

文章编号:1007-6735(2012)06-0517-03

# 混沌优化算法求解 Kepler 方程

杨玲玲, 马 良

(上海理工大学 管理学院, 上海 200093)

**摘要:**利用混沌运动的遍历性、随机性及规律性等特点,对 Kepler 方程设计了一种混沌优化算法,对其进行求解,并在计算机上予以实现.该算法易于实现且时间复杂度较小,在计算效率上有一定优势.经过一系列数值实验测试,验证了混沌优化算法对 Kepler 方程求解的可行性,并将计算结果与已有解法所得结果相比较,表明了该算法对此方程的有效性.

**关键词:**Kepler 方程;混沌;算法  
**中图分类号:** O 22      **文献标志码:** A

## Chaos Optimization Algorithm for Kepler's Equation

YANG Ling-ling, MA Liang

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:**Employing the property of ergodicity,randomness and regularity of chaotic movement,a chaos optimization algorithm for the Kepler's equation was presented and implemented. The time complexity of the method is relatively small and the algorithm is not difficult to realize,which gives advantage in computing efficiency. Comparing with the results of a series of tests and experiments, the numerical results verify the effectiveness of the method for solving the Kepler's equation.

**Key words:**Kepler's equation; chaos; algorithm

### 1 问题的提出

1619 年 Kepler 首次推导出了 Kepler 方程,该方程在物理、数学领域,特别是在经典天体力学研究中都起着重要作用.

在天体力学和轨道计算中,求解 Kepler 方程

$$f(x) = x - e \sin x - M = 0 \tag{1}$$

是经常遇到的问题,即在给定平近点角  $M(0 \leq M \leq \pi)$  和偏心率  $e$  后去求偏近点角  $x^{[1]}$ . 在椭圆轨道情

况下,  $0 \leq e \leq 1$ . 由于方程中含有超越函数  $\sin x$ ,因此,Kepler 方程是个超越方程,这意味着  $x$  无法用代数方法求出,只能用数值分析或级数求解,这就要求 Kepler 方程在实际求解中应尽可能地接近真值.

超越方程常用迭代方法求解,目前应用较多的数值迭代方法是不动点迭代法或牛顿迭代法,但前者收敛速度慢,后者在选取初值上要求较高,而且每次都要计算导数值<sup>[2]</sup>.

在实际应用中,由于对上千万年的天体轨道作积分需要反复求解 Kepler 方程,计算量大且耗时,因此,

收稿日期: 2011-06-19  
基金项目: 国家自然科学基金资助项目(70871081);上海市重点学科建设资助项目(S30504)  
作者简介: 杨玲玲(1987-),女,硕士研究生.研究方向:智能优化. E-mail: weenieyang@gmail.com  
马 良(联系人),男,教授.研究方向:智能优化、系统工程. E-mail: maliang@usst.edu.cn

研究快速求解 Kepler 方程的方法具有重要的意义<sup>[3]</sup>. 本文试图使用一种智能优化算法——混沌优化算法求解 Kepler 方程,探索从新的角度求解此类问题.

## 2 现有解法

尽管 Kepler 方程的提出已近 4 个世纪,但因其重要性而一直是研究的重要课题. 每 10~20 年就会提出一个新的求解方法. Ng<sup>[4]</sup> 在 1979 年提出一立方收敛的方法.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \left[ 1 + \frac{f(x_n)f''(x_n)}{2f'(x_n)^2} \right] \quad (2)$$

1983 年 Danby 和 Burkardt 提出了具有立方收敛性的 Halley 方法.

$$x_{n+1} = x_n - f(x_n) / \left[ f'(x_n) - \frac{f(x_n)f''(x_n)}{2f'(x_n)^2} \right] \quad (3)$$

同时,他们基于该方法提出了四次方和五次方的收敛性方法<sup>[5]</sup>. Colwell, Battin 和 Vallado 分别提出的方法也是以迭代或级数展开为基础的经典方法<sup>[6]</sup>.

## 3 算法设计

混沌运动具有遍历性、随机性及规律性等特点,它能在一定范围内,按其自身规律不重复地遍历所有状态<sup>[7]</sup>. 李兵和蒋慰孙<sup>[8]</sup>选用 Logistic 映射做算法

$$a_{n+1} = \mu a_n (1 - a_n) \quad (4)$$

式中,  $\mu$  为控制参量,取  $\mu = 4$ .

设  $0 \leq a_0 \leq 1$ ,  $n = 0, 1, 2, \dots$ , 当  $\mu = 4$  时,系统完全处于混沌状态,运用载波的方式将混沌状态引入至优化变量中,同时将遍历范围放大到优化变量的取值范围,然后利用混沌变量进行搜索<sup>[8]</sup>. 文献<sup>[8]</sup>中介绍了算法的基本步骤.

