第 6 章　方程式の数値解法

3 次や 4 次の方程式までは直接解法があるが、5 次以上の方程式や非線形方程式を解くには、一般には数値的に解くしかない。方程式の解を求める数値解法をマスターしよう。

6.1　Newton-Raphson 法

Newton-Raphson 法は、関数 $f(x)$ が単調連続で変曲点がなく、且つ導関数 $f'(x)$ が数値的に求められる時に利用できる数値解法で、収束が速く、広く用いられている。いま、曲線上の点 $(x, f(x))$ での接線と x-軸との交点を $(x_{n+1}, 0)$ とすると、

$$f'(x_n) = -\frac{f(x_n)}{x_{n+1} - x_n} \tag{6.1}$$

だから、解の適当な推定値から始めて、反復公式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (n = 0, 1, 2, \cdots). \tag{6.2}$$

を繰り返すと、漸近的に解に達する。

収束の判定条件としては、ある ε に対して、

1. 二つの近似解間の差 $|x_{n+1} - x_n| < \varepsilon$
2. 二つの近似解間の相対差 $|\frac{x_{n+1} - x_n}{x_n}| < \varepsilon$
3. 二つの近似解関数値間の差 $|f(x_{n+1}) - f(x_n)| < \varepsilon$

等が考えられるが、1 と 2 の条件では、$f'(x_n)$ の値が大きいと、真の解に未だ充分近くないのに満たされてしまう場合があるので、3 の条件を判定条件として用いる方が誤った解を出してしまう危険性がすくない。

以下の外部関数 `rtnewt` は、関数 $f(x)$ を `funcd` として、$f(x) = 0$ となる解を Newton-Raphson 法で解くプログラム例である[1] 。収束判定条件は上記 1 を使っている。

```
function rtnewt(funcd,x1,x2,xacc)
```

[1] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992, *Numerical Recipes in FORTRAN77* (*Second Edition*), (Cambridge University Press, Cambridge) から転載。尚、この本は、http://cfatab.harvard.edu/nr/nronline.html でオンラインで読む事が出来る。

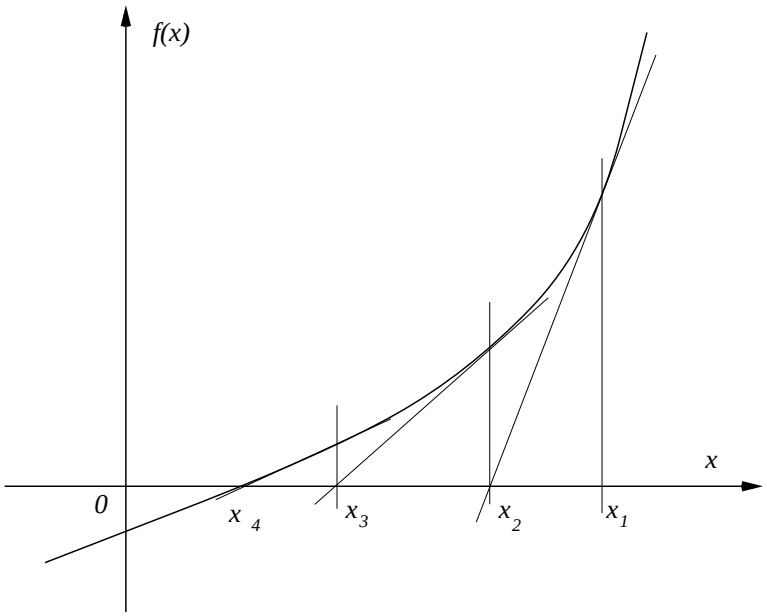

図 6.1: Newton-Raphson 法の概念図。

```
      Implicit NONE
      Integer JMAX
      Real rtnewt,x1,x2,xacc
      external funcd
      Parameter (JMAX=20)       !Set to maximum number of iterations.
! Using the Newton-Raphson method, find the root of a function known to
! lie in the interval [x1,x2]. The root 'rtnewt' will be refined until
! its accuracy is known within +-'xacc'. 'funcd' is a user-supplied
! subroutine that returns both the function value and the first
! derivative of the function at the point x.
      Integer j
      Real df,dx,f
      rtnewt=.5*(x1+x2)         !Initial guess.
      do j=1,JMAX
        call funcd(rtnewt,f,df)
        dx=f/df
        rtnewt=rtnewt-dx
        if((x1-rtnewt)*(rtnewt-x2).lt.0.)
     *    pause 'rtnewt jumped out of brackets'
```

```
  if(abs(dx).lt.xacc) return     !Convergence.
end do
pause 'rtnewt exceeded maximum iterations'
end
```

6.1.1 EXTERNAL 文

上記サンプルプログラムで、`external funcd` という行がある。これは、EXTERNAL 文と呼ばれる宣言文である。FUNCTION 文の引数は、通常はその関数の独立変数であるが、今の場合、`funcd` は、解きたい関数名である。そして、この関数は、別にサブルーチンで与える。そこで、FUNCTION 文の引数が、外部手続きサブルーチンの名である事を宣言しておくのである。

