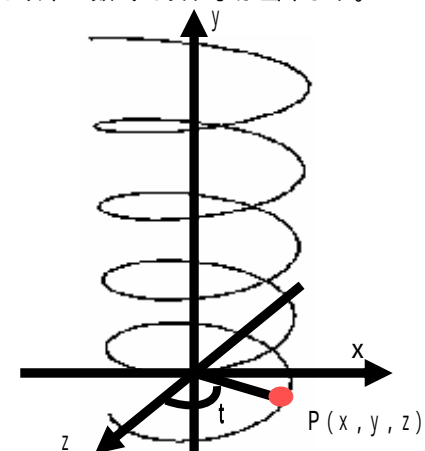


図3-9 バネ

この数式を基にグラフを表示する時は、次のように入力します。

```
(. .) def f(t)=sin(t)
(. .) def g(t)=t/PI-2.5
(. .) def h(t)=cos(t)
(. .) graph f:g:h[0,10*PI]
```

この結果をグラフウィンドウに図3-10のようにグラフが表示されます。



図3-10 バネのグラフ

### 3.1.1 3次元パラメトリック曲面グラフ表示

$$\begin{cases} x = f(s,t) \\ y = g(s,t) \\ z = h(s,t) \\ a \leq s \leq b \\ c \leq t \leq d \end{cases}$$

この数式のように媒介変数を二つに増やせば、図 3-11 のような 3 D パラメトリック曲面が描けます。

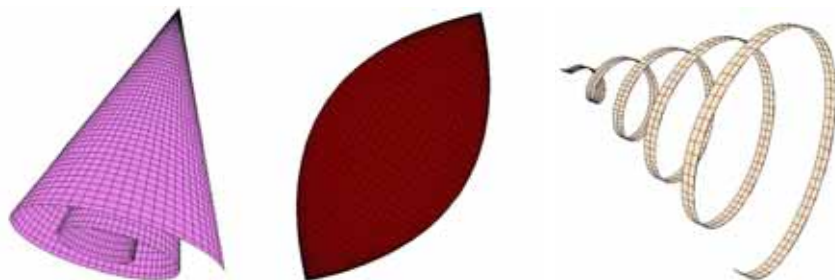


図 3-11 3 D パラメトリック曲面のグラフ例

3 D パラメトリック曲面グラフを表示の文法は次のようになります。

```
( . . ) graph 関数名 : 関数名 : 関数名 [媒介変数の範囲] [媒介変数の範囲]
```

3 D パラメトリック曲線と同様に関数名をコロンで区切り、大カッコの中に、媒介変数の範囲を入力します。媒介変数の範囲にはそれぞれに、始点と終点を入力します。また、その後にそれぞれ刻み幅を指定することも出来ます。刻み幅を細かくする事により、曲面が滑らかになります。

例えば、図 3-12 に示す円筒のグラフは以下の数式で表す事が出来ます。

$$\begin{cases} x = r \cdot \sin(t) \\ y = h \\ z = r \cdot \cos(t) \\ 0 \leq t \leq 2\pi \\ -2 \leq h \leq 2 \end{cases}$$

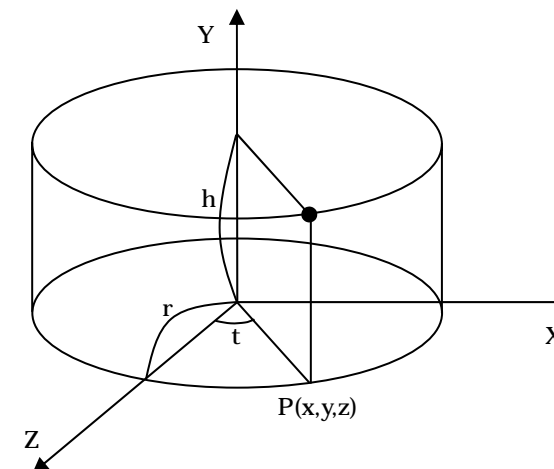


図 3-12 円筒

この数式を基にグラフを表示する時は、 $r=2$  の後に次のように入力します。

```
( . . ) def f(s,t)=r*sin(s)
( . . ) def g(s,t)=h
( . . ) def h(s,t)=r*cos(s)
( . . ) graph f:g:h[0,2*PI][-2,2]
```

この結果をグラフウィンドウに図 3-13 のようにグラフが表示されます。

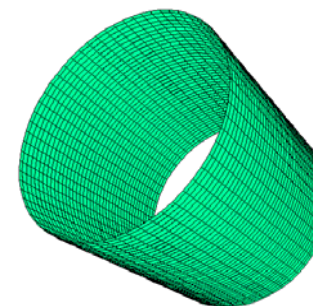


図 3-13 円筒のグラフ

3.1.2 回帰分析グラフ表示

実験データを直線、曲線の式に当てはめます。

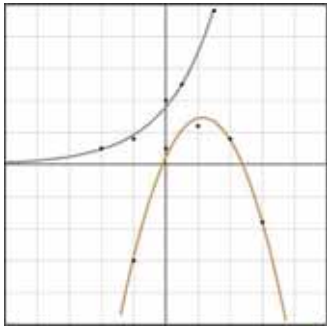


図 3-14 回帰分析のグラフ例

回帰分析グラフを表示の文法は次のようになります。

```
(. . .) regression 分類 (x0,y0)(x1,y1)(x2,y2)...(xn,yn)
```

regression コマンドの後に分類と要素を 2 組以上入力してください。回帰分析の分類として、直線回帰、2 次回帰～10 次回帰、指数回帰があります。その分類と、入力を表 3-4 に示します。

表 3-4 回帰分析の分類

分類	入力
多項式	1～10
$y = a \cdot e^{bx}$	e
$y = a \cdot b^x$	b
$a^y = b \cdot x$	log

例えば、表 3-5 実験データを減衰曲線に当てはめる場合について考えてみます。

表 3-5 実験データ例

分	3	6	9	12	15	18	21	24	27	30
温度差	60.8	44.8	28.8	22.4	16.0	9.6	8.0	6.4	4.8	3.2

減衰曲線の式は、

$$T = T_0 \cdot e^{-kt}$$

このように表されます。

- T<sub>0</sub> = 初期値  
e = 自然対数の底  
k = 係数  
t = 経過時間

表 3-5 のデータを減衰曲線に当てはめるには次のように入力します。(一部省略)

```
(. . .) regression e (3,60.8)(6,44.8)(9,28.8) ... (24,6.4)(27,4.8)(30,3.2)  
y = 80.1 * exp(-0.10758 * x)
```

