

复平面上计算超越函数零点的一种有效算法

王 则 柯
(数学力学系)

摘 要

超越方程数值求解,是数值分析中的一个重要课题。迄今已有多种求根方法出现,但 these 方法多限于处理实变量实值函数的情形,并限于求一个根。

基于采用 J_3 剖分和整数标号法的一种互补轮迴的不动点算法,本文提出在复平面上同时计算超越函数若干个零点的一种有效算法。

§1 引 言

超越方程的数值求解,即超越函数的零点计算,是科学研究和工程设计中经常遇到的问题。

不久以前,徐利治、朱自强^[1]就函数零点计算,提出三个值得进一步研究的问题,并估计(一)和(三)较为困难,(二)应不难解决。其(一)和(三)是:

(一)就有限阶超越整函数求复零点的问题,能否找到在一定范围内具有大范围收敛性的迭代求根公式?……

(三)能否找到不需要计算函数二阶导数的大范围收敛的迭代公式?……

随后,徐利治、陈永昌在[2]中,提出超越函数求实零点的一种迭代方法。方法要求,函数是有限阶整函数,对实变数取实值(即限制在实数域,是一个实变量实值函数)。算法的执行,是在使 $f(x_0) \approx 0$ 的初值 x_0 两侧,单调地收敛到 $f(z)$ 在 x_0 右侧及左侧最邻近的实零点。

基于不动点算法的先前的工作[5],本文提出求超越函数复零点的一种算法。不动点算法的要点是剖分——标号——互补轮迴。就迭代公式而言,则只需计算函数值,并不涉及任何一阶导数。我们只要求 $f(z)$ 连续。最后,我们的算法在复平面上寻找函数零点,大体上具有想找几个就找几个,想在什么地方附近找,就在什么地方附近找的能力。所以,本文的工作部分地回答了文[1]提出的问题。

§2 算 法

设函数 $f: C \rightarrow C'$ 是连续的,这里, C 是复数 $z = x + iy$ 平面, C' 是复数 $w = u + iv$ 平

面, $w = f(z)$ 。

不动点算法的基本做法是:对空间进行适当的单纯形剖分;对剖分的顶点进行适当的标号;用一种有规则的搜索方法,寻求所谓完全标号单纯形,作为近似解。第三代方法的特点之一,是搜索步长能自动加细,以求得符合任何精度要求的近似解。

我们就按剖分、标号、搜索规则的顺序,叙述我们的算法。

1. $C \times [0, 1)$ 的一种剖分

记 $C_d := C \times \{1 - 2^{-d}\}$, $d = 0, 1, 2, \dots$ 。选定参考点 $z_c = x_c + iy_c$ 。给定初始格距 (C_0 平面格距) $h > 0$ 。对 C_d 平面用直线族

$$\begin{aligned} x &= x_c + p \cdot h \cdot 2^{-d}, \\ y &= y_c + q \cdot h \cdot 2^{-d}, \\ y &= y_c \pm (x - x_c) + 2r \cdot h \cdot 2^{-d} \end{aligned}$$

进行三角剖分,这里 p, q, r 表整数(图1)。显然, d 越大, C_d 的剖分越细,这是因为 C_d 的剖分格距为 $h \cdot 2^{-d}$ 而剖分网径为 $\text{mesh } C_d = \sqrt{2} h \cdot 2^{-d}$ 。

易见, C_d 平面上一个正方形,与 C_{d+1} 平面上四个正方形上下相对。 C_d 与 C_{d+1} 之间这样一个方块,按图2规则剖分成四面体。

这样,就得到 $C \times [0, 1)$ 的一个单纯形剖分。我们把 $C \times [0, 1)$ 的这个单纯剖分记为 G , G 的顶点集记为 $V(G)$ 。算法将在 $V(G)$ 中进行。

为记号的方便, $(z, 1 - 2^{-d}) \in V(G)$ 简记为 z_d , 即: z_d 中的 z 表顶点的复数值,下标 d 表此顶点所在的 C_d 平面的层数 d 。

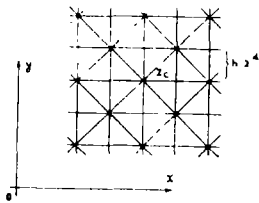


图 1

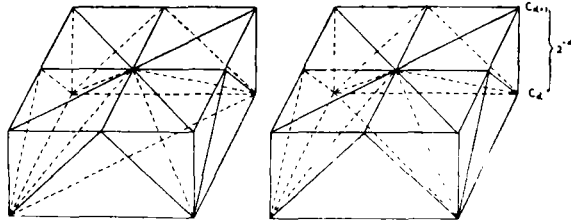


图 2

2. 标号法

设 n 为欲求根个数。

令 $f_0(z) = (z - z_c)^n$, $f_d(z) = f(z)$, $d \geq 1$ 。得到 $\{f_d\}: C_d \rightarrow C'$ 。

