# New Human Civilization Glittering Under Galaxy “Snowflakes”

Jin He
Department of Physics, Huazhong University of Science and Technology,
Wuhan, Hubei 430074, China
E-mail: mathnob@yahoo.com

**Abstract** What is human life? From the material sense, it is a material structure composed of oxygen, carbon and other atoms. From the biological point of view, it is a kind of advanced animal that understands the natural world, recognizes and creates products. In the last tens of thousands of years, human beings created languages and tools, and achieved a near-perfect understanding of the microscopic world of elementary particles. However, in the 21st century, mankind has experienced irreversible crises such as environmental pollution, erratic weather, food shortage, and population explosion. However, the crisis is also an opportunity. In the complexity of this world, a Chinese scientist opened a window for the understanding of human beings ourselves as well as the universe. The Earth is the direct environment for the survival of human life, but the root cause of human creation is Milky Way. Surprisingly enough, the life of galaxies is determined by a cubic algebraic equation. Therefore, the general public all have the potential to understand the lives of galaxies. Coincidentally, with human invention of computers, the general public has the potential to run the simple computer program (See Appendix of this paper) to generate and study galaxy snowflake chart (a simple graph expressing the internal structure of galaxies). Therefore, we see the hope of mankind: A harmonic general-public administrated society of new civilization rather than controlled by a few elites, will inevitably be born!

keywords: Life Origin, New Civilization, Galaxy Snowflake Chart, Cubic Equation, Galaxy Structure

PACS: 02.60.Lj, 98.52.Nr

What is the most precious thing in the world? Needless to say, it is life. What is the most precious life? It is certainly the human life. However, if human beings do not understand what life is, how do you expect people to cherish the life in its infancy?

## 1 What is the Most Precious Material Constituents in Life?

Life’s most precious material component is water, which is composed of two basic elements, hydrogen and oxygen, as shown in its molecular formula $H_2O$. Hydrogen is the lightest element in nature, with each hydrogen atom containing only one proton. However, each oxygen atom contains 8 protons and 8 neutrons. The human body is organic, as characterized by containing carbon atoms. Each carbon atom contains 6 protons and 6 neutrons. The human body mass contains 65% oxygen atoms, 18% carbon atoms, and only 10% hydrogen atoms.

Is life a rare being? It can be proved from the matter distribution in the Solar system that life is very rare. The Solar system contains the Sun, eight planets, asteroid belt, and other minor substances (such as comets, planet satellites). But the mass of the Sun itself is more than 98% of the total of the Solar System. What is the composition of the Sun? Analysis of the solar spectrum shows that more than 99% of the mass of the Sun are composed of two elements, hydrogen and helium. These two elements are the lightest ones in the natural world. In the universe, the most common material structures are similar to the Sun, i.e., stars. Therefore, only material structures other than stars in the universe can nurture life. In the universe, only galaxies are the structures greater than the solar system, while being relatively independent. The galaxy which hosts humans is called the Milky Way. There are only two types of galaxies (if we consider stable and weakly interacted entities only). One type has a three-dimensional elliptical shape; and the other has a flat spiral shape. Surprisingly, in elliptical galaxies, almost no substances other than stars can be observed. Therefore, the chance of life existing in an elliptical galaxy is almost zero. Human beings live in a flat spiral galaxy that is different from elliptical ones.

## 2 How Did Life Originate?

There is no satisfactory answer to this question. However, the nature does provide some inspirations. Can life exist without gravity? For example, in the International Space Station, astronauts are weightless around the Earth. There is no gravitational effect on both the Space Station and the people in it. In terms of single space travel, the only person who did the longest time is a Russian astronaut, who spent 437 days in space. For a person away from the Earth without gravity, within 2-3 days there is a 22% blood volume decrease or loss. Muscle atrophy occurs in space at 5% per week. Bone atrophy occurs in space at 1% per month [1]. It is obvious that it is hard for a human to survive in space longer than two years.

The human cannot survive in space, because without gravity, the body components are in a state of free motion. Without intercellular pressure, the cells in the human body are moving freely parallel to each other, and tending to be anti-structuring. For humans who live on the ground, intercellular pressure exists due to gravity. Moreover, the direction of this pressure is not simply towards the center of the Earth. Because of the Earth's self-rotation, the intercellular pressure has a horizontal component. This pressure not only induces the division and growth of human cells, but also affects the human body in blood circulation and helps improve the function and the development of organs; the pressure even helps the survival and growth of bacteria and viruses attached to human digestive tracts. Without the Earth's self-rotation and the gravity of Moon, life cannot exist. The human body is mainly composed of oxygen, carbon and other active atoms, which make the body soft and beautiful. In the outer shells of the metal atoms, e.g. copper and iron, there is only one free electron, which makes them not enough to form individual life beings. Rolling stones have no lasting vitality; only flexible car tires can help extend the car lifetime.

# 3 Galaxy "Snowflake" Chart

The universe is structured hierarchically. Humans live in a "microscopic" material world interacted by electromagnetic and nuclear forces. In this world, all the features of lives are functioned in terms of nuclear and electromagnetic forces. The effective range of nuclear interaction is very short. The sources of electromagnetic force, i.e. electric charges, can be positive or negative. In such a large-scale macroscopic world as the Solar system and galaxies, the total positive and negative charges are neutral. Therefore, electromagnetic and nuclear forces have no impact on the formation of large-scale structures and no impact on the formation of Solar system, not even the slightest one. The Solar system shows quantized structure but a galaxy at large-scale always shows a continuous shape.

