

グラフの描き方 (1)

桂田 祐史

1995 年 4 月 28 日

この情報処理 II では OpenWindows 環境下で¹グラフを描くための手段をいくつか提供します。今回紹介する方法は、複雑なことをするのには向いていませんが、使うための準備が簡単なので、覚えておくとなかなか便利です。

すでに環境設定を済ませている人は、

```
waltz11% getsample
```

としてファイルをコピーしてから (ls で確認してみよう) 以下の実習を行ないましょう。

1 コンピューターで描く曲線

コンピューターで滑らか曲線のグラフを描くにはどうしたらよいでしょう？実は原理はとても単純で

「十分細かい」折れ線を描けば人間の目には滑らかな曲線に見える

(だから折れ線を描けばいいんだ) というものです。折れ線で済ますのは、何かズルイような気がするかもしれませんが、もともとラスタ・ディスプレイでは飛び飛びのドットでしか表示できないわけですから、それで十分なわけです。

しかし「十分細かい (= 折れ線の頂点と頂点の間隔が十分小さい)」という点には注意しなくてはなりません。これが成立していないと、本来滑らかなはずの曲線がギクシャクしたものになってしまったり²、一見滑らかなようでも実は本当の姿とはかけはなれたものになってしまったりする可能性もあります。

2 mgraph コマンド

折れ線を表現するのに、各頂点の座標データを並べるというのは、自然な考え方です。ここで紹介する mgraph は、一言で説明すると、座標で指定された点を順に結んだ折れ線を描くためのコマンドです³。

¹実は OpenWindows の兄貴分である X Window System でも利用可能です。

²特にプリンターで印刷する場合は、プリンターの解像度がディスプレイの解像度よりも高いのが普通なため、ディスプレイ上に描画するよりも細かくしないとアラが目立ちます。

³mgraph は大抵の UNIX システムに用意されている graph コマンドの機能拡張版です (graph コマンドについては man コマンドで説明が読めます — “man graph” としてみましよう)。今回の例の多くは graph コマンドでも実行可能です。

入力データは、 x 座標と y 座標が並んでいるような行からなるテキストです。例えば

```
0.0 0.0
2.0 1.0
1.0 3.0
0.0 0.0
```

のような内容のテキストは、3 点 $(0, 0)$, $(2, 0)$, $(1, 3)$ を頂点とする三角形を描くためのデータということになります。

実際に画面に図を描くには、上のデータを収めたファイルの名前を “sankaku.data” とした時

```
waltz11% cat sankaku.data | mgraph | xplot
```

とします (図のウィンドウを消すには、マウス・カーソルをそのウィンドウまで移動して左ボタンをクリックします)。また同じ図をプリンターで印刷するには

```
waltz11% cat sankaku.data | mgraph | plot2ps | lpr
```

とします⁴ (紙がもったいないですから、どうしても必要な図だけ印刷するようにして下さい)。

解説 — 上の呪文のようなコマンドを理解するために —

前回のプリントで説明したパイプ機能を利用しています。mgraph は標準入力からデータを読み込み、標準出力にグラフィックス言語の一つである plot 形式のデータを書き出します。xplot は標準入力から plot 形式のデータを読み込み、画面に図を描くためのプログラムです。plot2ps は標準入力から plot 形式のデータを読み込み、標準出力にプリンター制御言語である PostScript 形式のデータを書き出します。

前回の最後で sin の関数表 “sintable” を作りなさいという問題を出しましたが、それがあれば次のようにして sin のグラフが描けます。

```
waltz11% cat sintable | mgraph | xplot
```

sin の表を作るプログラムとしては、例えば次のような FORTRAN プログラムが使えます (これで “sintable” を作るには “f77o mktable.f” そして “mktable > sintable” とします)。

```
* mktable.f -- sin の関数表を作る
  program mktable
    integer i
    real x,y,pi,degrad
*   円周率 の値 (= 4 * Arctan(1))
    pi = 4.0 * atan(1.0)
*   度 (degree) をラジアン (radian) に変換するための比例定数の計算
*   1 °= /180 radian、である。
    degrad = pi / 180.0
```

⁴lpr コマンドでは、“-Pプリンター名”として出力するプリンターが選べます。0609号室では“w1”, ..., “w4”という名前のプリンターがあります。例えば“w1”で印刷するには“lpr -Pw1”とします。

```

*      0 ~ 90 °までの値を計算
do i=0,90
  x = i * degrad
  y = sin(x)
  write(*,*) i,y
end do
end

