计算物理作业报告

报告主要内容

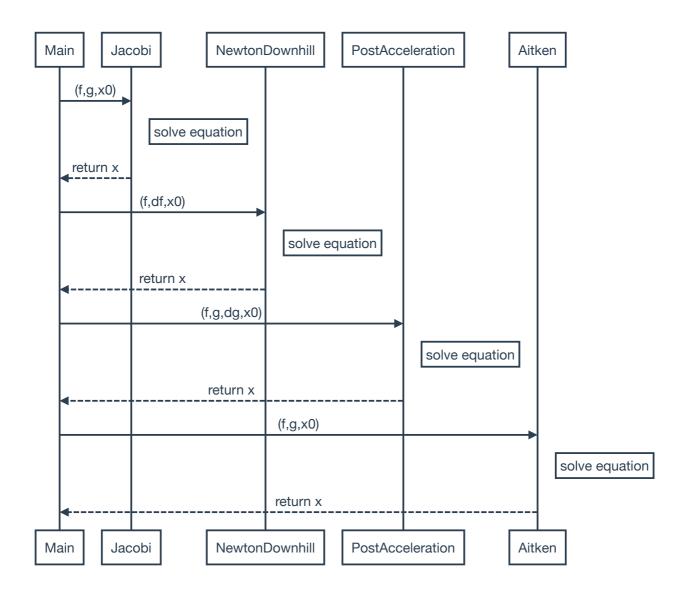
分别用雅科比迭代,牛顿下山法迭代,事后加速法迭代以及Aitken迭代等四种方法计算函数

$$f(x) = \frac{x^3}{3} - x = 0$$

的根。

Main program:

```
program main
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind),external :: f,df,g1,g2
real,external :: dg1
call Jacobi(f,g1,1.5)
call Jacobi(f,g2,1.5)
call Jacobi(f,g2,-1.5)
call Jacobi(f,g2,-1.5)
call NewtonDownhill(f,df,1.5)
call NewtonDownhill(f,df,0.5)
call PostAcceleration(f,g1,dg1,0.5)
call Aitken(f,g1,1.5)
end program main
```



The equations:

$$f(x) = \frac{x^3}{3} - x = 0$$

$$f'(x) = x^2 - 1$$

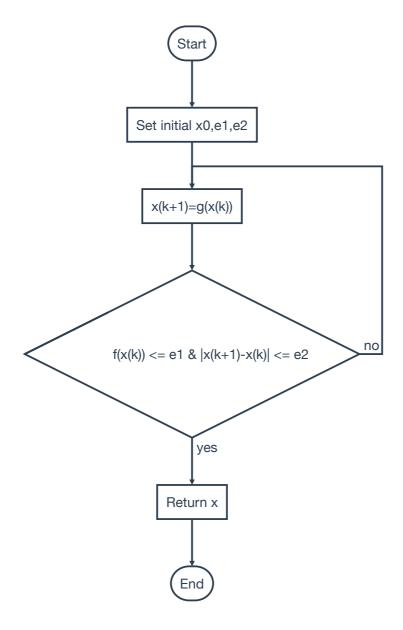
$$g_1(x) = \frac{x^3}{3}$$

$$g_2(x) = \sqrt[3]{3x}$$

$$g'_1(x) = x^2$$

```
!====== Functions ========
function f(x)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind) :: x,f
f=x*x*x/3.0_ikind-x
end function f
function df(x)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind) :: x,df
df=x*x-1
end function df
function g1(x)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind) :: x,f,g1
g1=f(x)+x
end function g1
function dg1(x)
implicit none
real :: x,dg1
dg1=x*x
end function dg1
function g2(x)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind) :: x,g2
g2=sign(abs(3.0_ikind*x)**(1.0_ikind/3._ikind), x)
end function g2
```