6.1.2 問題：Newton-Raphson 法を簡単な問題に適用する

Newton-Raphson 法で、

$$f(x) = 2x + \sin x - 2\pi = 0 \tag{6.3}$$

を解け。

6.1.3 問題：Mollwide 画法で天球図を描く

CfA カタログを使って、銀河の天球上の赤経・赤緯座標、銀経・銀緯座標、の二次元マップを、Mollwide 画法で描いて見よう。Mollwide 図法では経度・緯度 λ-ϕ を次の様に x-y 座標に変換する。

$$x = (\sqrt{8}/\pi)(\lambda - \lambda_0)\cos\theta, \tag{6.4}$$

$$y = \sqrt{2}\sin\theta. \tag{6.5}$$

但し、λ_0 は地図上の中心経度であり、θ は

$$2\theta + \sin 2\theta = \pi \sin\phi \tag{6.6}$$

で定義する[2] 。(経度 λ はファイル GALAXY.DAT から読み取った値を直接使用するのではない。適切な変換をしてから使用する。図 6.2、6.3 の x-軸を見て、良く考えよう。) 与えられた緯度 ϕ から上の式に基づいて θ を求めるには、先ず適当な推定値 θ を用意して、Newton-Raphson 法により、上の式を満たすように逐次修正して行けば良い。修正値 $\Delta\theta$ は、(6.2) 式を使って、

$$\Delta\theta = -(2\theta + \sin 2\theta - \pi\sin\phi)/2(1 + \cos 2\theta) \tag{6.7}$$

で求め、これを θ に加えたものを改めて θ として、修正値 $\Delta\theta$ が予め定めておいた限界値以下になるまで、上のプロセスを繰り返す。

[2] 各種の図法は以下の文献に詳しい。Snyder, J. P. 1987, "Map Projections —A Working Manual," *U.S. Geological Survey Professional Paper 1395*, (United States Goverment Printing Office, Washington).

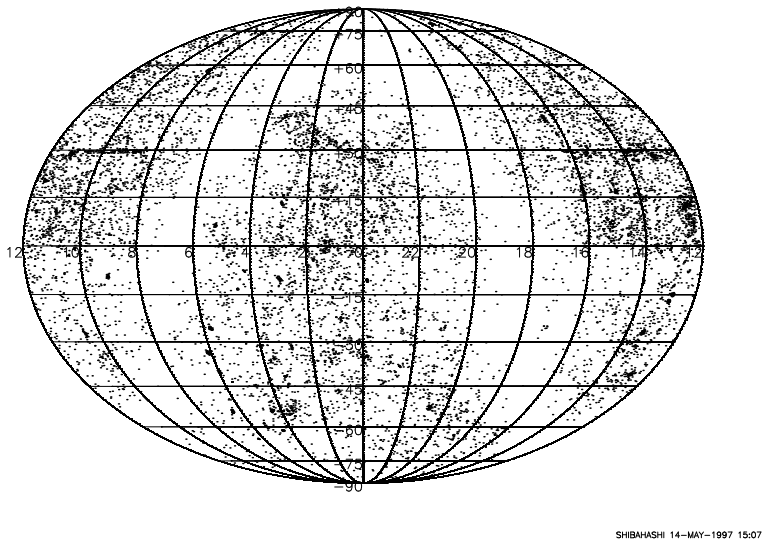

図 6.2: 銀河の天球上の分布。赤経、赤緯座標で図示。

赤経 (α)・赤緯 (δ) と銀経 (l)・銀緯 (b) の変換は、銀河の北極の赤経・赤緯を α_0 , δ_0 とし、

$$\Omega = 90^\circ + \alpha_0 \tag{6.8}$$

$$i = 90^\circ - \delta_0 \tag{6.9}$$

$$\alpha' \equiv \alpha - \Omega \tag{6.10}$$

として、

$$\cos b \cos l = \cos \delta \cos \alpha' \tag{6.11}$$

$$\sin b = \cos i \sin \delta - \sin i \cos \delta \sin \alpha' \tag{6.12}$$

$$\cos b \sin l = \sin i \sin \delta + \cos i \cos \delta \sin \alpha' \tag{6.13}$$

で与えられる。但し、$(\alpha_0, \delta_0) = (12^{\rm h}51^{\rm m}.4, +27^\circ 08'.0)$。

6.2　二分法

関数 $y = f(x)$ が区間 $[a, b]$ で連続であるとき、$f(a)f(b) < 0$ ならば、その区間に解が少なくとも一つ存在する。いま、その区間に解が一つしか存在しない場合を考えることにする。すると、$f(a)f(b) < 0$ のとき、a と b の中点 $c = \frac{a+b}{2}$ に対し、

1. $f(a)f(c) < 0$ ならば、解の存在区間は $[a, c]$ に狭まるし、逆に

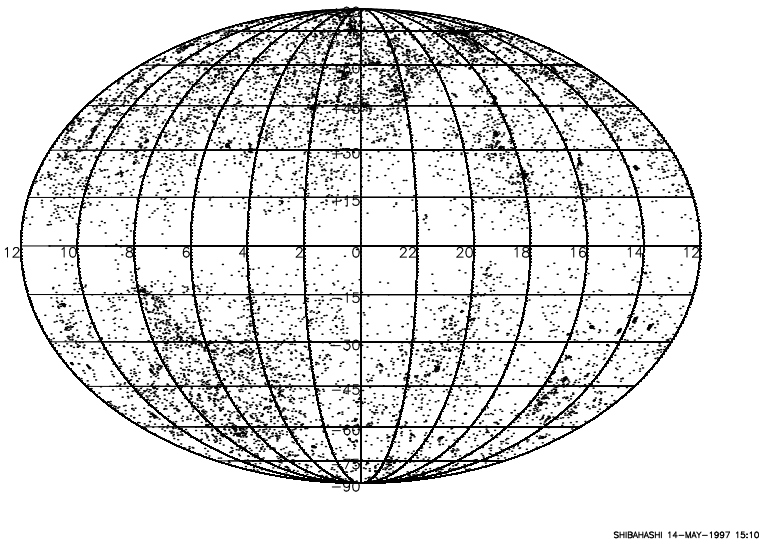

図 6.3: 銀河の天球上の分布。銀経、銀緯座標で図示。

2. $f(b)f(c) < 0$ ならば、解の存在区間は $[c, b]$ に狭まる。

そこで、狭まった解の存在区間を更に二分して、同様の判定を行ない、解の存在区間を更に狭める。以下、同様に、この操作を繰り返せば、解を得る事が出来る。収束判定条件としては、

