

数値計算法(2)

Numerical Analysis

防災・環境地盤工学研究室
村上 哲
murakami@mx.ibaraki.ac.jp

Numerical Analysis

先週のおさらい

■ 例題1を解いてみてください。

関数が次のように与えられている。

$$f(x) = x^2 - 2$$

この時、次式で定義される方程式に対し、ある適当な x_0 を数値として与え、求められた x を順次 x_0 に代入していくと、最終的に収束する x の値は？

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

□ x_0 の初期値を以下のようにしてみましょう。

- -2.0, -1.0, 1.0, 2.0

Numerical Analysis

■ 数値的に解を求めるとは...

関数

$$f(x) = x^2 - 2$$

において、 $f(x) = 0$ となる解を求めることを考える。

【まず準備！】

正解値を $x = x_0$ として、テーラー展開を施し、第1項までを考慮すると、近似的に、

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

となる。これより、

$$x = x_0 - \frac{f(x_0) - f(x)}{f'(x_0)} \xrightarrow{f(x)=0 \text{ なので}} x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Numerical Analysis

■ 数値的に解を求めるとは...

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} = x_0 - \frac{x_0^2 - 2}{2x_0}$$

初期値を $x_0 = 1.0$ として、 x を求める。	新しい値として $x_0 = 1.4182$ として、 x を求める。
$x = 1.0 - \frac{1.0^2 - 2}{2 \times 1.0} = 1.5$	$x = 1.4182 - \frac{1.4182^2 - 2}{2 \times 1.4182} = 1.4142$
新しい値として $x_0 = 1.5$ として、 x を求める。	新しい値として $x_0 = 1.4142$ として、 x を求める。
$x = 1.5 - \frac{1.5^2 - 2}{2 \times 1.5} = 1.3125$	$x = 1.4142 - \frac{1.4142^2 - 2}{2 \times 1.4142} = 1.4142$
新しい値として $x_0 = 1.3125$ として、 x を求める。	$x = x_0$ (入力値と出力値の差が十分小さい) となり、解を得る。
$x = 1.3125 - \frac{1.3125^2 - 2}{2 \times 1.3125} = 1.4182$	$x = 1.4142$

Newton法(反復法)による解法

Numerical Analysis

プログラミングすると

変数の説明

x :実数: 変数x
f :実数: 関数の値
df:実数: 導関数の値
i :整数: 繰り返し変数

```

program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,10
  f=x**2-2.0
  df=2.0*x
  x=x-f/df

!-----結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a
        
```

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$$f(x_0) = x_0^2 - 2$$

$$f'(x_0) = 2x_0$$

Numerical Analysis

プログラミングすると

フローチャート

```

program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,10
  f=x**2-2.0
  df=2.0*x
  x=x-f/df

!-----最終結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a
        
```

Numerical Analysis

プログラミングすると

実行結果 (初期値を50とした場合)

```

murakami@hcs[12]:./ex0101a
50
x= 25.02
x= 12.549968
x= 6.3546652
x= 3.3346972
x= 1.9672259
x= 1.491943
x= 1.4162384
x= 1.414215
x= 1.4142135
x= 1.4142135
murakami@hcs[13]:
        
```

初期値を-50とするとどうなるか?
初期値を??とするとどうなるか?

```

program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,10
  f=x**2-2.0
  df=2.0*x
  x=x-f/df

!-----最終結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a
        
```

Numerical Analysis

■ 解析的に解を求めること、数値的に解を求めること

関数 $f(x) = x^2 - 2$

において、 $f(x) = 0$ となる解を求めることを考える。

数学的知識を使って

$x = \pm\sqrt{2}$

厳密解

数学的知識に基づいた、反復計算を行って

$x = \pm 1.4142135$

近似解

Numerical Analysis

■ 問題

関数

$$f(x) = x^2 - 2$$

において、 $f(x) = 0$ となる解を求めることを考える。

関数

$$F(x) = \frac{1}{3}x^3 - 2x + 6$$

において、最小となる解を求めることを考える。

Numerical Analysis

■ どういうふうに求めているのか？考えてみましょう。

関数

$$f(x) = x^2 - 2$$

において、 $f(x) = 0$ となる解を求めることを考える。

【まず準備！】

正解値を $x = x_0$ として、テーラー展開を施し、第1項までを考慮すると、近似的に、

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

となる。これより、

$$x = x_0 - \frac{f(x_0) - f(x)}{f'(x_0)} \xrightarrow{f(x)=0 \text{ なので}} x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Numerical Analysis