针对 Kepler 方程设计下面的算法.

**Step 1** 等价转换式(1).

$$\begin{aligned} \min F(x) &= \min f^2(x) = \\ \min (x - e \sin x - M)^2 &\geq 0 \end{aligned} \quad (5)$$

将求解式(1)的问题转化为求  $x$ , 使得目标函数  $F(x) = (x - e \sin x - M)^2$  的值最小.

**Step 2** 确定  $x$  的取值范围.

已知  $0 \leq x \leq 2\pi$ , 进一步来说,在文献<sup>[9]</sup>中得出结论

$$M \leq x \leq M + e$$

因为  $f'(x) = 1 - e \cos x \geq 0$  恒成立,  $0 \leq e \leq 1$ , 函数  $f(x)$  单调递增,同时

$$f(M) = M - e \sin M - M = -e \sin M \leq 0$$

$$f(M + e) = (M + e) - e \sin(M + e)$$

$$- M = -e(1 - \sin(M + e)) \geq 0$$

故可得出  $M \leq x \leq M + e$ .  $f(x)$  的图像如图 1 所示.

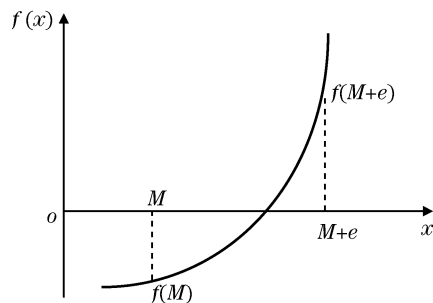


图 1  $f(x)$  的图形

Fig.1 Graph of  $f(x)$

**Step 3** 初始化.

设置  $k = 1$ ,  $a_0$ , 最大迭代次数  $n$ .

**Step 4** 根据式(4),产生混沌变量  $a_i$ . 通过式(6),利用载波方式将混沌变量放大到相应优化变量在 Step 2 中所得到的取值范围.

$$x_i = e a_i + M \quad (6)$$

**Step 5** 如果  $F(x_i) < F^*$ , 那么  $F^* = F(x_i)$ ,  $x^* = x_i$ ; 否则,放弃  $x_i$ .

$$i = i + 1$$

**Step 6** 如经若干次搜索后保持不变,则按式(7)进行第二次载波.

$$x(k) = x^* + \alpha a_k \quad (7)$$

式中,  $x^*$  为当前最优解;  $\alpha$  为较小的调节常数,本文中取小于 1 的常数.

**Step 7** 如果  $F[x(k)] < F^*$ , 那么  $F^* = F[x(k)]$ ,  $x^* = x(k)$ ; 否则,放弃  $x(k)$ .

$$k = k + 1$$

**Step 8** 当达到最大迭代次数时,终止搜索,输出最优解  $x^*$ ,  $F^*$ .

不难估算,该算法的时间复杂度为  $O(n)$ ,  $n$  为最大迭代次数.

## 4 数值测试

为检验求解效果,进行了一系列数值测试计算,并与已有方法进行比较. 实验所用的硬件为 Pentium(R)4 CPU 2.80 GHz, 256 MB 内存的微机,软件为 Windows XP 操作系统和 Matlab 7.0 运行环境.

在现有文献中,大多数计算实例是在  $M = 0$  和  $e = 1$  附近进行的,因为,在这一点有  $f' = f'' = 0$ ,在

运用传统方法时容易导致解法不收敛,本文首先求解了这类情况,现列出部分结果.表 1 和表 2 分别列出了迭代次数为 1 000 和 10 000 时,在临界情况(零平近点角和等于 1 的偏心率,即  $M = 0, e = 1$ )附近的测试结果,混沌初值  $\alpha_0 = 0.201\ 7$ .

表 1 最大迭代次数为 1 000 时的结果  
Tab.1 Results of  $n = 1\ 000$

| $e$      | $M = 0.001$          |                            |
|----------|----------------------|----------------------------|
|          | $x^*$                | $F^*$                      |
| 0.900 00 | 0.009 833 978 372 78 | 2.709 155 035 819 178e-010 |
| 0.990 00 | 0.088 765 515 958 12 | 9.075 884 848 936 042e-012 |
| 0.999 00 | 0.169 497 561 921 34 | 4.361 148 418 064 972e-010 |
| 0.999 90 | 0.180 543 362 250 29 | 7.906 393 831 236 803e-012 |
| 0.999 99 | 0.182 058 796 722 81 | 3.458 057 947 797 521e-011 |

从表 1 和表 2 中可以看出,迭代次数越大,求解精度越好.由于算法的时间复杂度较低,算法的运行时间基本可以忽略不计.

