

文章编号: 1004-5929(2008)02-0102-09

球形粒子 Mie 散射参量的 Matlab 改进算法

张合勇, 赵卫疆, 任德明, 曲彦臣, 宋宝安

(哈尔滨工业大学光电子信息科学与技术系, 哈尔滨 150080)

摘 要: 本文主要讲述了 Mie 散射物理参量的一种改进数值算法, 在抛弃了 Mie 散射物理参量的经典算法——连分式算法和后向递推算法之后, 在现有利用 Matlab 计算散射参量的基础上, 充分利用了 Matlab 内置命令集和函数集, 得出了任意折射率且尺度参数在 $10^{-4} \sim 10^4$ 的球形粒子散射参量的准确计算结果。收敛速度比改进后向递推快, 相应的程序简单易读且执行时间大为缩短, 比现有经典算法所用的递推关系较少, 因此在很大程度上避免了递推过程中误差的积累, 保证了计算结果的可靠性。

关键词: Matlab; Mie 理论; 散射参量; 尺度参数; 改进后向递推

中图法分类号: O432

文献标识码: A

Improved Algorithm of Mie Scattering Parameter Based on Matlab

ZHANG He-yong, ZHAO Wei-jiang, REN De-ming, QU Yan-chen, SONG Bao-an

*(Department of Optic-electronics Information Science and Technology,**Harbin Institute of Technology, Harbin 150080, China)*

Abstract: An improved numerical algorithm of Mie scattering parameter is presented in this paper. After giving up the classical algorithms—Continued Fraction Method and Modified Downward Recurrence Method, we have shown in this paper how to calculate scattering parameters using the functions and commands of Matlab. We obtained the correction results of spherical particles with size parameters from 10^{-4} to 10^4 and without any constrain on the refractive index. The speed of convergence is faster than improved Downward Recurrence Method. The program is easy to read and needs less time to program. This improved numerical algorithm has less recurrence than the classical algorithm, so we can avoid the errors that accumulated in the recurrence computation, and ensure the reliability of the results.

Key words: Matlab; Mie theory; scattering parameter; size parameter; Modified Downward Recurrence Method

1 引言

最早得出球体散射严格解的是 Mie^[1] 和 Debye, 但是当时并没有实现关于球体散射的数值计算, 因为 Mie 散射计算非常复杂, 另一方面是对 Mie 理论直接编程将会导致严重的计算错

误, 这尤其表现在其中复变量 Bessel 函数的计算上^[2]。到 1968 年 Dave 最先发表了完整的算法^[3], 以后 Lentz^[4] 和 Wiscombe^[5] 又针对部分计算提出了改进算法, 但 Dave 的算法不能应用到尺度参数的情况下, Wiscombe 的方案详细讨论了计算时间, 循环次数, 得到了准确的结果, 但

收稿日期: 2007-12-10; 收修改稿日期: 2008-01-08

基金项目: 哈尔滨工业大学优秀青年教师培养计划资助(编号 HITQNJ.S. 2006. 025)

作者简介: 张合勇(1982—), 男, 黑龙江省哈尔滨市人, 硕士生, 主要研究工作是激光多普勒雷达。E-mail: yonghezhang1116@

126.com

©1994-2019 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

是他没有考虑复折射率较大的粒子, 安庆师范学院的张杰等人对大气中具有复折射率气溶胶粒子的散射光学特性^[6], 余其铮等人利用连分式进行计算, 通过对计算次数进行拟合, 提高了计算的速度, 但是对于尺度参数在 100 以外的拟合公式没有讨论。王式民等利用后向递推办法大大提高了计算速度, 但是文中提到的递推表达式是和粒子的复折射率相关的表达式^[7]。朱震等人在以往 Mie 散射计算的基础上提出一种新的算法^[8], 此算虽然区别于传统的递推算法, 精度高, 但仍然使用了递推算法, 不可避免的会出现积累误差。前面所说的改进都是在同一种算法上改进, 而姬丰等人利用 Matlab 对散射参量进行数值模拟^[9], 得到了正确的结果, 但是它没有考虑折射率虚部较大和粒子尺度参数较大对计算结果的影响。而我们的数值计算完全不依赖连分式或者后向递推算法, 直接利用 Matlab 中内置的半整数阶 Besselj 函数, 半整数阶 Bessely 函数(由半整数阶 Besselj 和 Bessely 函数组合得到)函数, 连带 Legendre 函数通过简单有效的编程得到了准确的结果, 由于 Matlab 中基于矩阵的运算方式, 比起用后向递推或者改进后向递推的编程方式有很大的优势。

