第15卷第4期 数学研究与评论 Vol. 15 No. 4
1995年11月 JOURNAL OF MATHEMATICAL RESEARCH AND EXPOSITION Nov. 1995

关于Jain的隐式迭代法的注*

韩 波 张刚亮
(哈尔滨工业大学数学系，哈尔滨 150006)

摘 要 本文利用嵌入法思想构造了一类求解非线性方程组的隐式迭代法，分析了方法的收敛阶，给出了具体的计算格式，最后的计算结果表明了方法的有效性.

关键词 非线性方程组，隐式迭代法，嵌入法.
分类号 AMS(1991) 65H10/CCL O241.7

一 引 言

M. K. Jain 在[1]中针对非线性方程求根问题给出了一个隐式5阶迭代法. 之后，又对于非线性方程组求解问题提出了3阶隐式Newton法[2]. 针对一些敏感问题所进行的数值计算结果表明，隐式迭代法的收敛范围要明显比Newton型迭代法大，稳定性又不弱于Boggs的A-稳定方法[3]，但计算量要明显小于后者. 所以，隐式方法是可靠性和效率都很高的方法. 但在Jain的文章里，一是方法的构造过程十分复杂，难以形成一类方法，二是没能从理论上说明方法的收敛阶. 鉴于此，本文利用嵌入法思想给出了一类隐式迭代法，这类方法包含了Jain所给出的方法. 文中还给出了算法收敛阶的有关结论. 最后结合几个敏感问题从数值计算角度对方法进行了验证，计算结果表明了方法的有效性.

二 隐式迭代法

考虑非线性方程组

$$F(x)=0, \tag{1}$$

其中$F:D\subset R^n\to R^n$是充分光滑的向量函数，D为有界闭凸集. 设已求出(1)的孤立解x^*的第k个近似值x^k，考虑Newton同伦方程

$$H(x,t)=F(x)-(1-t)F(x^k)=0. \tag{2}$$

在一定条件下，当x^k充分接近x^*时，(2)的解$x=x(t)$，$t\in[0,1]$存在，且满足Cauchy问题

$$\begin{cases}\dfrac{dx}{dt}=-F'(x)^{-1}F(x^k),\\ x(0)=x^k,\end{cases} \tag{3}$$

* 1992年10月25日收到. 94年4月6日收到修改稿.

且 $x(1)=x^*$. 从 x^k 出发，以 1 为步长，应用 R 级隐式 Runge-Kutta 法来求解(3)，则可得到 x^* 的下一近似值，记为 x^{k+1}：

$$\begin{cases} x^{k+1} = x^k + \sum_{i=1}^{R} b_i K_i \\ K_i = -F'(x^k + \sum_{j=1}^{R} a_{ij} K_j)^{-1} F(x^k), \quad i = 1, \cdots, R. \end{cases} \tag{4}$$

对 $k=0,1,\cdots$ 重复上述过程，则(4)形式了一类隐式迭代法. 做为特例，由 1 级 2 阶、2 级 4 阶、3 级 6 阶隐式 R－K 法，可导出下列相应的方程组求解的迭代法：

$$\begin{cases} x^{k+1} = x^k + K_1, \\ K_1 = -F'(x^k + \frac{1}{2}K_1)^{-1} F(x^k), \quad k = 0,1,\cdots, \end{cases} \tag{5}$$

$$\begin{cases} x^{k+1} = x^k + \frac{1}{2}K_1 + \frac{1}{2}K_2, \\ K_1 = -F'(x^k + \frac{1}{4}K_1 + \frac{3-2\sqrt{3}}{12}K_2)^{-1} F(x^k), \\ K_2 = -F'(x^k + \frac{1}{4}K_2 + \frac{3+2\sqrt{3}}{12}K_1)^{-1} F(x^k), \quad k = 0,1,\cdots, \end{cases} \tag{6}$$

