

# Gnuplot OX server マニュアル

---

Edition : auto generated by oxgentexi on 20 July 2010

OpenXM.org

---

# 1 GNUPLOT 函数

この節では GNUPLOT の ox サーバ ox\_sm1\_gnuplot とのインタフェース関数を解説する。これらの関数はファイル ‘gnuplot.rr’ で定義されている。gnuplot.rr は ‘\$(OpenXM\_HOME)/lib/asir-contrib/’ にある。

```
[255] gnuplot.start();
0
[257] gnuplot.gnuplot("plot sin(x**2);");
0
```

関数 gnuplot.heat(dt,step) はわれわれの GNUPLOT インタフェース関数のデモである。この関数は熱伝導方程式

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad u(t, 0) = u(t, 1) = 1$$

を初期条件

$$u(0, x) = x, \quad (0 \leq x \leq 0.5), \quad u(1, x) = 1 - x, \quad (0.5 \leq x \leq 1)$$

で陽的差分法を用いて時間が  $0 \leq t \leq dt * step$  の範囲で解く。区間  $[0, 1]$  は Heat\_N 個に分割される。static 変数 Heat\_N は関数 gnuplot.set\_heat\_N で設定する。有名な Courant-Friedrichs-Levi 数  $dt * Heat\_N * Heat\_N$  が 0.5 以下であれば、陽的差分スキームは安定である。CFL を変えることにより、不安定性が生じるのを観察できる。

```
gnuplot.set_heat_N(20); gnuplot.heat(0.001,30);    (CFL number is 0.4)
gnuplot.set_heat_N(20); gnuplot.heat(0.003,30);    (CFL > 0.5 unstable)
```

Author of GNUPLOT: Thomas Williams, Colin Kelley. <http://www.gnuplot.info>

## 1.1 函数一覧

### 1.1.1 gnuplot.start

gnuplot.start()

:: Localhost で ox\_sm1\_gnuplot を起動する。

return 整数

- Localhost で ox\_sm1\_gnuplot を起動する。起動された ox\_sm1\_gnuplot の識別番号を戻す。
- Xm\_noX =1 としておくと, ox\_sm1\_gnuplot 用の debug window が開かない。
- 識別番号は, Gnuplot\_proc に格納される。

```
P = gnuplot.start();
```

参照 ox\_launch, gnuplot

### 1.1.2 gnuplot

gnuplot.gnuplot(s | proc=p)

:: GNUPLOT にコマンド s を実行してもらう。

return なし

p 数

*s*            文字列

- サーバは GNUPLOT のコマンド *s* を実行する. エラーがおきた場合 GNUPLOT 本体は終了してしまうが, `ox_sm1_gnuplot` は自動的に GNUPLOT 本体をリスタートする.
- GNUPLOT は長い多項式をただしく受けつけない.
- GNUPLOT は  $\wedge$  を受けつけない. かわりに, `**` を使う.

```
[232] P = gnuplot.start();
0
*Plot 3 dimensional graph.
[233] gnuplot.gnuplot("splot x**2-y**2;"|proc=P);
0
*Plot 2 dimensional graph.
[234] gnuplot.gnuplot("plot [-pi:pi] [-2:2] cos(x);");
0
*Output a graph as a postscript figure.
[235] gnuplot.output(|file="hoge.eps");
0
[236] gnuplot.gnuplot("plot sin(x)*cos(x);");
0
[237] gnuplot.gnuplot(|file="x11");
0

*Plot 3 dimensional graph hiding unvisible lines.
[236] gnuplot.gnuplot("set hidden3d");
0
[237] gnuplot.gnuplot("splot (x**2+y**2)*sin(x**2+y**2)");
0
[238] gnuplot.gnuplot("set isosamples 50");
0
[239] gnuplot.gnuplot("splot (x**2+y**2)*sin(x**2+y**2)");
```

参照            `ox_launch`, `gnuplot.start`, `rtostr`, `gnuplot.plot_dots`

参考書          矢吹道郎, 大竹つよし; 使いこなす GNUPLOT, テクノプレス, ISBN4-924998-11-7

### 1.1.3 gnuplot.plot\_dots

`gnuplot.plot_dots(d,s|proc=p)`  
                   :: 点の集合 *d* をスタイル *s* でプロットする.

*return*          なし

*p*                数

*d*                リスト

*s*                文字列 または 0

- 点集合 *d* をスタイル *s* でプロットする. *s* は次のような文字列: "style color point". ここで style には lines, points, linespoints, impulses, dots, steps, errorbars, boxes, boxerrorbars を選べる. color には 1 (red), 2 (green), 3 (blue), 4, ... , 8 を選べる. point は 1 から 8 の数を入れる. color, point は省略してよい.