2 散射理论基础与 Matlab 实现

若散射体为均匀球体^[10], 如图 1 所示, 照射光为线偏振平面波, 振幅为 E , 光强为 I_0 , 沿 z 轴传播, 其电场矢量沿 x 轴振动。散射体位于坐标原点 O , P 为观测点。散射光方向(OP 方向)与照射光方向(z 轴)所组成的平面称为散射面, 照射光方向至散射光方向之间的夹角 θ 称为散射角, 而 x 轴至 OP 在 xy 平面上投影线(OP')之间的夹角 φ 称为极化角。观测点与散射体相距 r 。

根据经典的 Mie 散射理论, 散射粒子的尺度参数为 $\alpha = 2\pi a / \lambda$, 其中 a 为球形粒子的半径, 散射粒子相对周围介质的折射率为 $m = m_1 + i * m_2$, 则散射光垂直于散射面和平行于散射面的两个分量的振幅函数为^[11]:

$$S_1 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \pi_n + b_n \tau_n) \tag{1}$$

$$S_2 = \sum_{n=1}^{\infty} \frac{2n+1}{n(n+1)} (a_n \tau_n + b_n \pi_n) \tag{2}$$

以上两式中:

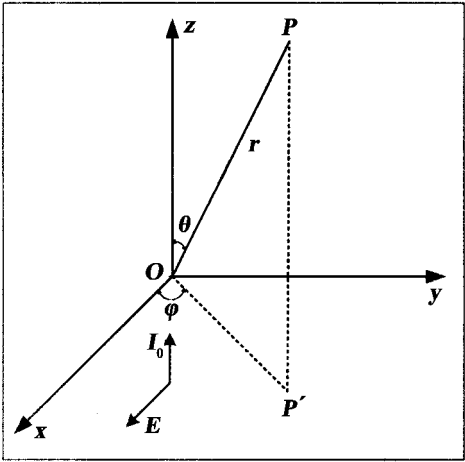


Fig. 1 Light scattering orthogonal reference frame

$$a_n = \frac{\Psi_n(\alpha) \Psi'_n(m\alpha) - m \Psi'_n(\alpha) \Psi_n(m\alpha)}{\xi_n(\alpha) \Psi'_n(m\alpha) - m \xi'_n(\alpha) \Psi_n(m\alpha)} \tag{3}$$

$$b_n = \frac{\Psi_n(\alpha) \Psi'_n(m\alpha) - \Psi'_n(\alpha) \Psi_n(m\alpha)}{m \xi_n(\alpha) \Psi'_n(m\alpha) - \xi'_n(\alpha) \Psi_n(m\alpha)} \tag{4}$$

$$\pi_n = \frac{P_n^{(1)}(\cos\theta)}{\sin\theta} \tag{5}$$

$$\tau_n = \frac{d}{d\theta} P_n^{(1)}(\cos\theta) \tag{6}$$

以上式中:

$$\psi_n(z) = \left\{ \frac{z\pi}{2} \right\}^{1/2} J_{n+1/2}(z) \tag{7}$$

$$\xi_n(z) = \left\{ \frac{z\pi}{2} \right\}^{1/2} [J_{n+1/2}(z) - i * Y_{n+1/2}(z)] \tag{8}$$

$$\psi'_n(z) = \psi_{n-1}(z) - \frac{n}{z} \psi_n(z) \tag{9}$$

$$\xi'_n(z) = \xi_{n-1}(z) - \frac{n}{z} \xi_n(z) \tag{10}$$

$J_{n+1/2}(z)$ 和 $Y_{n+1/2}(z)$ 分别为半整数阶的第一类, 第二类贝塞尔函数。 $P_n^{(1)}(\cos\theta)$ 为一阶 n 次第一类缔合勒让德函数; $P_n(\cos\theta)$ 为第一类勒让德函数。在数值模拟过程中选取初始条件如下:

$$\begin{cases} \xi_1(z) = \frac{1}{z}(\sin z + i \cos z) - (\cos - i \sin z) \\ \psi_1(z) = \frac{1}{z} \sin z - \cos z \\ \tau_1 = \cos \theta \\ \pi_1 = 1 \end{cases} \quad (11)$$

