

# 粒子散射计算-积分算法

## 一、计算流程及原理：

计算的公式如下：

$$\Theta = 2b \left[ \int_b^{r_{\max}} \frac{dr}{r^2} \left[ 1 - \frac{b^2}{r^2} \right]^{-1/2} - \int_{r_{\min}}^{r_{\max}} \frac{dr}{r^2} \left[ 1 - \frac{b^2}{r^2} - \frac{V}{E} \right]^{-1/2} \right]$$

其中， $r_{\max}$ 取3， $r_{\min}$ 通过搜索法计算如下积分式的根得到：

$$V(r) = 4V_0 \left[ \left( \frac{a}{r} \right)^{12} - \left( \frac{a}{r} \right)^6 \right]$$

而积分运算的过程采用抛物线积分公式：

$$\Delta S = \int_{x_{i-1}}^{x_{i+1}} f(x) dx \approx \int_{x_{i-1}}^{x_{i+1}} y(x) dx = \frac{1}{3} [y_{i-1} + 4y_i + y_{i+1}] \Delta x$$

计算核心过程为：以db为步长，从b\_min到b\_max，依次取不同的b值，计算Theta。每次计算Theta的时候先用findroot函数找到r\_min，再用抛物线法求积分。

## 二、程序源代码：

```
//物理学院 2012级 张楚珩 121120173
//源代码如下
//
// main.c
// Scattering
//
// Created by ZHANG CH on 14-4-2.
// Copyright (c) 2014年 NJU. All rights reserved.
//

#include <stdio.h>
#include <math.h>

#define NUM_OF_POINT (200)
#define R_MAX (4)
#define RAD_TO_DEGREE (57.2958)

double findRoot(double E, double b);
double V(double r);
double integral(double r_max, double r_min, double b, double E);
double f1(double r, double b);
double f2(double r, double E, double b);
void run(FILE * fp, double b_min, double b_max, double db, double E, double Theta[NUM_OF_POINT][NUM_OF_POINT]);
```

```
void init(double * b_min, double * b_max, double * db, double * E);

int main(int argc, const char * argv[])
{
    double b_min, b_max, E, db;
    double Theta[NUM_OF_POINT][NUM_OF_POINT];
    FILE * fp;
    fp = fopen("out.txt", "wb");
    init(&b_min, &b_max, &db, &E);
    run(fp, b_min, b_max, db, E, Theta);
    return 0;
}

void run(FILE * fp, double b_min, double b_max, double db, double E,
double Theta[NUM_OF_POINT][NUM_OF_POINT])
{
    int i, j;
    double b;
    double r_min, r_max;
    r_max = R_MAX;
    fprintf(fp, "[");
    for(b=b_min, j=0; b<=b_max; b+=db, j++)
    {
        if (j != 0) fprintf(fp, ",");
        r_min=findRoot(E, b);
        Theta[i][j]=integral(r_max, r_min, b, E);
        fprintf(fp, "%lf ", Theta[i][j]*RAD_TO_DEGREE);
    }
    fprintf(fp, "];");
}

void init(double * b_min, double * b_max, double * db, double * E)
{
    scanf("%lf%lf%lf%lf", b_min, b_max, db, E);
}

double findRoot(double E, double b)
{
    double r=4;
    double dr=0.1;
    while (dr>0.0001)
    {
        if (((1-((b*b)/(r*r)))-(V(r)/E)))<0)
        {
            r+=dr; dr/=2;
        }
        r-=dr;
    }
    return r;
}

double integral(double r_max, double r_min, double b, double E)
```

```
{
    double Int1=0;
    double Int2=0;
    double i;
    double di=2e-2;
    Int1=0;Int2=0;
    for(i=(b+2*di);i<r_max;i+=2*di)
    {
        Int1 += di*(f1(i-di,b)+4*f1(i,b)+f1(i+di,b))/3;
    }
    for(i=(r_min+2*di);i<r_max;i+=2*di)
    {
        Int2 += di*(f2(i-di,E,b)+4*f2(i,E,b)+f2(i+di,E,b))/3;
    }
    return 2*b*(Int1-Int2);
}

double f1(double r, double b)
{
    return 1/(r*r*sqrt(1-((b*b)/(r*r))));
}

double f2(double r, double E, double b)
{
    return 1/(r*r*sqrt(1-((b*b)/(r*r))-V(r)/E));
}

double V(double r)
{
    return 4*((1/pow(r,12))-((1/pow(r,6))));
}
```

