

第 2 回レポートの解説 (2)

桂田 祐史

1992 年 7 月 1 日

1 第 4 回の問題

1.1 問題 4-1 解説

この種の事は理工系の学科では常識になっていて欲しい、と思うのですが、、

reidai4-1b | mgraph -lx -ly | xplot で描いたグラフ (次ページの上段右側のグラフです) を見てみると、あるところから先は傾きが -1 の直線になっていることが読み取れます。両側対数のグラフを書いて傾きが m となるのは 2 番目の変数が 1 番目の変数の m 乗に比例することですから、この場合は誤差が分割数 N の -1 乗に比例していることになります。

確認: x と y が $y = ax^m$ (a, m は定数) という関係にあったとしましょう。両辺の対数¹を取ると $\log y = m \log x + \log a$ となります。よって、 $X = \log x$, $Y = \log y$, $A = \log a$ とおくと $Y = mX + A$ となり、 X - Y 図を描くと、傾き m の直線になることが分かります。(逆に X - Y 図でグラフが傾き m の直線となる時、 $y = ax^m$ となることも明らかです。)

問: 色々な m について、自分でデータを作って、グラフを描いてみて、上に述べたことを確かめなさい。

問: 二つの量 x, y が $y = a^x$ (a は正の定数) という関係にあることをチェックするにはどうしたらいいか?

¹容易に分かるように、自然対数でも (10 を底とする) 常用対数でも以下の議論は成立します。

普通にグラフを描いて出来る曲線を見て $y = c/x$ であるのか、それとも (例えば) $y = c/x^2$ であるかを区別するのは、普通の人間の能力を越えています。それに比べて直線の傾きを判定するのはずっと簡単なことは納得できるでしょう。後述の問題 4-5 では、Runge-Kutta 法による近似解の誤差が N の -4 乗に比例することを、この方法を用いてチェックします。

対数を使うことの利点を少し違った観点から考えてみます。科学で扱うデータの中には、値がかなり広い範囲に渡るものがしばしばあることに注意しましょう。例えば人間の成人の身長データなどは 1 桁 (=10 倍) 違うことはあり得ないですが、音の大きさや、地震のエネルギー等、日常出会う現象の中にも、値が桁違いであることが当たり前のものがたくさんあります。こういうものを「分かり易く」するにはデータの対数を取ることが有用です (デシベルとかマグニチュードなどの定義を見ると対数が登場します)。ここでの収束の問題では、0 に近づく量を扱うわけですが、そういうものを把握するにも対数は役立つ、ということです。

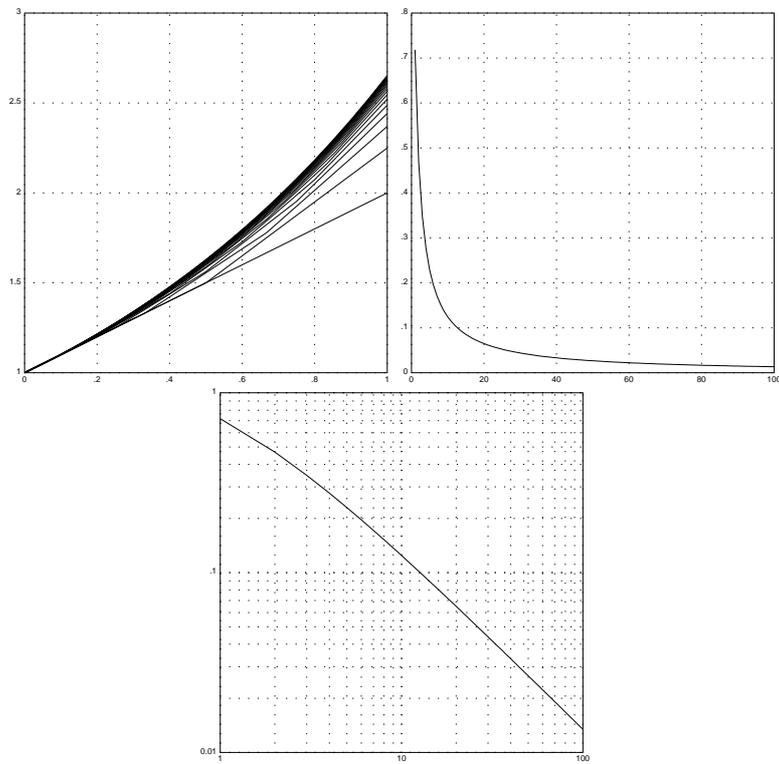
1.2 問題 4-2 解説

この問題を解いた人は案外少なく、また以後の分析が不十分だった人がほとんどでした。問題によっては、かなり大きな N を指定しないと十分精度の良い解が求まりません。前問の解説の繰り返しになりますが、収束のチェックには reidai4-1 のような解のグラフを描くだけでは不十分で、reidai4-1b のようなプログラムを作って誤差の解析をするべきでしょう。

