

非线性方程的基于重新参数化的裁剪求根方式

金佳培 陈小雕 史甲尔 陈立庚

(杭州电子科技大学计算机学院 杭州 310018)

摘 要 非线性方程的求根在计算机辅助几何设计、计算机图形学、信号处理、机器人等方面有着较为广泛的应用。文中提出基于重新参数化的三次裁剪求根算法,该算法可以用于非多项式方程的求根。首先,求解出插值四点的三次多项式;然后,寻找重新参数化函数,使得复合的插值多项式也插值对应的导数,从而提升对应的逼近阶和收敛阶。与已有的三次裁剪方法相比,所提方法能达到 9 次或更高的收敛阶。在区间内单根且有理三次裁剪方法需要计算包围多项式的某些情形下,所提方法可以包住对应的根。实例表明,在某些 Newton 方法失效的情形下,该方法也可以收敛到相应的实根。

关键词 非线性方程求根,重新参数化,三次裁剪,收敛阶

中图分类号 TP391.41 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.03.010

Reparameterization-based Clipping Method for Root-finding Problem of Non-linear Equations

JIN Jia-pei CHEN Xiao-diao SHI Jia-er CHEN Li-geng

(School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract The root-finding problem of non-linear equations has wide applications in computer aided geometric design, computer graphics, robotics, etc. This paper presented a reparameterization-based cubic clipping method for finding the roots of a non-linear equation. Firstly, it computes a cubic polynomial interpolating the given smooth function $f(t)$ at four points. Then, it searches two reparameterization functions so that the reparameterized functions have the same derivatives, which leads to higher approximation order and convergence rate. Compared with the prevailing cubic clipping methods, the new method can achieve the convergence rate 9 or higher for single root cases, and directly bound the root without computing the bounding polynomials. Numerical examples show that it can converge to the proper solution even in some cases that Newton's methods fail.

Keywords Root-finding of non-linear equations, Reparameterization, Cubic clipping, Convergence rate

1 引言

非线性方程求根问题在计算机辅助几何设计、计算机图形学、机器人和计算机辅助制造等方面有着广泛的应用^[1-2]。很多文献讨论了多项式方程的求根问题^[3-12]。理论上,结合 Sturm 定理和笛卡儿法则的二分法可以用于求解多项式的实根,但是其收敛速度过慢,导致计算效率较低。裁剪方法由于具有较快的收敛速度且能快速定位到实根,受到越来越多的关注^[4-7]。相对于多项式的幂级数表示的求根方法,基于 Bernstein 基函数表示的求根方法具有较好的计算稳定性^[3]。Barton 和 Liu 等人^[4-5]提出了多项式的 k -裁剪求根方法。一步裁剪过程的基本思路是:在给定的小区间内,先用降阶逼近的方法求得逼近给定多项式的 k 次多项式;然后估计逼近多项式和给定多项式的误差上下界,从而得到给定多项式的上

下包围多项式;最后利用包围多项式的实根裁剪掉不包围给定多项式实根的小区间。重复上述步骤,直到小区间的长度小于给定的误差或者小区间不包含实根为止。 k -裁剪方法可以求解出给定区间内的所有实根。 k -裁剪方法的收敛阶为 $(k+1)/s$,其中 s 为对应实根的重数。Chen 等人提出了基于 \mathbb{R}^2 空间二次曲线逼近的求根方法,相应的收敛阶可从 $3/s$ 提升为 $4/s$ ^[6]。随后,文献^[7]提出了有理三次裁剪方法,进一步将三次裁剪方法的收敛阶从 $4/s$ 提升为 $7/s$ 。上述方法在最坏情况下需要计算对应的包围多项式,相应的时间复杂度为 $O(n^2)$,其中 n 为给定多项式的次数。

文献^[8]讨论了线性复杂度的求根算法,该算法可以在线性时间内构造出给定多项式的包围多项式,相应的收敛阶为 $5/s$ 。

本文讨论基于重新参数化的裁剪求根方法。在区间内只

到稿日期:2017-07-18 返修日期:2017-08-14 本文受国家自然科学基金项目(61672009)资助。

金佳培(1991—),女,硕士生,主要研究方向为计算机图形学;陈小雕(1976—),男,博士,教授,硕士生导师,CCF 会员,主要研究方向为计算机辅助几何设计、计算机图形学,E-mail: xiaodiao@hdu.edu.cn(通信作者);史甲尔(1993—),男,硕士生,主要研究方向为计算机图形学;陈立庚(1998—),男,主要研究方向为计算机图形学。