---

### 三、运行结果及讨论:

分别运行输入参数

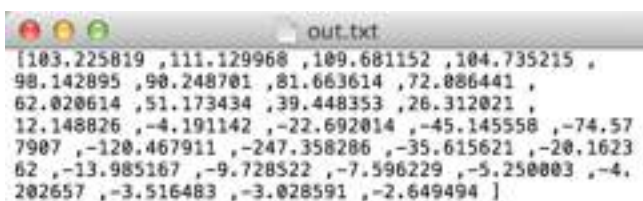
0.1 3.0 0.1 0.5

0.1 3.0 0.1 1.0

0.1 3.0 0.1 5.0

0.1 3.0 0.1 100

可以在可执行文件目录下找到



```
out.txt
[103.225819 ,111.129968 ,109.681152 ,104.735215 ,
98.142895 ,98.248701 ,81.663614 ,72.886441 ,
62.020614 ,51.173434 ,39.448353 ,26.312021 ,
12.148826 , -4.191142 , -22.692014 , -45.145558 , -74.57
7907 , -120.467911 , -247.358286 , -35.615621 , -20.1623
62 , -13.985167 , -9.728522 , -7.596229 , -5.250003 , -4.
202657 , -3.516483 , -3.028591 , -2.649494 ]
```

(仅标出一组数据, 其他相似)  
 这样的数据, 所得的原始数据如下:

```
[103.225819 ,111.129968 ,109.681152 ,104.735215 ,98.142895 ,90.248701 ,81.663614 ,
72.086441 ,62.020614 ,51.173434 ,39.448353 ,26.312021 ,
12.148826 , -4.191142 , -22.692014 , -45.145558 , -74.577907 , -120.467911 , -247.358286 , -35.6
15621 , -20.162362 , -13.985167 , -9.728522 , -7.596229 , -5.250003 , -4.202657 , -3.516483 , -3.
028591 , -2.649494 ]
[102.496474 ,109.656985 ,107.435352 ,101.690912 ,94.221059 ,85.384900 ,75.739967 ,
64.963773 ,53.457219 ,40.902857 ,26.999649 ,
11.040731 , -7.027604 , -28.989470 , -57.865926 , -103.550366 , -72.142107 , -28.648016 , -16.48
```

```
>> x = 0.1:0.1:2.9;
y1 = [111.129968 ,109.681152 ,104.735215 ,98.142895 ,90.248701 ,81.663614 ,72.086441
y2 = [102.496474 ,109.656985 ,107.435352 ,101.690912 ,94.221059 ,85.384900 ,75.739967
y3 = [100.883365 ,106.398089 ,102.485145 ,95.054867 ,85.693810 ,74.764382 ,62.771139
>> y4 = [97.712789 ,100.001086 ,92.817356 ,81.752047 ,68.639796 ,53.734504 ,38.038329
```

```
2206 , -11.272299 , -7.560707 , -6.386320 , -4.371284 , -3.528119 , -2.922703 , -2.515635 , -1.15
4858 , -2.042170 , -0.707048 ]
[100.883365 ,106.398089 ,102.485145 ,95.054867 ,85.693810 ,74.764382 ,62.771139 ,
49.320287 ,34.987843 ,18.975645 ,1.601011 , -14.320874 , -20.558375 , -15.837466 , -10.406904
, -7.070366 , -4.978664 , -3.650343 , -2.232651 , -2.231438 , -1.159018 , -1.659980 , -0.618999 ,
-1.335129 , -0.411684 , -1.289633 , -0.241095 , -1.276656 , -0.109855 ]
[97.712789 ,100.001086 ,92.817356 ,81.752047 ,68.639796 ,53.734504 ,38.038329 ,
22.089697 ,9.149166 ,
1.678314 , -0.671090 , -1.294814 , -0.812234 , -0.978179 , -0.409532 , -0.805072 , -0.163196 , -0
.738974 , -0.138599 , -0.764180 , -0.057943 , -0.763309 , -0.026390 , -0.899051 , -0.049129 , -0.
971538 , 0.002042 , -1.126417 , -0.007048 ]
```

将这些原始数据导入MATLAB中  
 再使用画图命令

```
plot(x,y1,x,y2,x,y3,x,y4)
```

即可得到最后的散射曲线图

散射