1. $|b - a| < \varepsilon$
2. $|f(c)| < \varepsilon$

がある。

この方法では、収束解を得るのに必要な反復計算回数 n を、初期値 a、b を用いて、

$$\frac{|b-a|}{2^n} < \varepsilon \tag{6.14}$$

から、予め知る事が出来る。したがって、n 回の反復計算の後も、収束解が得られない場合には、プログラムに誤りがあると判断できる。

以下の外部関数 `rtbis` は、関数 $f(x)$ を `func` として、$f(x) = 0$ となる解を二分法で解くプログラム例である[3] 。

```
function rtbis(func,x1,x2,xacc)
implicit NONE
```

[3] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992, *Numerical Recipes in FORTRAN77* (*Second Edition*), (Cambridge University Press, Cambridge) から転載。http://cfatab.harvard.edu/nr/nronline.html でオンラインで読む事も出来る。

```
      Integer JMAX
      Real rtbis,x1,x2,xacc,func
      external func
      Parameter (JMAX=40)      !Maximum allowed number of bisections.
! Using bisection, find the root of a function 'func' known to lie
! betwee x1 and x2. The root, returned as 'rtbis', will be refined
! until its accuracy is +-'xacc'.
      Integer j
      Real dx,f,fmid,xmid
      fmid=func(x2)
      f=func(x1)
      if(f*fmid.ge.0.) pause 'root must be bracketed in rtbis'
      if(f.lt.0.)then          !Orient the search so that f>0 lies at x+dx.
        rtbis=x1
        dx=x2-x1
      else
        rtbis=x2
        dx=x1-x2
      endif
      do j=1,JMAX              !Bisection loop.
        dx=dx*.5
        xmid=rtbis+dx
        fmid=func(xmid)
        if(fmid.le.0.)rtbis=xmid
        if(abs(dx).lt.xacc .or. fmid.eq.0.) return
      end do
      pause 'too many bisections in rtbis'
      end
```

6.3　Newton-Raphson 法と二分法のハイブリッド法

Newton-Raphson 法では、推定値 x_n の真の解 x からのずれを ε_n とすると、(6.2) 式より

$$\varepsilon_{n+1} = \varepsilon_n - \frac{f(x_n)}{f'(x_n)} \tag{6.15}$$

だが、

$$f(x+\varepsilon) = f(x) + \varepsilon f'(x) + \varepsilon^2 \frac{f''(x)}{2} + \cdots, \tag{6.16}$$

$$f'(x+\varepsilon) = f'(x) + \varepsilon f''(x) + \cdots \tag{6.17}$$

だから、

$$\varepsilon_{n+1} = -\varepsilon_n^2 \frac{f''(x_n)}{2f'(x_n)} \tag{6.18}$$

となり、良い推定値さえ得られていれば、収束が速いことが判る。しかし、解を捜す区間に変曲点があると、近傍の解には収束しないか、収束しても一般には収束は速くない。そこで、二分法と組み合わせて、確実に、しかも収束の速いプログラムを用意すると良い。以下のプログラムは、そんな例である[4] 。

```
      function rtsafe(funcd,x1,x2,xacc)
      implicit NONE
      Integer MAXIT
      Real rtsafe,x1,x2,xacc
      external funcd
      Parameter (MAXIT=100)   !Maximum allowed number of iterations.
! Using a combination of Newton-Raphson and bisection, find the root of a
! function bracketed between x1 and x2. The root, returned as the
! function value 'rtsafe', will be refined until its accuracy is known
! within +-'xacc'. 'funcd' is a user-supplied subroutine which returns
! both the function value and the first derivative of the function.
      Integer j
      Real df,dx,dxold,f,fh,fl,temp,xh,xl
      call funcd(x1,fl,df)
      call funcd(x2,fh,df)
      if((fl.gt.0..and.fh.gt.0.).or.(fl.lt.0..and.fh.lt.0.))
     * pause 'root must be bracketed in rtsafe'
      if(fl.eq.0.)then
        rtsafe=x1
        return
      else if(fh.eq.0.)then
        rtsafe=x2
        return
      else if(fl.lt.0.)then   !Orient the search so that f(x1)<0.
        xl=x1
        xh=x2
      else
        xh=x1
        xl=x2
```