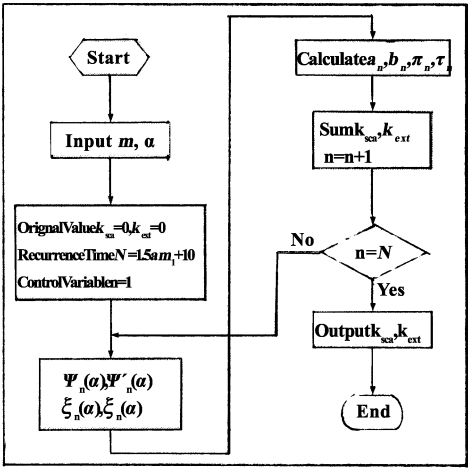


Fig. 2 Flow chart of the algorithm in this paper

通过设定符号变量 x 和两个符号函数 $\psi_n(x)$ 和 $\xi_n(x)$, 对于 $\psi_n(x)$ 的计算直接根据式(7)运用 Matlab 自带函数 `Besselj` 来实现; 对于 $\xi_n(x)$ 的计算直接根据式(8)运用 Matlab 自带函数 `Besselj` 和 `Bessely` 的组合来实现; 由于初值已给出, 所以两个导数项 $\psi'_n(x)$ 和 $\xi'_n(x)$ 分别用前面已经求出的 $\psi_n(x)$ 和 $\xi_n(x)$ 根据递推公式(9)和(10)

得出; 现在只需要运用 Matlab 自带函数 `Inline` 和 `Vectorize` 将符号变量和符号函数转化为数值变量 α 和 $m\alpha$ 数值函数。代入数值变量和就可以分别计算出 $\psi_n(\alpha)$, $\psi_n(m\alpha)$, $\psi'_n(\alpha)$, $\psi'_n(m\alpha)$, $\xi_n(\alpha)$, $\xi_n(m\alpha)$, $\xi'_n(\alpha)$, $\xi'_n(m\alpha)$, 代入(3)式和(4)式就分别可以求出 a_n 和 b_n , 系数 π_n 和 τ_n 可以由 Matlab 自带 `Legendre` 函数得出, 本文算法的具体程序流程如图 2 所示:

在这 4 个系数确定的情况下就可以计算粒子的散射系数和消光系数以及散射相位函数等物理参量。值得注意的是: Matlab 自带 `Legendre` 函数偶数行和实际值互为相反数, 这一点在本文作者编程的过程中深入分析才得以发现。

3 系数 a_n 和 b_n 的验证

在第 1 节中已经将计算 Mie 散射参量所需的四个最基本的系数通过编程计算得到, 还需对其准确性进行检验, 根据散射系数, 消光系数计算公式:

$$k_{sca} = \frac{2}{\alpha^2} \sum_{n=1}^{\infty} (2n+1) (|a_n|^2 + |b_n|^2) \quad (12)$$

$$k_{ext} = \frac{2}{\alpha^2} \sum_{n=1}^{\infty} (2n+1) \text{Re}(a_n + b_n) \quad (13)$$

我们计算了部分粒子的散射和消光系数与 Dave 和 Wiscombe 的结果作为对比如表 1(程序 Appendix 1.), 其中例 1~8 是 Dave(D)的计算结果, 例 9~18 是 Wiscombe(W)的计算结果。(表中画横线的表示没有提供相应数据)。