在 G 的顶点集 $V(G)$ 上,规定标号法 $L: V(G) \rightarrow \{0, 1, 2\}$ 如下:

设 $z_d = (z, 1 - 2^{-d}) \in V(G)$, 那末

$$L(z_d) = \begin{cases} 0, & \text{若 } -\pi/3 \leq \arg f_d(z) \leq \pi/3 \text{ 或 } f_d(z) = 0, \\ 1, & \text{若 } \pi/3 < \arg f_d(z) \leq \pi, \\ 2, & \text{若 } -\pi < \arg f_d(z) < -\pi/3, \end{cases}$$

这里, C' 平面上复数幅角取值范围规定为 $-\pi < \arg w \leq \pi$ 。

称 G 中一个三角形为完全标号三角形, 如果它的顶点具有0、1、2全部三个标号.

完全标号三角形三顶点的象, 在 C' 平面上分踞原点的“三侧”. 算法就是要找尺寸越来越小的完全标号三角形, 作为函数的零点.

3. 互补轮回的变维数算法

(1) 一维搜索 (DQ)

令 $m := \lceil 3(1 + \sqrt{2})n/4\pi \rceil + 1$. 在 C_0 平面上取定以 $z_c = x_c + iy_c$ 为中心、半边长 $m \cdot h$ 的正方形 Q_m . 注意在 C_0 上顶点的标号是由 $w = (z - z_c)^n$ 给出的, 易证正方形边界 ∂Q_m 上, 按逆时针方向顺序, 正好有 n 个棱, 标号由0变到1(参看[3]或[4]). 先把这 n 个0 \rightarrow 1棱找到, 作为一维搜索向二维搜索过渡的进口, 这种0 \rightarrow 1棱, 记作 (z^0, z^1) .

(2) 二维搜索 (IQ)

步1. 从每个 (z^0, z^1) 棱出发, 在 C_0 平面上, 有唯一的一个顶点 z' 使得 z^0, z^1, z' 成为剖分中的一个三角形的三个顶点, 并且 $z^0 \rightarrow z^1 \rightarrow z'$ 具有正的定向—— C_0 平面上的逆时针方向. 计算 z' 的标号 $L(z')$.

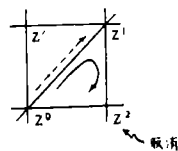


图 3

步2. 若 $L(z') = 2$, 令 $z^2 := z'$. 此时, C_0 平面上一个完全标号三角形已经找到, 转入三维搜索. 否则,

步3. 若 $L(z') = 1$, 令 $z^1 := z'$, 回到步1.

二维搜索总可以在有限步内找到 C_0 平面上一个完全标号三角形 (z^0, z^1, z^2) 作为转入三维搜索的进口而结束, 并且整个过程必在 Q_m 内进行, 所找到的完全标号三角形也必在 Q_m 内.

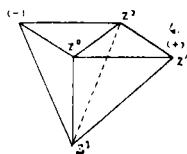


图 4

(3) 三维搜索 (LT)

步0. 当达到精度要求时, 打印 $z^* := (z^0 + z^1 + z^2)/3$ 作为一个零点.

步1. 若完全标号三角形在 C_0 平面上具有负的定向, 即 $z^0 \rightarrow z^1 \rightarrow z^2$ 构成顺时针方向, 则取消 z^2 , 变成0 \rightarrow 1棱 (z^0, z^1) , 回到二维搜索. (图3. 这种情况只有在从三维搜索回到 C_0 上 Q_m 内降为二维搜索时才会发生.) 否则,

步2. 存在 $V(G)$ 中唯一的顶点 z' 使得 z' 在 z^0, z^1, z^2 所决定的平面的正侧, 即 z^0, z^1, z^2, z' 符合右手螺旋法则(图4). 计算 $L(z')$. 若 $L(z') = 1$, 则令 $z^1 := z'$, 回到步0.

4. 主框图

算法的进行, 基于同标号顶点替换规则: 若 $L(z') = 1$, 则令 $z^1 := z'$, 而原来的 z^1 (如果有的话)取消, 这就是所谓互补轮回过程. 我们的算法从一维搜索开始, 进入二维搜索, 再进入三维搜索(可能有限次回退到二维搜索, 但终将重新进入三维搜索), 在三维搜索中解决问题. 这属于所谓变维数算法.