■ どういうふうに求めているのか？考えてみましょう。

正解値を $x = x_0$ として、テーラー展開を施し、第1項までを考慮すると、近似的に、

$$f(x) = f(x_0) + f'(x_0)(x - x_0)$$

となる。これより、

$$x = x_0 - \frac{f(x_0) - f(x)}{f'(x_0)} \xrightarrow{f(x)=0 \text{ なので}} x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

求められるxは、

$$f'(x_0)(x - x_0) + f(x_0) = 0$$

の解であるので、関数zを

$$z = f'(x_0)(x - x_0) + f(x_0)$$

とおくと、関数zのx軸との交点を求めていることになる。

Numerical Analysis

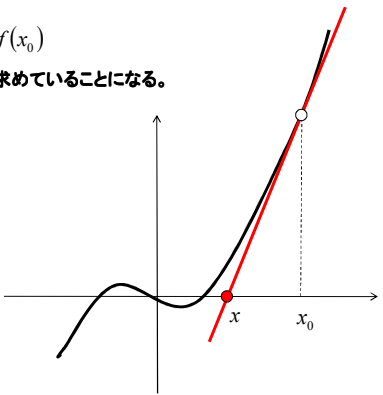
■ どういうふうに求めているのか？考えてみましょう。

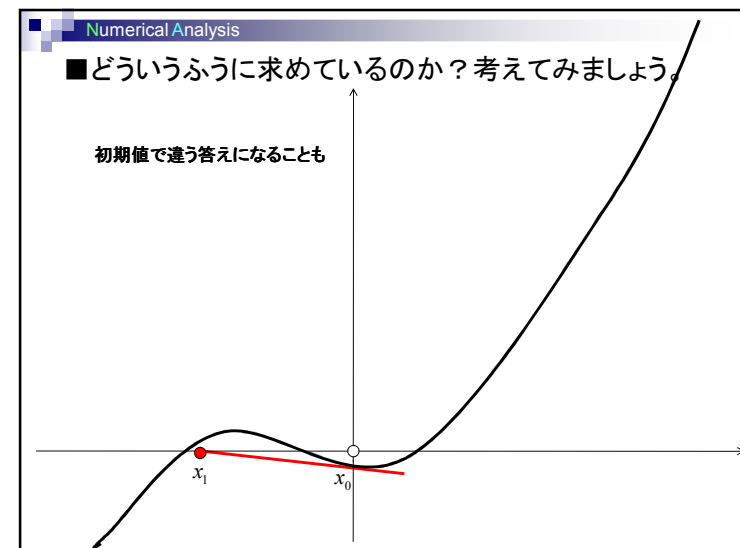
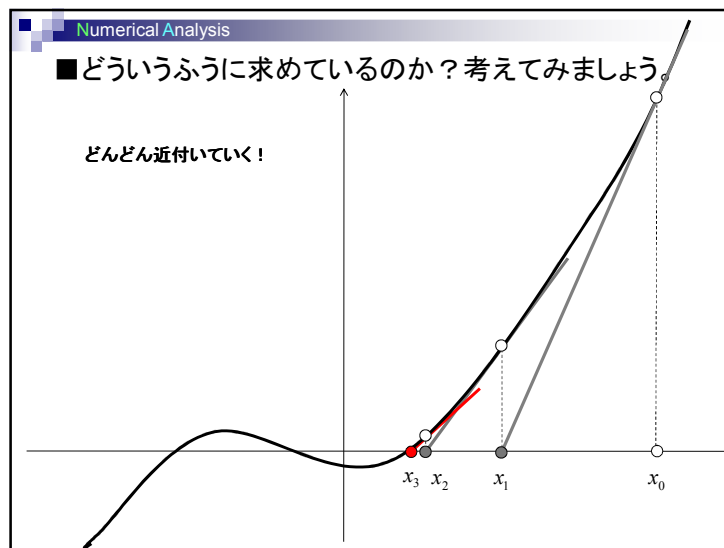
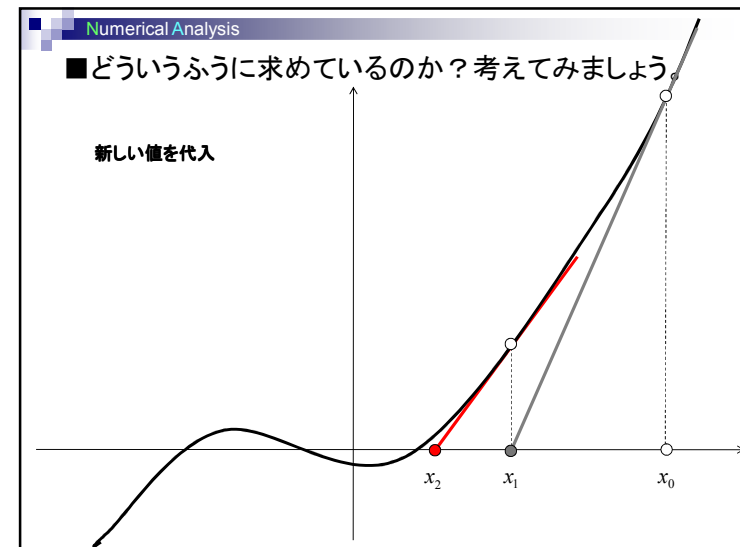
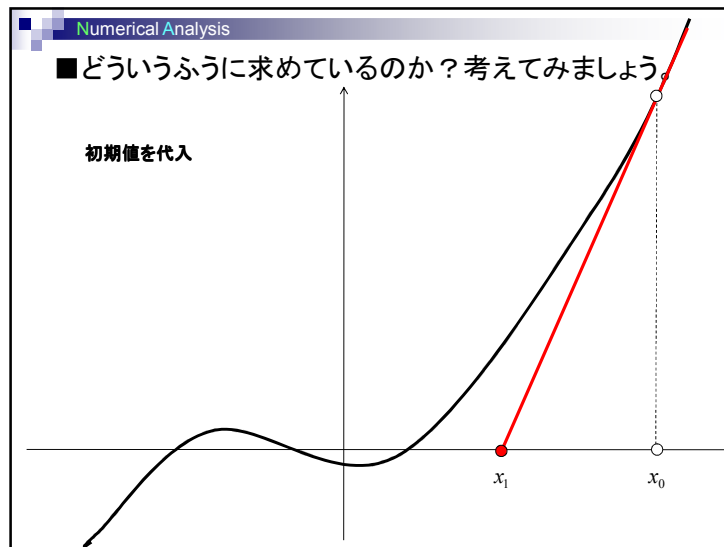
関数zを

