椭圆函数的精细积分改进算法*1)

姚 征
(大连海事大学, 交通与物流工程学院, 工程力学教研室, 辽宁大连 116026)

钟万勰
(大连理工大学, 运载工程与力学学部工程力学系, 工业装备结构分析国家重点实验室, 辽宁大连 116023)

摘 要

椭圆函数是一种特殊的双周期复变函数, 广泛应用于工程问题中, 尤其非线性问题中居多. 在工程中遇到的椭圆函数以二阶椭圆函数为主, 而且很多复杂的椭圆函数都可以通过变换由二阶椭圆函数得到. 二阶椭圆函数包括 Jacobi 椭圆函数和 Weierstrass 椭圆函数. 它们都可以进行幂级数展开, 直接计算很不方便. 椭圆函数的重要性质之一就是具有加法定理, 因此可利用精细积分法求解. 虽然椭圆函数的精细积分算法在精度和效率上取得了较大成功, 但椭圆函数的奇点问题仍然存在并对计算精度构成一定威胁. 在回顾并分析椭圆函数的精细积分算法的基础上, 通过对椭圆函数奇点的分析, 给出了椭圆函数可去奇点的近似公式, 并在此基础上进一步改进并完善了椭圆函数的精细积分算法.

关键词: 精细积分, Jacobi 椭圆函数, 双周期, 奇点, 可去奇点

MR (2000) 主题分类: 33E05

THE IMPROVED PRECISE INTEGRATION METHOD FOR ELLIPTIC FUNCTIONS

Yao Zheng
(*Transportation and Logistics Engineering College , Dalian Maritime University, Dalian 116026, Liaoning, China*)

Zhong Wanxie
(*Department of Engineering Mechanics, State Key Laboratory of Structural Analysis for Industrial Equipment, Dalian University of Technology, Dalian 116024, Liaoning, China*)

Abstract

Elliptic functions are a special kind of double period complex functions and are used in engineering widely, especially in nonlinear problems. Many elliptic functions in engineering problems are second order elliptic functions, and many complicated elliptic functions are found to be obtained from second order elliptic functions. The most familiar second order elliptic functions include Jacobi elliptic functions and Weierstrass elliptic functions. They can be expressed as power series expansion, directly calculating is inconvenient. One of the most important properties of elliptic functions is the additional theorem, so that the method of precise integration can be invoked. Although the precise integration method of elliptic

* 2006 年 11 月 28 日收到.
1) 国家重点基础研究专项经费资助项目 (2005CB321704) 和国家自然科学基金 (10632030) 资助项目.

functions has achieved many successes in precision and efficiency, but the singularity problem is still a big enemy of precision. The precise integration method of Jacobi elliptic function is reviewed first. The approximate formulae of removable singularity are educed after analyzing of the singularity. Then the improved precise integration method is presented based on those formulae and analyses.

Keywords: Precise integration method, Jacobi elliptic function, double periods, singularity, Removable Singularity

2000 Mathematics Subject Classification: 33E05

1. 引　　言

在工程问题中常遇到椭圆函数数值求解的问题 [1−4], 该问题可以通过查表和使用软件来解决, 例如 Matlab、Maple 和 IMSL 等软件就有求解椭圆函数的模块. 由于工程中的非线性问题的复杂性从而带来了对算法的速度与精度的要求, 尤其大量需要迭代的工程问题, 对算法的速度与稳定性要求很高. 椭圆函数在非线性问题中频繁出现, 因此研究椭圆函数的高效稳定算法是十分必要的.

精细积分法 (PIM) 首先解决了时不变系统的时间积分问题, 并取得了达到计算机精度的高精度结果. 精细积分法有 2 个要点: 1) 运用 2^N 类算法 [5−7]: 2) 把注意力放在增量上, 而不是全量. 解决时不变系统之后, 各种近似方法被广泛的应用于求解其他问题, 椭圆函数的精细积分算法就是其中之一 [8]. 通过利用椭圆函数的加法定理, 精细积分法成功的应用于两类二阶椭圆函数的数值求解. 与其他软件与算法相比, 在精度与速度上获得了较大的突破 [8,9]. 本文进一步分析了困扰各算法精度的奇点问题, 并给出了处理椭圆函数奇点的可行算法.

2. Jacobi 椭圆函数的精细积分算法

第二类 Legendre 椭圆积分可以定义为 [1]

$$u=\int_0^Z \frac{\mathrm{d}\xi}{\sqrt{(1-\xi^2)(1-m_o^2\xi^2)}}, \tag{2.1}$$

Jacobi 椭圆函数 $\mathrm{sn}(u,m_o)=z$ 定义为第二类 Legendre 椭圆积分的反函数. 另外两个基本 Jacobi 椭圆函数分别定义为

$$\mathrm{cn}u=\sqrt{1-\mathrm{sn}^2u}=\sqrt{1-z^2},\qquad \mathrm{dn}u=\sqrt{1-m_o^2\mathrm{sn}^2u}=\sqrt{1-m_o^2z^2}, \tag{2.2}$$