Plants and animals present orderly material structures. Similarly, galaxy structures are also regular. To humans surprise, galaxy structures are much simpler and regular. Spiral galaxies are flat shaped. The main structure of spiral galaxies is an axially symmetric disk, with the stellar density decreasing exponentially along radial direction. It is the so called exponential disk. Different from elliptical galaxies, spiral galaxies are subject to wavelike perturbation. The perturbation brings about arm structure of spiral shape. This is why they are named the spirals. In fact, the exponential disk of spiral galaxies can coexist with bar-shaped structure. There are two types of spiral galaxies. A barred galaxy has additional bar structure. Its body is a combination of exponential disk and bar structure. A spiral galaxy without bar structure (i.e., its body is simply the exponential disk) is called a normal spiral galaxy.

There has been no fundamental explanation of the simple structure of spiral galaxies since their discovery. I am honored to have made a historic breakthrough in this regard, based on my systematic analysis of galaxy images since 2000.

I proposed the concept of rational structure. The density of material distribution on a plane is not arbitrary. For a special net of orthogonal curves in the plane, if the ratio of the material densities on both sides of any curve from the net is constant along the curve, the curve is called a proportion one. The net of curves is called an orthogonal net of proportion curves. Such a structure of material distribution is called a rational structure. Exponential disks of spiral galaxies are all rational structures. They have many nets of orthogonal proportion curves; the curves are golden curves, i.e., logarithmic curves (or equiangular curves). My research showed that the bar of a barred galaxy is composed of two or three dual-handle structures. Each dual-handle structure is also a rational structure. Its orthogonal proportion curves are all confocal ellipses and hyperbolas [2]. A spiral arm is the perturbation to the galaxy body structure. In order to minimize the disturbance, the perturbation waves propagate along the orthogonal proportion curves or the non-orthogonal proportion curves [3]. However, spiral arms are usually broken-shaped (not connecting end to end), composed of segments which follow orthogonal or non-orthogonal proportion curves [3].

There has not been a mathematical proof if an exponential disk combined with several dual-handle structures is a rational structure. However, I found the equation that the slopes of the tangential lines to the orthogonal proportion curves satisfy. Surprisingly, it is a cubic algebraic equation with one variable, called the instinct equation [3]. In other words, although I did not find a general mathematical solution to rational structure, I found a complete solution to the slopes of the tangential lines of orthogonal proportion curves. We can use this complete solution to analyze galaxy images.

The first step is to find an image of a barred galaxy. In the second step, the body structure of the galaxy is fitted with an exponential disk plus two or three dual-handle structures. Here