有一个单根的情形下,本文方法采用直接包住多项式实根的方式,避免了包围多项式的求解,对应的时间复杂度是线性的。采用重新参数化的方法,可以将对应三次裁剪方法的逼近阶提升到10,将直接包住实根小区间的逼近阶提升到9。实例说明了本文方法具有较高的收敛阶和计算效率。

2 基于重新参数化的求根裁剪方法

理论上,可以使用已有的实根隔离方法或裁剪方法来隔离出给定多项式 $f(t)$ 在区间 $[a, b]$ 内所有的实根。本文假定 $f(t)$ 在 $[a, b]$ 内只有一个单根 t^* 。一步裁剪的过程就是裁剪掉不包含实根的小区间,并留下可能包含实根的小区间。基于重新参数化的求根裁剪过程可分为如下3个步骤:1)选取区间 $[a, b]$ 的两个端点以及两个内点,得到插值四点的三次多项式 $g(t)$,并计算其实根 t_0 ;2)渐进式计算对应的重新参数化函数 $\varphi_1(t)$ 和 $\varphi_2(t)$,使得插值多项式经过重新参数化,能插值相应的3或4个点处的导数值;3)计算 $\varphi_1(t_0)$ 和 $\varphi_2(t_0)$ 的值,然后通过 $2\varphi_2(t_0) - \varphi_1(t_0)$ 和 $\varphi_1(t_0)$ 直接包住 $f(t)$ 的实根 t^* 。

2.1 求解三次插值多项式

不妨设 $h=b-a$,4个插值点为 $s_i, i=1,2,3,4$ 。三次插值多项式 $g(t)$ 由式(1)确定。

$$f(s_i) = g(s_i), i=1,2,3,4 \tag{1}$$

由式(1)可得,存在 $\xi(s)$ 使得:

$$f(s) - g(s) = \frac{f^{(4)}(\xi(s))}{4!} (s-s_1)(s-s_2)(s-s_3)(s-s_4) \tag{2}$$

一般地,可简单地设 $s_i = a + \frac{i-1}{3}(b-a), i=1,2,3,4$,此时对应的逼近阶为4,即 $|f(s) - g(s)| = O(h^4)$ 。我们利用 $g(t)$ 的实根 t_0 来逼近 $f(t)$ 的实根 t^* 。注意到: $|f(t_0)| = |f(t_0) - g(t_0)| = O(\prod_{i=1}^4 (t_0 - s_i))$ 和 $|t_0 - t^*| = O(|f(t_0) - f(t^*)|) = O(|f(t_0)|)$,本文采用如下的插值点选取方法进行优化,虽然此过程中整体误差的逼近阶仍然为4,但可以使得 t_0 与 t^* 之间达到更高的逼近阶。本文将 s_2 选定为两端点连线和 x 轴的交点 t_1 (若 $s_2 \neq t_1$),则 $|t_0 - s_2| = |t_0 - t_1| = O(h^2)$,且对应应有 $|t_0 - t^*| = O(h^5)$;若 $s_2 = t_1$,则选择 $s_3 = t_2$,其中 t_2 为插值 $f(t)$ 于 $t=s_1, s_2, s_4$ 3点的二次多项式的根,其中 $|t_0 - s_3| = |t_0 - t_2| = O(h^4)$,则有 $|t_0 - t^*| = O(h^8)$ 。

2.2 求解对应的重新参数化函数 $\varphi_1(t)$ 和 $\varphi_2(t)$

通过上述步骤得到了三次多项式 $g(t)$ 。设 $\lambda_i = \frac{g'(s_i)}{f'(s_i)}$,则令 $\varphi_1(t)$ 和 $\varphi_2(t)$ 是未知系数,分别是由式(3)和式(4)确定的六次多项式和七次多项式,并可由多点泰勒展开公式渐进地求解得到,式(3)中的 α 为1或4且满足 $|s_\alpha - t_1| \geq |s_5 - a - t_1|$ 。

$$\varphi_1(s_i) = s_i, \varphi_1'(s_j) = \lambda_j, i=1,2,3,4; j=\alpha,2,3 \tag{3}$$

$$\varphi_2(s_i) = s_i, \varphi_2'(s_j) = \lambda_j, i=1,2,3,4; j=1,2,3,4 \tag{4}$$