参数 m_o 称为模数. 这三个 Jacobi 椭圆函数是全纯 (holonormic) 函数, 其幂级数展开式分别为 [1,2]

$$\begin{aligned}
&\mathrm{sn}(u,m_o)=u-(1+m_o^2)u^3/3!+(1+14m_o^2+m_o^4)u^5/5!-\cdots\\
&\mathrm{cn}(u,m_o)=1-u^2/2+(1+4m_o^2)u^4/4!-(1+44m_o^2+16m_o^4)u^6/6!-\cdots\\
&\mathrm{dn}(u,m_o)=1-m_o^2u^2/2+(4m_o^2+m_o^4)u^4/4!-(16m_o^2+44m_o^4+m_o^6)u^6/6!-\cdots
\end{aligned} \tag{2.3}$$

Jacobi 椭圆函数有加法定理 [1]

$$\begin{aligned}
\mathrm{sn}(u+v,m_o) &= \frac{\mathrm{sn}(u,m_o)\cdot\mathrm{cn}(v,m_o)\cdot\mathrm{dn}(v,m_o)+\mathrm{sn}(v,m_o)\cdot\mathrm{cn}(u,m_o)\cdot\mathrm{dn}(u,m_o)}{1-m_o^2\mathrm{sn}^2(u,m_o)\mathrm{sn}^2(v,m_o)},\\
\mathrm{cn}(u+v,m_o) &= \frac{\mathrm{cn}(u,m_o)\cdot\mathrm{cn}(v,m_o)-\mathrm{sn}(u,m_o)\cdot\mathrm{dn}(u,m_o)\cdot\mathrm{sn}(v,m_o)\cdot\mathrm{dn}(v,m_o)}{1-m_o^2\mathrm{sn}^2(u,m_o)\mathrm{sn}^2(v,m_o)},\\
\mathrm{dn}(u+v,m_o) &= \frac{\mathrm{dn}(u,m_o)\cdot\mathrm{dn}(v,m_o)-m_o^2\mathrm{sn}(u,m_o)\cdot\mathrm{cn}(u,m_o)\cdot\mathrm{sn}(v,m_o)\cdot\mathrm{cn}(v,m_o)}{1-m_o^2\mathrm{sn}^2(u,m_o)\mathrm{sn}^2(v,m_o)}.
\end{aligned} \tag{2.4}$$

当 $u=v$ 时

$$\begin{aligned}
\mathrm{sn}(2u,m_o) &= \frac{2\mathrm{sn}(u,m_o)\cdot\mathrm{cn}(u,m_o)\cdot\mathrm{dn}(u,m_o)}{1-m_o^2\mathrm{sn}^4(u,m_o)},\\
\mathrm{cn}(2u,m_o) &= \frac{\mathrm{cn}^2(u,m_o)-\mathrm{sn}^2(u,m_o)\cdot\mathrm{dn}^2(u,m_o)}{1-m_o^2\mathrm{sn}^4(u,m_o)},\\
\mathrm{dn}(2u,m_o) &= \frac{\mathrm{dn}^2(u,m_o)-m_o^2\mathrm{sn}^2(u,m_o)\cdot\mathrm{cn}^2(u,m_o)}{1-m_o^2\mathrm{sn}^4(u,m_o)}.
\end{aligned} \tag{2.5}$$

为了进行数值积分, 引入一个小的空间步长, 记为: η. 于是产生一系列等步长的空间间隔为

$$u_0=0\,,\quad u_1=\eta\,,\quad u_k=k\eta,\ \cdots,\quad u_{2^L}=2^L\eta=u. \tag{2.6}$$

假设这三个 Jacobi 椭圆函数位于 η 点的值都已经求得, 那么通过利用加法定理 (2.4) 就可方便的求出函数位于 u 点的值. 因此, 原问题变为求解 Jacobi 椭圆函数在 η 点的值. 令

$$\tau=\eta/m,\qquad m=2^N, \tag{2.7}$$

由于 η 本来是不大的空间区段, 则 $\tau=\eta/m$ 将是非常小的一个空间区段了. 因此对于空间区段 τ, 有:

$$\begin{aligned}
\mathrm{sn}\,\tau &\approx \tau-\frac{1}{3!}\left(1+m_o^2\right)\tau^3+\frac{1}{5!}\left(1+14m_o^2+m_o^4\right)\tau^5=S_1,\\
\mathrm{cn}\tau &\approx 1-\frac{1}{2!}\tau^2+\frac{1}{4!}\left(1+4m_o^2\right)\tau^4-\frac{1}{6!}\left(1+44m_o^2+16m_o^4\right)\tau^6=1+C_1,\\
\mathrm{dn}\tau &\approx 1-\frac{1}{2!}m_o^2\tau^2+\frac{1}{4!}\left(4m_o^2+m_o^4\right)\tau^4-\frac{1}{6!}\left(16m_o^2+44m_o^4+m_o^6\right)\tau^6=1+D_1.
\end{aligned} \tag{2.8}$$

因 τ 很小, 幂级数 4 项展开应以足够. 在上式中 S_1, C_1 和 D_1 相对于单位 1 是非常小的量. 因此在计算过程中至关重要的一点是数值的存储只能是 (2.8) 式中的 S_1, C_1 和 D_1, 而不是 $1+C_1$, $1+D_1$. 因为 C_1, D_1 很小, 当它们与单位 1 相加时, 就会成为尾数, 在计算机的舍入操作中, 其精度将丧失殆尽. S_1, C_1 和 D_1 就是增量 [5,7]. 这就是以上提到的精细积分法的第二个要点.