```

勘のいい人はここまでの知識で大抵のグラフが描けてしまうでしょう。

3 関数のグラフを描く

実変数の実数値関数 $y = f(x)$, ($a \leq x \leq b$) のグラフを描くには、定義区間 $[a, b]$ を適当に

$$a = x_0 < x_1 < x_2 < \cdots < x_N = b,$$

と分割して、各 x_j に対する f の値 $y_j = f(x_j)$ を計算して、

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$$

を順に結んだ折れ線を書けばいいでしょう。特に複雑な関数でない限り、 x_j は区間 $[a, b]$ の N 等分点に取るのが自然でしょう。

例題 2-1 $y = \sin x + \sin 3x + \sin 5x$ のグラフを描きなさい。

解答: 問題の関数は周期 2π ですから、 $0 \leq x \leq 2\pi$ で描けば良いでしょう。次の “reidai2-1.f” というプログラムは、前節で解説したような形式のデータを計算するためのプログラムです。実行すると最初に区間を何等分するか尋ねてくるようになっています。

```

* reidai2-1.f
  program rei21
  real x, y, h, Pi, a, b, f
  integer N, j
  parameter (Pi = 3.14159265)
  parameter (a = 0.0, b = 2.0 * Pi)
  read(*,*) N
  h = (b - a) / N
  do 100 j=0,N
    x = a + j * h
    y = f(x)
    write(*,*) x, y
100 continue
  end
*
  real function f(x)
  real x
  f = sin(x) + sin(3.0 * x) + sin(5.0 * x)
  end

```

この“reidai2-1”を用いてグラフを描くには、座標データをファイル“ftable”に記録してから、それを使うやり方

```
waltz11% reidai2-1 > ftable
100                                分割数として 100 を入力
waltz11% cat ftable | mgraph | xplot
```

でも良いし、あるいは全部パイプを使って次のように済ますことも出来ます。

```
waltz11% reidai2-1 | mgraph | xplot
100                                分割数として 100 を入力
```

問題 2-1 “reidai2-1.f”を読んで理解し、書き換えることによって、色々な関数のグラフを描きなさい。面白いものが出来たら印刷してみなさい。

問題 2-2 $f(x) = \sin x + \sin \frac{x}{1000}$ のような関数のグラフを描くことについて考察しなさい。(グラフの描きにくい関数である。なぜか? 区間や分割数を色々変えて実験してみよう。)

4 パラメーター曲線を描く

例題 2-2 楕円 $\frac{x^2}{4} + \frac{y^2}{9} = 1$ を描け。

解答: 方程式を局所的に y (あるいは x) について解くことによって ($y = \pm 3\sqrt{1 - \frac{x^2}{4}}$ とか)、例題 2-1 と同様に扱うことも可能ですが、ここではパラメーター表示

$$x = 2 \cos \theta, \quad y = 3 \sin \theta \quad (\theta \in [0, 2\pi])$$

を利用してみましょう。パラメーター θ の区間 $[0, 2\pi]$ を

$$0 = \theta_0 < \theta_1 < \theta_2 < \dots < \theta_N = 2\pi,$$

と分割して、楕円上の点 (x_j, y_j) を

$$x_j = 2 \cos \theta_j, \quad y_j = 3 \sin \theta_j$$

で定めて、 $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N)$ を順に結んだ折れ線を描けばいいでしょう。この考え方に従って座標を計算するプログラムは例えば、次のように書けます。

```
* reidai2-2.f
program rei22
real x, y, theta, h, Pi, a, b
integer N, j
parameter (Pi = 3.14159265, a = 0.0, b = 2.0 * Pi)
read(*,*) N
h = (b - a) / N
do 100 j=0,N
```

```

        theta = a + j * h
        x = 2.0 * cos(theta)
        y = 3.0 * sin(theta)
        write(*,*) x, y
100  continue
    end

```

このプログラム “reidai2-2” を用いてグラフを描くには

```

waltz11% reidai2-2 | mgraph -R -x -3.2 3.2 -y -3.2 3.2 | xplot
100                                100 等分を指定

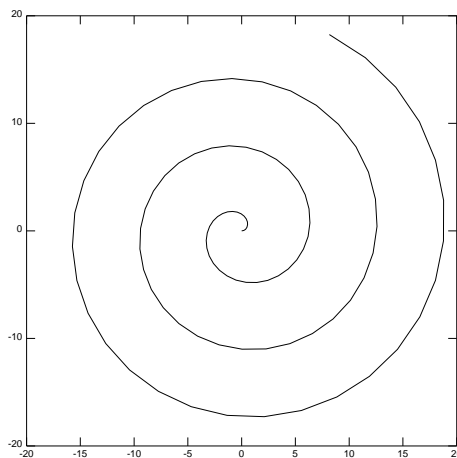
```