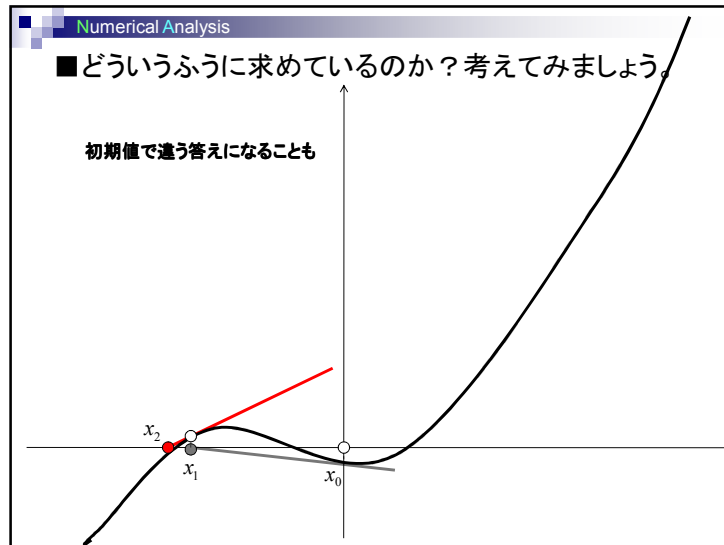
$$z = f'(x_0)(x - x_0) + f(x_0)$$

とおくと、関数zのx軸との交点を求めていることになる。

$x_0 \leftarrow x$







Numerical Analysis

実行結果

実行結果（初期値を50とした場合）

```
murakami@hcs[12]:./ex0101a
50
x= 25.02
x= 12.549968
x= 6.3546652
x= 3.3346972
x= 1.9672259
x= 1.491943
x= 1.4162384
x= 1.414215
x= 1.4142135
x= 1.4142135
murakami@hcs[13]:
```

```
program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

! -----初期値データの入力-----
read(*,*) x

! -----反復計算-----
do i=1,10

    f=x**2-2.0
    df=2.0*x
    x=x-f/df

! -----結果の出力-----
write(*,*) 'x=', x

end do

end program ex0101a
```

Numerical Analysis

実行結果

実行結果（初期値を50とした場合）

```
murakami@hcs[12]:./ex0101a
50
x= 25.02
x= 12.549968
x= 6.3546652
x= 3.3346972
x= 1.9672259
x= 1.491943
x= 1.4162384
x= 1.414215
x= 1.4142135
x= 1.4142135
murakami@hcs[13]:
```

収束！

```
program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

! -----初期値データの入力-----
read(*,*) x

! -----反復計算-----
do i=1,10

    f=x**2-2.0
    df=2.0*x
    x=x-f/df

! -----結果の出力-----
write(*,*) 'x=', x

end do

end program ex0101a
```

Numerical Analysis

実行結果

実行結果（初期値を100とした場合）

```
murakami@hcs[12]:./ex0101a
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
murakami@hcs[13]: ./ex0101a
```

収束していない！

```
program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

! -----初期値データの入力-----
read(*,*) x

! -----反復計算-----
do i=1,10

    f=x**2-2.0
    df=2.0*x
    x=x-f/df

! -----結果の出力-----
write(*,*) 'x=', x

end do

end program ex0101a
```

Numerical Analysis

100回繰り返すにはどこを変更するか？

実行結果（初期値を100とした場合）

```

murakami@hcs[12]:./ex0101a
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
murakami@hcs[13]:./ex0101a

```

収束していない！

```

program ex0101a
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,10
    f=x**2-2.0
    df=2.0*x
    x=x-f/df
!-----結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a

```

Numerical Analysis

100回繰り返すにはどこを変更するか？

実行結果（初期値を100とした場合）

```

murakami@hcs[12]:./ex0101b
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
murakami@hcs[13]:

```

```

program ex0101b
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,100
    f=x**2-2.0
    df=2.0*x
    x=x-f/df
!-----結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a

```

Numerical Analysis

100回繰り返すにはどこを変更するか？

実行結果（初期値を100とした場合）

```

murakami@hcs[12]:./ex0101b
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
murakami@hcs[13]:

```

```

program ex0101b
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,100
    f=x**2-2.0
    df=2.0*x
    x=x-f/df
!-----結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a

```

Numerical Analysis

100回繰り返すが収束したら途中でやめるには？

実行結果（初期値を100とした場合）

```

murakami@hcs[12]:./ex0101b
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
murakami@hcs[13]:

```

```

program ex0101b
!-----変数の宣言-----
real :: x, f, df
integer :: i

!-----初期値データの入力-----
read(*,*) x

!-----反復計算-----
do i=1,100
    f=x**2-2.0
    df=2.0*x
    x=x-f/df
!-----結果の出力-----
write(*,*) 'x=', x
end do
end program ex0101a