[4] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992, *Numerical Recipes in FORTRAN77* (*Second Edition*), (Cambridge University Press, Cambridge) から転載。http://cfatab.harvard.edu/nr/nronline.html でオンラインで読む事も出来る。

```
      endif
      rtsafe=.5*(x1+x2)       !Initialize the guess for root,
      dxold=abs(x2-x1)        !the "stepsize before last,"
      dx=dxold                !and the last step.
      call funcd(rtsafe,f,df)
      do j=1,MAXIT            !Loop over allowed iterations.
        if(((rtsafe-xh)*df-f)*((rtsafe-xl)*df-f).ge.0. !Bisect if Newton out of
     *     .or. abs(2.*f).gt.abs(dxold*df) ) then      !range, or not decreasing
          dxold=dx                                     !fast enough.
          dx=0.5*(xh-xl)
          rtsafe=xl+dx
          if(xl.eq.rtsafe)return        !Change in root is negligible.
        else                            !Newton step acceptable. Take it.
          dxold=dx
          dx=f/df
          temp=rtsafe
          rtsafe=rtsafe-dx
          if(temp.eq.rtsafe)return
        endif
        if(abs(dx).lt.xacc) return      !Convergence criterion.
        call funcd(rtsafe,f,df)         !The one new function evaluation per
        if(f.lt.0.) then                !iteration.
          xl=rtsafe                     !Maintain the bracket on the root.
        else
          xh=rtsafe
        endif
      end do
      pause 'rtsafe exceeding maximum iterations'
      return
      END
```

第 7 章　常微分方程式の数値解法：初期値問題

常微分方程式

$$\frac{dy_i}{dx} = f_i[x, y_i(x)] \tag{7.1}$$

$i, j = 1, 2, 3, ..., n$ を、初期値問題として数値的に解く事を考えよう。$x = x_0$ において初期値 $y_i(x_0) = y_{i,0}$ が与えられた時、$x = x_0 + h$ での y_i の値 $y_i(x_0 + h)$ が求められれば、次のステップでは、その (x, y_i) を新たな初期値と考えれば良いから、問題は $x = x_0 + h$ での y_i の値を数値的に求める事である。

7.1　Euler 法と Runge-Kutta 法

7.1.1　Euler 法

簡単のために、一元の微分方程式の場合を考えよう：

$$\frac{dy}{dx} = f(x, y). \tag{7.2}$$

$x = x_0$ の周りでの Taylor 展開を利用する事はすぐに思いつくだろう：

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{h^2}{2!}y''(x_0) + ... \tag{7.3}$$

ここで、$y'(x_0) = f(x_0, y_0)$ だから、$O(h)$ までは、Taylor 展開で求める事が出来る。即ち、

$$y(x_0 + h) = y(x_0) + hf(x_0, y_0) + O(h^2) \tag{7.4}$$

誤差は

$$\frac{h^2}{2!}y''(x_0) = \frac{h^2}{2}\left(\frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y}\right) \tag{7.5}$$

のオーダーである。この方法による計算は、Euler 法と呼ばれる。

7.1.2　Runge-Kutta 法

x_0 と $x_0 + h$ の間の、中間の x における導関数、$f(x, y)$ の関数値を何回か計算し、それらの適当な平均を用いる事によって、高次の Taylor 展開を達成しようというのが、Runge-Kutta 法である。

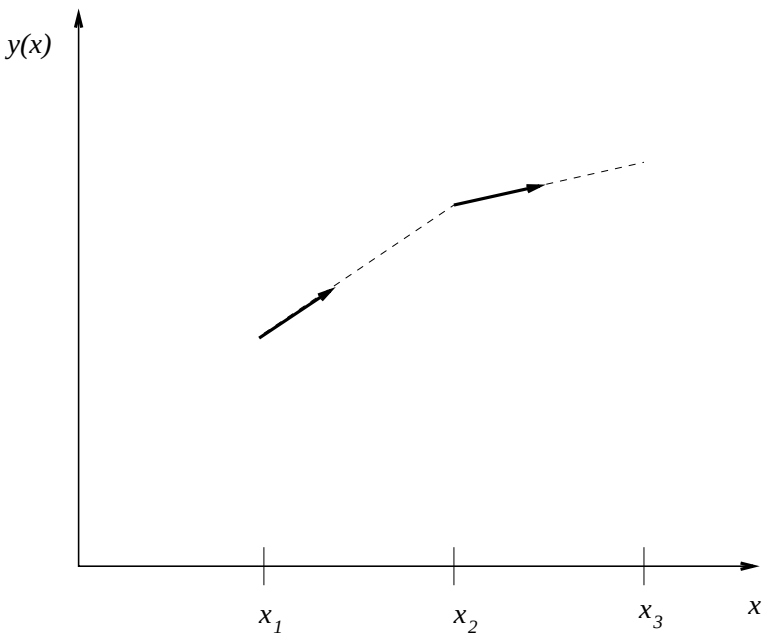

図 7.1: Euler 法による積分の概念図。

次の例で説明しよう。

$$y(x_0+h) = y(x_0) + \mu_1 h f(x_0, y_0) + \mu_2 h f(x_0+\alpha h, y_0+\beta k) \tag{7.6}$$

と表す。右辺第二項は、点 (x_0, y_0) での傾き $h \cdot f(x_0, y_0)$ に μ_1 の重みを掛けたものであり、第三項は、x_0 と x_0+h の間のある点 $(x_0+\alpha h, y_0+\beta k)$ での傾き $h \cdot f(x_0+\alpha h, y_0+\beta k)$ に μ_2 の重みを掛けたものである。即ち、第二項と第三項とで、x_0 と x_0+h の間の、中間の x における導関数、$f(x,y)$ の関数値の適当な平均を用いて、$y(x_0)$ から $y(x_0+h)$ を求める事を考える。ここで、

$$f(x_0+\alpha h, y_0+\beta k) = f_0 + \alpha h \left(\frac{\partial f}{\partial x}\right)_0 + \beta k \left(\frac{\partial f}{\partial y}\right)_0 \tag{7.7}$$