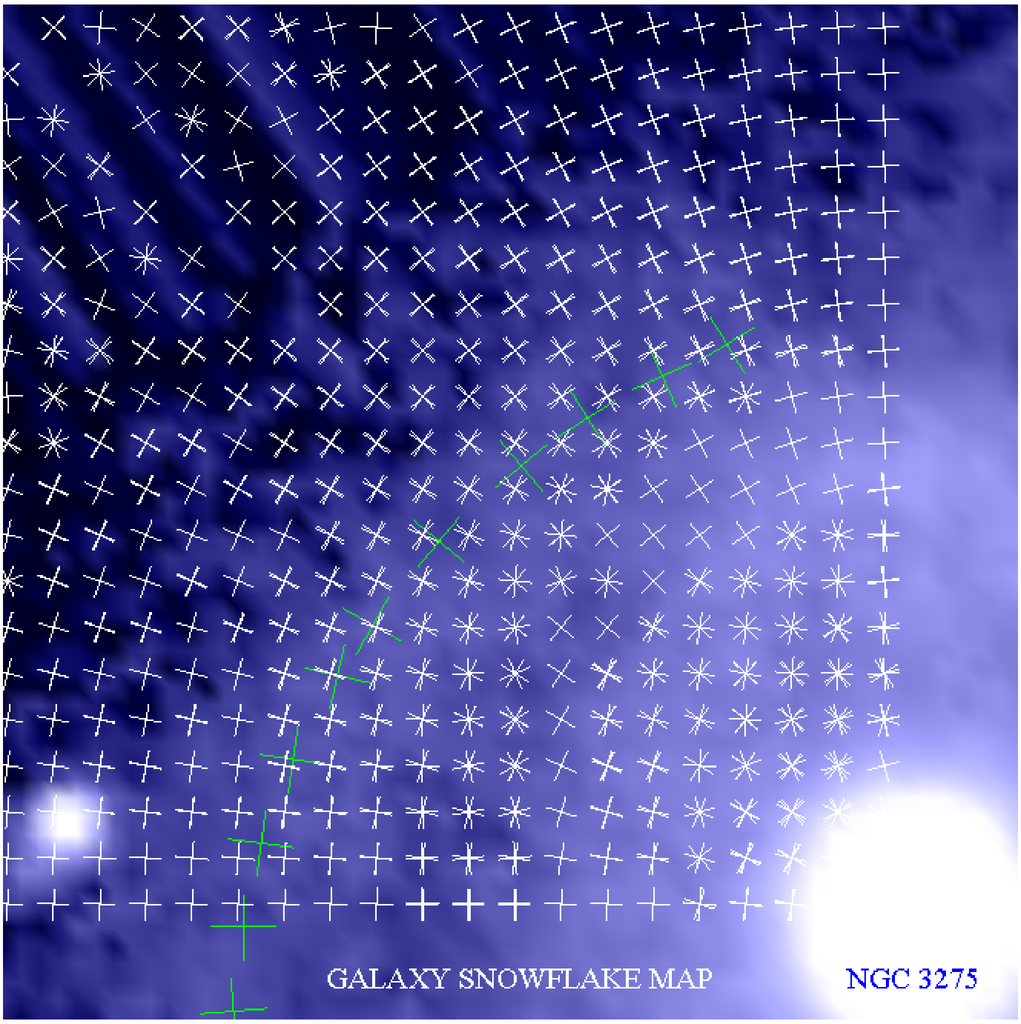

Figure 1: Galaxy snowflake chart for galaxy NGC 3275. The snowflakes far from the galaxy center are not real. They are the result of computing errors made by the computer. There is either a cluster of snowflakes near the galaxy center, or a queue of snowflakes at the periphery of galaxy bar. However, a unique net of orthogonal curves can be identified from the chart, and this should correspond to the global net of orthogonal proportion curves. The additional line of green cross to the chart demonstrates that the arm represented by the line follows a non-orthogonal proportion curve (see the upper panel of Figure 1 in [3]).

the body structure means the whole galaxy structure but ignoring the central bulge and the outer arms or rings. I found that the body structure can be well represented with a few parameters, which means that galaxies are rational. In the third step, with the simulated (analytical) body structure of the galaxy, we can apply calculus to the analytical expression of the structure. At any point in the galaxy, the coefficients of instinct equation can be calculated by taking differentiation to the expression. The roots of the instinctive equation are slopes of the tangential lines of the orthogonal proportion curves at that point. Each of the roots corresponds to a cross (i.e., the orthogonal tangential lines of the proportion curves at the point). If the equation has three roots, there are three crosses corresponding to the same point, forming a "snowflake" shape (see Fig.1). In a barred galaxy, if multiple points are selected, we get a "snowflake" chart for the galaxy. Note that, in the Figure, the snowflakes far from the galaxy center are not real. They are the result of computing errors made by the computer. It is better to calculate the coefficients of the instinct equation before calculating the "snowflake" chart. You will find that these coefficients taken far away from the galactic center, are almost zeros. The calculation accuracy of personal computers is usually sixteen digits after the decimal point. The instinct equation coefficients far away from the galaxy center, however, may be a decimal number with more than sixteen zeros after the decimal point! In Fig.1, there are several points where expected crosses are not drawn because the computer considered the corresponding coefficients "zero". The closer to the center of the galaxy, the greater the absolute value of all coefficients of the instinct equation. Near the galaxy center, the order of magnitude of the values is approximately 1.

Through investigating the images of different barred galaxies, we found that galaxy snowflake charts present regular patterns. This is an indirect proof that, for spiral galaxies, the combination of rational structures is still rational.

## 4 Life Characteristics in the Heaven

Curious readers may ask, since the exponential disk is a rational structure, can we calculate its snowflake chart? In fact, the coefficients of the instinct equation corresponding to the disk are all zero. The instinct equation has infinite solutions, hence no snowflake chart. The dual-handle structure is also a rational structure. The instinct equation corresponding to the double-handle structure has non-zero coefficients, and the equation usually has two or three roots. Therefore, the dual-handle structure has snowflake chart. And except for the places of the handle symmetry-axes, the dual-handle structure has snowflake solutions everywhere. It is difficult to identify the orthogonal net of confocal ellipses and hyperbolas in the snowflake chart.

Surprisingly (see Fig.1), the number of "snowflakes" is greatly reduced when the dual-handle structure is combined with the exponential disk. There is either a cluster of snowflakes near the galaxy center, or a queue of snowflakes at the periphery of galaxy bar. In other words, a global net of orthogonal curves can be identified from the galaxy snowflake chart, and this should correspond to the only net of orthogonal proportion curves. This indicates again (though indirectly) that galaxies are rational structure. In my paper [2], I listed twelve miracles of galaxy rational structure. In another paper of mine [3], I proved mathematically the simple equation a rational structure must satisfy, that is, the cubic algebraic equation (instinct equation). This can be regarded as the thirteenth miracle of rational galaxy structure. The galaxy snowflake chart shows that the cubic algebraic equation gives only one global net of orthogonal proportion curves. It was expected to have three global nets. This can be considered the fourteenth miracle.

The greatest miracle of galaxy rational structure, however, is its revelation of the life characteristics in the heaven. In the secular world on the Earth, there are miracles of life everywhere. However, humans do not even know what life is, and do not know how to cherish life. If life is considered rational, human beings even do not know the exact definition of rationality. The difficulties are caused by the fact that in the "microscopic world" where human beings survive, electromagnetic and nuclear forces dominate. Any independent many-particle system interacted by the electromagnetic and nuclear forces moves spontaneously towards a state of highest entropy, i.e., the highest degree of disorder in the microscopic state. However, no electromagnetic and nuclear forces are involved in the formation of galaxies. Galaxy structure is the only structure ever found whose rationality can be perceived, defined, and studied mathematically by humans.

Surprisingly, in the origin of life and growth, rationality is not the only cause. Another cause is the disturbance to the rational structure. Elliptical galaxies do not show much evidence of life, and the disturbance to their bodies is hard to be observed. For example, Elliptical galaxies present no significant arms. Images of spiral galaxies show that the greater the disturbance to the body structure, the more dusts and gases resulted, and the greater the events of star birth. Only dusts and gases contain significant amount of elements that are heavier than hydrogen and helium which the human body needs!