文献[10]中指出,用改进的牛顿法求解时可能

表 2 最大迭代次数为 10 000 时的结果  
Tab.2 Results of  $n = 10\ 000$

| $e$      | $M = 0.001$          |                            |
|----------|----------------------|----------------------------|
|          | $x^*$                | $F^*$                      |
| 0.900 00 | 0.009 998 074 922 30 | 1.814 320 220 821 478e-015 |
| 0.990 00 | 0.088 571 918 591 32 | 1.047 868 416 809 741e-013 |
| 0.999 00 | 0.170 913 310 600 05 | 9.401 719 068 406 652e-013 |
| 0.999 90 | 0.180 756 443 868 33 | 4.577 591 317 492 144e-013 |
| 0.999 99 | 0.181 778 738 526 03 | 1.578 420 258 579 867e-012 |

收敛至一个或多个错误的值,例如,当  $M = 1.347$ ,  $e = 0.66$ ,迭代序列在 0.448 902, 0.354 721, 0.469 946和 0.362 751 这 4 个值上循环,而此时的正确解是 1.958 11<sup>[10]</sup>.所以,本文也对该情况进行了求解,求解结果为

$$x^* = 1.958\ 119\ 411\ 280\ 24$$

$$F^* = 9.833\ 667\ 213\ 854\ 369\text{e} - 011$$

本文还测试了文献[3]和文献[10]中的一些算例,结果如表 3 所示.

表 3 部分算例测试结果  
Tab.3 Part of test results

| $M, e$                    | $x^*$                      | $F^*$                      |
|---------------------------|----------------------------|----------------------------|
| $M = 0, e = 0.1$          | 6.804 168 801 443 152e-009 | 3.750 033 759 361 120e-011 |
| $M = 0, e = 0.5$          | 3.402 084 400 721 576e-008 | 2.893 544 567 408 272e-016 |
| $M = 0, e = 0.95$         | 6.463 960 361 370 994e-008 | 1.044 569 588 834 414e-017 |
| $M = 0.799, e = 0.75$     | 1.548 819 052 075 65       | 2.919 886 554 764 162e-014 |
| $M = 2.577, e = 0.523\ 8$ | 2.768 124 480 647 76       | 3.156 692 685 024 609e-010 |
| $M = 0.45, e = 0.62$      | 0.956 726 957 148 30       | 2.767 126 519 921 110e-011 |

测试结果表明,混沌优化算法对于 Kepler 方程求解的有效性,该算法较低的时间复杂度提升了运算的效率,且免除了多次求导的复杂运算,避免了在临界状态容易不收敛的缺点.

5 结 论

常见的求解 Kepler 方程的方法是简单迭代法、牛顿法等经典方法,本文运用智能优化方法——混沌优化算法求解该问题.算法的时间复杂度较低,具有较高的运算效率,同时避免了传统方法中多次求导等复杂计算,且取得了满意的结果.对于 Kepler 方程,本文所给出的混沌优化算法是一种简单而又快速地获得有效解的途径.

参考文献:

[1] 周庆林,黄天衣. Kepler 方程的解法[J]. 天文学报, 1988,29(1):106-112.

[2] 束雄鹰,李红. Kepler 方程的一种迭代加速[J]. 航空计算技术,2005,35(1):41-44.

[3] 王玉诏,钟双英,孙威,等. Kepler 方程的六阶迭代解法[J]. 江西科学,2009,27(6):790-792.

[4] Colwell P. Solving Kepler's equation over three centuries[M]. Richmond: Willmann-Bell, 1993.

[5] Danby J M A, Burkardt T M. The solution of Kepler's equation[J]. Celestial Mechanics, 1983, 31 (2): 317-328.

[6] Davis J J. Sequential solution to Kepler' equation[J]. Celest Mech Dyn Astr,2010,108(7):59-72.

[7] 苗东升. 系统科学大学讲稿[M]. 北京: 中国人民大学出版社,2007.

[8] 李兵,蒋慰孙. 混沌优化方法及其应用[J]. 控制理论与应用,1997,14(4):613-615.

[9] Smith G R. A simple, efficient starting value for the iterative solution of Kepler's equation[J]. Celestial Mechanics,1979,19(2):163-166.

[10] 岳锦海,黄天衣. 椭圆 Kepler 方程求解中的非线性现象[J]. 南京大学学报(自然科学版),1998,34(1):21-28.