だから、(7.6) 式は、

$$\begin{aligned} y(x_0+h) &= y(x_0) + \mu_1 h f_0 + \mu_2 h \left[f_0 + \alpha h \left(\frac{\partial f}{\partial x}\right)_0 + \beta k \left(\frac{\partial f}{\partial y}\right)_0 \right] \\ &= y(x_0) + (\mu_1+\mu_2) h f_0 + \alpha\mu_2 h^2 \left(\frac{\partial f}{\partial x}\right)_0 + \beta\mu_2 hk \left(\frac{\partial f}{\partial y}\right)_0 \end{aligned} \tag{7.8}$$

となる。この式と (7.3) 式とを比べれば、

$$\mu_1 + \mu_2 = 1 \tag{7.9}$$

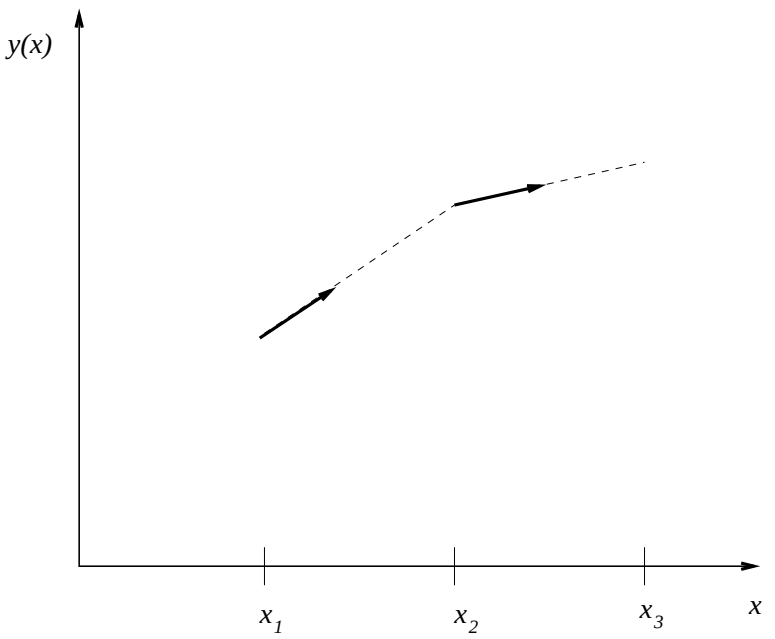

図 7.2: 修正 Euler 法による積分の概念図。

$$\alpha\mu_2 = \beta\mu_2 = 1/2 \tag{7.10}$$

$$k = hf_0 \tag{7.11}$$

ならば、(7.9) 式は、Taylor 展開に $O(h^2)$ の精度まで一致する事が判る。α , β や重み μ_1 , μ_2 はユニークではない。

特に、$\alpha = \beta = 1$, $\mu_1 = \mu_2 = 1/2$ の時は、

$$y(x_0 + h) = y(x_0) + \frac{1}{2}[hf_0 + hf_0(x_0 + h, y_0 + hf_0)] \tag{7.12}$$

で、これを改良 Euler 公式、或いは、Heun の二次公式と呼ぶ。$\alpha = \beta = 1/2$, $\mu_1 = 0$, $\mu_2 = 1$ の場合は、

$$y(x_0 + h) = y(x_0) + hf(x_0 + h/2, y_0 + h/2f_0) \tag{7.13}$$

で、これは修正 Euler 公式と呼ばれる。改良 Euler 公式も修正 Euler 公式も、幾何学的に考えれば、意味は明らかであろう。

ここでは証明は略すが、

$$y(x_0 + h) = y(x_0) + \frac{1}{6}\{k_1 + 2(k_2 + k_3) + k_4\} \tag{7.14}$$

但し、

$$k_1 = hf(x_0, y_0) \tag{7.15}$$

$$k_2 = hf(x_0 + h/2, y_0 + k_1/2) \tag{7.16}$$

$$k_3 = hf(x_0 + h/2, y_0 + k_2/2) \tag{7.17}$$

$$k_4 = hf(x_0 + h, y_0 + k_3) \tag{7.18}$$

とすると、(7.14) 式は、Taylor 展開の四次の精度までを持つ。通常、(7.14) 式を Runge-Kutta の公式と呼ぶ。

Runge-Kutta の公式のサブルーチンプログラムの例を下に掲げる。このプログラムでは、n 元連立一次方程式で、$x_0, y_0, (dy/dx)_0, h$ は与えられているとして、$y(x_0 + h)$ を計算する。中間点での導関数を与えるプログラムは、別のサブルーチンで与えられているとして、それを DERIVS と呼んでいる。この名前は仮の名前だから、引数として扱っている。メインプログラムでは、external 文を使って、導関数を与えるプログラムを指定しておくだけで良い。

```
      subroutine RK4(y,dydx,n,x,h,yout,DERIVS)

      implicit NONE
      real*8 y,dydx,yout,yt,dyt,dym,
     &     x,xh,h,hh,h6
      integer nmax,n,i
      parameter (nmax=2)
      dimension y(n),dydx(n),yout(n),yt(nmax),dyt(nmax),dym(nmax)

      hh=h*0.5
      h6=h/6.
      xh=x+hh
      Do i=1,n
        yt(i)=y(i)+hh*dydx(i)
      End do
      call DERIVS(xh,yt,dyt)
      Do i=1,n
        yt(i)=y(i)+hh*dyt(i)
      End do
      call DERIVS(xh,yt,dym)
      Do i=1,n
        yt(i)=y(i)+h*dym(i)
        dym(i)=dyt(i)+dym(i)
      End do
      call DERIVS(x+h,yt,dyt)
      Do i=1,n
```

```
        yout(i)=y(i)+h6*(dydx(i)+dyt(i)+2.*dym(i))
      End do

      Return
      End
```