设 $F_1(t) = f(\varphi_1(t)), F_2(t) = f(\varphi_2(t))$ 。结合式(1)、式(3)和式(4)可分别验证得到:

$$\begin{cases} F_1(s_i) = g(s_i), & i=1,2,3,4 \\ F_1'(s_j) = g'(s_j), & j=\alpha,2,3 \end{cases} \tag{5}$$

$$\begin{cases} F_2(s_i) = g(s_i), & i=1,2,3,4 \\ F_2'(s_j) = g'(s_j), & j=1,2,3,4 \end{cases} \tag{6}$$

由式(5)和式(6)可得到:

$$\begin{cases} f(\varphi_1(t_0)) = F_1(s) - g(s) = O(\frac{\prod_{i=1}^4 (s-s_i)^2}{(s-s_{5-a})}) \\ f(\varphi_2(t_0)) = F_2(s) - g(s) = O(\prod_{i=1}^4 (s-s_i)^2) \end{cases} \tag{7}$$

根据式(7),若 $s_2 \neq t_1$,有:

$$\begin{cases} |\varphi_1(t_0) - t^*| = O(h^9) \\ |\varphi_2(t_0) - t^*| = O(h^{10}) \end{cases} \tag{8}$$

2.3 求解包围小区间

若 $f(\varphi_1(t_0))$ 和 $f(\varphi_2(t_0))$ 异号,则 $\varphi_1(t_0)$ 和 $\varphi_2(t_0)$ 已经包住实根 t^* ;否则,由式(8)可知,当 h 比较小且使得:

$$|\varphi_1(t_0) - t^*| > 2|\varphi_2(t_0) - t^*| \tag{9}$$

时,可以证明, $\varphi_1(t_0)$ 和 $2\varphi_2(t_0) - \varphi_1(t_0)$ 可以包住 t^* 。具体过程为:设 $\varphi_1(t_0) < t^*$,由式(9)得, $2\varphi_2(t_0) - \varphi_1(t_0) - t^* = 2(\varphi_2(t_0) - t^*) + (t^* - \varphi_1(t_0)) \geq |\varphi_1(t_0) - t^*| - 2|\varphi_2(t_0) - t^*| > 0$,即 $\varphi_1(t_0) < t^* < 2\varphi_2(t_0) - \varphi_1(t_0)$ 。

2.4 数值实例

本节举例说明 M_1 (本文方法)的算法步骤及效果,并与已有方法 $M_1^{[5]}, M_2^{[7]}$ 和 $M_3^{[8]}$ 进行比较。

例1 给定 $f(t) = (t-0.2)(t+2)^3(t+5)^4, t \in [0, 1]$,如图1所示。首先计算端点连线与 x 轴的交点,并选定 $s_2 = t_1 = 0.0345$ 。选择 $s_1 = 0, s_2 = 0.0345, s_3 = 1/3, s_4 = 1$,计算得到三次插值多项 $g(t) = -1000 + 2790.67t + 6341.64t^2 + 19861.3t^3, g(t) = 0$ 的根为 $t_0 = 0.203768$ 。然后计算得到重新参数化函数,如图2所示, $t_2 = \varphi_1(t_0) = 0.199203, t_3 = \varphi_2(t_0) = 0.200888$ 。最后根据2.3节,得到小区间 $[0.199203, 0.200888]$ 。更多迭代次数及其对比结果如表1所列。

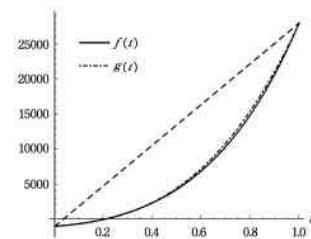


图1 给定曲线 $f(t)$ 、逼近曲线及端点连线

Fig. 1 The given curve $f(t)$, the approximation curve and the line segment connecting two endpoints