及

$$\begin{cases} x^{k+1} = x^k + \frac{5}{18}K_1 + \frac{4}{9}K_2 + \frac{5}{18}K_3, \\ K_1 = -F'(x^k + \frac{5}{36}K_1 + \frac{10-3\sqrt{15}}{45}K_2 + \frac{25-6\sqrt{15}}{180}K_3)^{-1} F(x^k), \\ K_2 = -F'(x^k + \frac{10+3\sqrt{15}}{72}K_1 + \frac{2}{9}K_2 + \frac{10-3\sqrt{15}}{70}K_3)^{-1} F(x^k), \\ K_3 = -F'(x^k + \frac{25+6\sqrt{15}}{180}K_1 + \frac{10+3\sqrt{15}}{45}K_2 + \frac{5}{36}K_3)^{-1} F(x^k), \end{cases} \tag{7}$$

其中(5)就是[2]中所给的隐式 Newton 法，(6)就是[1]中所给的隐式 5 阶法.

下面考虑(4)的收敛阶. 假设导出(4)的 R－K 方法在相容性[4]的意义下是 p 阶的.

令 $e=F(x^k)/\|F(x^k)\|$，$s=\|F(x^k)\|$. 考虑 $g(t):=x(t/s)$，则

$$\begin{cases} g'(t) = x'(\frac{t}{s})\frac{1}{s} = -F'(g)^{-1}e, \\ g(0) = x^k. \end{cases}$$

应用 p 阶 R－K 方法于(4)，有

$$\| g(h) - (g(0) + h\sum_{i=1}^{R} b_i K_i) \| \leqslant \Gamma h^{p+1},$$

其中 Γ 是和步长 h 无关的常数. 容易验证，当 $h=s$ 时，$g(0) + s\sum_{i=1}^{R} b_i K_i = x^{k+1}$. 因此有

$$\begin{aligned} \| x^* - x^{k+1} \| &= \| g(s) - x^{k+1} \| \leqslant \Gamma s^{p+1} \\ &\leqslant \Gamma \| F(x^*) - F(x^k) \|^{p+1} \leqslant \Gamma M \| x^* - x^k \|^{p+1}. \end{aligned}$$

这里 $M=\sup\limits_{x^*\in D}\sup\limits_{0\leq t\leq 1}\|F'(x^*+t(x^k-x^*))\|<+\infty$. 于是我们有

结论 如果导出迭代法(4)的 R－K 方法在相容性的意义下是 p 阶的，则相应的迭代法是 $p+1$ 阶的.

做为直接的推论，(5)，(6)，(7)分别为 3,5,7 阶的迭代法.

三 算法的实现

应用隐式迭代法时，每一步要解由 $K_i(i=1,\cdots,R)$所形成的非线性方程组. 可以采用简单迭代法(见[1],[2]). 这里，使用效率较高的 Chipman 的等效代换法[5]：

$$L^{(q)}=((A\otimes I)^{-1}-\mathrm{diag}(B_k))^{-1}\begin{bmatrix}-F'(x^k+L_1^{(q-1)})^{-1}F(x^k)-B_kL_1^{(q-1)}\\ \vdots \\ -F'(x^k+L_R^{(q-1)})^{-1}F(x^k)-B_kL_R^{(q-1)}\end{bmatrix}, \tag{8}$$

$$q=1,2,\cdots,M,$$

$$\overline{K}_i=(A\otimes I)^{-1}L^{(M)},\quad i=1,2,\cdots,R, \tag{9}$$

$$x^{k+1}=x^k+\sum_{i=1}^{R}b_i\overline{K}_i,\quad k=0,1,\cdots, \tag{10}$$

其中 M 为整数，或者事先给定，或者由 $\|L^{(q)}-L^{(q-1)}\|<\varepsilon$ 来决定，其中 ε 为给定的小数. $L^{(q)}=(L_1^{(q)},\cdots,L_R^{(q)})^T$，取 $L^{(0)}=0$. $B_k\in L(R^n)$可以取为函数$-F'(x)^{-1}F(x^k)$在 x^k 的 Jacobi 矩阵，这里用有限差商矩阵来近似它.

$$\mathrm{diag}(B_k)=\begin{bmatrix}B_k & & & \\ & B_k & & \\ & & \ddots & \\ & & & B_k\end{bmatrix}_{n\times R},$$

$A=(a_{ij})_{R\times R}$，I 为 $n\times n$ 单位阵，$A\otimes I$ 为 A 和 I 的 Kroecker 积.