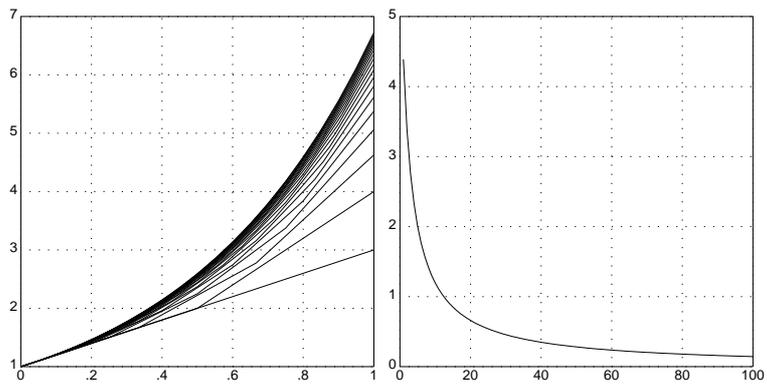
微分方程式の右辺が $f(t, x) = 2x$ のような場合は²、例題 4-1 の場合 ($f(t, x) = x$) と比べて、収束は若干遅くはなりますが本質的な違いはありません (証明はしていないわけですが、誤差が $O(N^{-1})$ となることが Euler 法による近似解の性質なわけで、このことは f に依りません)。

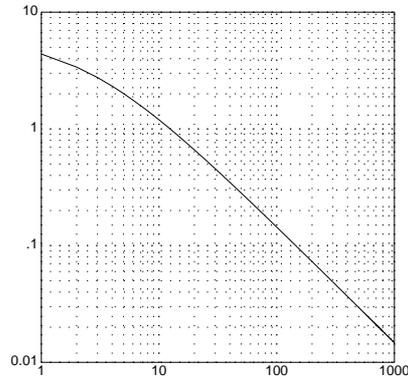
そのことを実際の計算で確かめてみましょう。そのために reidai4-1 | mgraph -M, reidai4-1b | mgraph, reidai4-1b | mgraph -lx -ly で描いた図 (それぞれ、 N は 20, 100, 1000 まで 1 刻みで増やしました。また初期条件は $x(0) = 1$ です。)

²これでは、例題で取り上げた関数 $f(t, x) = x$ とほとんど同じで、ちょっとツマラナイ、という気がしますが、レポートで取り上げた人が多かったので (こういうところくらいは他の人の真似をしないで欲しいな...) ここでもこの場合を調べてみることにしました。



と、 $f(t, x) = 2x$ の場合の同じ N に対応する図 (N や初期条件は上と同じです)

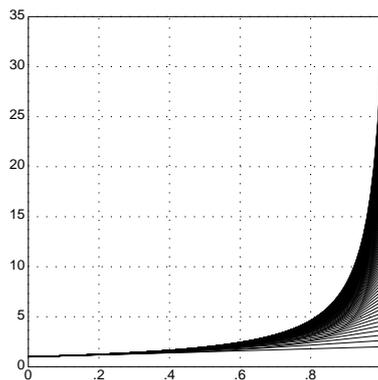




を見比べて見ましょう。

誤差の大きさ自体は解のグラフ（左側）を見てもあまり良く分かりません（この種のグラフは「分解能」が低いです）。中央、右側のグラフで同じ N に対応する誤差の値を見てみると、 $f(t, x) = 2x$ の場合の方がかなり大きいですが、収束の速さ（オーダー）はほぼ同じであることが分かります。（一番右側のグラフ（両側対数グラフ）で N が大きくなった時の傾きを比べると、ともにほぼ -1 になっていますね？）

