

求解非线性方程组的几种方法及程序实现

雍龙泉

(陕西理工大学 数学与计算机科学学院, 陕西 汉中, 723001)

摘 要: 以一个非线性方程组为例, 分别给出其 Jacobi 迭代法、Gauss-Seidel 迭代法、Newton 迭代法、五阶牛顿迭代法。在给出迭代公式和初始点后, 分别给出了每一种迭代方法对应的 MATLAB 代码与运行结果, 旨在提高学生编程的兴趣。

关键词: 非线性方程组; Jacobi 迭代; Gauss-Seidel 迭代; Newton 迭代; 五阶牛顿迭代

中图分类号: O241 **文献标志码:** A **文章编号:** 2095-4824(2021)03-0044-05

考虑非线性方程组

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0, \\ f_2(x_1, x_2, \dots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0. \end{cases} \quad (1)$$

记 $x = (x_1, x_2, \dots, x_n)$, 向量函数 $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$, 则方程组(1)等价于如下非线性方程

$$F(x) = 0. \quad (2)$$

很多复杂的问题都可以转化为非线性方程组的求解问题, 其最基本的求解方法是牛顿法^[1-7]。近年来, 相继出现了五阶牛顿迭代法^[8]、七阶牛顿迭代法^[9-10]、八阶牛顿迭代法^[11-12]、九阶牛顿迭代法等^[13-14]。对非线性方程组而言, 牛顿迭代法需要计算偏导数矩阵, 并依赖于初始点的选取和函数 $F(x)$ 的性态^[15-18], 而在一些实际问题中如何选取合适的初始点本身是一个比较困难的问题, 因此使用牛顿迭代法时具有一定的局限性^[19]。

本文假设非线性方程组的解存在, 在给定初始点后, 如何把数学公式转变为可以运行的代码, 让初学者对编程不再望而生畏; 进而提高学生编程的兴趣, 最终能够利用程序对实际问题进行求解, 培养与提高学生的科学计算能力。

下面通过一个具体的非线性方程组, 分别给出其 Jacobi 迭代、Gauss-Seidel 迭代、Newton 迭代、五阶牛顿迭代; 在给出迭代公式和初始点后, 并分别给出了 MATLAB 代码和对应的计算结果。

算例 分别用 Jacobi 迭代法、Gauss-Seidel 迭代法、Newton 迭代法、五阶牛顿迭代法求解如下非线性方程组

$$\begin{cases} 3x_1 - \cos(-x_2x_3) - \frac{1}{2} = 0 \\ x_1^2 - 81(x_2 + 1)^2 + \sin(x_3) + 1.06 = 0 \\ e^{-x_1x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0 \end{cases}$$

1 Jacobi 迭代

Jacobi 迭代法的迭代格式:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{3} \cos(x_2^{(k)} x_3^{(k)}) + \frac{1}{6} \\ x_2^{(k+1)} = \frac{1}{9} \sqrt{(x_1^{(k)})^2 + \sin x_3^{(k)} + 1.06} - 0.1, \\ k = 1, 2, \dots \\ x_3^{(k+1)} = -\frac{1}{20} e^{-x_1^{(k)} x_2^{(k)}} - \frac{10\pi - 3}{60} \end{cases}$$

取初值 $x^{(1)} = (0.1, 0.1, -0.1)^T$, 终止条件

收稿日期: 2020-12-10

基金项目: 国家自然科学基金(11401357); 陕西省教育厅重点科学研究计划项目(20JS021); 陕西理工大学科研项目(SLGKYQD2-14); 陕西理工大学教学改革研究项目(SLGYYJG2015)

作者简介: 雍龙泉(1980-), 男, 陕西洋县人, 陕西理工大学数学与计算机科学学院教授, 博士。

为 $\|x^{(k+1)} - x^{(k)}\|_{\infty} < 1 \times 10^{-5}$, Jacobi 迭代法的 MATLAB 代码和计算结果分别见表 1 和表 2。

表 1 Jacobi 迭代法的 MATLAB 代码

```
clc; clear all; close all;
x0=[0.1;0.1;-0.1];k=1;x(:,k)=x0;
while true
    x(1,k+1)=cos(x(2,k)*x(3,k))/3+1/6;
    x(2,k+1)=sqrt(x(1,k)^2+sin(x(3,k)+1.06))/9-0.1;
    x(3,k+1)=-exp(-x(1,k)*x(2,k))/20-(10*pi-3)/60;
    fanshu(:,k+1)=norm(x(:,k+1)-x(:,k),inf);
    if norm(x(:,k+1)-x(:,k))<0.0001
        break;
    end
    k=k+1;
end
figure(1),plot(x(1,:), 'o-'), figure(2),plot(x(2,:), 'o-'), figure(3),plot(x(3,:), 'o-')
TABLE_1=[x,fanshu]
```

