宇宙科学実習 I：重力二体軌道運動を数値的に解く]

これまで，重力を及ぼしあう 2 つの物体の運動を考えてきました．このような問題は**重力二体問題**と呼ばれます.[1] ここまで考えていた問題では質点の作る重力ポテンシャルは Newton 重力ポテンシャルで，質点からの距離 r のみに依存し，大きさは r^{-1} に比例していました．この場合，重力二体問題には解析的な厳密解が存在し，その軌道は二次曲線となることを学びました．

では，重力が単なる質点の Newton 重力ポテンシャルから導かれない場合は重力二体問題の解はどうなるでしょうか．例えば，重力法則が Newton 重力であったとしても，空間的に有限の範囲に広がった物質分布（一般に球対称ではない）を持つ天体はこの場合に当てはまります．全ての恒星，惑星は自転していると思われますが，それらの物質分布は自転の影響で自転軸と垂直な面に膨らむ傾向にあります．このような星の周りを公転する伴星，衛星の運動はどうなるのでしょうか．また，銀河の重力ポテンシャルも一般には球対称ではありません．銀河中の星やガスの運動は単純な二次曲線ではないと予想されますが，ではどんな軌道を描くのでしょうか．

一方，天体が球対称であっても重力法則が Newton 重力でなければ，やはり軌道決定問題の解は単純な二次曲線にはなりません．現在では，Newton の重力法則は一般相対論的重力法則の近似であることがわかっていますが，一般相対論的な重力二体問題の解はどうなるのでしょうか．

これらの場合，一般に解析解は存在しません．そこで今回はこれらの問題を数値的な近似解を求める問題に置き換えて解くことを考えてみましょう．

目的

”球対称な Newton 重力”の仮定を取り去った場合の重力二体問題の数値解を考える．

1 Newton 力学での基礎方程式

ここでは，以前に考えたように，重力相互作用する 2 物体のうち，ひとつの物体の質量が他の物体の質量に比べて無視できる極限を考えましょう（テスト粒子問題）．

重たい方の物体の質量を M とし，それを座標原点に置きます（このとき系の重心も原点にある）．軽いほうの物体の位置ベクトルを $\vec{r}$ とします．このとき，Newton 力学での運動方程式は

$$\frac{d^2\vec{r}}{dt^2} = -\vec{\nabla}\Phi \tag{1}$$

となります．ここで，スカラー関数 Φ は重力ポテンシャルで，点状重力源の場合は $\Phi = -\frac{GM}{|\vec{r}|}$ ですが，一般の場合はもっと複雑です．

この式をデカルト座標での成分について書き直すと $\vec{r} = (x, y, z)$ として

$$\frac{d^2x}{dt^2} = -\frac{\partial\Phi}{\partial x} \tag{2}$$

[1] これを一般化して，重力のみを及ぼしあう N 個の粒子系の運動を解く問題は，重力 N 体問題と呼ばれます．

$$\frac{d^2y}{dt^2} = -\frac{\partial \Phi}{\partial y} \tag{3}$$

$$\frac{d^2z}{dt^2} = -\frac{\partial \Phi}{\partial z} \tag{4}$$

となります.

1.1 方程式の無次元化：点状重力源の場合

まず．点状重力源の場合に基礎方程式を簡単化します．ここでは xy 平面内での運動の場合について考えましょう．この場合，$\Phi = -GM/r = -GM/\sqrt{x^2+y^2}$ なので，

$$\frac{d^2x}{dt^2} = -\frac{GM}{r^3}x \tag{5}$$

$$\frac{d^2y}{dt^2} = -\frac{GM}{r^3}y \tag{6}$$

となることは以前みました.

物理の問題では，基礎方程式の両辺に現れる物理量を，その次元を持つ適切な量（これは考えている系によって異なります）で割って無次元化することで式の取り扱いを簡単化します．上の方程式を無次元化してみましょう[2]

この問題では，次のように無次元化してみます.