# 5 New Human Civilization

If computing results are trustworthy, then the galaxy snowflake chart is a universal truth. If the chart is the truth, human beings will inevitably advance to a new age of civilization based on this truth. Of course, humans must avoid being eliminating themselves before entering a new era. Therefore, I call on the world for righteous people to stand out, to pass the galactic snowflake chart to the hands of every adult, so that the truth of the universe spreads to every corner of the world!

The truth of life is eternal as the existence of galaxies. On the road of exploring the truth, scientists still have a long way to go. For example, they need to analyze images of more barred galaxies, and prove with each galaxy that galaxy rings follow orthogonal proportion curves while arms follow non-orthogonal proportion curves. They need to find the mathematical formula for the unique net of orthogonal proportion curves as shown on galaxy snowflake charts. They need to find the possible correlation between barred galaxy arms and the queues of snowflakes at the periphery of the bars. They should provide a strict mathematical proof that the summation of rational structures is still rational for spiral galaxies .

Heroes use the universal truth to resolve social conflicts in human community. Children are innocent, because they enjoy the life with the instinct and various function endorsed in the universe. However, many adults, for various reasons, do not see the truth of the universe, living a bitter life than children. They acquire resources by using more powerful hands than children; obtain wealth with insidious means. Heroes use the truth of universe to illuminate every dark corner.

Heroes use truth of the universe to deal with things in the world. When meeting a government official, they will ask if he or she has seen the galaxy snowflake chart; will ask if his (her) political speeches follow the truth of the universe. When meeting a businessman, the heroes will ask him (her) if he (she) has seen the galaxy snowflake chart; will ask him (her) if his (her) products provide services to the life. When meeting a researcher, the heroes will ask if he (she) has seen the galaxy snowflake chart; will ask him (her) if the study reveals the truth related to life.

# 6 Acknowledgement

The author would like to thank to Gary G Li for his major English translation.

# References

[1] He J. Yahoo answers: http://answers.yahoo.com/question/index?qid=1006051621689

[2] He J. (2010) Electr. Journ. Theo. Phys. **24**, 361-372.