Jacobi Iteration Method



The code

```
!======= Subroutines ========
! Jacobi Method
! f: functions
! g: functions
! x0: initial value x
subroutine Jacobi(f,g,x0)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind),external :: f,g
real (kind=ikind) :: x,x_
real :: x0,e1,e2
integer :: k,k_max
print *,'========='
print *,'Jacobi...'
!Select proper initial value x0
print *,'Set initial x=',x0
e1=0.000000000001
e2=0.000000000001
print *,'Set e1=',e1,'e2=',e2
print *
k=0
k_max=100
write(*,"(5x,' k',5x,' x(k)',10x,' x(k)-x(k-1)')")
do
    k=k+1
   x_=x
   x=g(x)
   write(*,100) k,x,x-x_
   100 format(5x, i3, 5x, f19.15, 5x, f19.15)
    if (f(x) \le e1 \cdot and \cdot abs(x-x_) \le e2) then
       print *
       print *,'Iteration end!'
       print *,'x=',x
       exit
    else if (k>=k_max) then
       write(*,"(5x,'...')")
       write(*,"(5x,'Too many iterations!')")
       exit
end do
end subroutine Jacobi
```

$$x_0 = 1.5$$

$$g(x) = \frac{x^3}{3}$$

```
■ 桌面 - bash - 80×24
Eulars-MacBook-Pro:Desktop eular$ ./a
 Jacobi...
 Set initial x= 1.50000000
 Set e1= 9.99999996E-13 e2= 9.99999996E-13
                           x(k)
                                                 x(k)-x(k-1)
        1
                 1.125000000000000
                                                 -0.3750000000000000
                   0.474609375000000
                                                 -0.650390625000000
                  0.035635896027088
                                               -0.438973478972912
        3

      0.000015084877742
      -0.035620811149346

      0.0000000000000001
      -0.000015084877741

      0.0000000000000000
      -0.000000000000000

        5
        6
 Iteration end!
      0.00000000000000000
Eulars-MacBook-Pro:Desktop eular$
```

$$x_0 = 1.5$$

$$g(x) = \sqrt[3]{3x}$$

```
■ 桌面 - bash - 80×37
Eulars-MacBook-Pro:Desktop eular$ ./a
Jacobi...
 Set initial x= 1.50000000
Set e1= 9.99999996E-13 e2= 9.99999996E-13
                                       x(k)-x(k-1)
                    x(k)
      1
              1.650963624447313
                                       0.150963624447313
              1.704588626724398
                                       0.053625002277085
              1.722847936457236
                                       0.018259309732838
      3
              1.728977734722568
                                       0.006129798265331
                                       0.002048108814458
      5
              1.731025843537026
      6
              1.731709085476534
                                       0.000683241939508
              1.731936892712887
                                       0.000227807236353
                                       0.000075942404848
      8
              1.732012835117734
      9
              1.732038149992663
                                       0.000025314874929
     10
              1.732046588366528
                                       0.000008438373865
     11
              1.732049401166952
                                       0.000002812800424
              1.732050338768109
                                       0.000000937601156
     12
     13
              1.732050651301940
                                       0.000000312533831
     14
              1.732050755479897
                                       0.000000104177956
              1.732050790205883
                                       0.000000034725987
     16
              1.732050801781213
                                       0.000000011575329
                                       0.000000003858443
              1.732050805639656
     17
              1.732050806925803
                                       0.000000001286148
     18
     19
              1.732050807354519
                                       0.000000000428716
     20
              1.732050807497425
                                       0.00000000142905
     21
              1.732050807545060
                                       0.00000000047635
                                       0.000000000015878
              1.732050807560938
     22
              1.732050807566231
                                       0.000000000005293
     23
     24
              1.732050807567995
                                       0.00000000001764
     25
              1.732050807568583
                                      0.000000000000588
Iteration end!
     1.7320508075685832
X=
Eulars-MacBook-Pro:Desktop eular$
```