算法的具体过程是这样的: 从 ∂Q_m 右方中央的 $z^0 = z_c + mh + oi$ 开始, 逆时针方向在 ∂Q_m 上搜索 z^0 , 再找到 z^1 , 就取代原来的 z^0 , 直到找到 z^1 , 一维搜索结束. 转入二维搜索, 直至找到 z^2 . 然后开始三维搜索, 直至以 d 的增大为标志达到精度要求, 打印 z^* :

$= (z^0 + z^1 + z^2)/3$ 得到一个零点。这时将 z^0, z^1, z^2 清掉，回到 ∂Q_m 上，从上次 $0 \rightarrow 1$ 棱的下一点开始，重复上述过程(图 5)。

这样一个一个搜索零点。当在 ∂Q_m 上走了一圈，最后走到 $z^5 = z_c + mh - hi$ 时， ∂Q_m 上的 n 个 $0 \rightarrow 1$ 棱出发点已经用完，计算结束，停机。

在我们的程序中，在 ∂Q_m 上搜索的目前到达位置，用 z^5 表示。 z^0, z^1, z^2 专门记最新的标号为 0、1、2 的点。三维搜索时， (z^0, z^1, z^2) 三角形两侧各恰有一点 z^3, z^4 分别与 z^0, z^1, z^2 合成 G 中的一个四面体。 z^3 和 z^4 究竟哪个在 (z^0, z^1, z^2) 的正侧，用行列式判断。正侧的一点作为 z' (见图 4)，然后进行同标号替换。

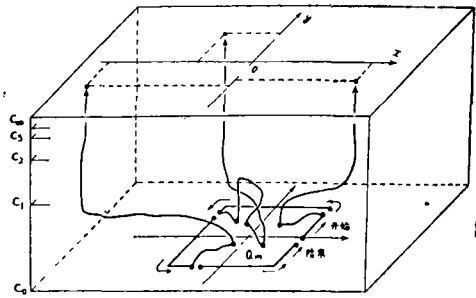
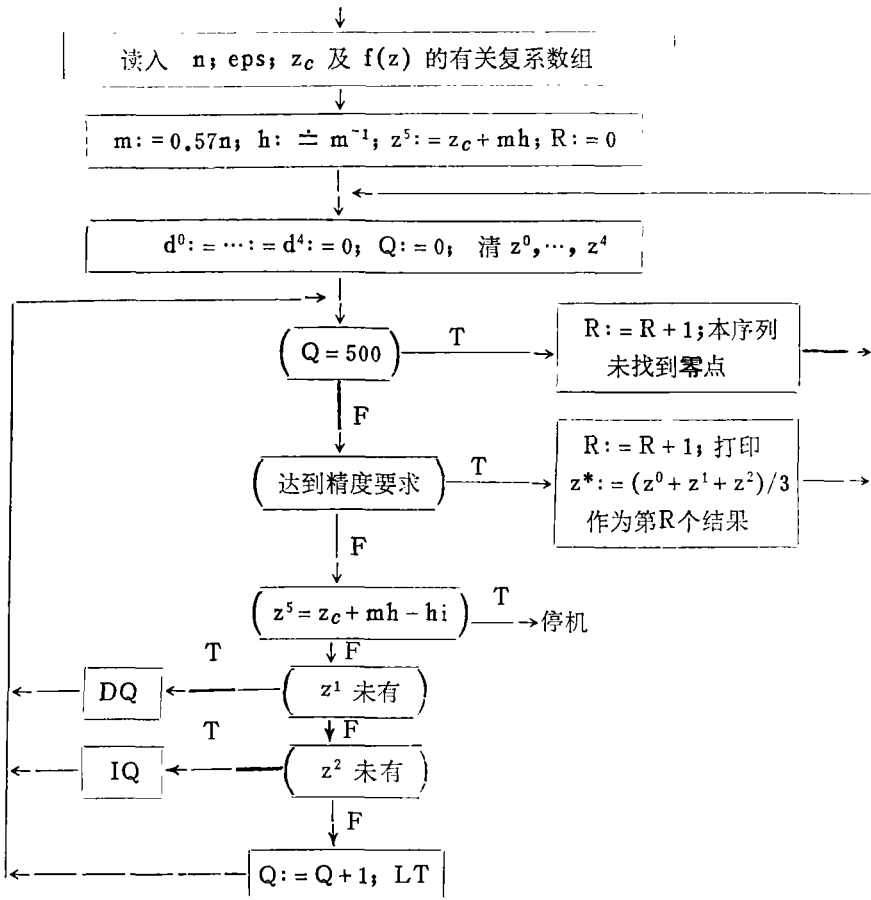