表 2 Jacobi 迭代计算结果

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\ x^{(k+1)} - x^{(k)}\ _{\infty}$
1	0.10000000	0.10000000	-0.10000000	
2	0.49998333	0.00117773	-0.52310127	4.23E-01
3	0.49999994	-0.00304278	-0.52356934	4.22E-03
4	0.49999958	-0.00306734	-0.52367490	1.06E-04
5	0.49999957	-0.00307314	-0.52367552	5.80E-06

2 Gauss-Seidel 迭代

Gauss-Seidel 迭代法的迭代格式:

$$\begin{cases} x_1^{(k+1)} = \frac{1}{3} \cos(x_2^{(k)} x_3^{(k)}) + \frac{1}{6} \\ x_2^{(k+1)} = \frac{1}{9} \sqrt{(x_1^{(k+1)})^2 + \sin(x_3^{(k)} + 1.06)} - 0.1, \\ x_3^{(k+1)} = -\frac{1}{20} e^{-x_1^{(k+1)} x_2^{(k+1)}} - \frac{10\pi - 3}{60} \end{cases}$$

式中: $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)})^T$ 是第 k 次迭代点, $k=1, 2, \dots$, 取初值 $x^{(1)} = (0.1, 0.1, -0.1)^T$, 终止条件为 $\|x^{(k+1)} - x^{(k)}\|_{\infty} < 1 \times 10^{-5}$ 。

Gauss-Seidel 迭代法的 MATLAB 代码和计算结果分别见表 3 和表 4。由于 Gauss-Seidel 迭代方法分别用 $x_1^{(k+1)}, x_2^{(k+1)}$ 替换 $x_1^{(k)}$ 和 $x_2^{(k)}$, 因此在相同的条件下, Gauss-Seidel 迭代方法比 Jacobi 迭代方法收敛更快。

表 3 Gauss-Seidel 迭代法的 MATLAB 代码

```
clc; clear all; close all;
x0=[0.1;0.1;-0.1];k=1;x(:,k)=x0;
while true
    x(1,k+1)=cos(x(2,k)*x(3,k))/3+1/6;
    x(2,k+1)=sqrt(x(1,k+1)^2+sin(x(3,k)+1.06))/9-0.1;
    x(3,k+1)=-exp(-x(1,k+1)*x(2,k+1))/20-(10*pi-3)/60;
    fanshu(:,k+1)=norm(x(:,k+1)-x(:,k),inf);
    if norm(x(:,k+1)-x(:,k))<0.00001
        break;
    end
    k=k+1;
end
figure(1),plot(x(1,:), 'o-')figure(2),plot(x(2,:), 'o-')figure(3),plot(x(3,:), 'o-')
TABLE_2=[x,fanshu]
```

表 4 Gauss-Seidel 迭代计算结果

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\ x^{(k+1)} - x^{(k)}\ _\infty$
1	0.10000000	0.10000000	-0.10000000	
2	0.49998333	0.02222979	-0.52304613	4.23E-01
3	0.49997747	0.00002815	-0.52359807	2.22E-02
4	0.50000000	0.00000004	-0.52359877	2.81E-05
5	0.50000000	0.00000000	-0.52359878	3.76E-08

3 Newton 迭代

Newton 迭代法的迭代格式:

$$x^{(k+1)} = x^{(k)} - F'(x^{(k)})^{-1} F(x^{(k)}), \quad k = 1, 2, \dots,$$

$$\text{这里 } F'(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix}$$

Newton 迭代法的 MATLAB 代码如表 5 所示。取初值 $x^{(1)} = (0.1, 0.1, -0.1)^T$, 终止条件为 $\|x^{(k+1)} - x^{(k)}\|_\infty < 1 \times 10^{-5}$, 计算结果见表 6。