$$x_0 = -1.5$$

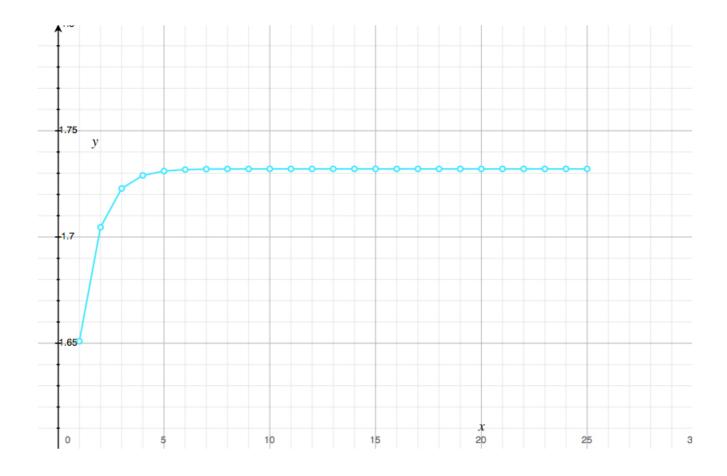
$$g(x) = \sqrt[3]{3x}$$

```
■ 桌面 - bash - 80×37
Eulars-MacBook-Pro:Desktop eular$ ./a
Jacobi...
 Set initial x = -1.50000000
Set e1= 9.99999996E-13 e2= 9.99999996E-13
                                      x(k)-x(k-1)
                    x(k)
      1
             -1.650963624447313
                                     -0.150963624447313
       2
             -1.704588626724398
                                     -0.053625002277085
             -1.722847936457236
                                     -0.018259309732838
      3
             -1.728977734722568
                                     -0.006129798265331
      5
             -1.731025843537026
                                     -0.002048108814458
      6
             -1.731709085476534
                                     -0.000683241939508
             -1.731936892712887
                                     -0.000227807236353
      8
             -1.732012835117734
                                     -0.000075942404848
      9
             -1.732038149992663
                                     -0.000025314874929
     10
             -1.732046588366528
                                     -0.000008438373865
     11
             -1.732049401166952
                                     -0.000002812800424
                                     -0.000000937601156
     12
             -1.732050338768109
     13
             -1.732050651301940
                                     -0.000000312533831
     14
             -1.732050755479897
                                     -0.000000104177956
             -1.732050790205883
                                     -0.000000034725987
     16
             -1.732050801781213
                                     -0.000000011575329
                                     -0.000000003858443
     17
             -1.732050805639656
                                     -0.000000001286148
             -1.732050806925803
     18
     19
             -1.732050807354519
                                     -0.000000000428716
     20
             -1.732050807497425
                                     -0.00000000142905
     21
             -1.732050807545060
                                     -0.000000000047635
                                     -0.00000000015878
     22
             -1.732050807560938
             -1.732050807566231
                                     -0.000000000005293
     23
     24
             -1.732050807567995
                                     -0.000000000001764
     25
             -1.732050807568583
                                     -0.000000000000588
Iteration end!
x = -1.7320508075685832
Eulars-MacBook-Pro:Desktop eular$
```

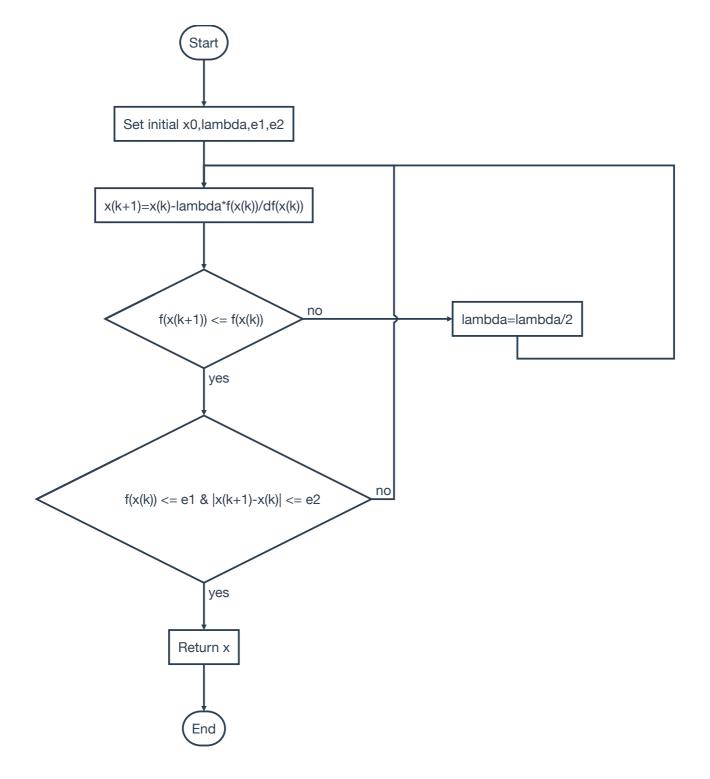
图例:

$$x_0 = 1.5$$

$$g(x) = \sqrt[3]{3x}$$



Newton Downhill Method



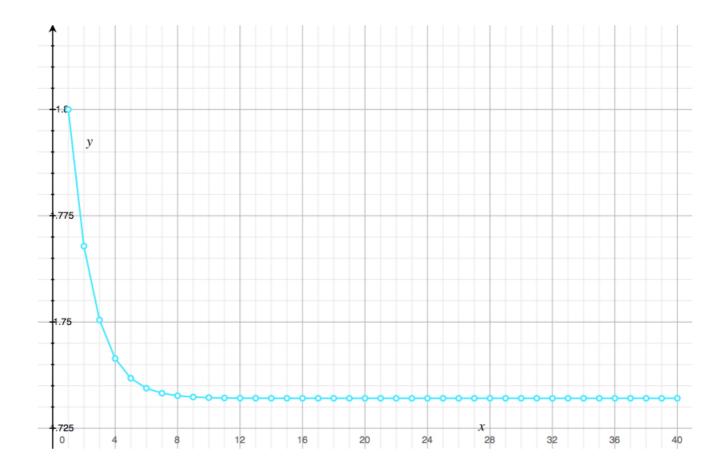
The code

```
! Newton Downhill Method
! f: functions
! df: functions
! x0: initial value x
subroutine NewtonDownhill(f,df,x0)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind),external :: f,df
real (kind=ikind) :: x,x_
real :: x0,delta,lambda,e1,e2,el
integer :: k
print *,'=========='
print *,'Newton Downhill...'
x=x0
lambda=1
print *,'Set initial x0=',x,'Lambda=',lambda
e1=0.0000000000001
e2=0.000000000001
el=0.000000000001
delta=0.000000000001
print *,'Set e1=',e1,'e2=',e2
print *,' el=',el,'delta=',delta
print *
k=0
write(*,"(5x,' k',5x,' x(k)',10x,' x(k)-x(k-1)')")
   k=k+1
   x = x
   x=x-lambda*f(x)/df(x)
   write(*,100) k,x,x-x_{-}
   100 format(5x, i3, 5x, f19.15, 5x, f19.15)
   if (f(x) \le f(x_{-})) then
       if (f(x_{-}) \le e1 \cdot and \cdot abs(x-x_{-}) \le e2) then
           print *
           print *,'Iteration end!'
           print *,'x=',x
           exit
       if (lambda > el) then
           lambda=lambda/2
           x=x_+delta
print *,'========='
end subroutine NewtonDownhill
```