图 5

有些超越函数，根本没有有限零点。鉴于超越函数零点分布的复杂性，程序中设置了“在规定时间内(例如迭代次数 $Q = 500$)本计算点列未找到函数零点”的保护性出口。主框图如下：



5. 算法分析

我们把从每个 ∂Q_m 上的 $0 \rightarrow 1$ 棱出发的互补轮迴过程所经过的完全标号三角形，顺次记作 $\{z^0(k), z^1(k), z^2(k)\}$ ，叫做一个计算点列。

当 $f(z)$ 是 n 次多项式时，[3]已证明，每个计算点列恰收敛到多项式的一个根，无一遗漏。而当 $f(z)$ 是超越函数时，情况要复杂得多。超越函数的零点的个数及分布，远不象多项式那样确定。一大类超越函数具有无穷多个零点，分布于复数平面的广大区域；而另一些超越函数，则没有有限零点。相应的收敛性条件，比较复杂，将另行讨论。这里，只证明下述命题。首先需要

引理 设 $\{z^0, z^1, z^2\}$ 是由 f 标号的一个完全标号三角形，并且对 $0 \leq i < j \leq 2$ 成立

$$|f(z^i) - f(z^j)| \leq \varepsilon,$$

那末，对 $i = 0, 1, 2$ ，有

$$|f(z^i)| \leq 2\varepsilon / \sqrt{3}.$$

证明 以 G_0, G_1, G_2 分别记 w 平面上幅角范围为 $[-\pi/3, \pi/3]$ 、 $(\pi/3, \pi]$ 、 $(-\pi, -\pi/3]$ 扇形区域，以 $\rho_{i0}, \rho_{i1}, \rho_{i2}$ 分别表示 $f(z^i)$ 与 G_0, G_1, G_2 的距离。按所设，必须 $\rho_{01} \leq \varepsilon, \rho_{02} \leq \varepsilon$ 。但 $f(z^0) \in G_0$ (图 6)， $f(z^0)$ 必在棱形 $OCAB$ 内，立得

$$|f(z^0)| \leq 2\varepsilon / \sqrt{3}.$$

同理可得 $|f(z^1)| \leq 2\varepsilon / \sqrt{3}, |f(z^2)| \leq 2\varepsilon / \sqrt{3}$ 。证完。

命题 设 $\{z^0(k), z^1(k), z^2(k)\}$ 是一个上述完全标号三角形计算点列，在 C 平面上有一个有限聚点 z^* ，那末， z^* 是 f 的一个零点： $f(z^*) = 0$ 。

证明 据命题所设，可知所论的计算点列有子序列收敛到 z^* 。由计算点列的构造法，不妨设此子序列仍具有 $\{z^0(k), z^1(k), z^2(k)\}$ 的形式。于是，有 $R > 0$ 使对任何自然数 k 及 $i = 0, 1, 2$ ，成立 $|z^i(k)| < R$ 。

记 $G(R) := \{z \mid |z| \leq R\}$ 。因 $f(z)$ 是全平面的连续函数， $f(z)$ 在 $G(R)$ 上一致连续；有 $M > 0$ 使对任何 $z', z'' \in G(R)$ 成立

$$|f(z') - f(z'')| \leq M \cdot |z' - z''|.$$

现在，计算点列在以 $\{z_c\} \times [0, 1)$ 为轴、半径为 R 的圆筒内无限发展，每次都走到新的四面体。但对任何自然数 d ，圆筒内 C_d 以下的四面体数目有限，所以计算点列必然无限向圆筒的上底 $G(R) \times \{1\}$ 靠近，其标志是点列所走到的层数 $d \rightarrow \infty$ ，当 $k \rightarrow \infty$ 。于是，对任意 $\varepsilon > 0$ ，有 K 使 $k \geq K$ 时，对 $0 \leq i < j \leq 2$ 成立

$$|z^i(k) - z^j(k)| < \varepsilon / M,$$

进而

$$|f(z^i(k)) - f(z^j(k))| \leq M \cdot |z^i(k) - z^j(k)| < \varepsilon.$$

复由引理，对 $k \geq K$ 及 $i = 0, 1, 2$ ，成立

$$|f(z^i(k))| \leq 2\varepsilon / \sqrt{3}.$$

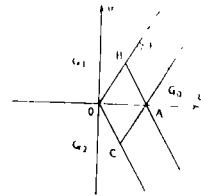


图 6

但 $\lim_{k \rightarrow \infty} z^i(k) = z^*$, $i = 0, 1, 2,$

$\therefore f(z^*) = 0$, 证完.