```

■ 収束判定

以下の条件となったとき計算を終了する。

$$EPS = \left| \frac{x - x_0}{x_0} \right| < 10^{-12}$$

収束判定の方法は問題に合わせて設定されることが多い。
必ずしも上記だけとは限らない。

```

program ex0101c
!-----変数の宣言-----
real :: x, x0, f, df, EPS
integer :: i
! -----初期値データの入力-----
read(*,*) x0
! -----反復計算-----
do i=1,100
    f=x0**2-2.0
    df=2.0*x0
    x=x0-f/df
! -----結果の出力-----
write(*,*) 'x=', x
    EPS=abs((x-x0)/x0)
    if(EPS<1.0E-12) then
        write(*,*) 'Finish!'
        exit
    else
        x0=x
    end if
end do
end program ex0101c
                
```

条件・判断(IF構文)

おさらい

IF構文

```

IF( e1 ) THEN
    (処理1)
ELSE
    (処理2)
END IF
                
```

```

graph TD
    A{条件 e1} -- Yes --> B[処理1]
    A -- No --> C[処理2]
                
```

- 条件を指定し、その条件の成立、不成立に応じて異なる処理を実行させたい場合に用いる。
- IF構文は、IF THEN文で始まり、END IF文で終わる構文である。

IF構文(特別な例)

```

IF( e1 ) THEN
    (処理1)
END IF

IF( e1 ) (処理1)
                
```

```

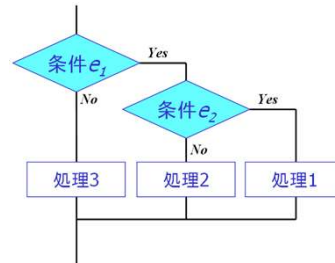
graph TD
    A{条件 e1} -- Yes --> B[処理1]
    A -- No --> C(( ))
    C --> B
                
```

- 条件を指定し、その条件の成立、不成立に応じて異なる処理を実行させたい場合に用いる。
- IF構文は、IF THEN文で始まり、END IF文で終わる構文である。

IF構文の2重構造

- IF構文の処理においても、さらにIF構文を用いることができる。

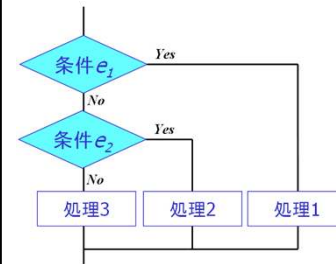
```
IF( e1 ) THEN
  IF( e2 ) THEN
    (処理1)
  ELSE
    (処理2)
  END IF
ELSE
  (処理3)
END IF
```



IF構文の2重構造

- IF構文の処理においても、さらにIF構文を用いることができる。

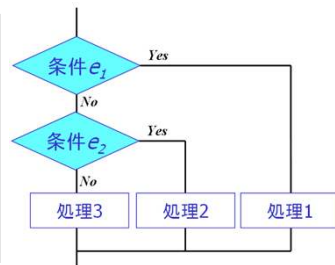
```
IF( e1 ) THEN
  (処理1)
ELSE
  IF( e2 ) THEN
    (処理2)
  ELSE
    (処理3)
  END IF
END IF
```



IF構文の2重構造

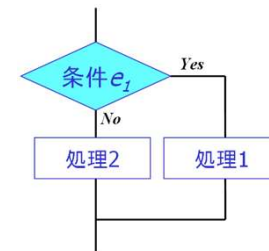
- IF構文の処理においても、さらにIF構文を用いることができる。

```
IF( e1 ) THEN
  (処理1)
ELSE IF( e2 ) THEN
  (処理2)
ELSE
  (処理3)
END IF
```