把 (2.8) 式代入 (2.5) 式, 并令 $\mathrm{sn}\,v=S_i$ 可以得到:

$$\begin{aligned}
\mathrm{sn}\,(2v) &= \mathrm{F_S}(S_i) \;= \frac{2S_i\,(1+C_i)\,(1+D_i)}{1-m_o^2S_i^4}=S_{2i},\\
\mathrm{cn}\,(2v) &= 1+\mathrm{F_C}(C_i)=1+\frac{k^2S_i^4+2C_i+C_i^2-S_i^2\,(1+D_i)^2}{1-m_o^2S_i^4}=1+C_{2i},\\
\mathrm{dn}\,(2v) &= 1+\mathrm{F_D}(D_i)=1+\frac{m_o^2S_i^4+2D_i+D_i^2-m_o^2S_i^2\,(1+C_i)^2}{1-m_o^2S_i^4}=1+D_{2i}.
\end{aligned} \tag{2.9}$$

因此, 通过 (2.9) 式便可得到 S_m, C_m 和 D_m, 从而求得 $\mathrm{sn}(\eta)$, $\mathrm{cn}(\eta)$ 和 $\mathrm{dn}(\eta)$. 这样的计算一共需要执行 N 次, 而且只有 S_i, C_i 和 D_i $(i = 2^0, 2^1, \ldots, 2^N)$ 被计算并存入内存. (2.9) 式的 N 次计算相当于一下语句:

$$\begin{aligned}&\mathrm{for}(\mathrm{iter} = 0; \mathrm{iter} < N; \mathrm{iter} + +)\\&\{i = 2^{iter};\ S_{2i} = \mathrm{F_S}(S_i);\ C_{2i} = \mathrm{F_C}(C_i);\ D_{2i} = \mathrm{F_D}(D_i);\}\end{aligned} \tag{2.10}$$

当以上语句的循环结束后, 再执行

$$\begin{aligned}&\mathrm{sn}(m\tau) = \mathrm{sn}(\eta) = S_m\\&\mathrm{cn}(m\tau) = \mathrm{cn}(\eta) = 1 + C_m\\&\mathrm{dn}(m\tau) = \mathrm{dn}(\eta) = 1 + D_m\end{aligned} \tag{2.11}$$

便可. 由于 N 次叠代后 S_m, C_m 和 D_m 已不再是很小的量了, 这个加法已没有严重的舍入误差了 [5,7]. 以上便是 Jacobi 椭圆函数的精细积分算法. Weierstrass 椭圆函数也有类似的精细积分算法, 由于篇幅关系这里便不再给出.

3. Jacobi 椭圆函数的奇点处理方法

Jacobi 椭圆函数存在奇点, 在一些特殊点也有给定值. 在奇点附近函数值会趋于无穷大, 因此常产生较大误差. 要进一步改进椭圆函数的精细积分算法, 就需要改善奇点处的误差. 如果模数 $m_o^2 < 1$, 则有

$$K = \int_0^1 [(1 - t^2)(1 - m_o^2 t^2)]^{-1/2} \mathrm{d}t, \tag{3.1}$$

模数 m_o 有补充模数 m_o', 它们有关系 $m_o^2 + {m_o'}^2 = 1$. 同时

$$K' = \int_0^1 [(1 - t^2)(1 - {m_o'}^2 t^2)]^{-1/2} \mathrm{d}t, \tag{3.2}$$

Jacobi 椭圆函数的 2 个周期就是由 K, K' 来表示的. 通过利用椭圆函数的双周期性质, 可将任意二阶椭圆函数转化到包含原点的胞腔里面. (3.1), (3.2) 两式可以利用级数展开或数值积分得到, 但精度不高. 实际上在可以用收敛很快的西他函数 (Theta) 来计算 m_o 的函数 K, K'[1]. 由于西他函数收敛很快, 所以可以快速得到精度很高的结果. 在自变量 u 的复平面上, Jacobi 椭圆函数 $\mathrm{sn}(u, m_o), \mathrm{cn}(u, m_o), \mathrm{dn}(u, m_o)$ 分别有周期的平行四边形胞腔. 例如 $\mathrm{sn}(u, m_o)$ 的平行四边形的四个角点为

$$[0, 4K, 4K + \mathrm{i}2K', \mathrm{i}2K'], \tag{3.3}$$

其中点 $u = \mathrm{i}K'$ 与 $u = 2K + \mathrm{i}K'$ 为 $\mathrm{sn}(u, m_o)$ 的单级罗朗 (Laurent) 奇点. 在奇点附近很小邻域内, 直接数值计算的精度不好. 应将罗朗级数的奇异部分扣除而成为可去奇点, 方可从容加以计算.