7.1.3 例題

Runge-Kutta 法と Euler 法と修正 Euler 法を使って、次の連立一階常微分方程式を解き、誤差を評価せよ。

$$\frac{d^2y}{dx^2} + y = 0. \tag{7.19}$$

但し、初期値は $x = 0$ で $y = 1$, $dy/dx = 0$ とし、区間 $x = [0, \pi/2]$ について解け。尚、先に挙げたサブルーチン RK4 の使用例として、Runge-Kutta 法でのサンプルプログラムを下に掲げる。

```
      program main

      implicit NONE
      real*8 y,dydx,x,h,yout,pi
      integer i
      parameter(i=2,h=0.001,pi=3.141593)
      dimension y(i),dydx(i),yout(i)
      external diff_eq

      open(unit=1,file='RK_sample1.dat',status='new')
c initial value
      x    = 0.
      y(1) = 1.
      y(2) = 0.
c do loop
      do while (x.le.pi/2.)
        write(1,100) x,y(1),y(2)
        call diff_eq(x,y,dydx)
        call rk4(y,dydx,2,x,h,yout,diff_eq)
        y(1) = yout(1)
        y(2) = yout(2)
        x    = x + h
      end do
100   format(11x,f6.3,1p2e12.4)
```

```
      close(1)
      stop
      end


      subroutine diff_eq(x,y,dydx)

      implicit NONE
      real*8 y,dydx,x,n
      dimension y(2),dydx(2)

      dydx(1)=y(2)
      dydx(2)=-y(1)
      return
      end
```

7.1.4　参考文献

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. 1992, *Numerical Recipes in FORTRAN 77* (*Second Edition*), (Cambridge University Press, Cambridge).

http://cfatab.harvard.edu/nr/nronline.html

7.2　ポリトロープガス球とLane-Emden方程式

常微分方程式の初期値問題の例として、ポリトロープガス球のLane-Emden方程式を解いてみよう。

7.2.1　恒星の内部構造の基本

恒星の内部構造の釣り合いを考えよう。

- 静水圧平衡
  - 星は自分自身の重力による収縮しようする力とガス圧力による対抗する力とがバランスして、力学平衡を保っている。r を中心からの距離、P をそこでの圧力、M_r をその半径より内側に含まれる質量とすると、G を重力定数として、この力学平衡は、次のように表す事が出来る。

$$\frac{dP}{dr} = -\frac{GM_r\rho}{r^2} \tag{7.20}$$

- 連続の式
  - 半径 r ,厚み dr の球殻に含まれる質量 dM_r は、そこでの密度を ρ とすると、$dM_r = 4\pi r^2 \rho dr$ であるから、次の連続の式を得る。

$$\frac{dM_r}{dr} = 4\pi r^2 \rho \tag{7.21}$$

- 状態方程式
  - ガスの圧力 P と密度 ρ は、ガスの成分、温度 T によって決まっている。即ち、

$$P = P(\rho, T, X_i) \tag{7.22}$$

    ただし、X_i は、ガスに含まれる i 種の元素の質量比とする。

- エネルギー輸達
  - 星は、自身の持つ熱エネルギーを外に放出している系である。温度 T は、エネルギーの輸達の効率によって決められる。エネルギーの輸達の取りえる形態には、輻射と対流と伝導があるが、通常の星の内部では伝導は効率が悪く、輻射か対流によっている。

$$\frac{dT}{dr} = \begin{cases} \frac{-3}{4ac}\frac{\kappa\rho}{T^3}\frac{L_r}{4\pi r^2} & \text{if radiative} \\ \left(1 - \frac{1}{\gamma}\right)\frac{T}{P}\frac{dP}{dr} & \text{if convective} \end{cases} \tag{7.23}$$

    ただし、ここで、a は輻射定数、c は光速、κ は吸収係数、L_r は半径 r での光度、γ は断熱指数を表す。

- 吸収係数
  - 吸収係数は、ガスの成分と状態による。

$$\kappa = \kappa(\rho, T, X_i) \tag{7.24}$$

- エネルギー保存式
  - 半径 r、厚み dr の球殻で発生するエネルギーを、$dL_r = 4\pi r^2 \rho\varepsilon dr$ と書くと、

$$\frac{dL_r}{dr} = 4\pi r^2 \rho\varepsilon \tag{7.25}$$

- 核反応生成率
  - ε は、エネルギー発生率を表す。星のエネルギー源は核反応である。これは、ガスの成分と状態によって決まる。

$$\varepsilon = \varepsilon(\rho, T, X_i) \tag{7.26}$$

7.2.2 Lane-Emden 方程式

状態方程式

$$P = P(\rho, T, X_i) \tag{7.27}$$

において、

$$P = K\rho^{1+\frac{1}{N}} \tag{7.28}$$

として解く事を考える。これを指数 N のポリトロープと呼ぶ。すると、上の基礎方程式で、力学平衡を表す静水圧平衡の式と連続の式の二本の式がエネルギーの式と分離する。即ち静水圧平衡の式より

$$M_r = -\frac{r^2}{G}\frac{1}{\rho}\frac{dP}{dr} \tag{7.29}$$

を得、これを連続の式に代入すれば

$$\frac{1}{r^2}\frac{d}{dr}\left(\frac{r^2}{\rho}\frac{dP}{dr}\right) = -4\pi G\rho \tag{7.30}$$