Table 1 The results of scattering coefficient and extinction coefficient

No	Refractive index	Size paramete	Rextinction coefficient (D or W)	Extinction coefficient (in this paper)	Scattering coefficient (D or W)	Scattering coefficient (in this paper)
1	1.50	0.0001	2.3068e-17	2.30680469545721e-017	2.3068e-17	2.30680469545721e-017
2	1.55	5.2182	3.10542	3.09263138572478	3.10542	3.09263138572478
3	1.50	100	2.094	42.09438781467652	2.0944	2.09438781467652
4	1.342	1570.7963	2.01294	2.01294397053586	2.01294	2.01294397053586
5	1.50-0.1i	0.0001	1.9925E-5	1.99251727053359e-005		2.40223709353470e-017
6	1.55-0.1i	5.2182	2.86165	2.85782076507058		1.66046624070259
7	1.50-0.1i	100	2.0898	2.08982183741646		1.13213397112473
8	1.50-0.1i	1570.7963	2.01445	2.01460913079986		1.10411063434223
9	0.75	0.099	7.41786e-6	7.41785911491118e-6	7.41786e-5	7.41785911491118e-6

No	Refractive index	Size paramete	Rextinction coefficient (D or W)	Extinction coefficient (in this paper)	Scattering coefficient (D or W)	Scattering coefficient (in this paper)
10	0.75	0.101	8.03354e-6	8.03353814855567e-6	8.03354e-6	8.03353814855567e-6
11	0.75	10	2.23226	2.23226484250203	2.23226	2.23226484250203
12	0.75	1000	1.99791	1.99790818424532	1.99791	1.99790818424532
13	1.33-i1e-5	100	2.10132	2.10132070588026	2.09659	2.09659350639351
14	1.33-i1e-5	10000	2.00409	2.00408893422900	1.72386	1.72385721774366
15	1.5-i	0.055	0.101491	0.10149104170528	1.13169e-5	1.13168723234955e-5
16	1.5-i	0.056	0.103347	0.10334669464987	1.21631e-5	1.21631094226684e-5
17	1.5-i	100	2.09750	2.09750175560618	1.28370	1.28369704937333
18	10-i10	1	2.53299	2.53299307789637	2.04941	2.04940500692548

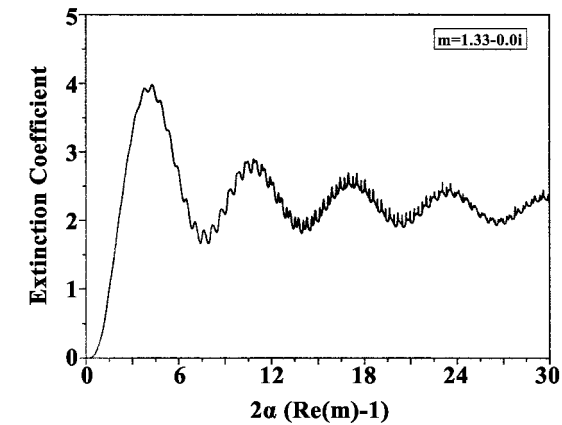


Fig. 3 Extinction coefficient with $m=1.33-0.0i$

由表可以看出, 我们所采用的简单的 Matlab 程序得出了任意折射率且尺度参数在 10^{-4} — 10^4 的球形粒子散射参量的准确计算结果, 与传统的计算结果吻合得很好。充分证明了系数 a_n 和 b_n 算法的正确性。而且我们的计算除了 (9), (10) 两式用到了简单递推以外, 其余根本不出现递推, 而传统的后向递推在处理粒子尺寸较大或者折射率虚部较大时, 往往出现计算速度慢或产生溢出和不收敛的现象, 本文的算法在根本上避免了以上问题。为了更充分的论证本算法的正确性和适用性, 本文又从尺度参数连续改变的角度画出了消光系数与尺度参数及折射率实部乘积的关系如图 3、4 所示 (程序 Appendix 2.):