結果の式に当てはめられました。図 3-15 に表示されるグラフを示します。

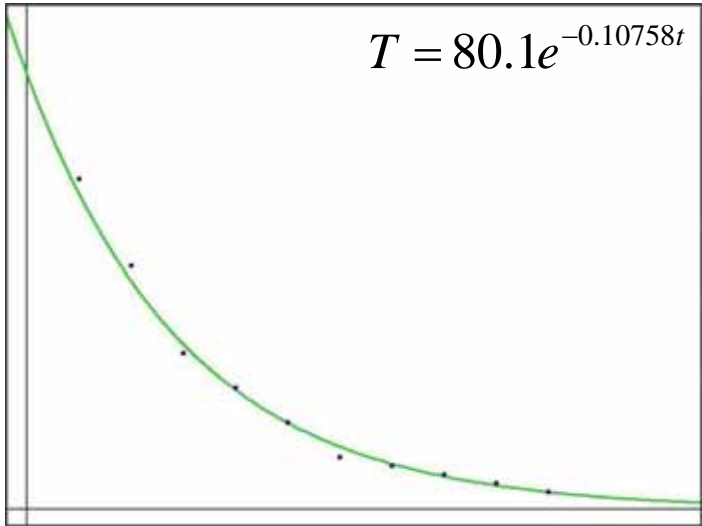


図 3-15 減衰曲線

### 3.1.3 積分グラフ表示

与えられた関数  $y=f(x)$  から、 $y = \int_0^x f(x)dx$  を近似計算して表示します。

積分グラフを表示の文法は次のようになります。

```
( . . ) graph integral(関数名)
```

ルンゲ・クッタ法を用いて積分を近似計算しているので、多少処理が遅い可能性があります。ご了承ください。

例えば、 $y=(x-1)^2-1$  と積分したグラフを表示する時は、次のように入力します。

```
( . . ) def f(x)=(x-1)^2-1  
( . . ) graph f  
( . . ) graph integral(f)
```

この結果をグラフウィンドウに図 3-16 のようにグラフが表示されます。

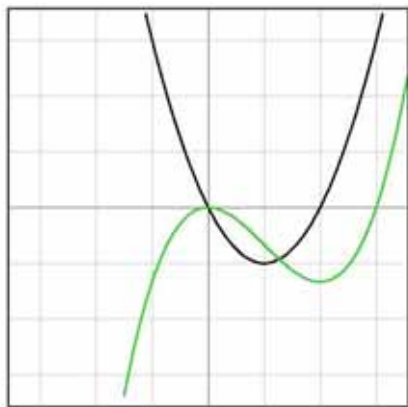


図 3-16 積分グラフ

図 3-16 では、 $y=(x-1)^2-1$  のグラフと  $y = \int_0^x \{(x-1)^2-1\}dx$  のグラフを表示させています。尚、わかり易いように、色分けをしています。

### 3.1.4 微分方程式のグラフ表示

多くの物理現象は微分方程式によって表されます。したがって、微分方程式を解けば、物理現象を解明できます。

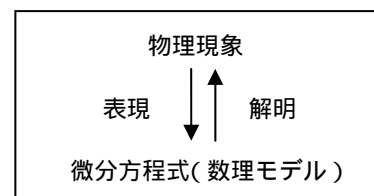


図 3-17 微分方程式と物理現象の関係

微分方程式、および初期値が次のように与えられているとします。

$$\frac{dy}{dx} = f(x, y)$$

$(x_0, y_0) \dots$  初期値

この場合、微分方程式を解き、グラフ化する時は、次のように入力します。

```
( . . ) graph diffsolve(x₀,y₀,f)
```

$x_0$  には  $x$  の初期値を、 $y_0$  には  $y$  の初期値を、 $f$  には引数 2 つの関数を入力します。

また、連立微分方程式や二階微分方程式を解くことも出来ます。まず、連立微分方程式を解き、グラフ化する時は、次のように入力します。

```
( . . ) graph simdiffsolve(x₀,y₀,z₀,f,g)
```

二階微分方程式を解き、グラフ化する時は、次のように入力します。

```
( . . ) graph diffsolve2(x₀,y₀,z₀,f,g)
```

$x_0$  には  $x$  の初期値を、 $y_0$  には  $y$  の初期値を、 $z_0$  には  $z$  の初期値を、 $f$  と  $g$  には引数 3 つの関数を入力します。



例えば、落下運動を解析する場合、物体の落下速度を  $v$  とし、運動方程式を立てると以下の式が得られます。

$$m \frac{dv}{dt} = mg - kv^2$$

$m$  = 重さ  
 $g$  = 重力加速度  
 $k$  = 空気抵抗係数  
 $v$  = 速さ  
 $t$  = 経過時間

この式を変形させると以下の式が得られます。

$$\frac{dv}{dt} = g - \frac{kv^2}{m}$$

罫の助には、この右辺の式と初期値  $t=0$   $v=0$  を与えます。

質量を 5 k g、空気抵抗係数を 0.2 として計算する時は次のように入力します。

```
(. .) m=5
(. .) k=0.2
(. .) def f(t,v)=g-k*v^2/m
(. .) graph diffsolve(0,0,f)
```

この結果をグラフウィンドウに図 3-19 のようにグラフが表示されます。次第に速度が増していき、一定の速度に収束していることが分かります。

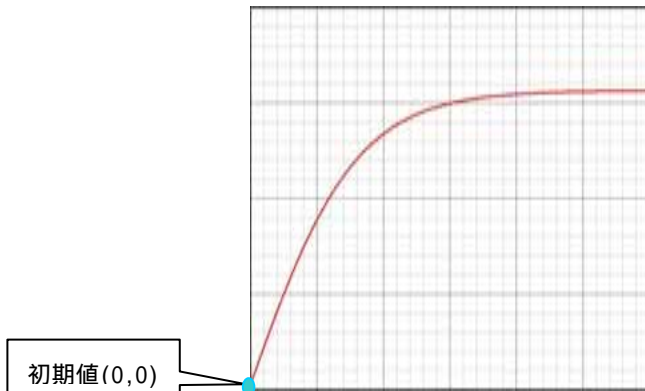


図 3-19 微分方程式グラフ

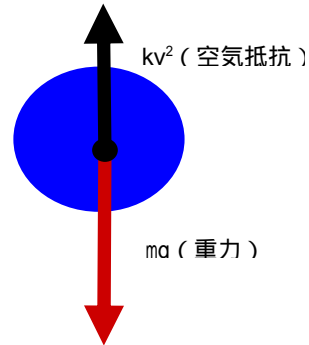


図 3-18 落下運動

### 3.15 ベクトル場のグラフ表示

$dy/dx = f(x,y)$  の式は二次元平面上のベクトル場を表します。

ベクトル場のグラフを表示の文法は次のようになります。

```
(. .) graph field(関数名)
```

例えば、 $dy/dx = \sin(x) + \cos(y)$  のベクトル場のグラフを表示する時は、次のように入力します。

```
(. .) def f(x,y)=sin(x)+cos(y)
(. .) graph field(f)
```