を得る。ここで

$$\rho = \rho_{\rm c}\theta^N \tag{7.31}$$

($\rho_{\rm c}$ は 星の中心密度) として、先の状態方程式を仮定すると

$$\left[\frac{(N+1)K}{4\pi G}\rho_{\rm c}^{\frac{1}{N}-1}\right]\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\theta}{dr}\right) = -\theta^N \tag{7.32}$$

を得る。

$$\alpha \equiv \left[\frac{(N+1)K}{4\pi G}\rho_{\rm c}^{\frac{1}{N}-1}\right]^{1/2} \tag{7.33}$$

で α を定義し、

$$r \equiv \alpha\xi \tag{7.34}$$

で 無次元変数 ξ を導入すると

$$\frac{1}{\xi^2}\frac{d}{d\xi}\left(\xi^2\frac{d\theta}{d\xi}\right) = -\theta^N \tag{7.35}$$

を得る。これを 指数 N の Lane-Emden equation と呼ぶ。

星の中心 $\xi = 0$ で

$$\theta = 1 \tag{7.36}$$

$$\frac{d\theta}{d\xi} = 0 \tag{7.37}$$

である。また $\theta = 0$ になる点が星の表面に相当する。

7.2.3 Lane-Emden 方程式の解の中心近傍での振舞

Lane-Emden 方程式 (7.35) は、$\xi = 0$ が、確定特異点になっている。この様な場合、特異点の近傍での解の Laurent 級数展開を考える必要がある。いま、$\xi = 0$ のまわりの形式解を

$$\theta(\xi) = 1 + a_2\xi^2 + a_4\xi^4 + \cdots \tag{7.38}$$

の形に仮定して、(7.35) 式の両辺に代入すれば、左辺は、

$$\mathrm{LHS} = 6a_2 + 20a_4\xi^2 + \cdots \tag{7.39}$$

一方、右辺は、

$$\mathrm{RHS} = -(1 + Na_2\xi^2 + \cdots) \tag{7.40}$$

となる。 両辺の ξ^0 と ξ^2 の係数を比較して、$a_2 = -1/6$, $a_4 = N/120$ を得る。よって、中心近傍での解は、

$$\theta(\xi) \simeq 1 - \frac{1}{6}\xi^2 + \frac{N}{120}\xi^4 + \cdots \tag{7.41}$$

と表される。

7.2.4 積分の方法

例えば、$y_1 = \theta$, $y_2 = \xi^2 d\theta/d\xi$ とおくと、(7.35) 式は次の二元連立一階微分方程式となる：

$$\frac{dy_1}{d\xi} = \xi^{-2} y_2, \tag{7.42}$$

$$\frac{dy_2}{d\xi} = -\xi^2 y_1^N. \tag{7.43}$$

この方程式を、$\xi = 0$ から外側に向かって、Runge-Kutta 法で積分していけば良い。中心は、明らかに確定特異点だから、初期値は、$\xi = 0$ ではなく、中心からわずかにずれた $\xi = h$ で設定する。そこでの y_i の値は、上で求めた展開から得られる

$$y_1 \simeq 1 - \frac{1}{6}\xi^2 + \frac{N}{120}\xi^4 + \cdots, \tag{7.44}$$

$$y_2 \simeq -\frac{1}{3}\xi^3 + \frac{N}{30}\xi^5 + \cdots \tag{7.45}$$

を使って求める。積分を進めて、$y_1 = 0$ になる点が表面である。その ξ を ξ_1 とすると、星の半径は

$$\begin{aligned} R &= \alpha\xi_1 \\ &= \left[\frac{(N+1)K}{4\pi G}\right]^{1/2} \rho_{\mathrm{c}}^{(1-N)/2N}\xi_1 \end{aligned} \tag{7.46}$$

となる。中心から ξ より内側に含まれる質量は

$$\begin{aligned} M(\xi) &= \int_0^{\alpha\xi} 4\pi\rho r^2 dr \\ &= 4\pi\alpha^3\rho_{\rm c}\int_0^{\xi}\xi^2\theta^N d\xi, \end{aligned} \tag{7.47}$$

或いは、(7.35) 式を使って書き直すと、

$$\begin{aligned} M(\xi) &= -4\pi\alpha^3\rho_{\rm c}\int_0^{\xi}\frac{d}{d\xi}\left(\xi^2\frac{d\theta}{d\xi}\right)d\xi \\ &= -4\pi\alpha^3\rho_{\rm c}\xi^2\frac{d\theta}{d\xi} \end{aligned} \tag{7.48}$$

で与えられる。従って、星の全質量は

$$M = -4\pi\left[\frac{(N+1)K}{4\pi G}\right]^{3/2}\rho_{\rm c}^{(3-N)/2N}\left(\xi^2\frac{d\theta}{d\xi}\right)_{\xi=\xi_1} \tag{7.49}$$

となる。

7.2.5　問題

1. 指数 $N = 1.5$、$N = 3$、及び $N = 4$ の場合について、Lane-Emden equation を Runge-Kutta 法を用いて数値的に解け。