[3] He J. (2010) viXra:1011.0057, http://vixra.org/abs/1011.0057

# 7 Appendix

The Appendix is a free c++ computer program written by the author. Its main purpose is to provide with readers the functions of differentiations given by the formulas (19) through (22) in [2], i.e. the formulas of coefficients of the instinct equation. If your computer is equipped with c++ software, you can open your software and copy the following text as a whatever.cpp file. You run the file and get two data files: endpointofcross and oppositeendpoint. Each line of a file contains two numbers that are the coordinates of an end point of a cross. The corresponding line of the other file contains two numbers that are the coordinates of the opposite end point of the cross. Connecting each pair of end points, you get the galaxy snowflake chart for the barred galaxy, NGC 3275, as shown on Fig. 1.

```
#include<stdio.h>
#include<math.h>
#include<iomanip.h>
#include<fstream.h>
long double pi=3.14159265358979323846,hpi=pi*0.5,tpi=pi*2,qpi=pi*0.25;
const int nb=4;
long double s0,d0,d1,b0[nb],b1[nb],b2[nb];

long double r0( long double x, long double y);
long double gi(int i, long double x, long double y);
long double ri(int i, long double x, long double y);
long double rh( long double x, long double y);
long double f0x( long double x, long double y);
long double f0y( long double x, long double y);
long double f0xx( long double x, long double y);
long double f0xy( long double x, long double y);
long double f0yy( long double x, long double y);
long double f0xxx( long double x, long double y);
long double f0xxy( long double x, long double y);
long double f0xyy( long double x, long double y);
long double f0yyy( long double x, long double y);
long double gix( int i, long double x, long double y);
long double giy( int i, long double x, long double y);
long double gixx( int i, long double x, long double y);
long double gixy( int i, long double x, long double y);
long double giyy( int i, long double x, long double y);
long double gixxx( int i, long double x, long double y);
long double gixxy( int i, long double x, long double y);
long double gixyy( int i, long double x, long double y);
long double giyyy( int i, long double x, long double y);
long double fix( int i, long double x, long double y);
long double fiy( int i, long double x, long double y);
long double fixx( int i, long double x, long double y);
long double fixy( int i, long double x, long double y);
long double fiyy( int i, long double x, long double y);
long double fixxx( int i, long double x, long double y);
long double fixxy( int i, long double x, long double y);
long double fixyy( int i, long double x, long double y);
long double fiyyy( int i, long double x, long double y);
long double fx( long double x, long double y);
long double fy( long double x, long double y);
long double fxx( long double x, long double y);
long double fxy( long double x, long double y);
long double fyy( long double x, long double y);
long double fxxx( long double x, long double y);
long double fxxy( long double x, long double y);
long double fxyy( long double x, long double y);
long double fyyy( long double x, long double y);
long double ca( long double x, long double y);
long double cb( long double x, long double y);
long double cc( long double x, long double y);
long double cd( long double x, long double y);
struct complx solvequad( long double p, long double q, long double r,
long double s, long double t, int isc);
struct complx
{
long double real;
long double img;
int iscomplex; // 0 = False, 1 = True
};
int main(){
ofstream of1(”endpointofcross”);
of1<<setw(16)<<setprecision (10)<< resetiosflags( ios::left )
<<setiosflags(ios::fixed — ios::showpoint);
ofstream of2(”oppositeendpoint”);
of2<<setw(16)<<setprecision (10)<< resetiosflags( ios::left )
<<setiosflags(ios::fixed — ios::showpoint);
```

```
int iscub;
long double a,b,c,d,x1,x2,x4,x5;
struct complx x3;

const int M=40,N=40,hM=M/2,hN=N/2;
long double dl=0.2,X,Y; //dl is size of cross
long double DI,DJ,epX,epY; //(epX epY) = end point of cross
int flag,m,n,nn; //flag =1: there exist at most one cubic root

s0=11; //fitted galaxy length on a side of the square image
d0=4000;d1=-1.3; //fitted galaxy exponential disk
b0[1]=134;b1[1]=1.76;b2[1]=-0.2; //fitted dual-handle structure
b0[2]=72; b1[2]=3.25;b2[2]=-0.095;//fitted dual-handle structure
DI=2*s0/M;DJ=2*s0/N;

for(m=1;m<=hM;m++){
X=(m-hM-0.5)*DI;
for(n=N;n>=hN+1;n–){
Y=(n-hN-0.5)*DJ;
a=ca(X,Y);
b=cb(X,Y);
c=cc(X,Y);
d=cd(X,Y);
if(a<0){a=-a;b=-b;c=-c;d=-d;}

if (a > 0){
if (d != 0 ){
x1=9; iscub=1; //value of x1 is arbitrarily chosen
//The cubic equation program is taken at
//http://www.dreamincode.net/code/snippet533.htm
/*The following for loop finds an approximate solution
for the equation by using Newton-Raphson Method:
x1=x0-(f(x0)/f'(x0))
x2=x1-(f(x1)/f'(x1))
...
...
the solution is more accurate with more iterations*/
for(nn=0;nn<=200;nn++){
x2=x1-
(((a*x1*x1*x1)+(b*x1*x1)+(c*x1)+d)/((3*a*x1*x1)+(2*b*x1)+c));
x1=x2;
}
x1=x2;
x3 = solvequad(a,b,c,d,x2, iscub);

if (x3.iscomplex == 0){
flag=0;
x4=x3.real + x3.img;
x5=x3.real - x3.img;
}else{
flag=1;
}
}else{ x1=0;
iscub=0;
x3 = solvequad(a,a,b,c,d, iscub);
if (x3.iscomplex == 0){
flag=0;
x4=x3.real + x3.img;
x5=x3.real - x3.img;
}else{
flag=1;
}
}
}else{
iscub = 0; //BE CAREFULL: we must avoid a = 0 !!!!
flag=1; x1=111111111111111111111111111111111111111111;
}

// give the endpoint position of galaxy snowflakes:
if( (m==(N-n+1))&&(m<8) ){; //we must avoid a = 0 !!!!
}else{
if(flag==0){ // three crosses
epX=dl*cos(0.5*atan(x1));
epY=dl*sin(0.5*atan(x1));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;