この結果をグラフウィンドウに図 3-20 のようにグラフが表示されます。

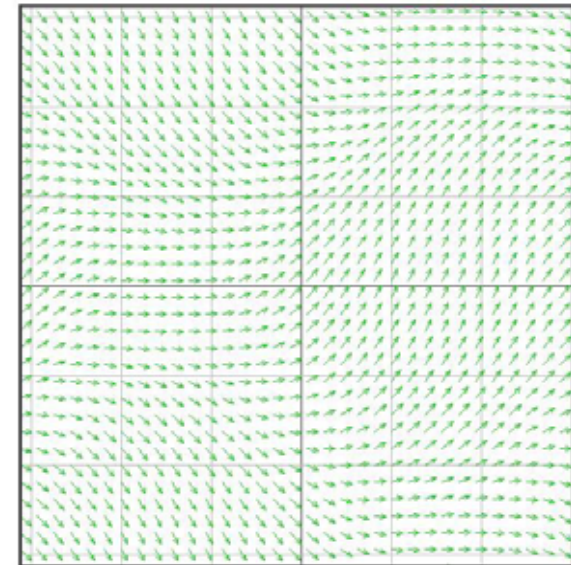


図 3-20 ベクトル場のグラフ



### 3.16 プロット、相関係数、折れ線

プロットを表示の文法は次のようになります。

```
( . . ) plot (x0,y0)(x1,y1)(x2,y2)...(xn,yn)
```

plot コマンドの後に打ちたい点の x 座標と y 座標を指定して下さい。

例えば、プロットを表示する時は、次のように入力します。

```
( . . ) plot (1,2)(2,3.5)(3,2.4)(4,4.2)(5,3)
```

この結果をグラフウィンドウに図 3-21 のようにグラフが表示されます。

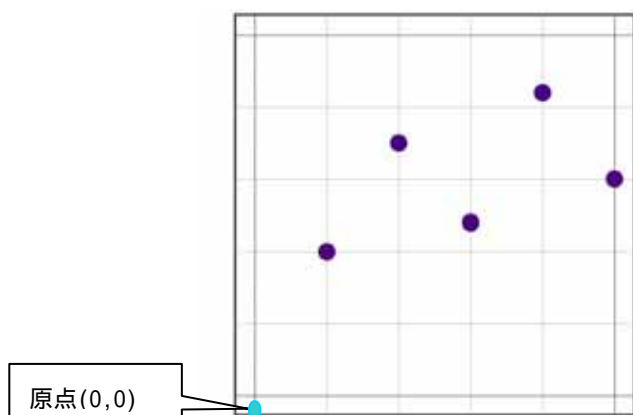


図 3-21 プロット

また、相関係数を求める事も出来ます。相関係数の文法は次のようになります。

```
( . . ) correlation (x0,y0)(x1,y1)(x2,y2)...(xn,yn)
```

correlation コマンドの後に打ちたい点の x 座標と y 座標を指定して下さい。

例えば、相関係数を求めるには次のように入力します。

```
( . . ) correlation (2,1)(-2,5)(3,8)(-1,-1)
```

```
0.364735074003
```

表示されるグラフはプロットと同様なので省略します。

折れ線グラフを表示の文法は次のようになります。

```
( . . ) graph line (x0,y0)(x1,y1)(x2,y2)...(xn,yn)
```

graph line コマンドの後に引きたい線の x 座標と y 座標を指定して下さい。

例えば、折れ線グラフを表示する時は、次のように入力します。

```
( . . ) graph line (1,2)(2,3.5)(3,2.4)(4,4.2)(5,3)
```

この結果をグラフウィンドウに図 3-22 のようにグラフが表示されます。

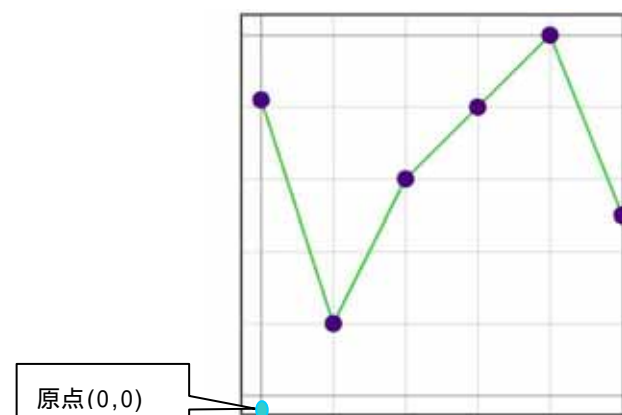


図 3-22 折れ線グラフ

### 3.17 入力支援ウィンドウ

鱒の助が提供している関数やコマンドを把握する時には時間がかかりますが、入力支援機能を使えば、関数やコマンドの説明を一覧表示して選択して入力することが出来ます。


メインウィンドウにある[挿入]メニューから[入力支援]をクリックするか、 ボタンをクリックします。すると、図 3-23 のような画面が表示されます。この画面から、いろいろな関数やコマンドが入力する事が出来ます。画面左下には、その関数やコマンドの説明が表示されます。







図 3-23 入力支援ウィンドウ

また、以下のように入力することでも関数やコマンド等の一覧表を見ることが出来ます。試してみてください。

定数表・・・const list	変数表・・・variable list
三角関数表・・・triangle list	演算表・・・operation list
統計関数表・・・statistics list	一般数学関数表・・・math list
自作関数表・・・function list	グラフ機能表・・・graph list
その他の関数表・・・another list	



### 3.18 ファイルの新規作成、保存、印刷

メインウィンドウにある[ファイル]メニューの[新規作成]をクリックするとメインウィンドウのテキスト画面の初期化などを行います。 ボタンでも同様の操作です。また、[ファイル]メニューの[名前を付けてテキストを保存]をクリックすると保存先ダイアログが表示されます。ファイル名を付けて保存すると、テキストファイルとして、現在表示されているテキストが保存されます。 ボタンでも同様の操作が出来ます。一度ファイルに名前を付けてテキストを保存すれば[ファイル]メニューの[上書き保存]をクリックすると同じファイルに上書き保存されます。 ボタンでも同様の操作ができます。