以上两图与参考文献[12] 完全吻合, 而上述两图中靠近坐标原点所对应的粒子尺度参数趋于无穷小, 即便在此情况下, 本文的算法得到的

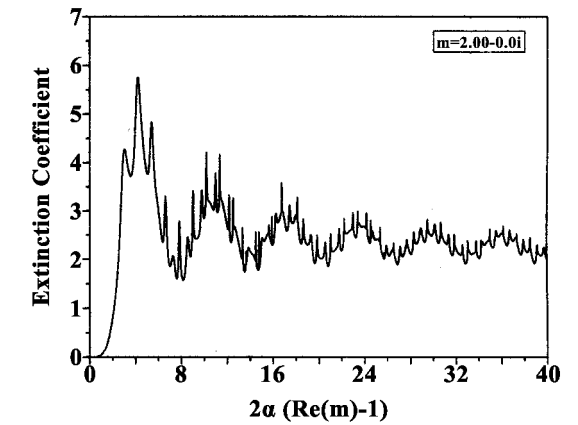


Fig. 4 Extinction coefficient with $m=2.00-0.0i$

结果仍然与前人算法吻合, 说明了本算法在粒子尺度参数很小的情况下仍然适用, 对于尺度参数较大的情况, 没有可对照的参考文献, 但是从表 1 中可以看到在尺度参数为 10000 时仍然给出了准确的计算结果, 至少说明在这个较大尺度范围内是准确的。

4 系数 a_n 和 b_n 的验证

散射相位函数是描述光经过粒子散射以后能量空间分布的物理量, 对于微粒散射特性的研究具有重要意义, 由参考文献[13] 中关于非偏振状态下散射相位函数的表达式:

$$P(\theta) = \frac{|S_1(\theta)|^2 + |S_2(\theta)|^2}{\sum_{n=1}^{\infty} (2n+1)(|a_n|^2 + |b_n|^2)}$$

(14)

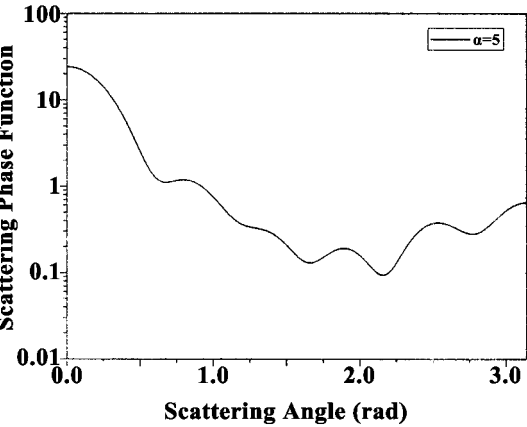


Fig. 5 Scattering phase function with $\alpha=5$

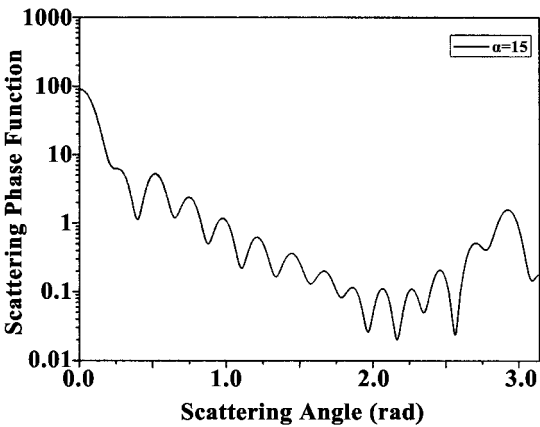


Fig. 6 Scattering phase function with $\alpha=15$

其中各个物理量的含义及其表达式在上文中已经提到, 由于粒子的散射是粒子本身的折射率以及尺度参数决定的, 与其它因素无关, 只须给出这两个物理量的具体值即可。本文选取折射率为 $m=1.5+0.005\times i$, 尺度参数分别为 5、15 如图 5、6 所示(程序 Appendix 3.):

以上两图与参考文献[11] 符合得很好, 由公式(1), (2)可以看出振幅函数种含有系数 a_n 、 b_n 、 π_n 、 τ_n , 而系数 a_n 和 b_n 的准确性已经通过散射系数和消光系数得以证明, 由散射相位函数计算的准确性可知系数 π_n 和 τ_n 计算的准确性。

5 收敛速度比较

在讨论收敛速时, 原则上讲应该是与前人算法在相同状态下进行比较, 但由于前人算法比较

繁琐, 况且程序执行效率低, 因此本文阐述的收敛速度是从收敛次数来考虑, 因为在传统的后向递推和改进后向递推中都用到递推次数的经验公式, 各个文献中所给的表达式不尽相同, 但是得出的收敛次数相差不大, 本文采用的后向递推的上限次数^[4]:

$$N=1.5\,m_1\alpha+10$$

(15)