IF構文、IF文の条件

- IF構文、IF文の条件は、関係式または理論式で記述する。
- 関係式・理論式のとり値は、“真(正しい)”か“偽(正しくない)”の2つ
 - 関係式・理論式の値が真 → 条件が成立
 - 関係式・理論式の値が偽 → 条件が不成立



関係組込み演算子

演算子	記号	意味
.LT. または <	<	小さい (L ess T han)
.LE. または <=	≤	以下 (L ess than or E qual)
.EQ. または ==	=	等しい (E Qual to)
.NE. または /=	≠	等しくない (N ot E qual to)
.GT. または >	>	大きい (G reater T han)
.GE. または >=	≥	以上 (G reater than or E qual)

理論組込み演算子

演算子	記号	意味
.AND.	∩	・・・かつ・・・ (論理積)
.OR.	∪	・・・または・・・ (論理和)
.NOT.	∼	・・・でない (論理否定)

収束判定

以下の条件となったとき計算を終了する。

$$EPS = \left| \frac{x - x_0}{x_0} \right| < 10^{-12}$$

収束判定の方法は問題に合わせて設定されることが多い。
必ずしも上記だけとは限らない。

```
program ex0101c
!-----変数の宣言-----
real :: x, x0, f, df, EPS
integer :: i
!-----初期値データの入力-----
read(*,*) x0
!-----反復計算-----
do i=1,100
    f=x0**2-2.0
    df=2.0*x0
    x=x0-f/df
!-----結果の出力-----
write(*,*) 'x=', x
EPS=abs((x-x0)/x0)
if(EPS<1.0E-12) then
    write(*,*) 'Finish!'
    exit
else
    x0=x
end if
end do
end program ex0101c
```

収束判定

実行結果（初期値を100とした場合）

```
murakami@hcs[12]:./ex0101c
100
x= 50.01
x= 25.024994
x= 12.552457
x= 6.355894
x= 3.3352813
x= 1.9674655
x= 1.4920009
x= 1.4162413
x= 1.414215
x= 1.4142135
x= 1.4142135
Finish!
murakami@hcs[13]:
```

```
program ex0101c
!-----変数の宣言-----
real :: x, x0, f, df, EPS
integer :: i
!-----初期値データの入力-----
read(*,*) x0
!-----反復計算-----
do i=1,100
    f=x0**2-2.0
    df=2.0*x0
    x=x0-f/df
!-----結果の出力-----
write(*,*) 'x=', x
EPS=abs((x-x0)/x0)
if(EPS<1.0E-12) then
    write(*,*) 'Finish!'
    exit
else
    x0=x
end if
end do
end program ex0101c
```

```
Numerical Analysis
■ 収束判定

実行結果 (初期値を0.0000001とした場合)

murakami@hcs[12]:./ex0101c
0.0000001
x= 10000000.0
x= 5000000.0
x= 2500000.0
x= 1250000.0
x= 625000.0
x= 312500.0
x= 156250.0
x= 78125.0
x= 39062.5
x= 19531.25
x= 9765.625
x= 4882.8125
x= 2441.4064

x= 1220.7037
x= 610.35266
x= 305.17797
x= 152.59225
x= 76.30268
x= 38.164447
x= 19.108427
x= 9.606546
x= 4.9073686
x= 2.6574594
x= 1.705029
x= 1.4390148
x= 1.4144273
x= 1.4142135
x= 1.4142135
Finish!
murakami@hcs[13]:

program ex0101c
!-----変数の宣言-----
real :: x, x0, f, df, EPS
integer :: i
! -----初期値データの入力-----
read(*,*) x0
! -----反復計算-----
do i=1,100

    f=x0**2-2.0
    df=2.0*x0
    x=x0-f/df

! -----結果の出力-----
write(*,*) 'x=', x

    EPS=abs((x-x0)/x0)
    if(EPS<1.0E-12) then
        write(*,*) 'Finish!'
        exit
    else
        x0=x
    end if
end do
end program ex0101c
```

```
Numerical Analysis

授業Web

http://wwwgeo.civil.ibaraki.ac.jp/murakami/class/
```