[ファイル]メニューの[印刷]をクリックすると印刷ダイアログが表示され、印刷することが出来ます。 ボタンからでも印刷が可能です。

### 3.19 文字のフォント、色を変える

文字のフォントを変更する場合、メインウィンドウにある[表示]メニューの[フォント]をクリックします。表示されるフォント名、サイズ、スタイルをそれぞれ選択します。

色を変更する場合、メインウィンドウにある[ツール]メニューの[オプション]をクリックします。[テキスト色]タブをクリックします。前景色、背景色、キャレット色、選択色の色をそれぞれ設定することが出来ます。また、テキストの文字はツールバーにある  ボタンをクリックすると太字に  ボタンをクリックすると斜体になります。

### 3.2.0 初期入力(プロファイル)を設定する

初期入力とは、鯖の助起動時に自動的に行われる一括入力の事です。起動するたびに同じ関数を定義したりする手間を省く時等に使用します。

実行ファイルと同じフォルダ内にある profile.txt にコマンドを入力しておけば、鯖の助起動時に自動的に一括実行されます。

profile.txt

```
def f(x) = x+3
def g(x) = x*2
a = 3
```

このようにファイルに入力されていれば起動時から関数  $f(x)$ ,  $g(x)$  や変数  $a$  が定義されている事になります。

また、起動時に初期入力を行う必要がない場合は読み込まないように設定する事が出来ます。[ツール]メニューの[オプション]をクリックし、[各種設定]タブをクリックします。[起動時に初期入力読み込み(profile.txt)]チェックボックスを外せば次回起動時から読み込まれません。

### 3.2.1 コメントを書く


入力した数式はどのような意味があるのか等、コメントとして残しておくことが出来ます。入力方法は、// (スラッシュを二つ続けて) の後にコメントを書き込んでください。

```
(・ ・) def f(z,x) = z^2+x^2 // y = z^2+x^2 の関数を定義
```

このように入力しておけば、ファイルに出力後、後から見直したりする時に便利です。

### 3.2.2 出力先をファイルに設定する

入力された数式や、その結果等を、指定したテキストファイルに出力することが出来ます。

[ファイル]メニューの[ロギングモード]をクリックします。後は入力していくだけです。終了するときは、もう一度[ロギングモード]をクリックします。メインウィンドウの  ボタンでも同様です。ロギングモード中は後から説明しますが、指定したファイルに指定した出力形式で追加出力されていきます。

まず、出力先ファイルを指定する時は、[ツール]メニューの[オプション]をクリックし、[各種設定]タブをクリックします。[ログファイル設定]の[ファイルパス]に保存するテキストファイルパスを指定します。( [参照] ボタンからテキストファイルを選択することも可能 ) 出力先ファイルは指定しなければ実行ファイルと同じフォルダの中にある log.txt に出力されます

出力形式を設定するときは、[各種設定]タブをクリックして [出力形式設定] コンボボックスの中から以下の三つの形式から選択します。

1. プロンプト、入力、出力結果をファイルに書き込む
2. 入力のみ、ファイルに書き込む
3. 出力結果のみ、ファイルに書き込む

例として、ロギングモード状態の時、プロンプトを (・ ・) として  $2*(3+1)$  と数式が入力した場合、指定したテキストファイルには以下のように出力されます。

1. (・ ・)  $2*(3+1)$   
8.0
2.  $2*(3+1)$
3. 8.0

2 はマクロファイルを作成する時に便利です。

### 3.2.3 マクロを使用する

マクロとは、あらかじめテキストファイルに記述しておいた数式やコマンド等を一括して実行する機能です。

使用する時は、メインウィンドウにある[ツール]メニューから[マクロ]をクリックして、表示されるダイアログからテキストファイルを選択します。

テキストファイルへの入力例を以下に示します。

sphere.txt

```
def x(u,v)=cos(u)*cos(v)
def y(u,v)=sin(u)
def z(u,v)=cos(u)*sin(v)
graph x:y:z[0,2*PI][-PI,PI]
```

マクロを使用して、このファイルを選択すると、グラフウィンドウに図 3-24 のようなスフィアが表示されます。

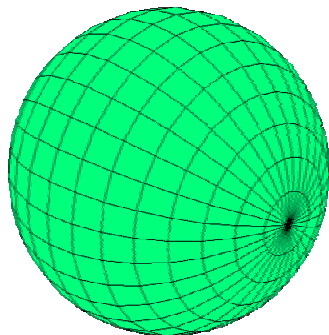


図3-24 スフィア

### 3.2.4 プロンプトの変更

プロンプトを自分の好きな様に変更することが出来ます。メインウィンドウにある[ツール]メニューのオプションをクリックして、[プロンプト]タブをクリックします。通常のプロンプトとエラー時のプロンプトを指定します。

### 3.2.5 度数法、ラジアン法で計算を行う

サインの計算をします。例えば  $\sin(\pi)$  と入力すればデフォルトでは計算結果は 0.0 と表示されます。これは計算がラジアン法を使用しているからです。この計算方法を度数法に変更すると、例えば、サイン 90 度の計算をする時は  $\sin(90)$  と入力できるようになります。計算方法を度数法、ラジアン法にそれぞれ変更する時には以下のように入力します。

degrees . . . 度数法	radians . . . ラジアン法
-------------------	---------------------

入力した次の行からは指定した計算方法で値が求められます。

尚、微積分やグラフ表示はラジアン法で行われています。

### 3.2.6 定義した関数、変数を削除する

定義した関数、変数が必要なくなれば削除する事が出来ます。関数、変数を削除する時、以下のように入力します。

function delete	関数名 . . . 自作関数を削除する。
function clear	. . . 自作関数をすべて削除する。
variable delete	変数名 . . . 変数を削除する。
variable clear	. . . 変数をすべて削除する。