- $d == [ ]$  のときはスクリーンがまず消去される.

```
[239] P = gnuplot.start();
0
[240] gnuplot.plot_dots([ ],0);
0
[241] for (I=0; I<10; I++) gnuplot.plot_dots([[I,I^2]], " lines ");
[242] A = [ ];
[]
[243] for (I=0; I<10; I++) A = append(A, [ [I,I^2]]);
[244] A;
[[0,0],[1,1],[2,4],[3,9],[4,16],[5,25],[6,36],[7,49],[8,64],[9,81]]
[245] gnuplot.plot_dots(A, " lines ");
0
```

参照 `gnuplot.start, plot "fileName" with options(GNUPLOT command),  
gnuplot.clean, gnuplot`

### 1.1.4 gnuplot.heat

`gnuplot.heat(dt,step)`  
:: 熱伝導方程式を数値的に解く.

*return* なし

*dt* 浮動小数点数

*step* 整数

- 熱伝導方程式  $du/dt = d^2 u/dx^2$ ,  $u(t,0) = u(t,1) = 0$  を初期条件  $u(0,x) = x$  ( $0 \leq x \leq 0.5$ ),  $u(0,x) = 1-x$  ( $0.5 \leq x \leq 1.0$ ) で解く.
- Heat\_N は空間方向でのメッシュの数.
- この関数は将来 `pde_heat_demo` と呼ばれる予定.

```
[232] Heat_N = 20$
[233] gnuplot.heat(0.001,30)$
```

### 1.1.5 gnuplot.output

`gnuplot.output(|file=s)`  
:: GNUPLOT にファイル *s* へポストスクリプトで出力するように頼む.

*return* Void

*s* String

- GNUPLOT にファイル *s* へポストスクリプトで出力するように頼む.
- *s* が "x11" または、この関数を引数無しでよぶと、以後、X11 の display に graphics が出力される.

```
[273] gnuplot.output(|file="hoge.eps");
Graphic output of GNUPLOT will be written to hoge.eps as a Poscript file.
0
[274] gnuplot.gnuplot("plot tan(x)+sin(x);");
0
```

```
[275] gnuplot.output();
Usage of gnuplot.output: gnuplot.output(|file="string")
                        gnuplot.output(|file="x11")
Output device is set to X11
```

参照 `gnuplot`

### 1.1.6 `gnuplot.plot_function`

```
gnuplot.plot_function(f|proc=p)
:: gnuplot サーバに  $f$  のグラフを書くように頼む.
```

戻り値 なし

$p$  数

$f$  多項式または多項式のリスト

- `gnuplot` サーバに  $f$  のグラフを書くように頼む.

```
[290] gnuplot.plot_function((x+sin(x))^2);
0
[291] gnuplot.plot_function([x,x^2,x^3]);
0
```

参照 `gnuplot.to_gnuplot_format`

### 1.1.7 `gnuplot.stop`

```
gnuplot.stop()
:: GNUPLOT を停止し, 通信用の fifo ファイルを消す.
```

return Void

$s$  String

- GNUPLOT を停止し, 一時ディレクトリの下に作成された通信用の fifo ファイルを消す.
- 通信用の fifo ファイル名は `gnuplot` で始まる.

```
[273] gnuplot.stop()
```

参照 `gnuplot.start`

### 1.1.8 `gnuplot.setenv`

```
gnuplot.setenv(key,value)
::
```

return Void

key String

value Object

- `key` は `"gnuplot.callingMethod"` または `"plot.gnuplotexec"`.

Use the old method to communicate with gnuplot (version 3).

This method does not use `mkfifo`, but we need a patched version of `gnuplot`.

```
[273] gnuplot.setenv("gnuplot.callingMethod",0);
```

```
[274] gnuplot.setenv("plot.gnuplotexec",getenv("OpenXM_HOME")+"/bin/gnuplot4ox");
```

Calling your own gnuplot binary.

```
[274] gnuplot.setenv("plot.gnuplotexec","/cygdrive/c/program files/gnuplot/pgnuplot
```

参照          gnuplot.start

# Index

(Index is nonexistent)

(Index is nonexistent)

## Short Contents

1	GNUPLOT 函数 .....	1
	Index .....	6



# Table of Contents

<b>1</b>	<b>GNUPLOT 函数</b>	<b>1</b>
1.1	函数一覧	1
1.1.1	gnuplot.start	1
1.1.2	gnuplot	1
1.1.3	gnuplot.plot_dots	2
1.1.4	gnuplot.heat	3
1.1.5	gnuplot.output	3
1.1.6	gnuplot.plot_function	4
1.1.7	gnuplot.stop	4
1.1.8	gnuplot.setenv	4
<b>Index</b>		<b>6</b>