- 距離：中心天体への最近接点の，原点からの距離 r_0
- 時間：最近接点距離 r_0 を半径とする円軌道の公転周期 P の定数 (α) 倍

円軌道の公転周期は，2π を角振動数 $\sqrt{GM/r_0^3}$ で割った，$P = 2\pi/\sqrt{GM/r_0^3}$ です．これらで無次元化を行うと（物理量 f の無次元化量を $\tilde{f}$ と書く），

$$\tilde{r} = r/r_0, \quad \tilde{x} = x/r_0, \quad \tilde{y} = y/r_0, \quad \tilde{t} = t/(\alpha P) \tag{7}$$

を用いて，

$$\frac{d^2\tilde{x}}{d\tilde{t}^2} = -(2\pi\alpha)^2\frac{\tilde{x}}{\tilde{r}^3} \tag{8}$$

$$\frac{d^2\tilde{y}}{d\tilde{t}^2} = -(2\pi\alpha)^2\frac{\tilde{y}}{\tilde{r}^3} \tag{9}$$

です．ここで，$\alpha = (2\pi)^{-1}$ と選べば式が簡単になります.

$$\frac{d^2\tilde{x}}{d\tilde{t}^2} = -\frac{\tilde{x}}{\tilde{r}^3} \tag{10}$$

$$\frac{d^2\tilde{y}}{d\tilde{t}^2} = -\frac{\tilde{y}}{\tilde{r}^3} \tag{11}$$

式 (10),(11) は 2 階の微分方程式の組なので，$2 \times 2 = 4$ 個の初期条件が与えられれば積分できます.

[2]同じ問題でも無次元化の仕方は一通りではありません．大抵の場合，予想される物理量が 1 のオーダーに近くなるような無次元化をします．たとえば太陽系内の天体の運動では，距離の無次元化に 1 メートルや 1 パーセクではなくて，1 天文単位を使うのが便利です.

1.2 初期条件

初期条件としては，時刻 $t = 0$ での座標と，速度成分を与えます．最近接点を x 軸上に設定しましょう：

$$(\tilde{x}(0),\ \tilde{y}(0)) = (1, 0). \tag{12}$$

最近接点での速度は位置ベクトルと直交していますので，x 成分は 0 です．y 成分の値を v とすると，この値を変えることで異なる軌道が実現します：

$$\left(\frac{d\tilde{x}}{d\tilde{t}}(0),\ \frac{d\tilde{y}}{d\tilde{t}}(0)\right) = (0, v). \tag{13}$$

2　数値計算ソフトウェア Octave

ここでは，Ubuntu にインストールされている Octave [3]を使ってみましょう．

2.1　Octave の起動

ターミナルで octave と入力すると，図 2.1 のようなウィンドウが立ち上がります．このウィンドウで Octave コマンドの実行，スクリプトファイルの編集ができます．