### 3.27 グラフウィンドウ

図 3-25 はグラフウィンドウの全項目を表します。表示領域上に出ているものはメニューバーから[グラフ]を選択したものです。

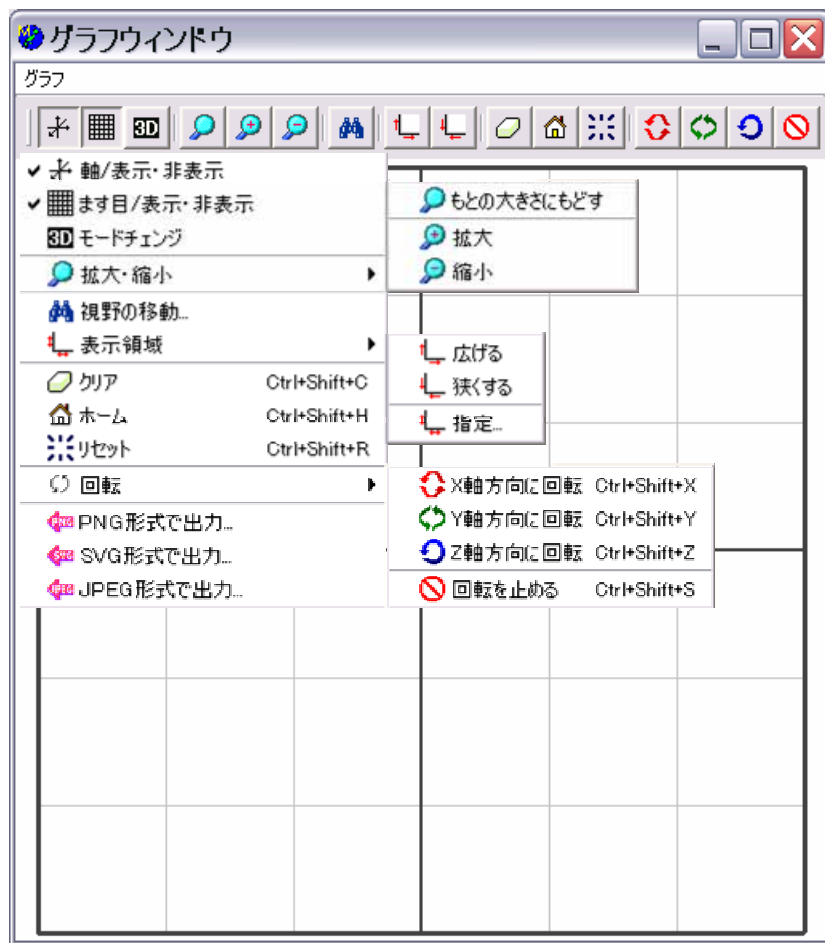




図 3-25 グラフウィンドウの全項目

### 3.28 軸、グリッドの表示・非表示の切り替え

軸の表示・非表示を切り替える時は、グラフウィンドウにあるツールバーから  ボタンをクリックします。表示された場合、2 D モードでは x 軸、y 軸が、3 D モードでは、x 軸、y 軸、z 軸表示されます。

同様にグリッドの表示・非表示を切り替える時は、 ボタンをクリックします。グリッドは 2 D モードの時のみ表示されます。3 D モードでは表示されません。軸とグリッドに関しては、コマンド入力でも可能です。軸を表示したい時は axis コマンドを使って、次のように入力します。

```
(. .) axis true
```

また、軸を非表示したい時は次のように入力します。

```
(. .) axis false
```

グリッドの場合は、measure コマンドの後 true か false を入力してください。

また、起動時の軸、グリッドの状態を指定する事が出来ます。指定する場合はメインウィンドウにある[ツール]メニューの[オプション]をクリックし、[グラフ設定 2] タブをクリックします。表示されている[起動時の軸初期状態]から 2 D、3 D モードの軸の状態を、[起動時のグリッド初期状態]の[グリッドの表示/非表示]から指定して下さい。

### 3.2.9 2D、3Dモード切り替え

グラフを2Dモード3Dモードに切り替える事が出来ます。2Dモードでは $y=f(x)$ 形式のグラフ等、3Dモードでは $y=f(z,x)$ 形式のグラフ等が表示されます。それぞれのグラフを表示すると、グラフの形式にあったモードへ自動的に切り替わります。2Dモードでグラフを表示していたとして、3Dモードでグラフを表示すると、先に表示していた2Dモードのグラフは表示されなくなります。もう一度2Dモードで表示されていたグラフを見るときにモードの切り替えが便利です。モード切り替えを行う時は2Dモードの時は $\text{3D}$ ボタンで3Dモードへ、3Dモードの時は $\text{2D}$ ボタンで2Dモードへ切り替えが行えます。

### 3.3.0 グラフの拡大、縮小表示

グラフを拡大する場合はグラフウィンドウにある[グラフ]メニューから[拡大・縮小]をポイントして、[拡大]をクリックします。 $\text{拡大}$ ボタンでも同様です。

縮小表示する場合は[縮小]をクリックします。 $\text{縮小}$ ボタンでも同様です。

スケール値を指定することも出来ます。[ツール]メニューから[オプション]を選択して、[グラフ設定2]タブをクリックして[もとの大きさにもどす選択時のScale 値]の値を指定してから[もとの大きさに戻す]をクリックすると、指定されたスケールに変わります。 $\text{戻す}$ ボタンでも同様です。

### 3.3.1 スケール

スケールとは、1論理単位あたり何ピクセルかを表した単位です。この値を大きくすると映し出されている画像が間近で見え、小さくすると離れたように見えます。スケール値を80にする時は次のように入力します。

```
( . . ) scale 80
```

### 3.3.2 グラフをいろいろな角度から見る

表示されたグラフをx軸、y軸、z軸方向に回転させる事が出来ます。

x軸を中心に回転させるときはグラフウィンドウのメニューバーにある[グラフ]メニューの[回転]をポイントして[X軸方向に回転]をクリックします。ツールバーにある $\text{X}$ ボタンでも同様です。同様にグラフは[Y軸方向に回転]をクリックまたは $\text{Y}$ ボタンでy軸、[Z軸方向に回転]をクリックまたは $\text{Z}$ ボタンでz軸方向にそれぞれ回転します。回転しているグラフを停止する時は $\text{停止}$ ボタンを押すと回転が停止します。

回転する間隔をミリ秒単位で指定する事が出来ます。指定する時は、[ツール]メニューの[オプション]をクリックし、[グラフ設定1]タブをクリックします。表示されている[2Dグラフ回転インターバル]のスライダーを変化させます。

グラフを回転させるときはマシンの性能により操作が重く感じる場合があります。その時はインターバルを大きく設定する等の対応が必要です。




また、コマンドによるグラフの回転ができ、文法は次のようになります。

```
( . . ) rotate 軸 角度
```

rotate コマンドの後、x,y,z のいずれか、それぞれ回転させたい軸を指定して、回転させる角度を指定します。例えば、x軸方向にグラフを $\pi/2$ ラジアン(90度)だけ回転させる時は次のように入力します。

```
( . . ) rotate x  $\pi/2$ 
```

### 3.3.3 グラフの表示領域を変更する

表示領域を変更することも出来ます。表示領域を変更するとは、グラフが見える範囲を変えるということです。実際に試してみてください。グラフウィンドウのツールバーにある  ボタンをクリックすると、表示領域が広くなり、 ボタンをクリックすると表示領域が狭くなります。数値で指定することも出来ます。 ボタンを押すと図 3-26 のような表示領域ダイアログが表示され、基準にする左下 x 座標、y 座標、基準からの視野の幅、高さをそれぞれ指定出来ます。尚、表示領域は 2 D モードでのみ変更できます。3 D モードでは変更することはできません。