ここで、“-x -3.2 3.2 -y -3.2 3.2” としているのは、描画する領域を $-3.2 \leq x \leq 3.2$, $-3.2 \leq y \leq 3.2$ に指定しています。

問 . “-x -3.2 3.2 -y -3.2 3.2” を指定しないとどうなるか？

問題 2-3 “reidai2-2.f” を読んで理解し、書き換えることによって、色々な曲線を描きなさい。面白いものが出来たら印刷してみなさい。

問題 2-4 極座標で $r = \theta$ と表される図形（螺旋の一種）を描け。



5 その他

例題 2-3 “mapdata” というファイルには世界地図のデータが入っています。普通のテキスト・ファイルですから、ちょっとのぞいて見て下さい⁵。実はこのデータは陸地の輪郭線の頂点の座標（経度・緯度）を順に並べたものですが、線を切りたいところで、二重引用符で囲った空白を置いてあります：

```

133 35
134 35
135 35

```

⁵“cat mapdata” とします。less を知っていたら、“less mapdata” として見ることを勧めます。

```
" "  
134 33  
133 33  
132 33
```

こうして作ったデータを表示するには

```
cat mapdata | mgraph -b | xplot
```

とします。“-b”を指定することにより線を途中で切ることが出来ます。

問 . “-b”を指定しないとどうなるか試さない。

問題 2-5 地図には目的に応じてさまざまな「図法」があることは社会科あるいは地理の時間に習ったことがあるでしょう。どれか一つの図法を（具体的にどう描くのか）調べて、その図法による世界地図を描きなさい。（FORTRAN でこの問題を解くのは面倒かもしれません。）

問題 2-6 何か面白いデータを入手して mgraph コマンドを用いて図を描きなさい。

6 mgraph コマンドについての補足

既に紹介した“-b”，“-x 数1 数2”，“-y 数1 数2”の他に以下のオプションが使えます。

- g 数 枠や軸、目盛をコントロールとします (数=0,1,2,3,4)。例えば -g 0 とすると枠も軸も描きません。
- lx, -ly 対数グラフを描くためのオプションです。-lx とすると、水平方向の軸を対数目盛にします。-ly とすると、垂直方向の軸を対数目盛にします。もちろん、二つを同時に指定して、両方の軸を対数目盛にすることも出来ます。
- a x 座標に相当する部分を省略したデータを作ることはよくあります。例えば、例題 2-1 のように x 座標が一定の量だけ増えていくような場合、一々記録するほどのこともないと考えて省略することが多いです。このようなデータを表示するため、自動的に等間隔の x 座標を補って表示するオプション-a があります。
- L 文字列, -X 文字列, -Y 文字列 その直後に来る引数をラベル⁶として表示します。-X は水平方向の軸につけるラベルの指定、-Y は垂直方向の軸につけるラベルの指定、-L はグラフ上部につけるラベル（つまりメインのラベル）の指定。（例えば“-L "FIGURE 1"”のように指定します。）。
- M x 座標の非単調性を検知して、折れ線を切ります。（一枚の図に複数の関数のグラフを描く場合のように、一つのグラフを描くためのデータが終って、次のグラフ用のデータに移るところで、 x 座標が「戻る」場合に、そこで一度折れ線を切ります。）

⁶graph コマンドに与えるデータで、二重引用符 " で囲まれた文字列をラベルといいます。

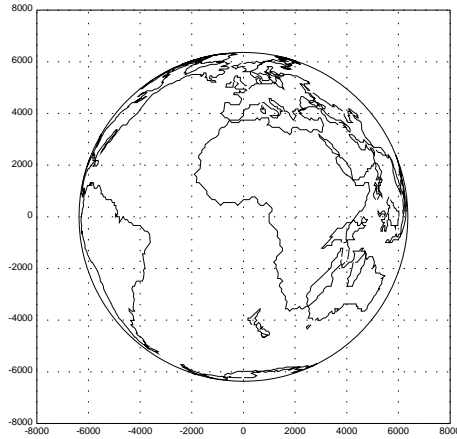


図 1:
透明な地球

問題 2-7 (getsample でコピーされたプログラム “reidai2-4.f” を使って)

```
f77o reidai2-4.f  
reidai2-4 | mgraph -M -lx -ly | xplot
```

として描けるグラフは一体何か? “reidai2-4.f” と上記の説明を読んで考えなさい。 “-lx -ly” を省略するとどうなるか?