第2章 非线性方程求根

计63 黄冰鉴 2016011296

第2题

解题思路

- 实现阻尼牛顿法的代码，指定不同的函数和初值，即可迭代计算出结果。因为λ要求每次折半，所以我将初值设为了0.5，方便运算。

实验过程

- 我使用python实现了阻尼牛顿法，残差和误差设为了`10^-8`。代码如下，

- ```
  def f(x):
      # return x*x*x-x-1
      return -pow(x, 3)+5*x

  def f_fdev(x):
      # return (x*x*x-x-1)/(3*x*x-1)
      return (-pow(x, 3)+5*x)/(-3*x*x+5)

  lda_0 = 0.5
  x_prev = 0
  y_prev = f(x_prev)
  # x_next = 0.6
  x_next = 1.2
  y_next = f(x_next)
  thres1 = 1e-8
  thres2 = 1e-8

  while abs(y_next) > thres1 or abs(x_next-x_prev) > thres2:
      y_prev = y_next
      x_prev = x_next
      s = f_fdev(x_prev)
      x_next = x_prev - s
      lda = lda_0
      while abs(f(x_next)) >= abs(f(x_prev)):
          x_next = x_prev - lda*s
          lda = lda/2
      y_next = f(x_next)
      print('final λ = %.8f'%(lda*2), 'approx x = ', x_next, 'approx f(x) = ',
  y_next)

  print(x_next, f(x_next))
  ```
  ```

- 运行结果如下，每一行为一次迭代中λ的取值与迭代得到的x和y，可以看到λ大部分迭代周期为1，接近超线性的收敛速度。

- ```
  x^3-x-1=0
  final λ = 0.03125000 approx x =  1.1406250000000009 approx f(x) =
  -0.6566429138183569
  final λ = 1.00000000 approx x =  1.3668136615928008 approx f(x) =
  0.1866397182002275
  final λ = 1.00000000 approx x =  1.3262798040083197 approx f(x) =
  0.006670401469929255
  final λ = 1.00000000 approx x =  1.324720225636056 approx f(x) =
  9.673876883775634e-06
  final λ = 1.00000000 approx x =  1.3247179572495411 approx f(x) =
  2.0449419935175683e-11
  final λ = 1.00000000 approx x =  1.324717957244746 approx f(x) =
  2.220446049250313e-16
  1.324717957244746 2.220446049250313e-16
  ```