ただし、厳密解が存在しないような場合は注意する必要があります。例えば、 $f(t, x) = x^2$ のような場合、厳密解は $x(t) = \frac{1}{1-t}$ ($0 \leq t < 1$) となるので、そもそも $x(1)$ は考えられない、つまり $t = 1$ までは解けないわけです。こういう f に対して Euler 法で計算してみると、 $0 \leq t < 1$ なる t に対しては Euler 法による解は厳密解に収束しますが、 $t = 1$ での値 x_N はどんどん大きくなって収束しません。



1.3 問題 4-3 解説

例えば、速さ 5m/sec で投げ上げた場合の、0~1 秒までを 100 分割してシミュレーションする場合、

```
waltz11% reidai4-2 > mondai4-3.data
5 100 0 1
```

のようにしてデータを作っておきます。こうして作ったファイル mondai4-3.data の先頭の行には、“VO,NSTEP,START_TIME,END_TIME=” のようなメッセージが入っていて、これは邪魔ですから nemacs 等で削除します。そして、

```
* print12.f
*   1 行にある 3 つの数のうち、一列目、二列目のものだけ出力する。

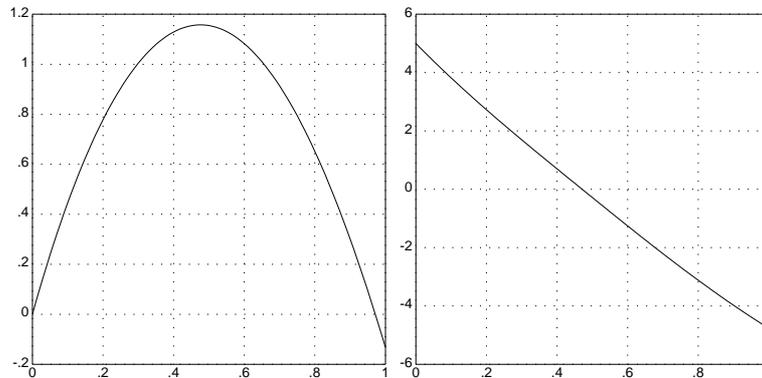
      program print12
      real x,y,z
*
      100  continue
      read(*,*,END=999) x,y,z          データが尽きたら文 999 に
ジャンプ
      write(*,*) x,y                  1,2列目のデータ(x,y)
のみ出力
      goto 100
*
      999  stop
      end
```

というプログラム “print12” を作って、

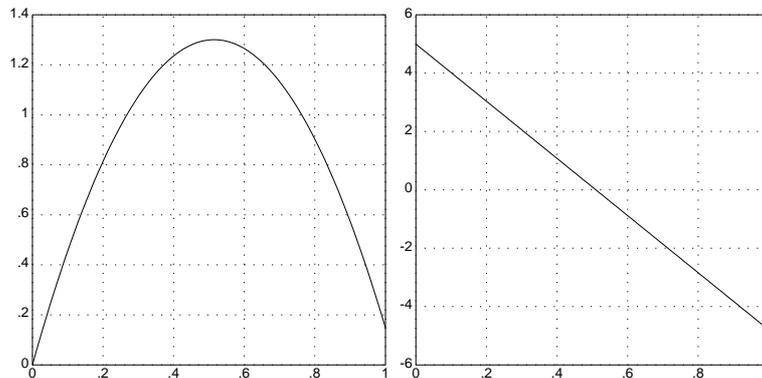
```
waltz11% cat mondai4-3.data | print12 | mgraph | xplot
```

とすれば、 $t-u$ 図が描けます³。同様に一列目、三列目を出力するプログラムを作れば、 $t-v$ 図が描けます。

³UNIX には awk というコマンドがあって、print12 の代わりに利用することができます: cat mondai4-3.data | awk 'print \$1,\$2' | mgraph | xplot



比較のために空気抵抗がない場合（関数 phi の中で定数 c を 0 にする）を見てみましょう。



例えば u について調べてみると、空気抵抗のある方（上段）が、(1) 最高点が低くなっている、(2) 地面に落ちてきた ($u = 0$ となった) 時刻が早くなる⁴などのことが分かります。