在一些特殊点, $\mathrm{sn}(u, m_o)$, $\mathrm{cn}(u, m_o)$, $\mathrm{dn}(u, m_o)$ 的值是已知的

$$\mathrm{sn}(\mathrm{i}K'/2, m_o) = \mathrm{i}/\sqrt{m_o},\ \ \mathrm{cn}(\mathrm{i}K'/2, m_o) = \sqrt{1 + m_o^{-1}},\ \ \mathrm{dn}(\mathrm{i}K'/2, m_o) = \sqrt{1 + m_o}, \tag{3.4}$$

这不是这些函数的奇点. 但还需要在 $u=\mathrm{i}K'/2$ 附近作 Taylor 级数展开. 根据 Taylor 级数展开的公式, 需要这些函数的各阶微商. 这可从微商公式

$$\begin{aligned}&\mathrm{d}[\mathrm{sn}(u)]/\mathrm{d}u=\mathrm{cn}(u)\cdot\mathrm{dn}(u),\\&\mathrm{d}[\mathrm{cn}(u)]/\mathrm{d}u=-\mathrm{sn}(u)\cdot\mathrm{dn}(u),\\&\mathrm{d}[\mathrm{dn}(u)]/\mathrm{d}u=-m_o^2\mathrm{sn}(u)\cdot\mathrm{cn}(u)\end{aligned}\tag{3.5}$$

导出各阶微商, 从而得到在 $u=\mathrm{i}K'/2$ 附近的 Taylor 展开式.

运用加法公式 $\mathrm{sn}(2u,m_o)=[2\mathrm{sn}(u,m_o)\cdot\mathrm{cn}(u,m_o)\cdot\mathrm{dn}(u,m_o)]/[1-m_o^2\mathrm{sn}^4(u,m_o)]$, 即知 $u=\mathrm{i}K'$ 是奇点. 在 $u=\mathrm{i}K'$ 附近可将 $\mathrm{sn}(u,m_o),\mathrm{cn}(u,m_o),\mathrm{dn}(u,m_o)$ 的 Taylor 级数展开式代入, 并将奇异部分减去, 剩下的便是可去奇点的复变函数.

在 $u=v=\mathrm{i}(\tau+K'/2)$ 附近有:

$$\begin{aligned}\mathrm{sn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)&=\mathrm{i}\big/\sqrt{m_o}+\mathrm{i}\tau\cdot\mathrm{cn}(\mathrm{i}K'/2,m_o)\cdot\mathrm{dn}(\mathrm{i}K'/2,m_o)+O(\mathrm{i}\tau^2)\\&=\mathrm{i}\big/\sqrt{m_o}[1+\tau\cdot(1+m_o)]+O(\mathrm{i}\tau^2),\\\mathrm{cn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)&=\sqrt{1+m_o^{-1}}-\mathrm{i}\tau\cdot\mathrm{sn}(\mathrm{i}K'/2,m_o)\cdot\mathrm{dn}(\mathrm{i}K'/2,m_o)+O(\mathrm{i}\tau^2)\\&=\sqrt{1+m_o^{-1}}[1+\tau]+O(\mathrm{i}\tau^2),\\\mathrm{dn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)&=\sqrt{1+m_o}-m_o^2\cdot\mathrm{i}\tau\cdot s\mathrm{n}(\mathrm{i}K'/2,m_o)\cdot c\mathrm{n}(\mathrm{i}K'/2,m_o)+O(\mathrm{i}\tau^2)\\&=\sqrt{1+m_o}[1+\tau\cdot m_o]+O(\mathrm{i}\tau^2),\end{aligned}\tag{3.6}$$

把 (3.6) 带入 (2.4), 利用加法公式并整理可以得到

$$\begin{aligned}&\mathrm{sn}(2(\mathrm{i}\tau+\mathrm{i}K'/2),m_o)\\&=\frac{2\mathrm{sn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)\cdot\mathrm{cn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)\cdot\mathrm{dn}(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)}{1-m_o^2\mathrm{sn}^4(\mathrm{i}\tau+\mathrm{i}K'/2,m_o)}\\&\approx-\frac{2(1+m_o)\mathrm{i}}{\tau m_o}\cdot\frac{1}{4(1+m_o)}\cdot\frac{1+a_1\tau+b_1\tau^2+c_1\tau^3}{1+a_0\tau+b_0\tau^2+c_0\tau^3}\\&=-\frac{\mathrm{i}}{2\tau m_o}-\mathrm{i}\frac{(a_1-a_0)+(b_1-b_0)\tau+(c_1-c_0)\tau^2}{2m_o(1+a_0\tau+b_0\tau^2+c_0\tau^3)}.\end{aligned}\tag{3.7}$$