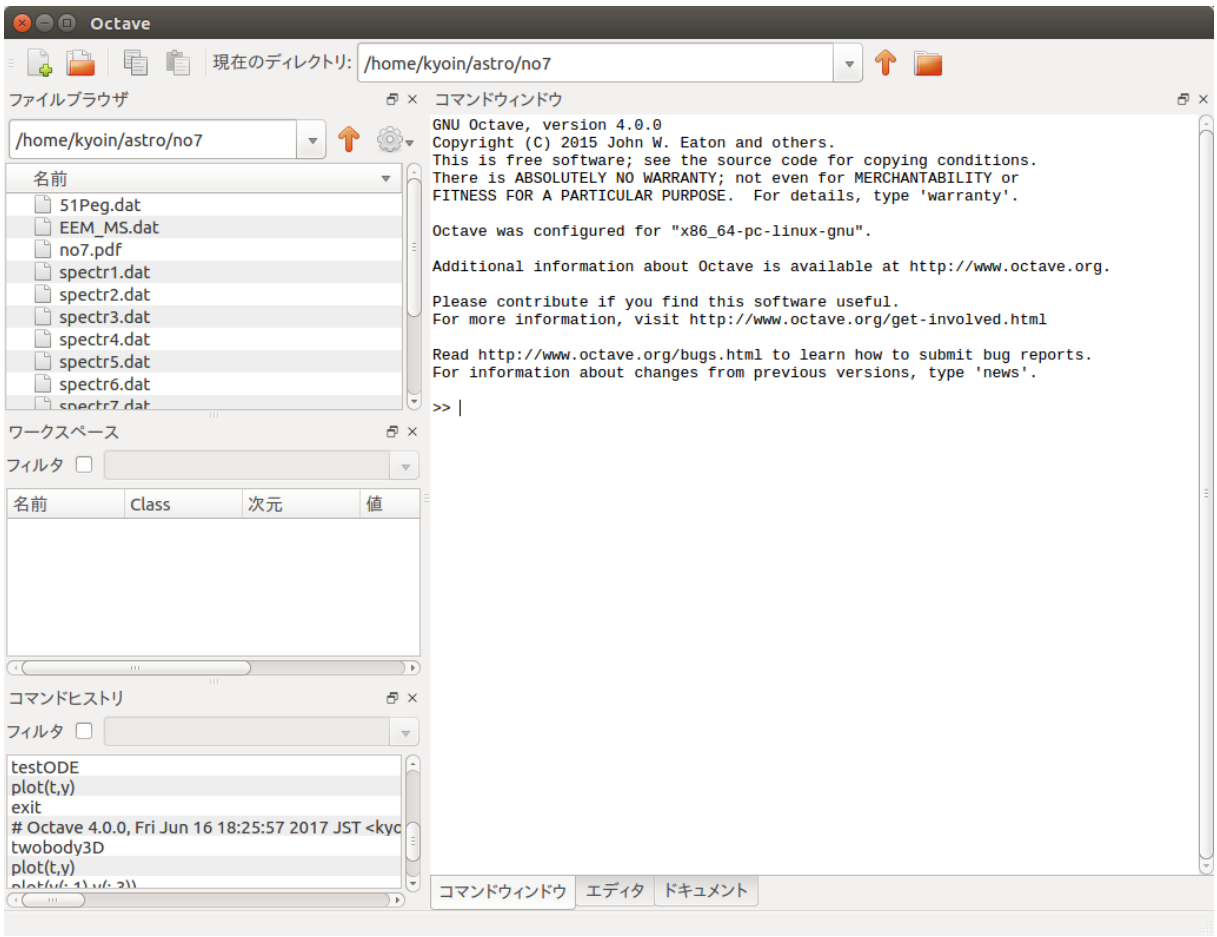

図 1: Octave のウィンドウ．右の「コマンドウィンドウ」を選択するとコマンド入力ができます．「エディタ」では Octave のスクリプトファイルを編集できます．編集したファイルはコマンドウィンドウで実行可能です．ファイルブラウザでは，このエディタで開きたいスクリプトファイルを選択できます．

2.2　基本演算

```
>> 1+2.5*sin(0.2*pi)/(1-0.3^2)
ans =  2.6148
```

[3]MathWorks 社の製品 MATLAB に似せて作られている数値計算用フリーソフトです．MATLAB と完全互換ではありませんが，文法はよく似ています．情報教育棟では純正の MATLAB が使えます．

exp（指数関数），sin（正弦関数）などの基本的な関数が使えます．またベキ乗は 3^2 などと書きます．
出力は，通常は 5 桁程度に限られていますが，

```
>> 8/3
ans =  2.6667
```

format long と指定すると，

```
>> format long
>> 8/3
ans =  2.66666666666667
```

出力桁数が伸びます．

また変数 x に値を代入して計算できます．

```
>> x=1.5
x =  1.50000000000000
>> x^5
ans =  7.59375000000000
```

なお「x=1.5;」として，「;」を末尾につけて実行すると x = 1.50000000000000 は表示されません．いちいち入力結果を表示されると煩雑になって見づらくなる場合，不必要な実行文には「;」を付けます．