- ```
  -x^3+5x = 0
  final λ = 0.50000000 approx x =  -1.9411764705882326 approx f(x) =
  -2.3912070018318916
  final λ = 1.00000000 approx x =  -2.3204623232388473 approx f(x) =
  0.892323097356627
  final λ = 1.00000000 approx x =  -2.240459436027981 approx f(x) =
  0.04404403706650051
  final λ = 1.00000000 approx x =  -2.236080855200313 approx f(x) =
  0.0001287781176912972
  final λ = 1.00000000 approx x =  -2.2360679776110337 approx f(x) =
  1.112439917960728e-09
  final λ = 1.00000000 approx x =  -2.23606797749979 approx f(x) =
  1.7763568394002505e-15
  -2.23606797749979 1.7763568394002505e-15
  ```

第3题

解题思路

- 书2.6.3节中给出了详细的MATLAB代码，因此只需理解代码之后将其复现，并找到十个包含零点的区间进行迭代即可。

实验过程

- 我使用了MATLAB复现了算法，通过plot()观察贝塞尔函数大于零的零点的粗略位置，确定了十个包含零点的区间，经过十组迭代得到结果。
- 代码如下，
- ```
  Xrange = -50:0.1:50;
  ```
  ```

```
plot(Xrange, bessel(Xrange))
%观察得
range = [0,4,7,10,13,16,19,23,26,29,32];
res = zeros(10, 1);
for i = 1:10
    x = range(i:i+1);
    res(i) = fzerotx(@bessel, x);
end
res

%同书2.6.3节
function b = fzerotx(F, ab, varargin)
a = ab(1);
b = ab(2);
fa = F(a, varargin{:});
fb = F(b, varargin{:});
if sign(fa) == sign(fb)
    error('Function change sign')
end
c = a;
fc = fa;
d = b-c;
e = d;

while fb ~= 0
    %调整abc的值使满足迭代条件
    if sign(fa) == sign(fb)
        a = c;
        fa = fc;
        d = b-c;
        e = d;
    end
    if abs(fa) < abs(fb)
        c = b;
        b = a;
        a = c;
        fc = fb;
        fb = fa;
        fa = fc;
    end

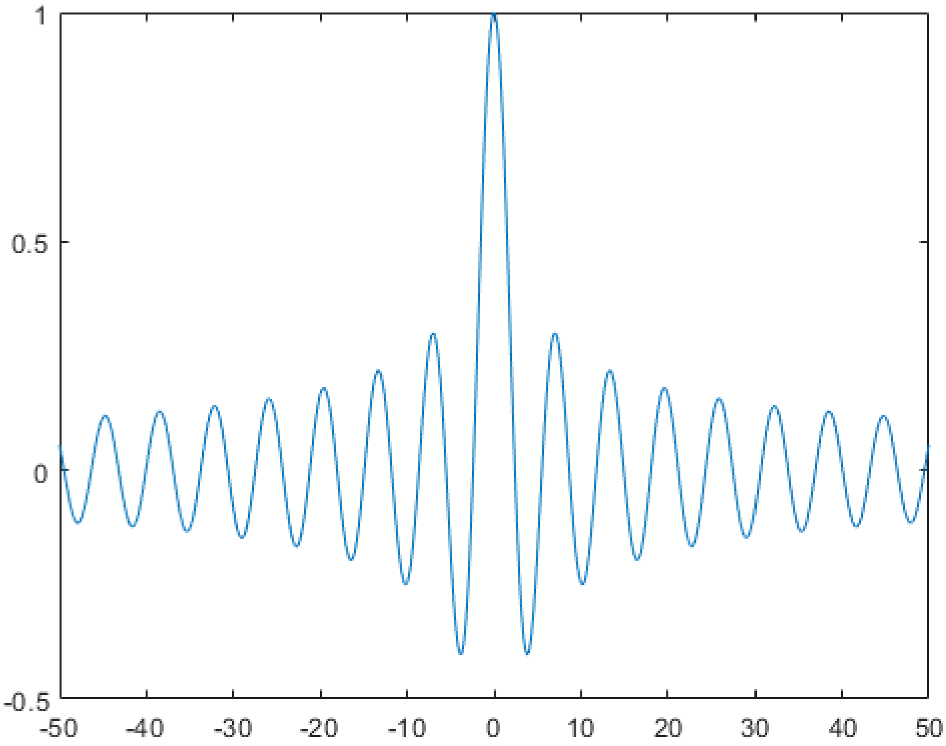
    m = 0.5*(a-b);
    tol = 2.0*eps*max(abs(b), 1.0);
    if (abs(m) <= tol) || (fb == 0)
        break
    end
    if ( abs(e) < tol) || (abs(fc) <= abs(fb))
        d = m;
        e = m;
    else
        s = fb/fc;
        if (a==c)
            p = 2.0*m*s;
```

```
            q = 1-s;
        else
            q = fc/fa;
            r = fb/fa;
            p = s*(2*m*q*(q-r)-(b-c)*(r-1));
            q = (q-1)*(r-1)*(s-1);
        end
        if p > 0
            q = -q;
        else
            p = -p;
        end
        if (2*p<3*m*q-abs(tol*q)) && (p<abs(0.5*e*q))
            e = d;
            d = p/q;
        else
            d = m;
            e = m;
        end
    end

    c = b;
    fc = fb;
    if abs(d) > tol
        b = b+d;
    else
        b = b-sign(b-a)*tol;
    end
    fb = F(b, varargin{:});
end
end


function y = bessel(x)
    y = besselj(0, x);
end
```

- 贝塞尔函数的曲线如图所示:
-



- 得到的**前十个正零点**如下，
- ```
  res =
      2.4048
      5.5201
      8.6537
     11.7915
     14.9309
     18.0711
     21.2116
     24.3525
     27.4935
     30.6346
  ```

实验结果分析

第2题通过实际例子说明了阻尼牛顿法对牛顿法的改进，以及它在实际使用过程中接近超线性的收敛速度。

第3题则是对bessel函数进行了求根。

2019/3/30
  ```