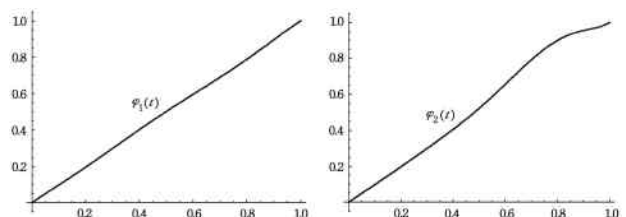


图2 重新参数化函数 $\varphi_1(t)$ 及 $\varphi_2(t)$

Fig. 2 Reparameterization function $\varphi_1(t)$ and $\varphi_2(t)$

表 1 不同方法针对例 1 的计算结果

Table 1 Computation results of different methods for example 1

Method	1	2	3	4	5	CR	T/ms
M ₁	5.9e-2	4.5e-7	1.4e-27	1.6e-109	2.1e-437	4	18.5
M ₂	8.7e-5	8.5e-34	6.9e-237	1.7e-1658	/	7	15.6
M ₃	1.1e-2	7.8e-14	7.9e-71	5.9e-357	1.0e-1788	5	4.8
M ₄	4.6e-5	1.0e-27	9.8e-251	2.4e-2352	/	9	1.6

3 更多的实例与讨论

二分法中,1 次运算约需要一次 $f(t)$ 的计算,使得区间长度缩短为原来的 2^{-1} 。本文裁剪方法中,一次裁剪过程需要 8 次 $f(t)$ 或 $f'(t)$ 的计算;设给定小区间的长度是 10^{-r} ,则 1 次裁剪运算后小区间长度的期望值变成 10^{-9r} ,约等于 $80^{r/3}$ 次二分运算的收敛效果。本文的例子中,裁剪算法的计算效率明显高于二分方法的效率,因此本文专注于裁剪方法之间的比较,省略与二分方法的比较。

例 2 采用下列 4 个多项式进行比较实验,初始给定区间为 $[0,1]$,根据对应控制多边形的实根来进行划分,得到不同的给定小区间,使得每个小区间包含更少的实根,从而提升对应的收敛速度。

$$f_2(t) = (t - 0.20001)^2(t + 0.5)^5(t - 0.7)(t - 1.1)^6, t \in [0.126, 0.310]$$

$$f_3(t) = (t - 0.2)^3(t - 5)^7(2 + t)^2(t + 7)^4, t \in [0, 1]$$

$$f_4(t) = (t - 0.2)(t + 6)^3(t - \frac{2}{3})(t - \frac{4}{5})(t - 3)^2, t \in [0.128, 0.584]$$

$$f_5(t) = (t - 0.2)(t + 6)^3(t - \frac{4}{5})^2(t - 3)^2, t \in [0.137, 0.653]$$

对于重根的情况,本文方法中使用 $F(t) = \frac{f(t)}{f'(t)}$ 来替代

原函数 $f(t)$ 。表 2 给出了不同方法针对例 2 的计算结果。表 3 给出了前 5 次迭代得到的包围小区间的长度,即误差 e_i 和第一次裁剪过程对应的平均计算时间。每次迭代的数值收敛阶可用 $\frac{\lg(e_{i+1})}{\lg(e_i)}$ 来近似估计。从表 1 可以看出,本文方法 M₄ 具有相对更好的收敛阶和更高的计算效率。