2.3 ベクトル，行列

Octave はベクトルや行列の計算に力を発揮します．$1 \times n$ の n 次元行ベクトルは

```
>> v = [1 2 3 4 5]
v =

   1   2   3   4   5
```

のように，角括弧を使い，要素の間にスペースを入れます．$n \times 1$ の n 次元列ベクトルは

```
>> u = [0 ; -1.5 ; -2.0 ; 3.1 ; 0.2]
u =

   0.000000000000000
  -1.500000000000000
  -2.000000000000000
   3.100000000000000
   0.200000000000000
```

のように定義します．これらベクトルの積は

```
>> v*u
ans =  4.40000000000000
```

```
>> u*v
ans =

    0.00000    0.00000    0.00000    0.00000    0.00000
   -1.50000   -3.00000   -4.50000   -6.00000   -7.50000
   -2.00000   -4.00000   -6.00000   -8.00000  -10.00000
    3.10000    6.20000    9.30000   12.40000   15.50000
    0.20000    0.40000    0.60000    0.80000    1.00000
```

のように計算できます．行列の計算は

```
>> A=[[2 1] ; [-1 1]]
A =

   2   1
  -1   1

>> B=[[0 1] ; [-1 0]]
B =

   0   1
  -1   0

>> B(2,1)       ← 行列 B の 2,1 成分
ans = -1

>> A + B - A * B
ans =

   3   0
  -1   2
```

のように行えます．

2.3.1 linspace() 関数

linspace という関数によって，ある区間に線形で等間隔に離れた成分を持つベクトルを生成できます．

```
>> y = linspace(0,3,50)       ←  0 から 3 までの区間を等間隔に 49 等分して 50 点とる
y =

 Columns 1 through 10:

   0.00000   0.06122   0.12245   0.18367   0.24490   0.30612   0.36735   0.42857   0.48980   0.55102

 Columns 11 through 20:
```

```
   0.61224   0.67347   0.73469   0.79592   0.85714   0.91837   0.97959   1.04082   1.10204   1.16327

 Columns 21 through 30:

   1.22449   1.28571   1.34694   1.40816   1.46939   1.53061   1.59184   1.65306   1.71429   1.77551

 Columns 31 through 40:

   1.83673   1.89796   1.95918   2.02041   2.08163   2.14286   2.20408   2.26531   2.32653   2.38776

 Columns 41 through 50:

   2.44898   2.51020   2.57143   2.63265   2.69388   2.75510   2.81633   2.87755   2.93878   3.00000

 >> y(17)                ← l 7番目の成分
ans =  0.97959
```

2.4 Octave の関数

Octave ではユーザーが関数を定義し，それを呼び出して使うことができます．ここでは次のようなベクトル値 2 変数関数 fxn を定義してみます．

```
function ydot = fxn(y,t)
ydot(1) = y(2);
ydot(2) = -0.5*y(2) - 1/(1+t^2)*y(1);
endfunction
```

これは，独立変数 t の関数である $(y_1(t), y_2(t))$ という 2 関数によって，$fxn(1) = y(2)$, $fxn(2) = -0.5y_2(t) - y_1(t)/(1+t^2)$ という二つの関数を定義しています．

一行目で，関数の値として返される値を格納する変数名 ydot，呼び出されるときの関数名 fxn，引数 y,t が記述されています．このとき，変数 ydot，y,t が複数の成分を持っていた（ベクトル，行列である）としても，その次元について宣言する必要はありません．二行目，三行目では ydot の値を具体的に y，t の関数として定義しています．この場合，多次元の変数（ここでは y）の成分をあからさまに書きます．四行目で関数の定義を終了します．この関数を呼び出して値を計算させてみましょう．

```
>> z = [1.5;-0.2]
z =

   1.50000
  -0.20000

>> t=1.3
t =  1.3000
>> fxn(z,t)
```

```
ans =

  -0.20000  -0.45762
```