其中 m_1 为粒子折射率的实部, a 为粒子的尺度参数。由此可以大致确定程序的循环次数, 然后在此基础上减少求和次数, 直到小数点后 14 位为稳定输出为止, 以此作为收敛的标准, 而 Dave 或者 Wiscombe 等人是将小数点后面 4 位或 5 位稳定输出作为最终计算结果。现将本文计算的级数求和次数 N_1 与后向递推的上限次数 N 进行对比如表 2。

Table 2 Comparison of the speed of convergence between Wiscombe algorithm and the improved algorithm in the paper

Size parameter	Refractive index	Wiscombe (N)	Im proved (N ₁)	Size parameter	Refractive index	Wiscombe (N)	Improved (N ₁)
0.0001	1.50	10	3	0.101	0.75	12	3
5.2182	1.55	22	11	10	0.75	22	18
100	1.50	235	118	1000	0.75	1135	1050
1570.7963	1.342	3173	1800	100	1.33-i1e-5	210	210
0.0001	1.50-0.1i	10	3	10000	1.33-i1e-5	19960	10050
5.2182	1.55-0.1i	22	11	0.055	1.5-i	11	9
100	1.50-0.1i	235	118	0.056	1.5-i	11	7
1570.7963	1.50-0.1i	3545	1820	100	1.5-i	235	129
0.099	0.75	11	3	1	10-i10	25	8

由表 2 可知: 本文的算法即使是在保证小数点后面 14 位输出稳定作为循环终止的前提下, 循环次数也比后向递推所用的上限次数少, 假如是在同样 5 位输出稳定的情况下, 我们的算法在求和次数上将会更少, 计算速度将会更快, 本算法在计算粒子尺度参数或者折射率实部较大的粒子方面更具有优势。

6 结论

本文抛弃了传统的关于经典 Mie 散射参量的后向递推和连分式法, 充分利用 Matlab 内部集成的函数模块, 对 Mie 散射消光系数, 散射系数进行了准确的计算, 计算结果和 Dave 或者 Wiscombe 的结果吻合得很好, 本算法的一大优点对于尺度参数在 $10^{-4} \sim 10^4$ 之间且折射率不限的粒子散射参量的收敛速度比传统后向递推或者改进后向递推算算法快, 本文在以小数点后 14 位稳定输出作为循环次数上限, 而 Dave 或者 Wiscombe 的算法都只是将小数点后 4 位或 5 位稳定输出时的递推次数作为上限。即便是在这样的条件下, 本文的算法在收敛次数上仍然低于改进的后向递推法, 特别是在尺度参数和折射率实部较大的情况下, 这样的优势就显得更明显。

参考文献:

[1] 吴健, 杨春平, 刘建斌, 等. 大气中的光传输理论 [M]. 北京: 北京邮电大学出版社, 2006. 11—12.

[2] 余其铮, 马国强, 刘晓彦. Mie 散射算法的改进 [J]. 哈尔滨工业大学学报, 1987, (3): 20—26.

[3] Dave J V. Scattering of visible light by large water spheres[J]. Appl. Opt, 1969, 8: 155—161.

[4] Lentz W J. Generating bessel functions in Mie scattering calculation using continued fractions [J]. Appl. Opt, 1976, 15: 668—673.

[5] Wicombe W J. Improved Mie scattering algorithms [J]. Appl. Opt, 1980, 19: 1505—1509.

[6] 张杰. 具有复折射率微粒的 Mie 散射光学特性研究 [J]. 光散射学报, 2006, 17(4): 359—365.

[7] 王小东, 吴健, 邱荣, 等. MIE 散射系数的改进算法[J]. 光电工程, 2006, 33(3): 24—27.

[8] 朱震, 叶茂, 陆勇, 等. 光散射粒度测量中 Mie 理论的高精度算法[J]. 光电子·激光, 1999, 10(2): 135—138.

[9] 姬丰, 郑刚, 屠其军, 等. 基于 MATLAB 的光散射物理量的数值计算方法[J]. 光学仪器, 2003, 25(1): 29—32.

[10] 孙景群. 激光大气探测[M]. 北京: 科学出版社, 1986. 31—32.

[11] 袁易君, 任德明, 胡孝勇. Mie 理论递推计算散射相位函数[J]. 光散射学报, 2006, 17(4): 366—371.