Newton 法具有二阶收敛速度, 其效率指数为 $\sqrt{2}$ 。实际计算时迭代初始点与根很靠近时, 二阶收敛才能很好地体现出来。

Newton 迭代法要求矩阵 $F'(x)$ 可逆, 若 $F'(x)$ 奇异或接近奇异, 可以采用阻尼 Newton 迭代法进行处理。

表 5 Newton 迭代法的 MATLAB 代码

```
clc; clear all; close all;
x0=[0.1;0.1;-0.1];k=1;x(:,k)=x0;
while true
    J=[3, x(3,k)*sin(x(2,k)*x(3,k)), x(2,k)*sin(x(2,k)*x(3,k));
        2*x(1,k), -162*(x(2,k)+0.1), cos(x(3,k));
        -x(2,k)*exp(-x(1,k)*x(2,k)), -x(1,k)*exp(-x(1,k)*x(2,k)), 20];
    f=[3*x(1,k)-cos(x(2,k)*x(3,k))-1/2;
        x(1,k)^2-81*(x(2,k)+0.1)^2+sin(x(3,k))+1.06;
        exp(-x(1,k)*x(2,k))+20*x(3,k)+(10*pi-3)/3];
    x(:,k+1)=x(:,k)-inv(J)*f;
    fanshu(:,k+1)=norm(x(:,k+1)-x(:,k),inf);
    if norm(x(:,k+1)-x(:,k),inf)<0.00001 break; end
    k=k+1;
end
figure(1),plot(x(1,:), 'o-')figure(2),plot(x(2,:), 'o-')figure(3),plot(x(3,:), 'o-')
TABLE_3=[x,fanshu]
```

表 6 Newton 迭代计算结果

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\ x^{(k+1)} - x^{(k)}\ _\infty$
1	0.10000000	0.10000000	-0.10000000	
2	0.49986967	0.01946685	-0.52152047	4.22E-01
3	0.50001424	0.00158859	-0.52355696	1.79E-02
4	0.50000011	0.00001244	-0.52359845	1.58E-03
5	0.50000000	0.00000000	-0.52359878	1.24E-05
6	0.50000000	0.00000000	-0.52359878	7.76E-10

4 五阶牛顿法迭代

文献[20]中给出了五阶牛顿迭代法:

$$\begin{cases} y^{(k)} = x^{(k)} - \frac{1}{2} [F'(x^{(k)})]^{-1} F(x^{(k)}) \\ z^{(k)} = x^{(k)} - [F'(y^{(k)})]^{-1} F(x^{(k)}), k = 1, 2, \dots \\ x^{(k+1)} = z^{(k)} + \left(\frac{[F'(x^{(k)})]^{-1} - 2[F'(y^{(k)})]^{-1}}{2[F'(y^{(k)})]^{-1}} \right) F(z^{(k)}) \end{cases}$$

这里 $y^{(k)}$ 与 $z^{(k)}$ 是两个中间量,五阶牛顿迭代法的 MATLAB 代码见表 7。

取初值 $x^{(1)} = (0.1, 0.1, -0.1)^T$, 终止条件为 $\|x^{(k+1)} - x^{(k)}\|_{\infty} < 1 \times 10^{-5}$, 计算结果见表 8。这里的五阶牛顿迭代法是一个三步迭代法, 每次迭代需要计算函数值 $F(x^{(k)})$, $F(z^{(k)})$, $[F'(x^{(k)})]^{-1}$, $[F'(y^{(k)})]^{-1}$ 计算工作量较大。

表 7 五阶牛顿迭代法的 MATLAB 代码

```
clc; clear all; close all
x0=[0.1;0.1;-0.1]; k=1;x=x0;
fun=@fun1; dfun=@dfun1;
while true
    yk=x(:,k)-0.5*inv(dfun(x(:,k)))*fun(x(:,k));
    zk=x(:,k)-inv(dfun(yk))*fun(x(:,k));
    x(:,k+1)=zk-(2*inv(dfun(yk))-inv(dfun(x(:,k))))*fun(zk);
    fanshu(:,k+1)=norm(x(:,k+1)-x(:,k),inf);
    if norm(x(:,k+1)-x(:,k))<1e-5 break; end
    k=k+1;
end
figure(1),plot(x(1,:), 'o-')figure(2),plot(x(2,:), 'o-')figure(3),plot(x(3,:), 'o-')
TABLE_4=[x,fanshu]
```

表 8 五阶牛顿法迭代计算结果

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$\ x^{(k+1)} - x^{(k)}\ _{\infty}$
1	0.10000000	0.10000000	-0.10000000	
2	0.50001207	0.00135046	-0.52354774	4.24E-01
3	0.50000000	0.00000000	-0.52359878	1.35E-03
4	0.50000000	0.00000000	-0.52359878	1.08E-11

需要求解的非线性方程组 fun1 与偏导数矩阵 dfun1 的代码见表 9 和表 10。

表 9 非线性方程组 fun1 的代码

```
function [f]=fun1(x)
f(1,1)=3*x(1)-cos(x(2)*x(3))-1/2;
f(2,1)=x(1)^2-81*(x(2)+0.1)^2+sin(x(3))+1.06;
f(3,1)=exp(-x(1)*x(2))+20*x(3)+(10*pi-3)/3;
end
```

表 10 偏导数矩阵 dfun1 的代码

```
function [df]=dfun1(x)
df=[3,x(3)*sin(x(2)*x(3)),x(2)*sin(x(2)*x(3));
    2*x(1),-162*(x(2)+0.1),cos(x(3));
    -x(2)*exp(-x(1)*x(2)), -x(1)*exp(-x(1)*x(2)),20];
end
```