算法的效率依赖于 M 和具体要求解的问题.

四 应用实例

采用的算法是由隐式 Newton 法(5)做为基本迭代所形成的算法(8)－(10). 终止迭代的准则为 $\|x^{k+1}-x^k\|\leqslant 10^{-10}$.

例 1 考虑方程组[3]

$$F(x)=\begin{bmatrix}x_1^2-x_2+1\\ x_1-\cos(\frac{\pi x_2}{2})\end{bmatrix}=0, \tag{11}$$

初值 $x^0=(1,0)$，解 $x^*=(0,1)$. Newton 迭代法不收敛于该解，而收敛于(－1,2). 阻尼 Newton 法需要 321 次函数值计算，107 次迭代法才收敛于 x^*. 而[3]中最成功的 A-稳定方法

PE_BCE_B 需要 17 次迭代，36 次函数计算. 本文算法仅需 5 次迭代，10 次函数计算. 这里取 $M=2$. 迭代结果见表 1

表 1

k	x_1^k	x_2^k
1	2.153312E−2	5.187562E−1
2	−2.851191E−2	1.006743E+0
3	−6.546050E−5	1.000042E+0
4	4.911271E−8	1.000000E+0
5	7.549789E−8	1.000000E+0

例 2 考虑方程组

$$F(x)=\begin{bmatrix}\frac{1}{2}[\sin(x_1x_2)-\frac{x_2}{2\pi}-x_1] \\ (1-\frac{1}{4\pi})(e^{2x_1}-e)+\frac{ex_2}{\pi}-2ex_1\end{bmatrix}=0, \tag{12}$$

初值 $x^0=(0.4,3)$，所求解在 (0.30, 2.8) 附近. Newton 法收敛于 (−0.26, 0.26). PE_BCE_BC[3] 需要 46 次函数计算才收敛于此解. 而我们的算法仅需 13 次函数计算，总的迭代次数为 7. 其中当 $k=0,1,\cdots,5$ 时，取 $M=2$；当 $k=6$ 时，取 $M=1$. 计算结果见表 2.

表 2

k	x_1^k	x_2^k
1	3.890950E−1	2.957794E+0
2	5.663928E−1	2.946072E+0
3	3.240205E−1	2.876179E+0
4	3.007634E−1	2.840173E+0
5	2.994495E−1	2.836929E+0
6	2.994495E−1	2.836928E+0
7	2.994487E−1	2.836928E+0

另外，还针对 [6] 所给例子将本文方法和光滑与阻尼方法进行了比较，无论在计算量方面还是在收敛速度方面，本文方法均优于 [6] 中方法.

参 考 文 献

[1] M.K.Jain, *Fifth order implicit multipoint method for solving equations*, BIT, 1985, 2, 250—255.

[2] M.K.Jain, *Implicit Newton iterative method*, J. Math. Phy. Sci., 1986, 2, 169—173.

[3] P.T.Boggs, *The solutions of nonlinear systems of equations by A-stable integration techniques*, SIAM J. Numer. Anal., 1971, 28(8), 767—785.

[4] R.Gekeler, *Diser. Methods for Stable IVP's*, Springer Lecture Notes in Math, 1044, 1982.

[5] F.H.Chipman, *A-stable Runge-Kutta processes*, BIT, 11(1971), 384—388.

[6] R.P.Twearson, *Use of smoothing and damping techniques in the solution of nonlinear equations*, SIAM Review, 33(1) (1977), 35—45.

A Note On Jain's Implicit Method

Han Bo Zhang Gangliang

(Dept. of Math., Harbin Institute of Technology, Harbin 150001)

Abstract

By the principle of embedding method, a class of implicit methods for solving nonlinear equations is constructed. The theory analysis on the convergent order is given. Comparisions of computational results are made with other well-known methods on a number of difficult problems.

Keywords nonlinear equations, implicit iterative methods, embedding method.