この程度の軽い問題では（問題の大きさに比べて使っている計算機が十分速いので）、毎回計算をやり直してグラフを描いてもいいのですが、もっと大きな問題を調べる場合は、計算は一度だけ遂行して、出力ファイルを色々な方法（大抵は小さなプログラムを使います）で解析する、という手順で作業をすることになります。そういうやり方を紹介してみた、ということです。

⁴空気抵抗のある方は $t = 1$ の時に既に $u < 0$ となっていますが、抵抗のない方は $t = 1$ でも $u > 0$ 、つまり地面に落ちていませんね。

1.4 問題 4-4 解説

この問題では $f(t, x) = x$ 、そして $x_0 = 1$ ですから、

$$k_1 = hf(t_0, x_0) = hx_0 = h \cdot 1 = h$$

$$k_2 = hf(t_0 + h/2, x_0 + k_1/2) = h(x_0 + k_1/2) = h(1 + h/2) = h + h^2/2$$

$$k_3 = hf(t_0 + h/2, x_0 + k_2/h) = h(x_0 + k_2/2) = h[1 + (h + h/2)/2] = h + h^2/2 + h^3/4$$

$$k_4 = hf(t_0 + h, x_0 + k_3) = h(x_0 + k_3) = h[1 + (h + h^2/2 + h^3/4)] = h + h^2 + h^3/2 + h^4/4$$

よって

$$\begin{aligned} x_1 &= x_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ &= 1 + [h + 2(h + h^2/2) + 2(h + h^2/2 + h^3/4) + (h + h^2 + h^3/2 + h^4/4)]/6 \\ &= 1 + (h + 2h + h^2 + 2h + h^2 + h^3/2 + h + h^2 + h^3/2 + h^4/4)/6 \\ &= 1 + (6h + 3h^2 + h^3 + h^4/4)/6 \\ &= 1 + h + h^2/2 + h^3/6 + h^4/24, \end{aligned}$$

つまり $x_1 = 1 + h + h^2/2! + h^3/3! + h^4/4!$ となります。Euler 法で同じことをすると $x_1 = 1 + h$ となります。厳密解では $x_1 = e^h = \sum_{n=0}^{\infty} h^n/n!$ ですから、Euler 法による近似解は 1 次の項まで、Runge-Kutta 法による近似解は 4 次の項まで、厳密解と一致していることとなります⁵。誤差については、Euler 法の場合 $O(h^2)$ 、Runge-Kutta 法の場合 $O(h^5)$ となり、当然 Runge-Kutta 法による結果の方が良好になるはずです。

1.5 問題 4-5 解説

まずプログラムの書き換えは

```
do 200 i=0,N-1
  x=x+h*f(t,x)
  t=t+h
200 continue
```

を

⁵元はと言えば、こうなるように Runge-Kutta 法は作られました。実は Runge-Kutta 法には兄弟と言える公式（それらも Runge-Kutta 型公式と呼ばれます）がたくさんあり、ここで紹介した「元祖 Runge-Kutta 法」をそういうものとはっきり区別したい場合は、（次数を明示して）4 次の Runge-Kutta 法と呼びます。

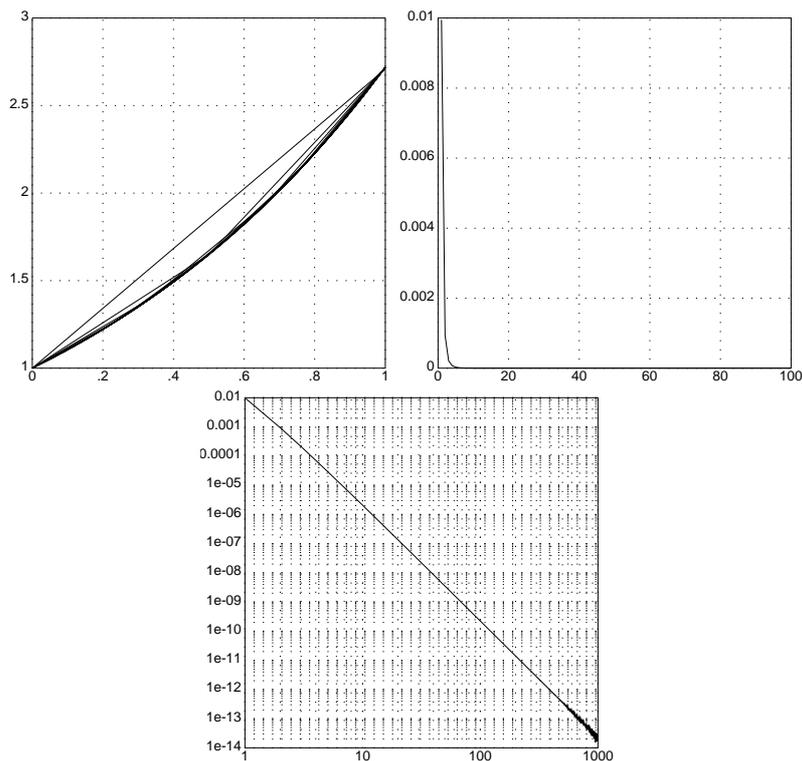
```

real k1,k2,k3,k4
....
do 200 i=0,N-1
  k1 = h*f(t,x)
  k2 = h*f(t+h/2, x+k1/2)
  k3 = h*f(t+h/2, x+k2/2)
  k4 = h*f(t+h, x+k3)
  x = x + (k1 + 2.0 * k2 + 2.0 * k3 + k4) / 6.0
  t = t + h
200 continue