上式中

$a_1=2+2m_o;\quad db_1=m_o^2+3m_o+1;\quad c_1=m_o(1+m_o);$

$a_0=3(1+m_o)/2;\quad b_0=(1+m_o)^2;\quad c_0=(1+m_o)^3/4.$

同理可以求得

$$\begin{aligned}&\mathrm{cn}(2(i\tau+\mathrm{i}K'/2),m_o)\\&=\frac{\mathrm{cn}^2(i\tau+\mathrm{i}K'/2,m_o)-\mathrm{sn}^2(i\tau+\mathrm{i}K'/2,m_o)\mathrm{dn}^2(i\tau+\mathrm{i}K'/2,m_o)}{1-m_o^2\mathrm{sn}^4(i\tau+\mathrm{i}K'/2,m_o)}\\&\approx-\frac{1}{2\tau m_o}-\frac{(a_2-a_0)+(b_2-b_0)\tau+(c_2-c_0)\tau^2}{2m_o(1+a_0\tau+b_0\tau^2+c_0\tau^3)}.\end{aligned}\tag{3.8}$$

其中

$a_2=2+2m_o;\quad b_2=1+3m_o+3m_o^2;\quad c_2=m_o(1+m_o)(1+2m_o);$

$a_0=3(1+m_o)/2;\quad b_0=(1+m_o)^2;\quad c_0=(1+m_o)^3/4.$

$$\begin{aligned}&\mathrm{dn}(2(i\tau+\mathrm{i}K'/2),m_o)\\&=\frac{\mathrm{dn}^2(i\tau+\mathrm{i}K'/2,m_o)-m_o^2\mathrm{sn}^2(i\tau+\mathrm{i}K'/2,m_o)\mathrm{cn}^2(i\tau+\mathrm{i}K'/2,m_o)}{1-m_o^2\mathrm{sn}^4(i\tau+\mathrm{i}K'/2,m_o)}\\&\approx-\frac{1}{2\tau}-\frac{(a_3-a_0)+(b_3-b_0)\tau+(c_3-c_0)\tau^2}{2(1+a_0\tau+b_0\tau^2+c_0\tau^3)}.\end{aligned}\tag{3.9}$$

其中

$$\begin{array}{lll}a_3=2+2m_o; & b_3=3+3m_o+m_o^2; & c_3=m_o(1+m_o)(2+m_o);\\ a_0=3(1+m_o)/2; & b_0=(1+m_o)^2; & c_0=(1+m_o)^3/4\end{array}.$$

式 (3.7)-(3.9) 即为 Jacobi 椭圆函数的可去奇点公式, 当 $\tau\to 0$ 的时候可以用以上公式计算极点的值.

Jacobi 椭圆函数存在以下的恒等式:

$$\mathrm{sn}^2u+\mathrm{cn}^2u=1,\qquad \mathrm{dn}^2u+m_o^2\mathrm{sn}^2u=1.\tag{3.10}$$

将 (3.7)-(3.9) 式带入 (3.10) 式即得到奇点处的恒等式的值:

$$\mathrm{sn}^2u+\mathrm{cn}^2u\,|_{\mathrm{Singularity}}=0\,;\qquad \mathrm{dn}^2u+m_o^2\mathrm{sn}^2u\,|_{\mathrm{Singularity}}=0,\tag{3.11}$$

我们发现在奇点处恒等式得值由 1 变为 0, 但仍然是有限值.

因为在某些实际应用中, 可能发生函数在运算时奇点部分相互抵消的情况, 从而只剩下已去奇点的函数. 因而推导和求解椭圆函数可去奇点的是一项很有意义的工作. 通过引入可去奇点公式, 原有椭圆函数的精细积分算法可以得到改进, 在奇点附近可选择利用可去奇点公式输出已去奇点的函数值.

但在实际计算中如何判定何种情况下需要使用可去奇点公式 (3.7)-(3.9) 对函数值进行修正呢? 当函数靠近奇点时数值便会急速膨胀, 数值计算所产生的误差也会迅速加大, 从而无法满足恒等式 (3.10), (3.11) 的要求. 本文采用的判定依据就是将数值结果带入恒等式所得出的相对误差. Jacob i 椭圆函数的奇点和周期如表 1 所示 [1]:

表 1 Jacobi 椭圆函数的极点、零点和周期分布

函数	奇点 (极点)	零点	周期
$\mathrm{sn}\,u$	$\mathrm{i}K'$, $2K+\mathrm{i}K'$	0, $2K$	$4K$, $2\mathrm{i}K'$
$\mathrm{cn}\,u$	$\mathrm{i}K'$, $2K+\mathrm{i}K'$	K, $3K$	$4K$, $2K+2\mathrm{i}K'$
$\mathrm{dn}\,u$	$\mathrm{i}K'$, $3\mathrm{i}K'$	$K+\mathrm{i}K$, $K+3\mathrm{i}K$	$2K$, $4\mathrm{i}K'$