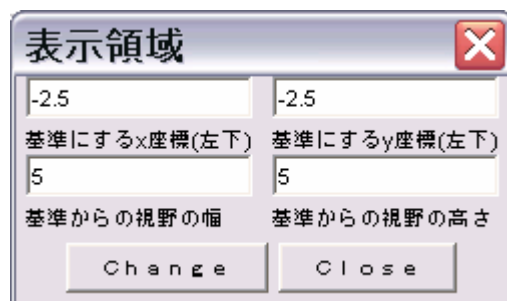


図 3-26 表示領域ダイアログ

また、コマンドによる表示領域の変更ができ、文法は次のようになります。

```
( . . ) view x,y,幅,高さ
```

view コマンドの後に基準にする左下座標(x,y)と基準からの視野の幅、高さを入力します。例えば、左下座標(-2.5,-2.5)から幅 5,高さ 5 の表示領域を指定する時は次のように入力します。

```
( . . ) view -2.5,-2.5,5,5
```

幅、高さは省略可能です。省略した場合は、現在の幅と高さのままです。

### 3.3.4 視野の移動



視野を移動する場合はグラフウィンドウのツールバーにある  ボタンをクリックします。図 3-27 のようなダイアログが表示されます。



図 3-27 視野の移動ダイアログ

移動幅を指定して矢印をクリックすればその方向に移動幅分移動します。例えば移動幅に 2 を入力して ボタンをクリックすれば視点が x 軸方向に 2 移動した事になります。

指定した座標へ視点を移動する事も出来ます。移動する場合は、グラフウィンドウのツールバーにある  ボタンをクリックします。デフォルトでは(0,0)を中心とする座標へ視点が移動します。この座標を変更することも出来ます。この場合[ツール]メニューの[オプション]をクリックし、[グラフ設定 2]タブをクリックします。表示されている[ホーム実行時移動する中心座標]で 2 D モード時、3 D モード時、それぞれの中心 x、y 座標を指定します。





### 3.3.5 グラフの生成範囲を変える

小さな生成範囲だと高速に表示される、大きな生成範囲だと広範囲のグラフを見る事が出来ます。このように必要に応じてグラフの生成範囲を変える事が出来ます。これらを変える場合はメインウィンドウにある[ツール]メニューの[オプション]をクリックし、[グラフ設定1]タブをクリックします。表示されている[3Dグラフ生成範囲]のスライダーを変化させて下さい。

### 3.3.6 滑らかな曲面、曲線のグラフを表示する

2Dモードではプロットの刻み幅を、3Dモードではグラフ生成時の面の荒さを変える事で、滑らかな曲線や曲面を描くグラフを表示させたり、多少荒くなっても時間をかけず高速にグラフを表示させたりする事が出来ます。それらを変える場合は、メインウィンドウにある[ツール]メニューの[オプション]をクリックし、[グラフ設定1]タブをクリックします。表示されている[2Dグラフプロットの刻み幅]、[3Dグラフ生成時の面の荒さ]のスライダーを変化させて下さい。

### 3.3.7 一度描いたグラフを消す

一度かかれたグラフを消す時は、グラフウィンドウのツールバーにある ボタンをクリックします。また、回転させている角度も同時に戻すときは ボタンを押します。

### 3.3.8 グラフの色を変更する

グラフを表示させる時の色を変える事が出来ます。色を変えられるのは以下の三つです。

1. グラフの線の色・・・2Dモード時の色
2. グラフの面の色・・・3Dモード時の色
3. グラフのプロットの色

グラフの線、面の色を変える文法は次のようになります。

```
(. .) graph color 色
```

graph color コマンドの後に色を入力すれば2D・3Dモード時、それぞれの時に線、面の色が指定した色に変わります。

グラフのプロットの色を変える文法は次のようになります。

```
(. .) plot color 色
```

plot color コマンドの後に色を入力すればプロットの色が変わります。

色は全部で140種類あり、代表的な色としては以下のような色があります。

aqua	・・・	水色	black	・・・	黒色	blue	・・・	青色
brown	・・・	茶色	gold	・・・	金色	gray	・・・	灰色
green	・・・	緑色	indigo	・・・	藍色	orange	・・・	橙色
pink	・・・	桃色	purple	・・・	紫色	red	・・・	赤色
silver	・・・	銀色	white	・・・	白色	yellow	・・・	黄色

また、罫の助を実行して入力支援機能を使用してみてください。他にもたくさん色を用意しています。



### 3.3.9 論理単位を変更する

グリッドの1升を / 2 単位にするか1 論理単位にするか指定する事が出来ます。論理単位を変える文法は次のようになります。


```
( . . ) measuremode true/false
```

measuremode コマンドの後が true ならば / 2 単位で false ならば1 論理単位になります。論理単位の起動時の状態を指定する事が出来ます。指定する場合はメインウィンドウにある[ツール]メニューの[オプション]をクリックし、[グラフ設定 2]タブをクリックします。表示されている[グリッドの幅設定、 / 2 単位・1 論理単位]のチェックを / 2 単位にするなら付けて、1 論理単位にするなら外して下さい。

### 3.4.0 グラフをファイルに出力する

グラフを PNG,SVG,JPEG 形式でファイル出力する事が出来ます。出力する時はグラフウィンドウにある[グラフ]メニューの[PNG 形式で出力]、[SVG 形式で出力]、[JPEG 形式で出力]をそれぞれクリックしてください。保存ダイアログが表示されるので、ファイル名を入力して保存します。

### 3.4.1 グラフウィンドウの表示・非表示切り替え

グラフウィンドウの表示・非表示にすることが出来ます。その場合、メインウィンドウのツールバーにある  ボタンをクリックする事でグラフウィンドウを表示・非表示に切り替えます。コマンド入力でのグラフ表示は次のようになります

```
( . . ) show
```

また、グラフ非表示は次のようになります。

```
( . . ) hide
```

### 3.4.2 魔法陣を表示する

n 行 n 列の升目に、1 から  $n \times n$  までの自然数を配置し、各行・各列・各対角線の和が、どれも等しくなるものが魔方陣です。(但し、n は自然数) 鱒の助では、3 ~ 200 までの魔方陣が作れます。例えば、 $6 \times 6$  の魔方陣を表示するには次のように入力します。

```
( . . ) magic 6
```

magic コマンドの後に、値を入力してください。

表 3-6  $6 \times 6$  の魔方陣例

4	3	35	36	28	5
6	26	12	13	23	31
30	15	21	20	18	7
29	19	17	16	22	8
10	14	24	25	11	27
32	34	2	1	9	33

