

情報処理 II 第 1 回 グラフの描き方 (1) 修正版

桂田 祐史

1992 年 4 月 22 日

1 はじめに

この講義ではグラフを描くための手段を二つ提供します。今回紹介する方法は、あまり複雑な図形を描く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。

注意：二つの道具のいずれも、OpenWindows の下で使うことを念頭においています。以下の例題プログラムを試したりする場合は、必ず OpenWindows を起動して行なって下さい¹。

2 コンピューターで描く曲線

コンピューターで、数学の教科書に出て来るような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。折れ線を描いただけで済むのは、何かズルイような気がするかもしれませんが、ラスター・ディスプレイの原理を思い出せば、あまり心配する必要はないことが分かるでしょう。

¹実は OpenWindows の兄貴分である X Window System でも利用可能です。ただし SunView で使うことは出来ません。

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、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。ここらへんは真面目に考えると結構難しい問題になります。

3 mgraph コマンド

折れ線を指定するのに、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で折れ線を描くためのコマンドです²。

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。(毎回 “mgraph | xplot” と入力するのは面倒だろうということで、mgraph と xplot を連結したプログラム Xgraph を作りました³。これを利用すると上と同じことが

```
waltz11% cat sankaku.data | Xgraph
```

²mgraph は大抵の UNIX 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 で説明が読めます)。今回の例のほとんどは、graph コマンドでも実行可能です。

³実はセンターには xgraph という名前 (大文字・小文字が違うだけ!) の別のコマンドが存在するので、あまり良い命名とは言えないのですが。

としても出来ます⁴。)

ここでは、UNIX のパイプ機能を利用しています。これは “|” という記号で、コマンドを連結すると、“|” の左側にあるコマンドの標準出力を、“|” の右側にあるコマンドの標準入力に接続するというものです。

補足： 同じ図をプリンターに出力するには

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

とします。(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。)

4 関数のグラフを描く

実一変数の実数値関数 $y = f(x)$, ($a \leq x \leq b$) 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 1-1 $y = \sin x + \sin 3x + \sin 5x$ のグラフを描きなさい。

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。“reidai1-1.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データです。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。コンパイルして実行してみましょう。

```
waltz11% f77o reidai1-1.f
waltz11% reidai1-1
100                                分割数 N として 100 を入力
```

この後グラフ上の点の座標データがたくさん (101 行分) 表示されます。このデータを元にグラフを描くには

⁴入出力のリダイレクションを使えば、もっと短く `Xgraph < sankaku.data` とも出来ます。

```
waltz11% reidai1-1 | mgraph | xplot
100                               分割数として 100 を入力
```

とすれば OK です。

問 . 分割数 N を色々変えて (特に N を小さくして) 出来上がる図を比べて見なさい。

問題 1-1 “reidai1-1.f” を読んで理解し (20 行程度で簡単なものです) 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 1-2 $f(x) = \sin x + \sin 1000x$ 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?)

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

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、例題 1-1 と同様に扱うことも可能ですが、ここではパラメータ表示

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

を利用してみましょう。パラメータ θ の区間 $[0, 2\pi]$ を

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

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。そのために、座標を計算するプログラム reidai1-2.f を用意しました。

```
waltz11% f77o reidai1-2.f
waltz11% reidai1-2 | 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。

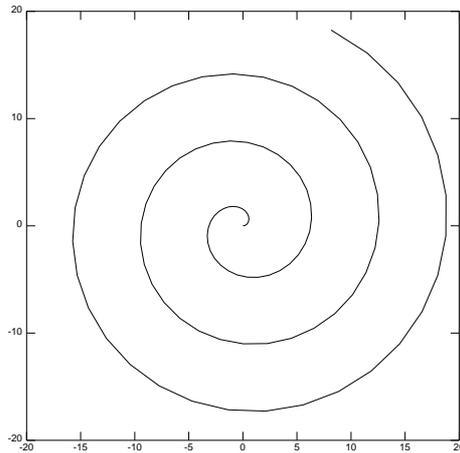
問 . “-x -3.2 3.2 -y -3.2 3.2” を指定しないとどうなるか?

問題 1-3 “reidai1-2.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 1-4 極座標で

$$r = \theta$$

と表される図形（螺旋の一種）を描け。



例題 1-3 “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
" "
134 33
133 33
132 33
```

こうして作ったデータを表示するには

```
waltz11% 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しなさい。

⁵cat 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。

問題 1-5 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が。）

問題 1-6 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。

5 補足

-g オプション `(m)graph` コマンドでは、枠や軸、目盛を描きますが、それをコントロールするには、`-g` 数 とします。例えば `-g 0` とすると枠も軸も描きません。

-a オプション 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 1-1 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。

-M オプション 例題 1-1,1-2 では一筆書きをしています。しかし、それだけでは複雑な図は書けません。例題 1-3 では、ダミーのラベル⁶を使ってこの問題を解決しました（`-b` はラベルが現われたら線を切る、ということを指定するオプションです）。もう一つ `-M` というオプションが有効な場合があります。

一枚の図に複数の関数のグラフを描く場合を考えましょう。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」ことに気がきます（それまで単調に増加していたのが、急に小さくなる、とか）。こういう x 座標の非単調性を察知して、折れ線を切るようにするオプションが `-M` です。

⁶`graph` 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見なします。