epX=dl*cos(hpi+0.5*atan(x1));
epY=dl*sin(hpi+0.5*atan(x1));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;

epX=dl*cos(0.5*atan(x4));
epY=dl*sin(0.5*atan(x4));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;
epX=dl*cos(hpi+0.5*atan(x4));
epY=dl*sin(hpi+0.5*atan(x4));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;

epX=dl*cos(0.5*atan(x5));
epY=dl*sin(0.5*atan(x5));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;
```

```
epX=dl*cos(hpi+0.5*atan(x5));
epY=dl*sin(hpi+0.5*atan(x5));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;

}else{ // one cross

epX=dl*cos(0.5*atan(x1));
epY=dl*sin(0.5*atan(x1));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;
epX=dl*cos(hpi+0.5*atan(x1));
epY=dl*sin(hpi+0.5*atan(x1));
of1<<X-epX<<” ”<<Y-epY<<endl;
of2<<X+epX<<” ”<<Y+epY<<endl;
}
} // avoid a = 0 claimed by your computer !!!!
} // n
} // m
return 0;
}

struct complx solvequad( long double p, long double q,
long double r, long double s, long double t, int isc)
{ long double a,b,c,i;
int iscu = isc;
struct complx x;
if (iscu == 1){
a=p; b=q+(t*p); c=(-s)/t; x.real =((-b)/(2*a));
i=((b*b)-(4*a*c));
if (i>=0)
{ x.img =(sqrt(i))/(2*a);
x.iscomplex = 0;
}else{
x.img =(sqrt(-i))/(2*a);
x.iscomplex = 1;
}
} else {
a=q; b=r; c=s; x.real =((-b)/(2*a));
i=((b*b)-(4*a*c));
if (i>=0)
{ x.img =(sqrt(i))/(2*a);
x.iscomplex = 0;
}else{
x.img =(sqrt(-i))/(2*a);
x.iscomplex = 1;
}
```

```
}
return x;
}

long double r0( long double p, long double q){
long double x=p, y=q;
return d0*exp(d1*pow(x*x+y*y,0.5));
}
long double gi(int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.5*(x*x+y*y-b1[i]*b1[i]
+pow((x*x+y*y-b1[i]*b1[i])*(x*x+y*y-b1[i]*b1[i])+4*b1[i]*b1[i]*x*x,0.5));
}
long double ri(int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return b0[i]*exp((b2[i]/3)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5));
}
long double f0x( long double p, long double q){
long double x=p, y=q;
return d1*x/pow(x*x+y*y,0.5);
}
long double f0y( long double p, long double q){
long double x=p, y=q;
return d1*y/pow(x*x+y*y,0.5);
}
long double f0xx( long double p, long double q){
long double x=p, y=q;
return -d1*x*x/pow(x*x+y*y,1.5)+d1/pow(x*x+y*y,0.5);
}
long double f0xy( long double p, long double q){
long double x=p, y=q;
return -d1*x*y/pow(x*x+y*y,1.5);
}
long double f0yy( long double p, long double q){
long double x=p, y=q;
return -d1*y*y/pow(x*x+y*y,1.5)+d1/pow(x*x+y*y,0.5);
}
long double f0xxx( long double p, long double q){
long double x=p, y=q;
return 3*d1*x*x*x/pow(x*x+y*y,2.5)-3*d1*x/pow(x*x+y*y,1.5);
}
long double f0xxy( long double p, long double q){
long double x=p, y=q;
return 3*d1*x*x*y/pow(x*x+y*y,2.5)-d1*y/pow(x*x+y*y,1.5);
}
```

```
long double f0xyy( long double p, long double q){
long double x=p, y=q;
return 3*d1*x*y*y/pow(x*x+y*y,2.5)-d1*x/pow(x*x+y*y,1.5);
}
long double f0yyy( long double p, long double q){
long double x=p, y=q;
return 3*d1*y*y*y/pow(x*x+y*y,2.5)-3*d1*y/pow(x*x+y*y,1.5);
}
long double gix( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return x+(x*x*x+x*y*y+x*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double giy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return y+(y*y*y+x*x*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double gixx( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 1-2*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+(3*x*x+y*y+b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double gixy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -2*(x*x*x+x*y*y+x*b1[i]*b1[i])*(y*y*y+x*x*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+2*x*y/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double giyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 1-2*(y*y*y+x*x*y-y*b1[i]*b1[i])*(y*y*y+x*x*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+(3*y*y+x*x-b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double gixxx( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 12*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],2.5)
-6*(x*x*x+x*y*y+x*b1[i]*b1[i])*(3*x*x+y*y+b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+6*x/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double gixxy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 12*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*y+y*y*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],2.5)
-8*(x*x*x+x*y*y+x*b1[i]*b1[i])*x*y/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
-2*(3*x*x+y*y+b1[i]*b1[i])*(x*x*y+y*y*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+2*y/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double gixyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 12*(y*y*y+x*x*y-y*b1[i]*b1[i])*(y*y*y+x*x*y-y*b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],2.5)
-8*(y*y*y+x*x*y-y*b1[i]*b1[i])*x*y/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
-2*(x*x+3*y*y-b1[i]*b1[i])*(x*x*x+x*y*y+x*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+2*x/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double giyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 12*(y*y*y+x*x*y-y*b1[i]*b1[i])*(y*y*y+x*x*y-y*b1[i]*b1[i])*(y*y*y+x*x*y-y*b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],2.5)
-6*(y*y*y+x*x*y-y*b1[i]*b1[i])*(3*y*y+x*x-b1[i]*b1[i])/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],1.5)
+6*y/
pow(x*x*x*x+2*x*x*y*y+2*b1[i]*b1[i]*x*x+y*y*y*y-2*b1[i]*b1[i]*y*y+b1[i]*b1[i]*b1[i]*b1[i],0.5);
}
long double fix( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*
((1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y));
}
```

```
long double fiy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y);
}