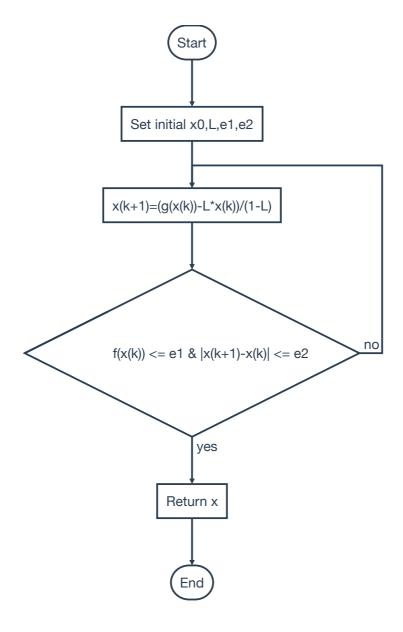
```
🛅 桌面 - bash - 80×53
Eulars-MacBook-Pro:Desktop eular$ ./a
Newton Downhill...
Lambda=
                                                     1.00000000
          9.9999996E-13 e2= 9.99999996E-13
Set e1=
    el=
                                  9.9999996E-13
          9.99999996E-13 delta=
                     x(k)
                                       x(k)-x(k-1)
                                      0.3000000000000000
       1
              1.8000000000000000
              1.767857142857143
                                     -0.032142857142857
       3
              1.750483603043618
                                     -0.017373539813525
       4
              1.741410765197173
                                      -0.009072837846445
                                      -0.004642515413679
              1.736768249783494
      6
              1.734419104204341
                                     -0.002349145579153
       7
              1.733237376857049
                                     -0.001181727347292
                                     -0.000592675958253
      8
              1.732644700898796
      9
              1.732347906839345
                                      -0.000296794059451
     10
              1.732199395409975
                                     -0.000148511429370
                                     -0.000074284362258
     11
              1.732125111047717
     12
              1.732087961698724
                                      -0.000037149348993
              1.732069385231514
                                     -0.000018576467210
     14
              1.732060096549638
                                      -0.000009288681877
              1.732055452096620
                                      -0.000004644453018
     15
              1.732053129842089
                                     -0.000002322254530
     16
     17
              1.732051968707818
                                      -0.000001161134271
              1.732051388138932
                                     -0.000000580568887
     18
     19
              1.732051097854050
                                      -0.000000290284881
                                     -0.000000145142550
     20
              1.732050952711500
              1.732050880140198
     21
                                     -0.000000072571303
     22
              1.732050843854540
                                      -0.000000036285658
     23
              1.732050825711709
                                     -0.000000018142831
     24
              1.732050816640293
                                      -0.000000009071416
                                      -0.000000004535708
     25
              1.732050812104585
              1.732050809836731
                                      -0.000000002267854
     26
     27
              1.732050808702804
                                      -0.000000001133927
              1.732050808135841
     28
                                      -0.000000000566964
     29
              1.732050807852359
                                      -0.000000000283482
     30
              1.732050807710618
                                     -0.000000000141741
     31
              1.732050807639748
                                      -0.00000000070870
     32
              1.732050807604313
                                      -0.00000000035435
     33
              1.732050807586595
                                     -0.00000000017718
     34
              1.732050807577736
                                      -0.000000000008859
              1.732050807573307
                                     -0.000000000004429
     35
              1.732050807571092
                                     -0.000000000002215
     36
     37
              1.732050807569985
                                      -0.000000000001107
     38
              1.732050807569431
                                     -0.000000000000554
              1.732050807569154
                                     -0.000000000000277
     39
              1.732050807569016
     40
                                     -0.000000000000138
Iteration end!
     1.7320508075690157
Eulars-MacBook-Pro:Desktop eular$
```

$$x_0 = 0.5$$

```
■ 桌面 - bash - 80×53
Eulars-MacBook-Pro:Desktop eular$ gfortran -o a LocationOfRoot.f90
Eulars-MacBook-Pro:Desktop eular$ ./a
Newton Downhill...
Set initial x0= 0.5000000000000000
                                           Lambda=
                                                      1.00000000
           9.9999996E-13 e2= 9.99999996E-13
Set e1=
           9.99999996E-13 delta= 9.99999996E-13
    el=
                     x(k)
                                       x(k)-x(k-1)
                                      -0.61111111111111111
              -0.1111111111111111
       1
       2
              -0.055092592592593
                                       0.056018518518518
       3
              -0.027490387705163
                                       0.027602204887430
       4
              -0.013738263590324
                                       0.013752124114838
       5
              -0.006868267312563
                                       0.006869996277761
      6
              -0.003434025652042
                                       0.003434241660521
       7
                                       0.001717026324800
              -0.001716999327243
      8
                                       0.000858501350914
              -0.000858497976329
      9
              -0.000429248777255
                                       0.000429249199074
              -0.000214624362264
                                       0.000214624414991
      10
      11
              -0.000107312177836
                                       0.000107312184427
      12
              -0.000053656088506
                                       0.000053656089330
      13
              -0.000026828044202
                                       0.000026828044305
      14
                                       0.000013414022107
              -0.000013414022094
      15
                                       0.000006707011048
              -0.000006707011046
      16
              -0.000003353505523
                                       0.000003353505523
      17
              -0.000001676752762
                                       0.000001676752762
      18
              -0.000000838376381
                                       0.000000838376381
      19
              -0.000000419188190
                                       0.000000419188190
                                       0.000000209594095
      20
              -0.000000209594095
      21
              -0.000000104797048
                                       0.000000104797048
      22
              -0.000000052398524
                                       0.000000052398524
      23
              -0.000000026199262
                                       0.000000026199262
      24
              -0.000000013099631
                                       0.000000013099631
              -0.000000006549815
                                       0.000000006549815
      25
      26
              -0.000000003274908
                                       0.000000003274908
      27
              -0.000000001637454
                                       0.000000001637454
      28
              -0.000000000818727
                                       0.000000000818727
      29
              -0.00000000409363
                                       0.00000000409363
             -0.000000000204682
                                       0.000000000204682
      30
              -0.000000000102341
                                       0.000000000102341
      31
      32
              -0.000000000051170
                                       0.00000000051170
      33
              -0.000000000025585
                                       0.000000000025585
      34
              -0.00000000012793
                                       0.00000000012793
             -0.000000000006396
                                       0.000000000006396
      35
              -0.00000000003198
                                       0.00000000003198
      36
      37
              -0.00000000001599
                                       0.00000000001599
      38
              -0.000000000000800
                                       0.000000000000800
      39
              -0.000000000000400
                                       0.000000000000400
Iteration end!
x= -3.9976901090832452E-013
Eulars-MacBook-Pro:Desktop eular$
```