入力するベクトル変数 z は列ベクトルとして定義しておきます．結果は，fxn(1)　fxn(2) のように列記されています．

2.5　グラフの描画

Octave では gnuplot を用いてグラフを描くことができます．２次元プロットでは，関数の独立変数を表すベクトルと，その関数の値のベクトルを用意します．

```
>> x = linspace(0,pi,100);
>> y = x.^(1/2).*sin(x.^2);
>> plot(x,y)
```

一行目で，0 から π の区間を 99 等分して 100 点とり，それらを要素として並べたベクトルとして x を定義しています．二行目ではそれらの点での関数 $y = \sqrt{x}\sin x^2$ の値を計算しています．ここで注意すべきは，二行目の x の各演算子 (「^ (1/2)」,「*」,「^ 2」) に .(ピリオド) がついていることです．これが無いとエラーとなります．ピリオドの後の演算はベクトル x の各要素に対して演算することを意味します．例えば n 次正方行列 A に対して A^ 2 とすると，これは行列としての 2 乗を行います．他方で，A.^ 2 とすると，行列 A の n^2 個の各成分を 2 乗して新しい n 次正方行列を作るということです．三行目で y を x の関数としてプロットしています．

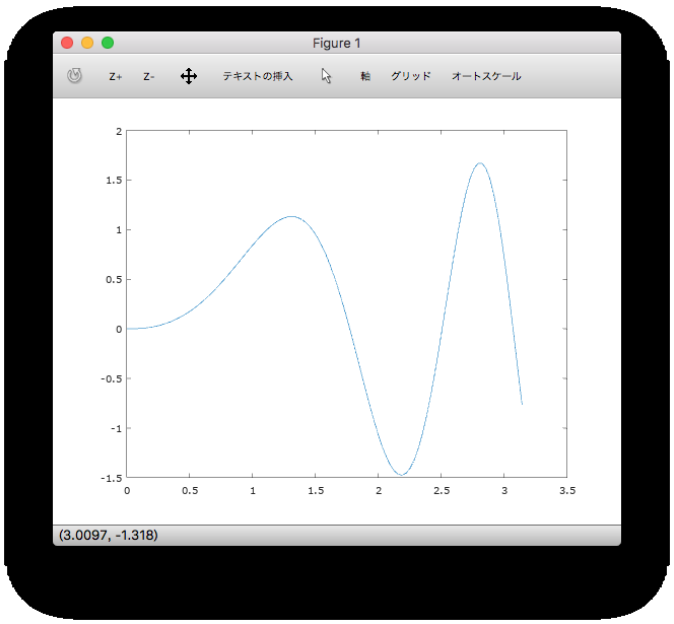

図 2: Octave によるプロット例.

2.6 スクリプト

Octave で複雑な計算を行う場合，コマンドを記述した手続きファイル（スクリプトファイル）を作成し，これを読み込むようにしましょう．これによって間違いを発見し，修正をしやすくなります．

スクリプトファイルは「□ × △.m」というように，拡張子「.m」をつけた名前で作りましょう．

以下の例は，関数 $y = x^a \sin x^2$ をプロットするスクリプトです．

```
## figure example
x = linspace(0,pi,100)
a = input(" non negative number a ")
y = x.^a.*sin(x.^2);
plot(x,y)
```

#で始まる一行目はコメント行です．二行目では，” ”で囲まれた文字列を画面に表示し，入力待ちの状態になります．ここで数字を入力すると，以下の文が実行されて関数がプロットされます．この例をエディタで作成，保存して実行してみてください．

3 Octaveによる常微分方程式の解

ここでは Octave によって常微分方程式を解いて，それをプロットする方法を学びましょう．

3.1 一階常微分方程式の解

まず初めに簡単な微分方程式

$$\frac{d}{dt}y(t) = -y(t), \quad y(0) = 0.5 \tag{14}$$

を $0 \leq t \leq 1$ で解いてみます．

微分方程式 $\frac{dy}{dt} = -y,\ y(0) = 0.5$ の解

```
### ODE: dy/dt = -y
y0 = [0.5];
t = linspace(0,1,100);
#
function ydot = rhs(y,t)
ydot(1) = -y(1);
endfunction
#
y = lsode("rhs",y0,t);
plot(t,y)
###
```