3.4.3 パスカルの三角形を表示する

$(a + b)^n$  ( $n=0,1,2,\dots$ )の展開式で各項の係数だけを取り出して順に並べます。こうして得られる三角形状の数の配列をパスカルの三角形といいます。パスカルの三角形には、自然数、三角数、四面体数などのいろいろな数列が現れます。0～33 までのパスカルの三角形が作れます。例えば、6 のパスカルの三角形を表示するには次のように入力します。

```
(. .) pascal 6
```

pascal コマンドの後に、値を入れてください。

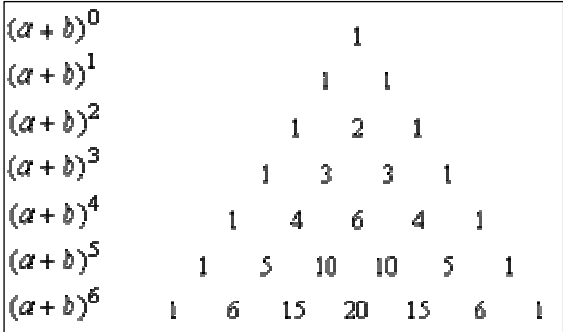


図 3-28 パスカルの三角形例

3.4.4 カレンダーを表示する

鱒の助では、カレンダー表示も可能です。しかし、グレゴリオ暦の使われ始めた 1752 年 9 月以前もずっとグレゴリオ暦が使われていたという仮定のカレンダー表示になりますので実際とは異なります。

カレンダーを表示する文法は次のようになります。

```
(. .) cal 月 年
```

cal コマンド後に、英語での月を頭文字 3 文字取り、その後に数字で西暦年数を入力してください。月の頭文字 3 文字が分からない時は、これを参考にして下さい。

Jan	・・・ 1 月	Feb	・・・ 2 月	Mar	・・・ 3 月
Apr	・・・ 4 月	May	・・・ 5 月	Jun	・・・ 6 月
Jul	・・・ 7 月	Aug	・・・ 8 月	Sep	・・・ 9 月
Oct	・・・ 10 月	Nov	・・・ 11 月	Dec	・・・ 12 月

例えば、2004 年の 2 月のカレンダーを表示する場合は次のように入力します。

```
(. .) cal Feb 2004
```

この結果は図 3-29 のように表示されます。



図 3-29 カレンダー

## 4 組込関数

### 4.1 一般数学関数

#### (1) 平方根と立方根

x の平方根を求める時の入力例を以下に示します。

例: `sqrt(x)`

x の立方根を求める時の入力例を以下に示します。

例: `cbrt(x)`

#### (2) 指数と対数

x の自然対数を求める時、`ln` を用います。この入力例を以下に示します。

例: `ln(x)`

また、指数(Exponential)は `exp` で求める事ができます。 $e^x$  を求める時の入力例を以下に示します。

例: `exp(x)`

`log` は常用対数を求める時に用います。常用対数とは、10 を底とした対数のことです。 $\log_{10}x$  を求める時の入力例を以下に示します。

例: `log(x)`

また、 $\log_yx$  のような式の結果を求める時の入力例を以下に示します。

例: `log(x,y)`

`log` の後の最初の引数は数値、次の引数は底を表します。

#### (3) 絶対値

x の絶対値を求める時の入力例を以下に示します。

例: `abs(x)`

#### (4) 符号

x の符号を求める時の入力例を以下に示します。

例: `sign(x)`

x の値が正なら 1、負なら -1、ゼロなら 0 が表示されます。

#### (5) 乱数

一様乱数を表示する時の入力例を以下に示します。

例: `rand()`

0 ~ x までの整数値の乱数を求める時の入力例を以下に示します。

例: `rand(x)`

x ~ y までの整数値の乱数を求める時の入力例を以下に示します。

例: `rand(x,y)`

#### (6) 階乗

x の階乗を求める時の入力例を以下に示します。

例: `fact(x)`

この結果は  $x!$  と入力した時と同じです。

x の素数階乗を求める時の入力例を以下に示します。

例: `primorial(x)`

この結果は  $x\#$  と入力した時と同様です。

#### (7) 累乗

x の y 乗の値を求める時の入力例を以下に示します。

例: `pow(x,y)`

( 8 ) 剰余

$x$  を  $y$  で割った余りを求める時の入力例を以下に示します。

例 :  $\text{mod}(x, y)$

( 9 ) 順列

$n$  個から  $r$  個を取り出して並べた順列の総数  ${}_nP_r$  を求める時の入力例を以下に示します。

例 :  $P(n, r)$

( 10 ) 組合せ

$n$  個から  $r$  個を取り出した組合せの総数  ${}_nC_r$  を求める時の入力例を以下に示します。

例 :  $C(n, r)$

( 11 ) 重複組合せ

$n$  個から  $r$  個を取り出した重複組合せ  ${}_nH_r$  の総数を求める時の入力例を以下に示します。

例 :  $H(n, r)$

( 12 ) 最大公約数

$x, y, \dots$  の最大公約数を求める時の入力例を以下に示します。

例 :  $\text{GCM}(x, y, \dots)$

( 13 ) 最小公倍数

$x, y, \dots$  の最小公倍数を求める時の入力例を以下に示します。

例 :  $\text{LCM}(x, y, \dots)$

( 14 ) 素数

$2 \sim x$  までの素数と素数の合計を求める時の入力例を以下に示します。

例 :  $\text{prime}(x)$

( 15 ) 素因数分解

$2 \sim x$  までの素因数分解した数と素因数分解した数の合計を求める時の入力例を以下に示します。

例 :  $\text{factor}(x)$

( 16 ) 四捨五入

$x$  の整数値による四捨五入をした値を求める時の入力例を以下に示します。

例 :  $\text{round}(x)$

( 17 ) 切り上げ

$x$  の整数値による切り上げをした値を求める時の入力例を以下に示します。

例 :  $\text{ceil}(x)$

( 18 ) 切り捨て

$x$  の整数値による切り捨てをした値を求める時の入力例を以下に示します。

例 :  $\text{floor}(x)$

## 4.2 三角関数

### (1) 正弦・余弦・正接( $\sin, \cos, \tan$ )

x の正弦関数(サイン)・余弦関数(コサイン)・正接関数(タンジェント)を求める時のそれぞれの入力例を以下に示します。

例： $\sin(x), \cos(x), \tan(x)$

### (2) 割三角関数( $\operatorname{cosec}, \sec, \cot$ )

x の正割関数・余割関数・余接関数を求める時のそれぞれの入力例を以下に示します。

例： $\sec(x), \operatorname{cosec}(x), \cot(x)$

### (3) 逆三角関数( $\arcsin, \arccos, \arctan \dots \text{etc}$ )

x の逆正弦関数・逆余弦関数・逆正接関数また、逆正割関数・逆余割関数・逆余接関数を求める時のそれぞれの入力例を以下に示します。

例： $\arcsin(x), \arccos(x), \arctan(x),$   
 $\operatorname{arcsec}(x), \operatorname{arccosec}(x), \operatorname{arccot}(x)$