Post Accelerating Method



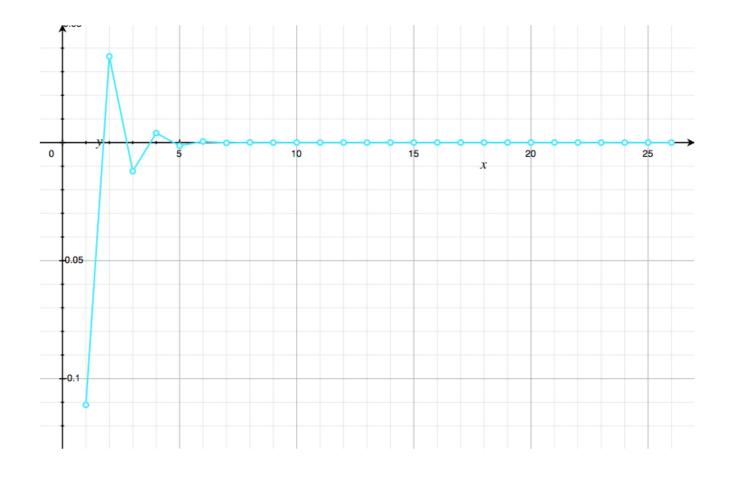
The code:

```
! Post Accelerating Method
! f: functions
! g: functions
! dg: functions
! x0: initial value x
subroutine PostAcceleration(f,g,dg,x0)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind),external :: f,g
real (kind=ikind) :: x,x_
real, external :: dg
real :: x0,e1,e2,L
integer :: k,k_max
print *,'Post Accelerating...'
x=x0
L=dg(x0)
print *,'Set initial x0=',x,'L=',L
e1=0.000000000001
e2=0.000000000001
print *,'Set e1=',e1,'e2=',e2
k=0
k_max=100
write(*,"(5x,' k',5x,' x(k)',10x,' x(k)-x(k-1)')")
   k=k+1
   x = x
   x=(g(x)-L*x)/(1-L)
   write(*,100) k,x,x-x_{-}
   100 format(5x, i3,5x, f19.15,5x, f19.15)
   if (f(x_{-}) \le e1 \cdot and \cdot abs(x-x_{-}) \le e2) then
       print *,'Iteration end!'
      print *,'x=',x
   else if (k>=k_max) then
       write(*,"(5x,'...')")
       write(*,"(5x,'Too many iterations!')")
   end if
end subroutine PostAcceleration
```