第二行は微分方程式の初期条件をベクトル（次元＝ 1 ）の形で与えています．第三行で独立変数 t を $0 \leq t \leq 1$ の範囲で等間隔にし，100 点とります．第 5 から 7 行目では関数として微分方程式の右辺（名前は rhs）を定義しています．第九行目で Octave にあらかじめ定義されている関数 lsode を呼び出して微分方程式を解

いています[4]．lsode の第一引数は微分方程式の右辺を定義する関数の名前，第二引数は初期条件のベクトル，第三引数は独立変数です．結果はベクトル y に代入されます．この内容のファイルをエディタでつくって，実行してみてください．

※　lsode について
lsode は，Lawrence Livermore 研究所の Alan C. Hindmarsh が作成した常微分方程式パッケージ ODEPACK の副プログラムの一つです．ODEPACK そして lsode は元々数値計算言語 Fortran によって書かれていましたが，他のプログラミング言語にも移植されて広く使われています．lsode は通常 Adams 法で微分方程式を解いています．Adams 法については Appendix を参照してください.[5]

3.2　連立常微分方程式

連立微分方程式も，変数や方程式の右辺をベクトルにすることで同様に解けます．次の例は $\vec{Y} = {}^T(Y(1), Y(2))$ の微分方程式，

$$\frac{d\vec{Y}}{dt} = \vec{g}(Y,t), \quad \vec{g} = {}^T(Y(2), \ -4Y(2) - 13Y(1)) \tag{15}$$

を $0 \leq t \leq 4$ の区間を 999 等分して，初期条件 $Y(1) = 1, Y(2) = 1$ の元で解くプログラムです．

連立微分方程式の解

```
### ODE
y0 = [1;1];
t = linspace(0,4,1000);
#
function ydot = g(y,t)
ydot(1) = y(2);
ydot(2) = -4*y(2) -13*y(1);
endfunction
#
y = lsode("g",y0,t);
###
plot(t,y)
```

最後の行で二次元プロットを行っています．この行を plot(t,y(:,1)) と書くと，ベクトル変数 y の第一列，つまり y(1) のみをプロットします．

3.3　高階微分方程式の解

高階の常微分方程式は，一階の連立常微分方程式系に帰着することで解くことができます．例として次の微分方程式

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + 13y = 0, \quad \text{ただし } y(0) = 1, \ \frac{dy}{dt}(0) = 1 \tag{16}$$

[4] MATLAB では，lsode は使えません．代わりに ode45 など，複数のソルバーが使えます．
[5] 詳しくは，共立数学講座 12「数値解析」森正武（共立出版）などを読んでください．

を考えましょう．この方程式は，$\frac{dy}{dt} \equiv z$ という新しい変数を導入すると，次の２つの方程式の組に書き直せます．

$$\frac{dy}{dt} = z \tag{17}$$

$$\frac{dz}{dt} = -4z - 13y \tag{18}$$

対応する初期条件は $y(0) = 1,\ z(0) = 1$ です．この方程式は上述の連立常微分方程式の解法で解けます．実際，この方程式系は前章の例と同じものです（$\vec{Y} = (Y(1), Y(2)) = (y, z)$ と書き直す）．

高階常微分方程式が複数連立されている場合も同様に，補助的な従属変数を定義すれば常に一階の常微分方程式の組に直して解くことができます．

3.4　点運動のアニメーション

微分方程式の解ベクトルをプロットするとき，一点をを描く毎に図をリフレッシュすることを繰り返してアニメーションをつくることができます．

```
###-- animation ---
for i=1:1000
  hold on;
  plot(y(i,1),y(i,2),"+");
  refresh;
end
hold off;
```

"+"の指定で，点のプロットを+で行っている．これを"."や"*"に変えると，プロットの点の形が変わります．

4　Newton 重力での二体問題–質点の場合

ここでは，Newton 重力での二体問題 (10), (11) を解いてプロットしてみましょう．

課題 (A)