```

とするだけで OK です。

例題 4-1 の 3 つのプログラムを Runge-Kutta 法に書き換えて計算してみると、



となります。特に右図でグラフの傾きが、ほぼ -4 になっていることが分かります。よって $|x_N - x(1)| = O(N^{-4})$ です。Euler 法の時 $|x_N - x(1)| = O(N^{-1})$ でしたから、Runge-Kutta 法の方がかなり速いことが分かります (問題 4-1 参照)。つまり誤差を $1/10000$ 倍にするために、Euler 法で

は N を 10000 倍する必要がありますが、Runge-Kutta 法では N を 10 倍にするだけで良い、ということになります。Runge-Kutta 法では 1 ステップ進むための計算に、Euler 法の 4 倍以上の計算が必要になりますが（公式を比較すると、 f の計算回数だけで勘定して 4 倍、他の計算も明らかに多いです）、この程度の不利はすぐに挽回出来るものであることが分かります。

注意： 実は reidai4-1b.f に相当するプログラムは、上に説明した書換えの他に、浮動小数データを倍精度型にしています。これは Runge-Kutta 法の精度が良過ぎるため、単精度を使った場合は、 N が小さいところで（この問題の場合は大体 12, 3 程度）限界に近い誤差に到達してしまって調べ難いせいです。（Fortran の単精度型浮動小数は 10 進にして 7 桁程度の精度しかありませんが、倍精度型では 10 進 16 桁程度の精度を持っています）。実際真ん中の図では、早い段階でグラフが N 軸についてしまっています（あんまり早いので $N = 100$ までグラフを描いた意味がほとんどなくなっています）。ここでも右側の図のようにデータの対数を取ることの必要性が分かります。

2 第5回の問題の解説

2.1 問題 5-1 解説

問題 5-1 で具体例を上げた行列で説明します。

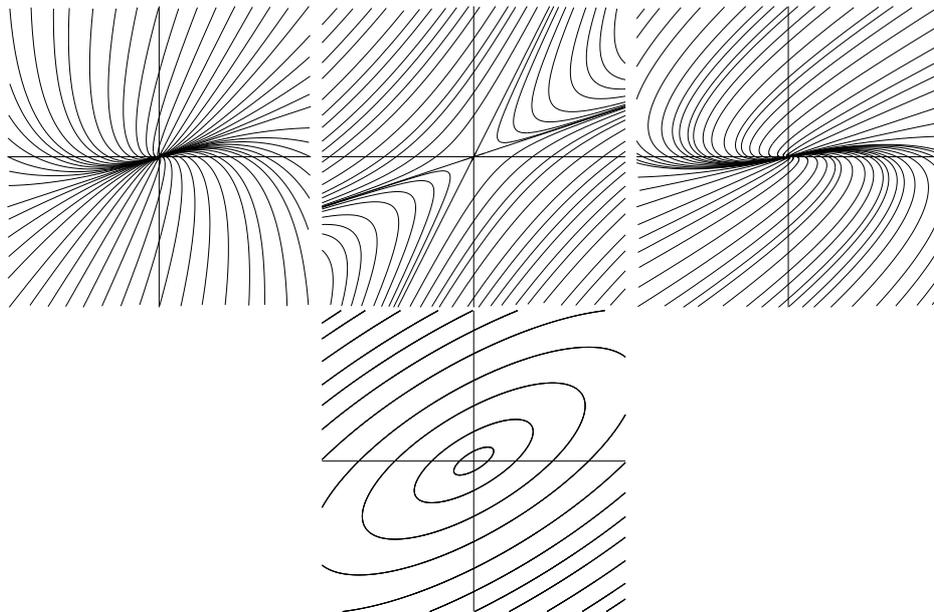
$$A_1 = \begin{pmatrix} -\frac{4}{5} & -\frac{3}{5} \\ \frac{2}{5} & -\frac{11}{5} \end{pmatrix}, \quad A_2 = \begin{pmatrix} \frac{8}{5} & -\frac{9}{5} \\ \frac{6}{5} & -\frac{13}{5} \end{pmatrix}, \quad A_3 = \begin{pmatrix} \frac{2}{5} & \frac{9}{5} \\ -\frac{1}{5} & \frac{8}{5} \end{pmatrix}, \quad A_4 = \begin{pmatrix} -1 & 2 \\ -1 & 1 \end{pmatrix}.$$