また、直交座標(x,y)に対応する極座標(r, theta)の theta の成分を求める時の入力例を以下に示します。

例： $\arctan2(x, y)$

### (4) 双曲線関数( $\sinh, \cosh, \tanh \dots \text{etc}$ )

x の双曲線正弦関数・双曲線余弦関数・双曲線正接関数を求める時のそれぞれの入力例を以下に示します。

例： $\sinh(x), \cosh(x), \tanh(x)$

### (5) 割双曲線関数( $\operatorname{sech}, \operatorname{cosech}, \coth$ )

x の双曲線正割関数・双曲線余割関数・双曲線正接関数を求める時のそれぞれの入力例を以下に示します。

例： $\operatorname{sech}(x), \operatorname{cosech}(x), \coth(x)$

### (6) 逆双曲線関数( $\operatorname{arcsinh}, \operatorname{arccosh}, \operatorname{arctanh} \dots \text{etc}$ )

x の逆双曲線正弦関数・逆双曲線余弦関数・逆双曲線正接関数・逆双曲線正割関数・逆双曲線余割関数・逆双曲線余接関数を求める時のそれぞれの入力例を以下に示します。

例： $\operatorname{arcsinh}(x), \operatorname{arccosh}(x), \operatorname{arctanh}(x),$   
 $\operatorname{arcsech}(x), \operatorname{arccosech}(x), \operatorname{arcoth}(x)$

### (7) 頂点三角関数( $\operatorname{versin}, \operatorname{vercos}, \operatorname{vorsec}, \operatorname{vercosec}$ )

x の頂点正弦関数・頂点余弦関数・頂点正割関数・頂点余割関数を求める時のそれぞれの入力例を以下に示します。

例： $\operatorname{versin}(x), \operatorname{vercos}(x), \operatorname{vorsec}(x), \operatorname{vercosec}(x)$

### (8) 半頂点三角関数( $\operatorname{havesin}, \operatorname{havcos}, \operatorname{havsec} \dots \text{etc}$ )

x の半頂点正弦関数・半頂点余弦関数・半頂点正割関数・半頂点余割関数を求める時のそれぞれの入力例を以下に示します。

例： $\operatorname{havesin}(x), \operatorname{havcos}(x), \operatorname{havsec}(x), \operatorname{havcosec}(x)$

### 4.3 統計関数

#### (1) 要素の和、要素の積

$x, y, \dots$  と与えられた値の要素の数や要素の和や要素の積を求める時のそれぞれの入力例を以下に示します。

例: `count(x,y,...), sum(x,y,...), product(x,y,...)`

#### (2) 代表値

$x, y, \dots$  と与えられた値の平均値、中央値、最頻値を求める時のそれぞれの入力例を以下に示します。

例: `mean(x,y,...), median(x,y,...), mode(x,y,...)`

平均値の入力例は `avarage(x,y,...)` としても同じ結果が得られます。

また、中央値・中位値とは、要素の中間の値を返します。要素が奇数の場合は、真ん中の値を返します。要素が偶数の場合は、真ん中の前後の値を二分した値を返します。最頻値とは、最も頻繁に出現する値を返します。要素に同じ数値が2つ以上存在しない時は、エラーになります。

#### (3) 散布度

$x, y, \dots$  と与えられた値の範囲、最大最小値の平均値、平均偏差、分散、標準偏差を求める時のそれぞれの入力例を以下に示します。

例: `range(x,y,...), mid_range(x,y,...), medev(x,y,...),  
variance(x,y,...), stdev(x,y,...)`

範囲とは、要素の最大値と最小値との差です。平均偏差とは、偏差の絶対値をとって平均をしたものです。分散とは、偏差の二乗をとって平均したものです。標準偏差とは、偏差の二乗をとって平均したものの正の平方根です。

### 4.4 その他の関数

#### (1) 総和、直積

関数  $f$  に対して、総和や直積を求めます。 $x$  から  $y$  まで総和は加算し、直積では乗算を行います。求める時のそれぞれの入力例を以下に示します。

例: `sigma(x,y,f), pi(x,y,f)`

#### (2) 数値解を求める

関数  $f$  の数値解を  $x$  から  $y$  までの間で求めます。割り出せない場合はエラーになります。求める時の入力例を以下に示します。

例: `solve(x,y,f)`

#### (3) 等差数列、等比数列

等差数列の和と等比数列の和を求める時の入力例を以下に示します。

例: `AP(x,y,z), AP2(x,y,z), GP(x,y,z), GP2(x,y,z)`

#### (4) 微分

関数  $f$  を  $x$  について微分します。求める時の入力例を以下に示します。

例: `differ(f(x))`

#### (5) 積分

関数  $f$  を  $x$  から  $y$  まで積分します。求める時の入力例を以下に示します。

例: `integral(x,y,f)`

( 6 ) 微分方程式

関数  $f$  を用いて初期値  $(x_0, y_0)$  の  $x$  の微分方程式を解きます。求める時の入力例を以下に示します。

例 : `diffsolve(x0,y0,f,x)`

( 7 ) 連立微分方程式

関数  $f, g$  を用いて初期値  $(x_0, y_0, z_0)$  の  $x$  の連立微分方程式を解きます。求める時の入力例を以下に示します。

例 : `simudiffsolve(x0,y0,z0,f,g,x)`

( 8 ) 二階微分方程式

関数  $f, g$  を用いて初期値  $(x_0, y_0, z_0)$  の  $x$  の二階微分方程式を解きます。求める時の入力例を以下に示します。

例 : `diffsolve2(x0,y0,z0,f,g,x)`

## 5 感想・バグ報告等連絡先

感想やバグ報告等は作者までメールでお寄せ下さい。

メールアドレス：aikawa@hokuriku-pc.ac.jp

速やかにバグ対応できないかも知れませんが、ご了承下さい。

## 6 困ったときは

( 1 ) どんな関数があるのか分からない

組込関数を参照してください。また、入力支援機能を使ってみてください。定数や関数、コマンド、とその説明が表示されます。

( 2 ) 全角文字の使用

半角文字にて入力してください。ただしコメントとして入力する場合はどのような文字でも構いません。

範囲選択をして全角入力を行った場合、テキストエリアの表示がおかしくなってしまう場合があります。

---

鱒の助取扱説明書

2004 年 7 月 4 日 初版発行

著者：新浜 裕幸、田中 信博、相川 政和

---