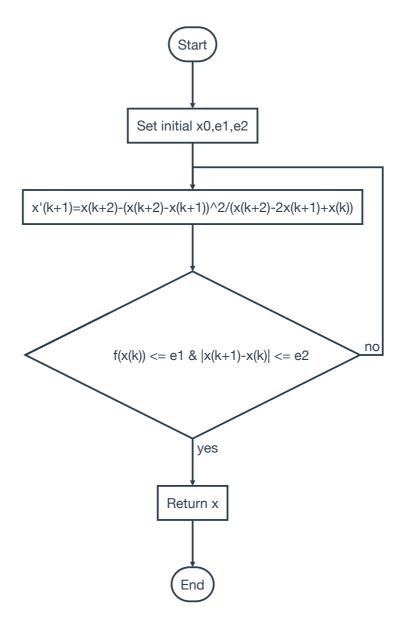
$$x_0 = 0.5$$

$$L = 0.25$$

```
■ 桌面 - bash - 80×39
Eulars-MacBook-Pro:Desktop eular$ gfortran -o a LocationOfRoot.f90
Eulars-MacBook-Pro:Desktop eular$ ./a
 Post Accelerating...
 L= 0.250000000
 Set e1= 9.99999996E-13 e2= 9.99999996E-13
      k
                    x(k)
                                      x(k)-x(k-1)
             -0.1111111111111111
      1
                                     -0.6111111111111111
              0.036427373875934
      2
                                     0.147538484987045
      3
             -0.012120974654769
                                    -0.048548348530703
              0.004039533422845
                                     0.016160508077614
      4
      5
             -0.001346481844761
                                     -0.005386015267606
      6
              0.000448826196614
                                     0.001795308041375
                                    -0.000598434888635
             -0.000149608692021
      7
      8
             0.000049869562519
                                     0.000199478254539
      9
                                    -0.000066492749970
             -0.000016623187451
     10
              0.000005541062482
                                      0.000022164249933
             -0.000001847020827
                                     -0.000007388083309
     11
     12
             0.000000615673609
                                     0.000002462694436
     13
             -0.000000205224536
                                    -0.000000820898145
     14
                                     0.000000273632715
             0.000000068408179
     15
             -0.000000022802726
                                     -0.000000091210905
                                      0.000000030403635
              0.000000007600909
     16
             -0.000000002533636
                                     -0.000000010134545
     17
     18
             0.000000000844545
                                      0.000000003378182
     19
                                    -0.000000001126061
             -0.000000000281515
     20
              0.00000000093838
                                      0.000000000375354
                                     -0.000000000125118
             -0.000000000031279
     21
             0.000000000010426
                                     0.000000000041706
     22
     23
             -0.00000000003475
                                    -0.00000000013902
     24
             0.000000000001158
                                     0.000000000004634
     25
             -0.00000000000386
                                     -0.00000000001545
     26
              0.00000000000129
                                      0.000000000000515
 Iteration end!
     1.2872205716664194E-013
 \mathbf{x} =
Eulars-MacBook-Pro:Desktop eular$
```



Aitken Method



The code:

```
! Aitken Method
! f: functions
! g: functions
! x0: initial value x
subroutine Aitken(f,g,x0)
implicit none
integer,parameter :: ikind=selected_real_kind(p=15)
real (kind=ikind), external :: f,g
real (kind=ikind) :: x,x_
real :: x0,e1,e2
integer :: k,k_max
print *,'Aitken...'
print *,'Set initial x0=',x
e1=0.000000000001
e2=0.000000000001
print *,'Set e1=',e1,'e2=',e2
k=0
k_max=100
write(*,"(5x,' k',5x,' x(k)',10x,' x(k)-x(k-1)')")
   k=k+1
   x_=x
   x=g(x_{-})
   x=g(x)-(g(x)-x)**2/(g(x)-2*x+x_{-})
   write(*,100) k,x,x-x_
   100 format(5x, i3, 5x, f19.15, 5x, f19.15)
   if (f(x) \le e1 \cdot and \cdot abs(x-x_-) \le e2) then
      print *,'Iteration end!'
   else if (k>=k_max) then
       write(*,"(5x,'...')")
      write(*,"(5x,'Too many iterations!')")
   end if
end do
end subroutine Aitken
```

```
■ 桌面 - bash - 80×25
Eulars-MacBook-Pro:Desktop eular$ ./a
Aitken...
Set e1= 9.99999996E-13 e2= 9.99999996E-13
                   x(k)
                                   x(k)-x(k-1)
      1
             2.010638297872340
                                   0.510638297872340
                                   -0.151569048272452
             1.859069249599888
             1.765106642447591
      3
                                  -0.093962607152298
             1.734704213224771
                                  -0.030402429222820
             1.732068997319653
      5
                                  -0.002635215905118
      6
             1.732050808428462
                                  -0.000018188891191
             1.732050807568877
                                  -0.000000000859585
             1.732050807568877
                                   0.0000000000000000
Iteration end!
x= 1.7320508075688772
Eulars-MacBook-Pro:Desktop eular$
```

图例