文献[20-23]中分别采用该五阶牛顿迭代法求解线性规划、线性互补、非线性两点边值、投资

组合优化等问题。

更多的实验结果表明:各类迭代的收敛与否、收敛快慢均与初始点的选取有关;收敛阶数越高,需要付出的计算代价(包括占用内存、函数调用次数)也就越大;即使高阶的牛顿法,也只有初始点与根很靠近时,高阶收敛性才能很好地体现出来。

[参 考 文 献]

- [1] 李庆扬,莫牧中,祁力群. 非线性方程组的数值解法[M]. 北京:科学出版社,1987.
- [2] 范金燕,袁亚湘. 非线性方程组数值方法[M]. 北京:科学出版社,2018.
- [3] 马元婧. 非线性方程组的一种修正牛顿法及其连续型[D]. 哈尔滨:哈尔滨工业大学,2009.
- [4] 萨和雅. 非线性方程组的锥模型方法研究[D]. 呼和浩特:内蒙古大学,2017.
- [5] 雍龙泉. 基于正弦余弦算法的非线性方程组求根[J]. 陕西理工大学学报(自然科学版), 2019, 35

- (6):64–69.
- [6] 雍龙泉. 一种改进的和声搜索算法求解非线性方程组[J]. 重庆理工大学学报(自然科学), 2020, 34(10):231–237.
- [7] 伍渝江, 陈亮. 非线性方程组的非交替 Newton–PHSS 迭代法[J]. 应用数学与计算数学学报, 2017, 31(2):153–162.
- [8] 薛霜. 两类五阶收敛的牛顿迭代格式[J]. 安徽师范大学学报(自然科学版), 2013, 36(5):430–436.
- [9] 刘雅妹, 王霞. 一类新的求解非线性方程的七阶方法[J]. 数学的实践与认识, 2011, 41(14):239–245.
- [10] NARANG M, BHATIA S, KANWAR V. New efficient derivative free family of seventh-order methods for solving systems of nonlinear equations[J]. Numerical Algorithms, 2017, 76(1):1–25.
- [11] 王晓锋, 张铁. 一类求解非线性方程最优的 8 阶收敛迭代法[J]. 吉林大学学报(理学版), 2013, 51(4):568–572.
- [12] BI W, REN H, WU Q. Three-step iterative methods with eighth-order convergence for solving nonlinear equations[M]. Journal of Computational and Applied Mathematics, 2009, 225:105–112.
- [13] HU Z, LIU G, TIAN L. An iterative method with ninth-order convergence for solving nonlinear equations[J]. International Journal of Contemporary Mathematical Sciences, 2011, 6(1):17–23.
- [14] 张旭, 檀结庆, 艾列富. 一种求解非线性方程组的 3p 阶迭代方法[J]. 计算数学, 2017, 39(1):14–22.
- [15] 李庆扬, 王能超, 易大义. 数值分析[M]. 5 版. 北京:清华大学出版社, 2008.
- [16] 王开荣, 杨大地. 数值分析[M]. 2 版. 北京:科学出版社, 2014.
- [17] 雷金贵, 李建良, 蒋勇. 数值分析与计算方法[M]. 2 版. 北京:科学出版社, 2017.
- [18] 施妙根, 顾丽珍. 科学和工程计算基础[M]. 北京:清华大学出版社, 1999.
- [19] 雍龙泉. 非线性方程的三种牛顿迭代方法[J]. 高等数学研究, 2020, 23(1):76–79.
- [20] 雍龙泉. 基于上方一致光滑逼近函数的高阶牛顿法求解线性规划[J]. 吉林大学学报(理学版), 2019, 57(2):265–270.
- [21] 雍龙泉. 一类高阶牛顿迭代法及其在线性互补问题中的应用[J]. 数学的实践与认识, 2019, 49(14):160–167.
- [22] 雍龙泉. 一类高阶牛顿迭代法及其在非线性的两点边值问题中的应用[J]. 重庆理工大学学报(自然科学), 2019, 33(7):242–247.
- [23] 雍龙泉. 一种高阶牛顿法求解投资组合优化模型[J]. 数学的实践与认识, 2019, 49(19):216–221.

Several Methods and Program Implementation for Solving Nonlinear Equations

Yong Longquan

(School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong, Shaanxi 723001, China)

Abstract: Taking a nonlinear equations as an example, the Jacobi iteration, Gauss-Seidel iteration, Newton iteration and fifth-order Newton iteration are given respectively. After the iteration formulas and initial point are given, the MATLAB code corresponding to each iteration method and computational results are given respectively, which can improve students' interest in programming.

Key Words: nonlinear equations; Jacobi iteration; Gauss-Seidel iteration; Newton iteration; fifth-order Newton iteration

(责任编辑:熊文涛)