从表 1 不难看出这三个 Jacobi 椭圆函数的奇点的分布拥有相同的规律性, 都位于 $2nK+(2m+1)\mathrm{i}K'$, $m,n\in\mathrm{Z}$. 不失一般性, 我们考虑位于第一象限且包含原点的单位胞腔, 此时存在位于 $\mathrm{i}K'$ 的奇点. 利用未作可去奇点处理的精细积分算法计算奇点附近的函数值, 并带入恒等式 (3.10) 求出最大相对误差. 奇点附近误差增加很快, 已无法满足双精度的误差要求, 本文的控制精度采用的是 1×10^{-8} 和 1×10^{-10}. 实际计算得出, 如果需要相对误差满足 1×10^{-10} 量级, 那么要在以奇点为中心, 以 $\frac{5}{1000}K'$ 为半径的圆内采用可去奇点公式; 如果只需要满足 1×10^{-8} 的量级, 那么只需要在奇点为中心, 以 $\frac{5}{10000}K'$ 为半径的圆内采用可去奇点公式即可. 图 1 给出了 $m_o^2=0.5$ 时, 虚轴上奇点 $\mathrm{i}K'=1.85407\mathrm{i}$ 附近未作奇点处理的精细积分算法得

出的相对误差图, 作为对比给出了 IMSL 库函数得出的相对误差图. 从图中可以看出奇点处理只需在靠近奇点的小范围内进行即可, 而且未作处理的精细积分算法在奇点附近的精度比 IMSL 普遍高出大约一个量级 (注意: 图线的纵坐标是对数坐标).

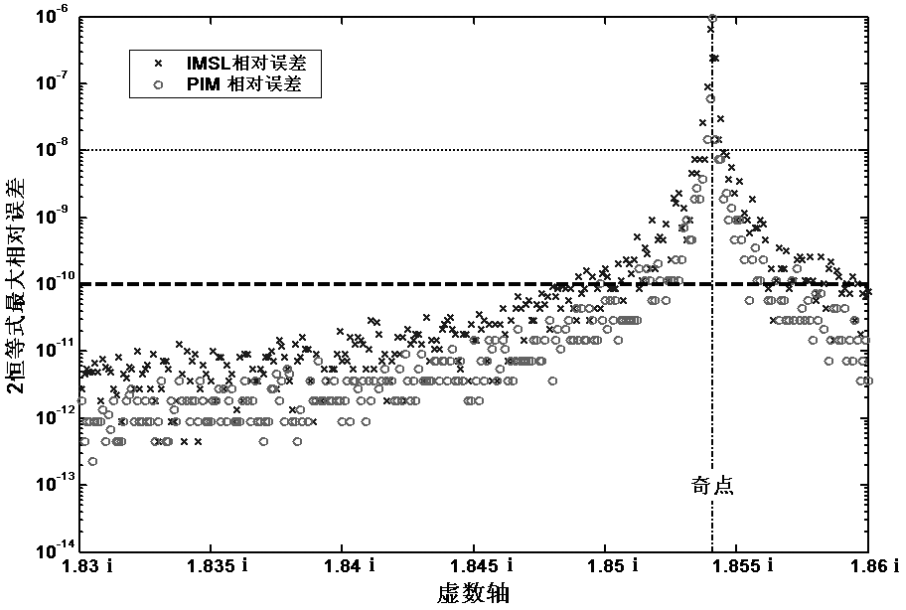

图 1 近奇点处相对误差

注. 任何格阵形状的 Weierstrass 椭圆函数的奇点都分布在单胞四边形边界的 4 个顶点上, 每个单胞包含一个奇点, 坐标原点是 Weierstrass 椭圆函数的一个奇点. 因此利用周期 $(2\omega,\ 2\omega')$ 表示的 Weierstrass 椭圆函数可直接利用周期性化到包含原点的单胞内, 而 Weierstrass 椭圆函数精细积分算法的增量部分恰好就是相应的已去奇点部分 [8]. 而利用不变量 $(g_2,\ g_3)$ 表示的 Weierstrass 椭圆函数由于周期 $(2\omega,\ 2\omega')$ 难以精确求出, 不方便处理, 可转化到 Jacobi 椭圆函数计算. 如能找到用不变量 $(g_2,\ g_3)$ 求解周期值的高精度数值算法, 则可利用周期性直接套用相应的精细积分算法予以计算, 这是进一步研究需要考虑的问题.

4. 精细积分计算过程中可能遇到的奇点和处理方法

如果在计算时发现所待求得函数值位于奇点, 那么就可以利用上文给出的可去奇点公式计算其已去奇点的函数值. 对于非奇点的计算, 由于精细积分算法要执行 N 次迭代运算, 分别计算椭圆函数在 $\tau,\ 2\tau,\ 2^2\tau,\cdots,\ 2^N\tau$ 点的值, 如果恰巧 $2^j\tau\ (0<j<N)$ 位于奇点或很临近奇点那么会导致椭圆函数在 $2^j\tau$ 点的值出现趋近无穷或数值过大的情况, 从而导致 $2^{j+1}\tau$ 点的值无法精确求得.