A_1 の固有値は $-1, -2$ で、対応する固有ベクトルはそれぞれ $(1, 2), (3, 1)$ です。従って Case I-2 に属し、すべての解軌道は原点に吸い込まれていき、原点は沈点になります。また直線 $y = 2x, y = \frac{1}{3}x$ は流れで不変になっています。

A_2 の固有値は $-2, 1$ で、対応する固有ベクトルはそれぞれ $(1, 2), (3, 1)$ です。従って Case I-3 に属し、原点は鞍状点になります。また直線 $y = 2x, y = \frac{1}{3}x$ は流れで不変です。

A_3 の固有値は 1 (重根) で、固有ベクトルは $(3, 1)$ のみ（したがって対角化不可能）です。Case III に属します。この場合固有値は正ですから、原点は湧出点です。

A_4 の固有値は $\pm i$ で、対応する固有ベクトルはそれぞれ $(1+i, 1)$, $(1-i, 1)$ です。Case IV 3 に属するので、すべての解軌道は楕円になり、原点は渦心点になります。



2.2 問題 5-2 解説

Runge-Kutta 法を Euler 法に書き換える仕方は簡単ではありますが、結構うっかりした人も多かったようですね⁶。一時変数を導入して、例えば

$$\begin{aligned} \text{newx} &= x + h * \text{fx}(x,y) \\ y &= y + h * \text{fy}(x,y) \\ x &= \text{newx} \end{aligned}$$

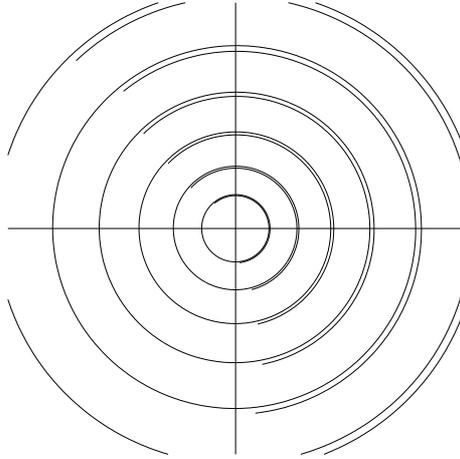
のようにする必要があります (y についての計算をする時に古い x の値が必要になるので、最初に $x=x+h*\text{fx}(x,y)$ としてはいけないのです)。

その結果得られたプログラムで、Case IV-3 に属する

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

の場合を試してみました。

⁶実は私も最初はどうもうっかりしていました。



本来は（単振動の方程式に由来する問題で、周期運動となるはず）閉軌道⁷となるべきものが、一回まわることにはずれてしまって螺旋のようになっていることが分かります。時間間隔 h を小さくすると改善できますが、Euler 法の一つの限界の現れと見るべきでしょう。

2.3 問題 5-3 解説

単振動の方程式は Case IV の 3、すなわち純虚数根を持つ場合に属します。すべての解軌道は閉じた楕円になり、周期現象であることを示唆します。

一方抵抗のある場合の微分方程式は、係数行列の固有方程式が $\lambda^2 + \gamma\lambda + \omega^2 = 0$ で、その判別式は $\gamma^2 - 4\omega^2$ となります（ γ, ω は正の定数です）。