§3 例子

对形如 $\sum_{k=1}^n P_k(z) \sin(A_k + B_k z) = 0$ 的方程及形如 $\sum_{k=1}^n P_k(z) \exp(A_k + B_k z) = 0$ 的方程进行了数值试验, 这里, $P_k(z)$ 是复变量 z 的复系数多项式, A_k, B_k 是复系数. 按照所编的程序, 只要输入欲求根个数 n , 预期点 z_c 的坐标, 精度要求 eps 以及超越函数的复系数组, 均可输出 n 个计算结果.

在以下例子中, 所有计算在国产 DJS-21 机上以单精度进行.

〔例1〕 $f(z) = 3z - 1 - \cos z$, 在原点 $(0, 0)$ 附近求6个零点, 得

$$\begin{aligned} z_1 &= -34.404666 && + 5.3512989i, \\ z_{2,6} &= 5.6824440 && \pm 3.6605012i, \\ z_{3,5} &= -2.2466720 && \pm 3.2069886i, \\ z_4 &= 0.60710164 && - 0.99341074 \times 10^{-9}i; \end{aligned}$$

而在 $(0, -4)$ 附近求7个零点, 得

$$\begin{aligned} z_1 &= 18.596141 && + 4.7290072i, \\ z_2 &= 0.60710164 && - 0.99341074 \times 10^{-9}i, \\ z_{3,4} &= -2.2466720 && \pm 3.2069886i, \\ z_5 &= -15.424482 && + 4.5897234i, \\ z_6 &= -9.0097238 && - 4.1148543i, \\ z_7 &= 5.6824440 && - 3.6605012i. \end{aligned}$$

〔例2〕 $f(z) = z - \pi + i + i \cos z$, 在 $(0, 0)$ 附近求4个零点, 得

$$\begin{aligned} z_1 &= 3.1415926 && + 1.6161375i, \\ z_2 &= -17.049109 && + 3.7255550i, \\ z_3 &= -1.3063376 && - 2.2325410i, \\ z_4 &= 3.1415926 && - 0.14901161 \times 10^{-8}i; \end{aligned}$$

而在 $(7, 4)$ 附近求6个零点, 得

$$\begin{aligned} z_1 &= 10.521699 && + 2.8123008i, \\ z_2 &= 3.1415926 && + 1.6161375i, \\ z_3 &= -4.2385140 && + 2.8123008i, \\ z_4 &= 20.273070 && - 3.5457825i, \\ z_5 &= 3.1415926 && - 0.99341074 \times 10^{-9}i, \\ z_6 &= 7.5895229 && - 2.2325410i. \end{aligned}$$

注意在本例中, 变量 z 取实值时, 函数 $f(z)$ 不取实值.

常见的超越函数, 许多是以指数运算为特征的, 所以, 在计算中, 溢出是容易发生的. 这自然不是算法方面的问题. 但作为应用软件, 尚可进一步改善.

参 考 文 献

- [1] 徐利治、朱自强, 关于某些大范围收敛的迭代方法的评述, 高等学校计算数学学报, 1979, 2, 131—142.
- [2] 徐利治、陈永昌, 关于整函数求实零点的一族大范围收敛的迭代法, 计算数学, 2(1980), 2, 172—179.
- [3] Kuhn, H.W., *Fixed Points: Algorithms and Applications*, Ed. S. Karamardian, Academic Press, New York (1977), 11—40.
- [4] 王则柯, 代数基本定理的一个构造性证明, 数学的实践与认识, 1980, 4.
- [5] 王则柯, Kuhn算法的程序实施与数值试验, 数值计算与计算机应用, 2 (1981).

An Efficient Algorithm for Computing the Zeros of Transcendental Functions in the Complex plane

Wang Zeke

Abstract

The finding of numerical solutions of transcendental equations is a problem of interest in Numerical Analysis. Up to now, various methods have been presented in finding the roots of the equations. However, these methods are mainly limited to deal with the case of real-value functions in a real variable, and are mostly limited to determine only one root.

This paper proposes a method to compute simultaneously several zeros of a transcendental function in the complex plane. It is done through a special application of the complementary pivoting algorithm using a J_3 triangulation and an integer labelling. The details of the algorithm with necessary mathematical proofs are given. Finally, some numerical results are reported to show the efficiency of the method.