表 2 不同方法针对例 2 的计算结果

Table 2 Computation results of different methods for example 2

Exam	Method	1	2	3	4	5	CR	T/ms
f ₂	M ₁	2.1e-2	2.7e-4	4.3e-8	1.1e-15	6.9e-31	2.0	24.3
	M ₂	3.4e-3	2.9e-9	1.6e-30	6.3e-105	2.5e-365	3.5	21.6
	M ₃	6.4e-3	1.6e-6	1.2e-17	9.1e-51	3.1e-150	3.0	6.3
	M ₄	2.0e-8	4.3e-69	8.0e-641	3.5e-6029	/	9.0	3.6
f ₃	M ₁	5.0e-1	8.9e-2	4.6e-3	6.6e-5	2.3e-7	1.6	26.4
	M ₂	9.7e-2	3.3e-4	5.4e-10	1.8e-23	5.3e-55	2.4	22.6
	M ₃	2.9e-1	2.9e-2	5.3e-4	4.2e-7	1.4e-12	1.7	6.1
	M ₄	1.4e-8	4.8e-80	3.2e-751	6.3e-7078	/	9.0	3.8
f ₄	M ₁	1.3e-3	3.1e-14	1.0e-56	1.1e-226	1.7e-906	4.0	19.1
	M ₂	6.9e-7	8.3e-48	3.2e-334	3.5e-2339	/	7.0	18.2
	M ₃	2.0e-4	3.1e-21	2.4e-105	6.3e-526	8.4e-2629	5.0	6.1
	M ₄	7.9e-4	1.6e-26	1.6e-237	1.4e-2133	/	9.0	2.0
f ₅	M ₁	1.7e-3	5.9e-14	8.1e-56	2.8e-223	3.8e-893	4.0	18.8
	M ₂	1.3e-6	5.7e-46	1.6e-321	2.6e-2250	/	7.0	18.5
	M ₃	3.0e-4	1.5e-20	5.0e-102	2.0e-509	2.0e-2546	5.0	5.5
	M ₄	1.2e-3	2.0e-25	4.5e-229	4.5e-2058	/	9.0	1.9

例 3 本文方法也适用于非多项式表示的曲线。给定 $f_6(t) = 10^{(150-5t^2)} - 1$,它在初始区间 $[5.464, 5.494]$ 中有一个实根,如图 3 所示。对 f_6 应用类牛顿方法 MLi^[14]、Msha^[15] 以及本文方法 M₄ 得到的计算结果如表 3 所列。从表 3 可以看到,M₄ 能较好地处理非多项式情况,对应的收敛阶为 9。而 MLi 以及 Msha 方法在初始点 $t_0 = 5.494$ 处发散,这在一定程度上说明了本文方法具有相对较好的稳定性。

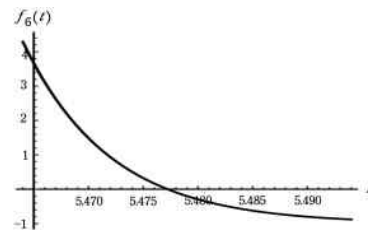


图 3 例 3 中的给定曲线 $f_6(t)$

Fig. 3 The given curve $f_6(t)$ of example 3

表 3 例 3 的计算结果

Table 3 Computation results of example 3

f ₆	1	2	3	4	CR	T/ms
M ₄	4.5e-5	1.4e-27	1.5e-233	2.5e-2084	9	2.3
MLi	1.6e-147	溢出				17.5
Msha	41.5	溢出				0.47

结束语 本文讨论了基于重新参数化的三次裁剪方法,并将其用于非线性方程的求根计算。与已有的三次裁剪方法相比,本文方法在单步裁剪过程中具有较高的计算效率和收敛阶。本文方法还可以直接应用于非多项式方程的求根。与

Newton方法相比,本文方法在Newton方法发散的某些情况下仍可以快速收敛到正确的根。

本文方法仍有不足,即目前假定给定的小区间内只有一个单实根。理论上,对于单变量的多项式方程,有很多方法可以用于隔离出所有的实根。但是,对于非多项式的方程,今后还需要研究相应的实根隔离方法。另外,对于重实根,目前只是简单地应用 $F(t) = \frac{f(t)}{f'(t)}$ 来处理。在小区间内当 $f'(t)$ 有实根时, $F(t)$ 变成无界,从而会削弱上述方法的处理效率。研究更好的重根处理方法以提高相应的收敛速度和计算稳定性,也将成为今后的工作之一。