分析 Jacobi 椭圆函数的奇点和周期表 1 不难发现, 3 个基本 Jacobi 椭圆函数在同一胞腔内的奇点相同, 也就是说在计算过程中会同时出现. 由于椭圆函数的精细积分算法对这三基本椭圆函数是同时计算的, 所以在运算时也是同时遭遇奇点. 因此可在算法中对奇点问题统一

修改.

一般说来 τ, 2τ 都是非常小的值, 不会靠近奇点. 假设已得到了 $2^{j-2}\tau$, $2^{j-1}\tau$ 的值, 而在进行 $2^j\tau$ 的计算时发现公式 (2.4) 的分母部分的模过小, 说明 $2^j\tau$ 位于极点附近, 则可由下列步骤处理:

1) 利用 $2^{j-2}\tau$, $2^{j-1}\tau$ 求得各 Jacobi 椭圆函数在 $(2^{j-2}+2^{j-1})\tau = 3\cdot 2^{j-2}\tau$ 点的值;

2) 利用加法公式求出 $6\cdot 2^{j-2}\tau = 3\cdot 2^{j-2}\tau + 3\cdot 2^{j-2}\tau$ 点的值;

3) 再利用加法公式求出 $8\cdot 2^{j-2}\tau = 6\cdot 2^{j-2}\tau + 2^{j-1}\tau = 2^{j+1}\tau$.

从而跳过奇点附近的运算.

下面给出算例说明算法改进后的稳定性. Jacobi 椭圆函数在每个单胞内存在 2 个奇点, 在靠近奇点的区域是无论是 IMSL 模块还是精细积分法都会产生误差. 如果在复空间划分一个矩形区域 $(-10-10\mathrm{i},\ 10-10\mathrm{i},\ 10+10\mathrm{i},\ -10+10\mathrm{i})$, 当模数 $m_o^2 = 0.5$ 时, 在该区域内三种 Jacobi 椭圆函数都存在 10 多个完整的单胞. 在该区域内沿实轴等间距分别作 n 条虚轴的平行线, 同理沿虚轴也作条实轴的平行线. 空间被刨分出 $(n+1)\times(n+1)$ 个点, 这些点都应满足恒等式 (3.10), 可以分别计算出每个点的值代入 (3.10) 式所得的相对误差的最大值. 随着 n 的增加, 近奇点区域内的点也会增加, 相对误差大的点也会增多. 按照以上算法改进原有 Jacobi 椭圆函数的算法, 并和原有算法以及 IMSL 数学库函数相对比的结果如表 2 所示. (注: 此算例中并未采用可去奇点公式, 仅在精细积分算法中采用了本节的跳过奇点处理)

表 2 改进后精细积分算法的稳定性对比 ($m_o^2 = 0.5$)

点的个数	相对误差 $> 10^{-12}$ 点的个数			最大相对误差		
$n\times n$	改进算法	最初算法	IMSL	改进算法	最初算法	IMSL
40401	2	19	8065	1.455902×10^{-12}	1.025090×10^{-11}	1.290573×10^{-9}
160801	31	99	32319	1.455192×10^{-11}	3.637979×10^{-11}	7.805284×10^{-9}
1002001	158	611	201695	7.275958×10^{-11}	5.238690×10^{-10}	2.393499×10^{-7}

5. 椭圆函数精细积分算法的适用范围和相关比较

从前文的推导中我们不难看出, 所有的推导都是基于椭圆函数的数学定义且未引入假设. 在精细积分算法中, 对于 Jacobi 椭圆函数的模数 m_o 以及 Weierstrass 椭圆函数的不变量 g_2, g_3 并没有作选取范围的限制, 所以精细积分算法可以计算任意的二阶椭圆函数. 作为对比, 在下表中给出了几个常用软件: Matlab、Mathematica、IMSL 数学库和 Maple 对二阶椭圆函数的计算范围与精度的对比.

对于二阶椭圆函数的计算, 这几款软件中只有 Maple 可以和精细积分算法一样计算 Weierstrass 椭圆函数的数值解, 其他都只能计算 Jacobi 椭圆函数, 如表 3 所示:

表 3 各常用软件对二阶椭圆函数的计算范围

算法名称	精细积分	Matlab	IMSL 库	Mathematica	Maple
Jacobi 椭圆函数	可计算	可计算	可计算	可计算	可计算
Weierstrass 椭圆函数	可计算	不可计算	不可计算	不可计算	可计算

表 3 中的算法与软件是都可以对 Jacobi 椭圆函数进行计算, 但对于模数 m_o 的选取范围和结果却有很大差别, 如表 4 所示.