(10), (11) を初期条件 (12)，(13) の下で解くプログラムを書きなさい．次に，軌道が楕円になるような初期速度 $v > 0$ の範囲を決めなさい．v をこの範囲の値で与えて数値計算を行い，軌道をプロットしてみなさい．

5　非球対称分布する質量の周囲での３次元的軌道

自転等によって星が球対称から変形している場合 (ただし軸対称性はもっているとする)，その周囲を運動する衛星，惑星の運動は次のような重力ポテンシャルによって決まります．

$$\Phi = -\frac{GM}{r} + \frac{2}{5}\frac{GMa^2}{r^3}\epsilon P_2(\cos\theta) \tag{19}$$

ここで，(r, θ) は対称軸を天頂方向とした球座標で，$P_2(\cos\theta) = \frac{1}{2}(3\cos^2\theta - 1)$ です．[6]また M，a は星の質量と平均半径，$\epsilon \geq 0$ は扁平度と呼ばれるパラメータで，星が球状からどの程度変形しているかを表します．

[6] $P_\ell(\cos\theta)$ は ℓ 次のルジャンドル多項式と呼ばれる特殊関数で，物理学の問題には頻繁に出てきます．参考：「岩波 数学公式３」（森口，宇田川，一松　共著）

$\epsilon = 0$ のときは星は球状になっています．(19) はこの場合，球対称の場合のポテンシャル $-\frac{GM}{r}$ に等しくなります．

課題 (B)

ポテンシャル (19) の下で運動する質点の運動方程式をデカルト座標 (x, y, z) で書き下し，適宜無次元化しなさい．次に，この方程式を解いて軌道をプロットするプログラムを書き，実行してみなさい．

3 次元プロットには `plot3` が使えます．例えば，変数 `Q` が $n \times 3$ 行列として定義されていて（n は微分方程式を積分した時間ステップの数），これを 3 次元デカルト座標の軌跡としてプロットするには，

```
plot3(Q(:,1), Q(:,2), Q(:,3))
```

とします．

A　Adams 法について

$Y(t)$（一般にはベクトル変数）の常微分方程式

$$\frac{dY}{dt} = F(Y;t), \quad Y(T_0) = Y_0 \text{ (初期条件)} \tag{20}$$

を解く方法は様々なものがあります．ここでは `lsode` が使っている Adams 法について概要をみてみましょう．いま，t の区間 $T_0 \le t \le T$ でこの区間を $N-1$ 等分し，これら離散的な N 点での近似解を求めることを考えます．$h = \frac{T-T_0}{N-1}$ とします．第 i 番目 $(t = t_i)$ と $i+1(t = t_{i+1})$ 番目の間の区間 $(t_1 = T_0$ とする$)$ で (20) を形式的に解くと

$$Y(t_{i+1}) - Y(t_i) = \int_{t_i}^{t_{i+1}} F(Y(t');t')dt' \tag{21}$$

とかけます．(21) の右辺は未知関数 Y に依存しているのでそのままでは解けませんが，**この区間で F の関数系が多項式として近似できるなら**ば積分は容易です．いま，$t = t_i$ までの区間で近似解 $Y(t_j)$ $(1 \le j \le i)$ が得られているとすると，$F(Y(t_j);t_j)$ も既知です．したがって，t_{i-p+1} から t_i までの p 個の点 $(t_j,\ F(Y(t_j);t_j))$　(ただし $i-p+1 \le j \le i$) をとって，これらを通るような $p-1$ 次近似多項式 $\tilde{F}(t)$ がつくれます．この F の近似多項式 $\tilde{F}(t)$ を (21) の右辺に代入すれば，被積分関数は t の $p-1$ 次多項式なので，簡単に積分できます．すると，

$$Y(t_{i+1}) = Y(t_i) + [\text{右辺の多項式近似の積分}] \tag{22}$$

のように，$t = t_{i+1}$ での Y が求まります．これを順次繰り返して区間の右端 $t = T$ まで Y を計算します．