long double fixx( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.25*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*pow(
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y),2)
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
(-4*b1[i]*b1[i]*x/(gi(i,x,y)*gi(i,x,y))+(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gix(i,x,y))*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y) );
}
long double fixy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.25*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*(
pow(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)),2)*gix(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y)*2*b1[i]*b1[i]*x/gi(i,x,y) )
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gix(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/pow(gi(i,x,y),2))*gixy(i,x,y)
-(2*b1[i]*b1[i]*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y) );
}
long double fiyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return 0.25*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
pow(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)),2)*giy(i,x,y)*giy(i,x,y)
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giy(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/pow(gi(i,x,y),2))*giyy(i,x,y) );
}
long double fixxx( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -0.125*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*pow(
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y),3)
+0.75*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
( (1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y) )*(
(-4*b1[i]*b1[i]*x/(gi(i,x,y)*gi(i,x,y))+(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gix(i,x,y))*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y) )
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
-(4*b1[i]*b1[i]/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)
+(12*b1[i]*b1[i]*x/pow(gi(i,x,y),3))*gix(i,x,y)*gix(i,x,y)
```

```
-(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),4))*pow(gix(i,x,y),3)
+(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gixx(i,x,y)*gix(i,x,y)
-(6*b1[i]*b1[i]*x/pow(gi(i,x,y),2))*gixx(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixxx(i,x,y)
-(2*b1[i]*b1[i]/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y) );
}
long double fixxy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -0.125*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y)*pow(
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y),2)
+0.5*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
( (1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y) )*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giy(i,x,y)*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixy(i,x,y)
-(2*b1[i]*b1[i]*x/pow(gi(i,x,y),2))*giy(i,x,y) )
+0.25*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y)*(
(-4*b1[i]*b1[i]*x/(gi(i,x,y)*gi(i,x,y))+(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gix(i,x,y))*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y) )
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
8*(b1[i]*b1[i]*x/pow(gi(i,x,y),3))*gix(i,x,y)*giy(i,x,y)
-(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),4))*pow(gix(i,x,y),2)*giy(i,x,y)
+(4*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gixy(i,x,y)*gix(i,x,y)
-(4*b1[i]*b1[i]*x/pow(gi(i,x,y),2))*gixy(i,x,y)
+(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gixx(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixxy(i,x,y)
-(2*b1[i]*b1[i]/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y) );
}
long double fixyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -0.125*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*
pow((1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y),2)*
( (1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y) )
+0.5*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y)*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giy(i,x,y)*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixy(i,x,y)
-(2*b1[i]*b1[i]*x/pow(gi(i,x,y),2))*giy(i,x,y) )
+0.25*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
( (1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y) )*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giy(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giyy(i,x,y) )
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
4*(b1[i]*b1[i]*x/pow(gi(i,x,y),3))*giy(i,x,y)*giy(i,x,y)
```

```
-(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),4))*pow(giy(i,x,y),2)*gix(i,x,y)
+(4*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*gixy(i,x,y)*giy(i,x,y)
-(2*b1[i]*b1[i]*x/pow(gi(i,x,y),2))*giyy(i,x,y)
+(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giyy(i,x,y)*gix(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*gixyy(i,x,y) );
}
long double fiyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -0.125*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*pow(
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y),3)
+0.75*(b2[i]/pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*
(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giy(i,x,y)*(
(2*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giy(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giyy(i,x,y) )
+0.5*b2[i]*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*(
-(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),4))*pow(giy(i,x,y),3)
+(6*b1[i]*b1[i]*x*x/pow(gi(i,x,y),3))*giyy(i,x,y)*giy(i,x,y)
+(1-b1[i]*b1[i]*x*x/(gi(i,x,y)*gi(i,x,y)))*giyyy(i,x,y) );
}
long double rh( long double p, long double q){
long double x=p, y=q;
return r0(x,y)+ri(1,x,y)+ri(2,x,y);
}
long double fx( long double p, long double q){
long double x=p, y=q;
return (r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))/rh(x,y);
