数値計算法概論:No.9(ニュートン・ラフソン法)

1　非線形方程式の解法:ニュートン・ラフソン法

任意の関数 $f(x)$ について、$f(x)=0$ となる点 x を求めよう。図のように適当な初期値 x_0 において $f(x)$ に接線を引けば、接線の方程式は

$$y - f(x_0) = f'(x_0)(x - x_0) \tag{1}$$

であり、したがってこの接線と x 軸との交点 x_1 は $y=0$ とおいて

$$x_1 = x_0 - f(x_0)/f'(x_0) \tag{2}$$

で与えられる。

次に x_1 での $f(x)$ への接線と x 軸との交点を x_2 とする、という操作を繰り返すと、交点は $f(x)=0$ の解に近付く。i 番目の繰り返しでは、

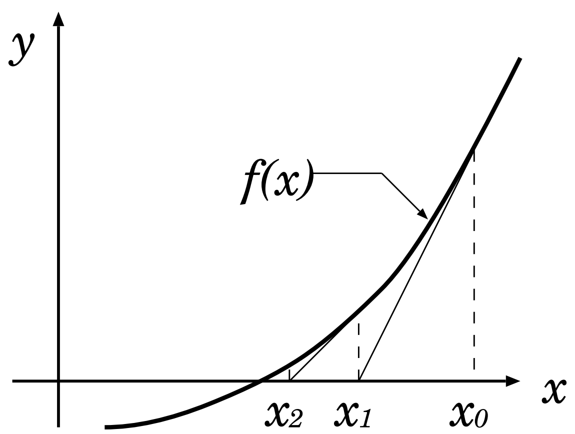
$$x_{i+1} = x_i - f(x_i)/f'(x_i) \tag{3}$$

になるので、適当な値 ϵ(収束半径) を決めておき、$|x_{i+1} - x_i| < \epsilon$ になったら、x_{i+1} を解とみなす。

ニュートン法の収束は、2 次収束

$$|x_{i+1} - x_\infty| < c|x_i - x_\infty|^2 \tag{4}$$

で、真の値に近付くと極めて早い。大体、i が 1 増えると有効桁数が倍になる。



2 Fortran 文法

2.1 do while 文

do ...end do 構文で、ある一定回数繰り返すのではなく、ある条件が満たされる時のみ、繰り返したい時がある。そういう場合に便利なのが do while 文である。

```
do while（論理式）
実行文
end do
```

論理式が真であるあいだ、do block を繰り返す。(注：これは、Fortran 90 の文法である。F77 では標準ではない)

例：newton.f を次のようにする。

```
c Newton-Raphson method
      implicit real*8 (a-h,o-z)
c
      epsilon = 1.0 e-10
      read(*,*) xnew
      xold = xnew + 0.1
      do while (abs(xnew-xold) > epsilon)
         xold=xnew
         xnew=xold-f(xold)/df(xold)
         write(*,1000) "result=",xnew
      end do
      write(*,1000) "result=",xnew
 1000 format(a,f20.16)
      end
```

これはニュートン法のメインルーチンであり、xnew と xold の差の絶対値が、epsilon より小さくなるまで実行を繰り返す。

2.2 分割コンパイル

これまでの積分や、ニュートン法では、メインルーチンは同じで、関数の部分だけを関数副プログラム、又はサブルーチン文として与え、変更することができた。これは別に 1 つのファイルとする必要はない。例えば、function.f を

```
      real*8 function f(x)
      implicit real*8 (a-h,o-z)
      integer n
c
      n=3
```

```
      f=x**n-n
      return
      end
c
      real*8 function df(x)
      implicit real*8 (a-h,o-z)
      integer n
c
      n=3
      df=n*x**(n-1)
      return
      end
```

として、

```
%frt newton.f function.f
```

としてコンパイルすると、$f(x) = x^n - n = 0$ (この場合 $n = 3$) の解、つまり $n^{1/n}$ が得られる。関数部分を変えると任意の関数についてニュートン法が実行できる。

2.3 指数部つきの表現

6.67×10^{-11} のような指数のついた数値は、Fortran では

```
6.67e-11
```

で表す。倍精度で表すには、

```
epsilon=3.141592d-14
```

のようにする。

3 課題

ニュートン法について、自分で選んだ関数で試してみよ。また、収束性についても議論すること。