表 4 不同软件对 Jacobi 椭圆函数的模数计算范围

模数 m_o 的范围	$m_o^2<0$	$0<m_o^2<1$	$1<m_o^2<\infty$	m_o 取复数域
精细积分法	可计算	可计算	可计算	可计算
Matlab	—	可计算	—	—
IMSL 库	不准确	可计算	可计算	—
Mathematica	可计算	可计算	可计算	可计算
Maple	不准确 [1)]	可计算	可计算	可计算

1) Maple 在 $m_o^2<0$ 的计算结果与 Mathematica 和精细积分的结果相比, 实部与虚部颠倒.

从以上的表格不难看出, 椭圆函数的精细积分法较其它常用软件有很大的适用范围.

这里对于椭圆函数的应用举一个简单的例子说明, 其他应用可参考文献 [1, 9].

算例 1. 考虑简单的单摆问题. 如图 2 所示, 摆重为 m, 摆长为 l. 假设初始时刻单摆位于最底端, 初始速度为 v_0. 求单摆的运动轨迹.

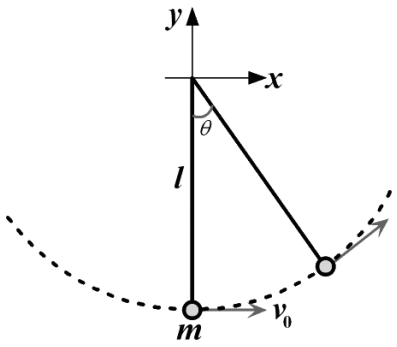

图 2 单摆

设重力加速度为 g, 原点为势能零点, 通过推导可以得到运动方程为

$$l^2(\frac{d\theta}{dt})^2=v_0^2-2gl(1-\cos\theta), \tag{5.1}$$

所以

$$t=l\int_0^\theta\frac{d\theta}{\sqrt{2g(a+l\cos\theta)}}=\frac{l\sqrt{2}}{\sqrt{g(a+l)}}\int_0^\theta\frac{d\frac{\theta}{2}}{\sqrt{1-\frac{2l}{a+l}\sin^2\frac{\theta}{2}}}. \tag{5.2}$$

其中 $a=v_0^2/2g-l$, 令 $z=\sin(\theta/2)$, $m_o^2=2l/(a+l)$, $c=m_o^2\sqrt{(a+l)/2g}$ 得到

$$t=c\int\frac{dz}{\sqrt{(1-z^2)(1-m_o^2z^2)}}. \tag{5.3}$$

求 (5.3) 式的反函数就得到 Jacobi 椭圆函数, 故单摆的运动轨迹是用椭圆函数来表示的. 一般情况下摆角不到 90 度, 那么

$$m_o^2=\frac{2l}{a+l}=\frac{4gl}{v_0^2}>2, \tag{5.4}$$

此时 Matlab 便已经无法进行计算了. 而在实际的工程问题中模数 m_o 为负数或复数的情况常有出现, 此时利用精细积分求解就可以快速得到高精度的正确结果.

6. 结论与展望

本文首先回顾了 Jacobi 椭圆函数的精细积分算法, 指出了精细积分算法在奇点处理部分上需要改进. 进而利用椭圆函数的性质在奇点附近对椭圆函数进行展开, 推导得出了 Jacobi 椭圆函数的可去奇点公式. 从而改善了 Jacobi 椭圆函数的精细积分算法. 在此基础上进一步分析了精细积分法在运算过程中受奇点影响而可能产生的误差, 并给出了解决方案. 通过大量的算例指出, 改进后的椭圆函数精细积分算法有更好的稳定性和精度. 本文还分析了目前常用到的各种工程软件对椭圆函数计算的适用范围, 从而说明了精细积分算法不仅高效稳定而且还拥有很好的适用范围, 在计算椭圆函数的已有算法中具有很大优势.

但还有部分后续工作需要继续深入:

1. 对已有软件的进一步分析和对算法的进一步改进;
2. 将精细积分法应用于其它特殊函数的求解.

参 考 文 献

[1] 高本庆. 椭圆函数及其应用 [M]. 北京: 国防工业出版社, 1991.

[2] Serge Lang. Elliptic Functions-Second Edition [M]. New York: Springer-Verlag, 2003.

[3] Keigo, Iizuka. Elements of photonics vol.II-for fiber and integrated optics [M]. New York: John Wiley and Sons Inc, 2002.

[4] 石顺祥, 陈国夫, 赵卫, 刘继芳. 非线性光学 [M]. 西安: 西安电子科技大学出版社, 2002.

[5] Wanxie Zhong. On precise integration method [J]. Journal of Computational and Applied Mathematics, 2004, 163(1): 59-78.

[6] 钟万勰. 结构动力方程的精细时程积分法 [J]. 大连理工大学学报, 1994, 43(2): 1865-1872.

[7] 钟万勰. 应用力学对偶体系 [M]. 北京: 科学出版社, 2002.

[8] 钟万勰, 姚征. 椭圆函数的精细积分算法: 应用力学进展 [C]. 北京: 科学出版社, 2004: 106-111.

[9] Wanxie Zhong, Zheng Yao. The precise integration method for Jacobi elliptic functions and application: Computational Method[C]. Springer, 2006.