}
long double fy( long double p, long double q){
long double x=p, y=q;
return (r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))/rh(x,y);
}
long double fxx( long double p, long double q){
long double x=p, y=q;
return -(1/pow(rh(x,y),2))*pow(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y),2)
+(1/rh(x,y))*(r0(x,y)*(f0x(x,y)*f0x(x,y)+f0xx(x,y))
+ri(1,x,y)*(fix(1,x,y)*fix(1,x,y)+fixx(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fix(2,x,y)+fixx(2,x,y)) );
}
long double fxy( long double p, long double q){
long double x=p, y=q;
return -(1/pow(rh(x,y),2))*(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))
(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))
+(1/rh(x,y))*(r0(x,y)*(f0x(x,y)*f0y(x,y)+f0xy(x,y))
+ri(1,x,y)*(fix(1,x,y)*fiy(1,x,y)+fixy(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fiy(2,x,y)+fixy(2,x,y)) );
}
```

```
long double fyy( long double p, long double q){
long double x=p, y=q;
return -(1/pow(rh(x,y),2))*pow(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y),2)
+(1/rh(x,y))*(r0(x,y)*(f0y(x,y)*f0y(x,y)+f0yy(x,y))
+ri(1,x,y)*(fiy(1,x,y)*fiy(1,x,y)+fiyy(1,x,y))
+ri(2,x,y)*(fiy(2,x,y)*fiy(2,x,y)+fiyy(2,x,y)) );
}
long double fxxx( long double p, long double q){
long double x=p, y=q;
return (2/pow(rh(x,y),3))*pow(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y),3)
-(3/pow(rh(x,y),2))*(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))*
(r0(x,y)*(f0x(x,y)*f0x(x,y)+f0xx(x,y))
+ri(1,x,y)*(fix(1,x,y)*fix(1,x,y)+fixx(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fix(2,x,y)+fixx(2,x,y)) )
+(1/rh(x,y))*(r0(x,y)*(pow(f0x(x,y),3)+3*f0x(x,y)*f0xx(x,y)+f0xxx(x,y))
+ri(1,x,y)*(pow(fix(1,x,y),3)+3*fix(1,x,y)*fixx(1,x,y)+fixxx(1,x,y))
+ri(2,x,y)*(pow(fix(2,x,y),3)+3*fix(2,x,y)*fixx(2,x,y)+fixxx(2,x,y)) );
}
long double fxxy( long double p, long double q){
long double x=p, y=q;
return (2/pow(rh(x,y),3))*pow(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y),2)*
(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))
-(2/pow(rh(x,y),2))*(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))*
(r0(x,y)*(f0x(x,y)*f0y(x,y)+f0xy(x,y))
+ri(1,x,y)*(fix(1,x,y)*fiy(1,x,y)+fixy(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fiy(2,x,y)+fixy(2,x,y)) )
-(1/pow(rh(x,y),2))*(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))*
(r0(x,y)*(f0x(x,y)*f0x(x,y)+f0xx(x,y))
+ri(1,x,y)*(fix(1,x,y)*fix(1,x,y)+fixx(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fix(2,x,y)+fixx(2,x,y)) )
+(1/rh(x,y))*(r0(x,y)*((f0x(x,y)*f0x(x,y)+f0xx(x,y))*f0y(x,y)+2*f0x(x,y)*f0xy(x,y)+f0xxy(x,y))
+ri(1,x,y)*((fix(1,x,y)*fix(1,x,y)+fixx(1,x,y))*fiy(1,x,y)+2*fix(1,x,y)*fixy(1,x,y)+fixxy(1,x,y))
+ri(2,x,y)*((fix(2,x,y)*fix(2,x,y)+fixx(2,x,y))*fiy(2,x,y)+2*fix(2,x,y)*fixy(2,x,y)+fixxy(2,x,y)) );
}
long double fxyy( long double p, long double q){
long double x=p, y=q;
return (2/pow(rh(x,y),3))*pow(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y),2)*
(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))
-(2/pow(rh(x,y),2))*(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))*
(r0(x,y)*(f0x(x,y)*f0y(x,y)+f0xy(x,y))
+ri(1,x,y)*(fix(1,x,y)*fiy(1,x,y)+fixy(1,x,y))
+ri(2,x,y)*(fix(2,x,y)*fiy(2,x,y)+fixy(2,x,y)) )
-(1/pow(rh(x,y),2))*(r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y))*
(r0(x,y)*(f0y(x,y)*f0y(x,y)+f0yy(x,y))
+ri(1,x,y)*(fiy(1,x,y)*fiy(1,x,y)+fiyy(1,x,y))
+ri(2,x,y)*(fiy(2,x,y)*fiy(2,x,y)+fiyy(2,x,y)) )
+(1/rh(x,y))*(r0(x,y)*((f0y(x,y)*f0y(x,y)+f0yy(x,y))*f0x(x,y)+2*f0y(x,y)*f0xy(x,y)+f0xyy(x,y))
+ri(1,x,y)*((fiy(1,x,y)*fiy(1,x,y)+fiyy(1,x,y))*fix(1,x,y)+2*fiy(1,x,y)*fixy(1,x,y)+fixyy(1,x,y))
+ri(2,x,y)*((fiy(2,x,y)*fiy(2,x,y)+fiyy(2,x,y))*fix(2,x,y)+2*fiy(2,x,y)*fixy(2,x,y)+fixyy(2,x,y)) );
}
long double fyyy( long double p, long double q){
long double x=p, y=q;
return (2/pow(rh(x,y),3))*pow(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y),3)
-(3/pow(rh(x,y),2))*(r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y))*
(r0(x,y)*(f0y(x,y)*f0y(x,y)+f0yy(x,y))
+ri(1,x,y)*(fiy(1,x,y)*fiy(1,x,y)+fiyy(1,x,y))
+ri(2,x,y)*(fiy(2,x,y)*fiy(2,x,y)+fiyy(2,x,y)) )
+(1/rh(x,y))*(r0(x,y)*(pow(f0y(x,y),3)+3*f0y(x,y)*f0yy(x,y)+f0yyy(x,y))
+ri(1,x,y)*(pow(fiy(1,x,y),3)+3*fiy(1,x,y)*fiyy(1,x,y)+fiyyy(1,x,y))
+ri(2,x,y)*(pow(fiy(2,x,y),3)+3*fiy(2,x,y)*fiyy(2,x,y)+fiyyy(2,x,y)) );
}
long double ca( long double p, long double q){
long double x=p, y=q;
return (pow(fy(x,y),4)-pow(fx(x,y),4))*pow(fyy(x,y)-fxx(x,y),2)
+(pow(fy(x,y),2)-pow(fx(x,y),2))*(pow(fy(x,y),2)-pow(fx(x,y),2))*(
fx(x,y)*fxxx(x,y)-fx(x,y)*fxyy(x,y)+fy(x,y)*fxxy(x,y)-fy(x,y)*fyyy(x,y) );
}
long double cb( long double p, long double q){
long double x=p, y=q;
return 2*fx(x,y)*fy(x,y)*(pow(fy(x,y),2)+pow(fx(x,y),2))*pow(fyy(x,y)-fxx(x,y),2)
+4*fx(x,y)*fy(x,y)*(pow(fy(x,y),2)-pow(fx(x,y),2))*(
fx(x,y)*fxxx(x,y)-fx(x,y)*fxyy(x,y)+fy(x,y)*fxxy(x,y)-fy(x,y)*fyyy(x,y) )
-2*(pow(fy(x,y),2)-pow(fx(x,y),2))*(pow(fy(x,y),2)-pow(fx(x,y),2))*(
fx(x,y)*fxxy(x,y)+fy(x,y)*fxyy(x,y) )
+4*(pow(fy(x,y),4)-pow(fx(x,y),4))*fxy(x,y)*(fyy(x,y)-fxx(x,y));
}
long double cc( long double p, long double q){
long double x=p, y=q;
return 4*fx(x,y)*fx(x,y)*fy(x,y)*fy(x,y)*(
fx(x,y)*fxxx(x,y)-fx(x,y)*fxyy(x,y)+fy(x,y)*fxxy(x,y)-fy(x,y)*fyyy(x,y) )
+8*fx(x,y)*fy(x,y)*(pow(fy(x,y),2)+pow(fx(x,y),2))*fxy(x,y)*(fyy(x,y)-fxx(x,y))
-8*fx(x,y)*fy(x,y)*(pow(fy(x,y),2)-pow(fx(x,y),2))*(fx(x,y)*fxxy(x,y)+fy(x,y)*fxyy(x,y))
+4*fxy(x,y)*fxy(x,y)*(pow(fy(x,y),4)-pow(fx(x,y),4));
}
long double cd( long double p, long double q){
long double x=p, y=q;
return -8*fx(x,y)*fx(x,y)*fy(x,y)*fy(x,y)*(fx(x,y)*fxxy(x,y)+fy(x,y)*fxyy(x,y) )
+8*fx(x,y)*fy(x,y)*fxy(x,y)*fxy(x,y)*(pow(fy(x,y),2)+pow(fx(x,y),2));
}
```