[12] Jian—Qi Zhao, Yin—Qiao Hu. Bridging techniaue for caculating the extinction efficiency of arbitrary shaped particles [J]. Appl. Opt. 20 August, 2003, 42(24): 4937—4944.

[13] Mishchenko Michael I, Mackowski Daniel W, Travis Larry D. Scattering of light by bispheres withtouching and separated components[J]. Appl. Opt., 1995, 34(21): 4589—4598.

[14] 王式民, 朱震, 叶茂, 等. 光散射粒度测量中 Mie 理论两种改进的数值计算方法[J]. 计量学报, 1999, 20(4): 279—285.

Appendix 1.

```
function scafactor
clc
alpha=5. 2182;          %散射微粒尺度参数
n=1. 55;                 %折射率
e=0;                    %散射截面递推公式求和项
f=0;                    %消光截面递推公式求和项
for l=1:50               %级数求和次数
    syms x f1 f2
    if l== 1
        f1=sin(x). / x—cos(x);
        f1dot=diff(f1,x,1);
```

```
f2=(sin(x)+i*cos(x))./x-cos(x)+i*sin(x);
```

```
f2dot=diff(f2,x,1);
```

```
else
```

```
f1=(pi*x/2).*besselj(l+0.5,x);
```

```
f1dot=(pi*x/2).*besselj(l-0.5,x)-l*(pi*x/2).*besselj(l+0.5,x)./x;
```

```
f2=(pi*x/2).*(besselj(l+0.5,x)-i*bessely(l+0.5,x));
```

```
f2dot=(pi*x/2).*(besselj(l-0.5,x)-i*bessely(l-0.5,x))-l*(pi*x/2).*(besselj(l+0.5,x)-i*bessely(l+0.5,x))./x;
```

```
end
```

```
g1=inline(vectorize(f1),'x'); %将符号变量转换为函数
```

```
g1dot=inline(vectorize(f1dot),'x');
```

```
g2=inline(vectorize(f2),'x');
```

```
g2dot=inline(vectorize(f2dot),'x');
```

```
al=(g1dot(n*alpha).*g1(alpha)-n*g1(n*alpha).*g1dot(alpha))./(g1dot(n*alpha).*g2(alpha)-n*g1(n*alpha).*g2dot(alpha));
```

```
bl=(n*g1dot(n*alpha).*g1(alpha)-g1(n*alpha).*g1dot(alpha))./(n*g1dot(n*alpha).*g2(alpha)-g1(n*alpha).*g2dot(alpha));
```

```
e=e+(2*l+1)*(abs(al).^2+abs(bl).^2); % (求散射系数)
```

```
f=f+(2*l+1)*real(al+bl); % (求消光系数)
```

```
end
```

```
kext=2/alpha.^2.*f
```

```
ksca=2/alpha.^2.*e
```

Appendix 2.

```
function kext
```

```
clc
```

```
alpha=1:0.1:50;
```

```
n=1.33+0.0i;
```

```
f=0; %消光截面递推公式求和项
```

```
for l=1:50 %级数求和次数
```

```
syms x f1 f2
```

```
if l==1
```

```
f1=sin(x)./x-cos(x);
```

```
f1dot=diff(f1,x,1);
```

```
f2=(sin(x)+i*cos(x))./x-cos(x)+i*sin(x);
```

```
f2dot=diff(f2,x,1);
```

```
else
```

```
f1=(pi*x/2).*besselj(l+0.5,x);
```

```
f1dot=(pi*x/2).*besselj(l-0.5,x)-l*(pi*x/2).*besselj(l+0.5,x)./x;
```

```
f2=(pi*x/2).*(besselj(l+0.5,x)-i*bessely(l+0.5,x));
```

```
f2dot=(pi*x/2).*(besselj(l-0.5,x)-i*bessely(l-0.5,x))-l*(pi*x/2).*(besselj(l+0.5,x)-i*bessely(l+0.5,x))./x;
```

```
end
```

```

g1=inline(vectorize(f1), 'x');      %将符号变量转换为函数
g1dot=inline(vectorize(f1dot), 'x');
g2=inline(vectorize(f2), 'x');
g2dot=inline(vectorize(f2dot), 'x');
al=(g1dot(n * alpha) * g1(alpha) - n * g1(n * alpha) * g1dot(alpha)) / (g1dot(n * alpha) * g2(alpha) - n * g1(n * alpha) * g2dot(alpha));
bl=(n * g1dot(n * alpha) * g1(alpha) - g1(n * alpha) * g1dot(alpha)) / (n * g1dot(n * alpha) * g2(alpha) - g1(n * alpha) * g2dot(alpha));
f=f+(2 * l+1) * real(al+bl);          % (求消光系数)
end
kext=2. / alpha. 2. * f
plot(2 * alpha * (real(n)-1), kext, 'r')
axis([0 30 0 5])
xlabel('2α(n-1)')
ylabel('消光系数')