(1) $\gamma > 2\omega$ ならば相異なる 2 実根 $-\gamma \pm \sqrt{\gamma^2 - 4\omega^2}$ で、これは共に負ですから Case I の 2（沈点の場合）に相当します。

(2) $\gamma = 2\omega$ ならば、重根 $-\gamma$ を持ち、Case III であることが分かります。これも原点は沈点となり、解軌道は原点に吸い込まれていきます。

(3) $\gamma < 2\omega$ ならば、2 虚根 $-\gamma \pm i\sqrt{4\omega^2 - \gamma^2}$ を持ち、特にその実部は負ですから、Case IV の 2 に属することが分かります。原点は安定渦状点となり、解軌道は対数螺旋を描きながら原点に吸い込まれていきます。

もう紙がもったいないので図は省略します。

⁷閉曲線となる解軌道のこと。単振動の場合の軌道は、（プリントにも書きましたが）楕円になるはずで。

いずれの場合も解軌道は原点に吸い込まれていきます。これは時間が経過すると、抵抗によりエネルギーが消費されて 0 に近づくことを意味しています。しかし抵抗がそれほど大きくない場合は、抵抗がない場合のように、解軌道が原点の周りを（無限回）回ります。これは減衰しながらも振動はしているということです。

2.4 問題 5-4 解説

色々な考え方があると思いますが、、、

マウスを用いた場合、何と云っても目で確認しながら位置を指定出来るので、多くの場合作業がとても楽です。ただし解軌道が初期値に敏感に依存する場合は細かいコントロールが難しくなります。キーボードから数値で入力する場合は 6,7 桁の精度⁸は簡単に確保できますが、マウスでは 2 桁か、良くて 3 桁がせいぜいでしょう。また数値を入力する場合は、実験結果を再現することは難しくありませんが、マウスでは「さっきと同じ点を指定する」のは大変です（もっともこれはマウスで読み取った値をファイルにきちんと保存しておくようにプログラムを作れば済むことです）。もう一つ、キーボードからの入力の利点は UNIX の場合、入出力リダイレクションの利用が出来て、ファイルに納めたデータを利用しやすいということです（実際、例題 5-1 では “reidai5-1.data” というテキスト・ファイルをリダイレクションを利用して読み込みました: `cat reidai5-1.data | reidai5-1`）。

こうして並べると、マウスの利点が少ないようですが、実際はかなりマウスによって使い易くなっていることが分かると思います。特に、たくさん解軌道を描きたい時はマウスを使わないでやるのは面倒でしょう。（reidai5-1.f 自体はどっちつかずのプログラムになってしまったので、全体として非常に不細工な、使いにくいプログラムになってしまっています。ごめんなさい。）

3 レポートの書き方について

いくつか思いつくままあげておきます。

- どの問題を解いたか、はっきり書いて下さい。

⁸単精度をフルに利用できるということです。

- 関数等を自分で選ぶような問題では、どれを選んだか、文章中にはっきり書きましょう（プログラムを読むしかなかったレポートがありますが、これはいけません）。また入力データも出来る限り添えるべきです。
- 例題プログラムを変更した場合は、ただ印刷したリストを添付するだけでなく、どこを変更したか書いて下さい。
- 何等かの説明をつけるのは当然として、結果の分析も分かる範囲でして下さい。
- 比較する問題では、「 λ は $\times \times$ なのに、 μ は \dots 」のように自分が読み取ったことをなるべく具体的に書いて下さい。
- 問題が途中までしか解けていなくても、出来たところまで正直にきちんと書いたものはレポートとして合格です（良いレポートだと考えます）。

4 訂正事項

第4回のプリント⁹に間違いがありました。Case III. の解説で、「 $\lambda = 0$ であれば平面上のすべての点が平衡点、 $\lambda \neq 0$ であれば原点を通る一つの直線上の点が平衡点全体となります。」と書いたのは全くのウソです。このあたりの解説は、ある本から機械的に抜き出したものなのですが、寝惚けていて他のところの説明を混入させてしまいました。ごめんなさい。

⁹そもそもタイトルからして「常微分方程式の数値シミュレーション (2)」でなく、「常微分方程式の初期値問題 (2)」なのでした。