参 考 文 献

- [1] CHOI Y K, WANG W, LIU Y, et al. Continuous Collision Detection for Two Moving Elliptic Disks[J]. IEEE Transactions on Robotics, 2006, 22(2): 213-224.
- [2] ELBER G, KIM M S. Geometric constraint solver using multivariate rational spline functions[C]// ACM Symposium on Solid Modeling and Applications. DBLP, 2001: 1-10.
- [3] FAROUKI R T, GOODMAN T. On the optimal stability of the Bernstein basis [J]. Mathematics of Computation, 1996, 65(216): 1553-1566.
- [4] BARTOŃ M, JÜTTLER B. Computing roots of polynomials by quadratic clipping[J]. Computer Aided Geometric Design, 2007, 24(3): 125-141.
- [5] LIU L, ZHANG L, LIN B, et al. Fast approach for computing roots of polynomials using cubic clipping[J]. Computer Aided Geometric Design, 2009, 26(5): 547-559.
- [6] CHEN X D, MA W. A planar quadratic clipping method for computing a root of a polynomial in an interval[J]. Computers & Graphics, 2015, 46(C): 89-98.
- [7] CHEN X D, MA W, YE Y. A rational cubic clipping method for computing real roots of a polynomial [J]. Computer Aided Geometric Design, 2015, 38: 40-50.
- [8] CHEN X D, MA W. Rational cubic clipping with linear complexity for computing roots of polynomials[J]. Applied Mathematics & Computation, 2016, 273: 1051-1058.
- [9] EFREMOV A, HAVRAN V, SEIDEL H P. Robust and numerically stable Bézier clipping method for ray tracing NURBS surfaces[C]// Spring Conference on Computer Graphics. 2005: 127-135.
- [10] JÜTTLER B. The dual basis functions for the Bernstein polynomials[J]. Advances in Computational Mathematics, 1998, 8(4): 345-352.
- [11] MØRKEN K, REIMERS M. An unconditionally convergent method for computing zeros of splines and polynomials [J]. Mathematics of Computation, 2007, 76(258): 845-865.
- [12] WEI F F, ZHOU F, FENG J Q. Survey of real root finding of univariate polynomial equation in CAGD/CG [J]. Journal of Computer-Aided Design and Computer Graphics, 2011, 23(2): 193-207. (in Chinese)
卫飞飞, 周飞, 冯结青. CAGD/CG 领域中一元多项式方程求根问题综述[J]. 计算机辅助设计与图形学学报, 2011, 23(2): 193-207.
- [13] DAVIS P J. Interpolation and approximation[J]. Unt Theses & Dissertations, 2001, 3(1): 59-67.
- [14] LI X, MU C, MA J, et al. Sixteenth-order method for nonlinear equations[J]. Applied Mathematics & Computation, 2010, 215(10): 3754-3758.
- [15] SHARMA J R, ARORA H. A new family of optimal eighth order methods with dynamics for nonlinear equations[J]. Applied Mathematics & Computation, 2016, 273: 924-933.
- (上接第 57 页)
- [10] JEONG B, KIM H, LEE Y J, et al. Exponential polynomial reproducing property of non-stationary symmetric subdivision schemes and normalized exponential B-splines[J]. Advances in Computational Mathematics, 2013, 38: 647-666.
- [11] LI B J, YU Z L, YU B W, et al. Non-stationary Subdivision for Exponential Polynomials Reproduction[J]. Acta Mathematicae Applicatae Sinica(English Series), 2013, 29(3): 567-578.
- [12] YU B W. A Family of Subdivision Schemes for Exponential Polynomials Reproduction[D]. Dalian: Dalian University of Technology, 2008. (in Chinese)
郁博文. 一类重构指数多项式的细分方法[D]. 大连: 大连理工大学, 2008.
- [13] LI B J. Construction of Exponentials Reproducing Subdivision Schemes and Rapid Evaluation of Interpolatory Subdivision Surfaces[D]. Dalian: Dalian University of Technology, 2009. (in Chinese)
李宝军. 指数多项式曲线细分重构与插值细分曲面快速计算[D]. 大连: 大连理工大学, 2009.
- [14] DYN N, LEVIN D, LUZZATTO. Exponentials reproducing subdivision schemes[J]. Foundations of Computational Mathematics, 2003, 3: 187-206.
- [15] DYN N, HORMANN K, SABIN M, et al. Polynomial reproduction by symmetric subdivision schemes[J]. Journal of Approximation Theory, 2008, 15: 28-42.
- [16] ROMAN S. The formula of Faà di Bruno[J]. American Mathematical Monthly, 1980, 10: 805-809.
- [17] CONTI C, ROMANI L, YOON J. Approximation order and approximate sum rules in subdivision[J]. Journal of Approximation Theory, 2016, 207(C): 380-401.