```

Appendix 3.

```

function phasedangelizitheta2D
clc
lambda=1.06;
n=1.5+0.005 * i          %折射率
alpha=15                  %散射微粒尺度参数
theta=linspace(0.001, 0.999, 1000) * pi; %散射角 theta(弧度)
ph=pi/4;
s1=0;
s2=0;
e=0;
for l=1:100               %级数求和次数
    syms x f1 f2
    if l==1
        f1=sin(x) / x - cos(x);
        f1dot=diff(f1, x, 1);
        f2=(sin(x) + i * cos(x)) / x - cos(x) + i * sin(x);
        f2dot=diff(f2, x, 1);
        pil=1;
        taul=cos(theta);
    else
        f1=(pi * x/2) * besselj(l+0.5, x);
        f1dot=(pi * x/2) * besselj(l-0.5, x) - 1 * (pi * x/2) * besselj(l+0.5, x) / x;
        f2=(pi * x/2) * (besselj(l+0.5, x) - i * bessely(l+0.5, x));
        f2dot=(pi * x/2) * (besselj(l-0.5, x) - i * bessely(l-0.5, x)) - 1 * (pi * x/2) * (besselj(l+0.5, x) - i * bessely(l+0.5, x)) / x;
        h=legendre(l, cos(theta)); % 1 * length(theta)
    end
end

```



```
pl1=-h(2,:);
```

%legendre 函数的偶数行的值均是实际值的相反数。这

```
pl2=h(3,:);
```

```
taul=cot(theta). *pl1-pl2;
```

```
pil=pl1./sin(theta);
```

```
end
```

```
g1=inline(vectorize(f1),'x'); %将符号变量转换为函数
```

```
g1dot=inline(vectorize(f1dot),'x');
```

```
g2=inline(vectorize(f2),'x');
```

```
g2dot=inline(vectorize(f2dot),'x');
```

```
al=(g1dot(n *alpha). *g1(alpha)-n *g1(n *alpha). *g1dot(alpha)). /(g1dot(n *alpha). *g2(alpha)-n *g1(n *alpha). *g2dot(alpha));
```

```
bl=(n *g1dot(n *alpha). *g1(alpha)-g1(n *alpha). *g1dot(alpha)). /(n *g1dot(n *alpha). *g2(alpha)-g1(n *alpha). *g2dot(alpha));
```

```
s1=s1+(2 *l+1)/(1 *(l+1)) *(al' *pil+bl' *taul); %length(alpha) *length(theta)
```

```
s2=s2+(2 *l+1)/(1 *(l+1)) *(al' *taul+bl' *pil);
```

```
e=e+(2 *l+1) *(abs(al). ^2+abs(bl). ^2);
```

```
end
```

```
i1=abs(s1). ^2; %散射光强度函数(垂直), length(alpha) *length(theta)
```

```
i2=abs(s2). ^2; %散射光强度函数(平行)
```

```
sigmas0=(lambda. ^2/(2 *pi)) *e; %单个粒子散射截面
```

```
beta0=(lambda/(2 *pi)). ^2 *(i1 *sin(ph). ^2+i2 *cos(ph). ^2); %单个粒子微分散射
```

截面

```
phasefun=4 *pi *(beta0./sigmas0); %单个粒子的散射相位函数(与入射波长无关)
```

```
semilogy(theta, phasefun)
```

```
axis([0 pi 1e-2 1e3])
```

```
xlabel(' Scattering Angle(弧度)')
```

```
ylabel(' Phase Function')
```

```
title(' 单个粒子散射相位函数与 theta 关系')
```