2. 太陽が $N = 3$ のポリトロープで近似できるとして、その内部構造を求めよ。但し太陽半径 $R_\odot = 6.96\times10^5$ km、太陽質量 $M_\odot = 1.99\times10^{30}$ kg とし、内部は完全気体の状態方程式

$$P_{\rm gas} = NkT \tag{7.50}$$

に従うものとする。ここで N は単位体積当たりの全粒子数、k はボルツマン定数。今、水素、ヘリウム、それ以外の元素が質量比 (X, Y, X_i) である完全電離理想気体を考える。水素原子の質量を $m_{\rm H}$ で表わすとすると、水素については核の数密度は $\rho X/m_{\rm H}$、電子の数密度は同じく $\rho X/m_{\rm H}$ 、ヘリウムについては核の数密度は $\rho Y/(4m_{\rm H})$、電子の数密度は $\rho Y/(2m_{\rm H})$、それら以外の元素については、それぞれの元素の atomic weight を A_i、atomic number を Z_i とすると、核の数密度は $\rho X_i/(A_i m_{\rm H})$、電子の数密度は $\rho X_i Z_i/(A_i m_{\rm H})$ だから、全粒子数 N は

$$N = \frac{\rho}{m_{\rm H}}\left(2X + \frac{3}{4}Y + \sum_i\frac{Z_i+1}{A_i}X_i\right) \tag{7.51}$$

となる。重元素については $(Z_i+1)/A_i \sim 1/2$ だから、水素とヘリウム以外の元素の質量比を Z $(Z = 1-X-Y = \sum_i X_i)$ と表わす事にすれば、

$$N = \frac{\rho}{\mu m_{\rm H}} \tag{7.52}$$

であり、ガス圧は

$$P_{\rm gas} = \frac{\rho}{\mu m_{\rm H}} kT \tag{7.53}$$

$$= \frac{\mathcal{R}}{\mu}\rho T \tag{7.54}$$

ここに μ は

$$\mu^{-1} = 2X + \frac{3}{4}Y + \frac{1}{2}Z \tag{7.55}$$

で、平均分子量と呼ぶ。また $\mathcal{R} \equiv k/m_{\rm H} = 8.314510\times10^3\ {\rm J\cdot kg^{-1}\cdot K^{-1}}$。($k = 1.380658\times10^{-23}$ ${\rm J\cdot K^{-1}}$、$m_{\rm H} = 1.672623\times10^{-27}\,{\rm kg}$)。太陽の水素、ヘリウム、金属の質量比はそれぞれ $X = 0.75$、$Y = 0.23$、$Z = 0.02$ とする。

3. 白色矮星の内部では、電子は縮退し、その縮退圧が星を支えるようになる。電子が完全に縮退した場合、分布関数は

$$n_{\rm e}(p)dp = \begin{cases} \frac{2}{h^3}4\pi p^2 dp & \text{for } p < p_0 \\ 0 & \text{for } p > p_0 \end{cases} \tag{7.56}$$

全電子数 $n_{\rm e}$ は

$$n_{\rm e} = \int_0^{p_0} n_{\rm e}(p)dp = \frac{8\pi}{3h^3}p_0^3 \tag{7.57}$$

だから、与えられた電子数に対しては p_0 は

$$p_0 = \left(\frac{3\pi^3}{8\pi}n_{\rm e}\right)^{1/3} \tag{7.58}$$

$p_0 \ll m_{\rm e}c^2 = 0.51{\rm MeV}$ であれば、$v = p/m_{\rm e}$ として、電子の縮退による圧力は

$$P = \frac{1}{3}\int_0^{p_0} p v_p n(p)dp$$

$$= \frac{1}{3}\int_0^{p_0} \frac{8\pi}{m_{\rm e}h^3}p^4 dp$$

$$= \frac{8\pi}{15 m_{\rm e}h^3}p_0^5$$

$$= \frac{h^2}{20m_{\rm e}}\left(\frac{3}{\pi}\right)^{2/3} n_{\rm e}^{5/3} \tag{7.59}$$

$$= \frac{h^2}{20m_{\rm e}}\left(\frac{3}{\pi}\right)^{2/3} N_0^{5/3}\left(\frac{\rho}{\mu_{\rm e}}\right)^{5/3}$$

$$= 1.004\times10^7\left(\frac{\rho}{\mu_{\rm e}}\right)^{5/3}\ {\rm Pa} \tag{7.60}$$

となり、$N = 1.5$ のポリトロープに相当する。ここで、$\mu_{\rm e}^{-1} = (X + \frac{1}{2}Y + \frac{1}{2}Z) = \frac{1}{2}(1+X)$。白色矮星の状態方程式を (7.60) 式とした時、その半径と質量はどれぐらいになるか。中心密度は $\rho_{\rm c} = 10^9{\rm kg\cdot m^{-3}}$ を目安とせよ。また、半径の質量依存性はどうなるか。

相対論的縮退の場合は、

$$\begin{aligned} P &= \frac{hc}{8}\left(\frac{3}{\pi}\right)^{1/3} N_0^{4/3}\left(\frac{\rho}{\mu_{\rm e}}\right)^{4/3} \\ &= 1.243\times 10^{10}\left(\frac{\rho}{\mu_{\rm e}}\right)^{4/3} \quad {\rm Pa} \end{aligned} \tag{7.61}$$

となる。状態方程式が (7.60) 式で表される程の高密度の白色矮星では、その半径と質量はどれぐらいになるか。同じく中心密度は $\rho_{\rm c} = 10^9 {\rm kg\cdot m^{-3}}$ を目安とせよ。質量が中心密度によらない事を導け。この事の意味する所を考察せよ。

7.2.6　参考文献

Hansen, C. J. and Kawaler, S. D. 1994, *Stellar Interiors*, (Springer-Verlag, New York).