The Structuring Force of Galaxies

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

Abstract The concept of rational structure was suggested in 2000. A flat material distribution is called the rational structure if there exists a special net of orthogonal curves on the plane, and the ratio of mass density at one side of a curve (from the net) to the one at the other side is constant along the curve. Such curve is called a proportion curve. Such net of curves is called an orthogonal net of proportion curves. Eleven years have passed and a rational sufficient condition for given material distribution is finally obtained. This completes the mathematical basis for the study of rational structure and its galaxy application. People can fit the stellar distribution of a barred spiral galaxy with exponential disk and dual-hand structure by varying their parameter values. If the conjecture is proved that barred galaxies satisfy a rational suffcient condition then the assumption of galaxy rational origin will be established.

keywords: Rational Structure, Spiral Galaxy, Galaxy Conjecture

PACS: 02.60.Lj, 98.52.Nr

1 Introduction to Rational Structure

All galaxies demonstrate the structure of uneven material distribution. The density distribution of a relatively independent galaxy is always simple and orderly. Thus, discovery of the fundamental law to galaxy mass distribution is the basic step toward the understanding of galaxies. Relatively independent galaxies fall into two categories. One is the three-dimensional elliptical galaxies, and the other is the flat-shaped spiral galaxies. 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. In fact, the exponential disk of spiral galaxies can coexist with bar-shaped structure. There are two types of spiral galaxies. A barred spiral 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.

This paper studies spiral galaxies only. Because spiral galaxies are flat-shaped, we use the function of two variables

$$\rho(x, y) \tag{1}$$

to represent the stellar density distribution of spiral galaxies. Modern galaxy images are generally digital ones. Thus, a long-wavelength image of face-on spiral galaxy is essentially an array of positive numbers proportional to the stellar density $\rho(i, j)$. The size of the array is dependent on the resolution of the image. We use the array to draw galaxy image with the image brightness at each point being proportional to the corresponding value of the array.

Since the ancient times, humans have not known which density distribution an independent natural material system should take. Scientists who study a flat material distribution generally consider its level curves, i.e., the contours of constant density. This resembles the situation that ancient mathematicians could not study mathematics beyond Algebra. Since Newton and Leibniz discovered Calculus in the seventeenth century, a revolutionary advance of mathematical science has been made. Back to the case of spiral galaxies, I want to study the change in the stellar density. Therefore I consider not the contours of constant density, but the variance rate of the density. Starting at any point on a plane are infinite directions along each of which we can calculate the variance rate of the density. The law of galaxy structure must be some property of invariance governing the rate of variance. I suggested a simple property of invariance. Among the infinite directions starting at a point there exist two special ones which are mutually perpendicular. I consider only the variance rates along the two special directions. At every point on the plane are there two such special directions, and I always consider the variance rates along the two special directions. In fact, I never consider the variance rate of the density $\rho(x, y)$ but the variance rate of the logarithmic density,

$$f(x, y) = \ln \rho(x, y) \tag{2}$$

(the variance rate of $f(x, y)$ along the perpendicular direction to a curve is proportional to the density ratio of $\rho(x, y)$ explained in Abstract of the paper). Therefore, considering the variance rate of the function $f(x, y)$, we get two functions $u(x, y)$ and $v(x, y)$ which record the variance rates along the two special directions. Because these directions can connect into curves on the plane, the two special directions connect into two sets of parallel curves which are mutually perpendicular, called an orthogonal net of curves. The property of invariance I suggested is that the two sets of curves are, respectively, the level curves of the two functions $u(x, y)$ and $v(x, y)$. That is, the variance rate of $f(x, y)$ along the normal direction of any curve (from the net) is constant along the curve. This is my concept of rational galaxy structure.

Definition of rational structure (see paper [1]): The logarithmic density distribution $f(x, y)$ on a plane is not arbitrary. There exists a special net of orthogonal curves on the plane. The variance rate of $f(x, y)$ along the normal direction of a curve (from the net) is constant along the curve. The curve is called a proportion curve. The net of curves is called an orthogonal net of proportion curves. This kind of density distribution $\rho(x, y)$ is called a rational structure.

It is certain that the mathematical result presented in Section 2 completes the necessary mathematical basis for rational structure study. It would be entertaining for me to give a simple chronological account of how the rational structure concept came up. I am not a gifted person but I was born with a huge mind of unrealistic dream. The dream grew up through my ten-year period of primary and secondary education which coincided with the movement of Chinese Cultural Revolution. The dream led my choice of Mathematics department for me to start my college education. But I realized that physical science is more attractive to me. I took ten years to pass the examination to enter a Master-degree program to study elementary particle theory, which proved that I am not a gifted person. During the program I was impressed by the scientists' enthusiasm at that time that nonlinear science would be the center stage of future scientific exploration which might solve such hard problem as biological phenomena. Later on I joined reluctantly the 'realistic' PhD program of Astronomy in China. My mind, however, was still full of such grandiose ideas as quantum gravity. Fortunately I used some 'realistic' method provided by nonlinear science to analyze astronomic data, and got some results published to earn the degree. The nonlinear method is very simple: taking among a distribution of data the ratio of two individual values at each pair of relevant positions. In 1998 I was accepted to enter an Astronomic PhD program in USA before I was approved the first PhD in China. The PhD program in USA was the most realistic one: analysis of galaxy images. Then it was straightforward for me to use the same method to study galaxy images as explained in Abstract of the paper. This led naturally the concept of rational structure. The image analysis suggested that spiral galaxy arms would be some rational structure.

I was greatly challenged to search for the sufficient condition which, if satisfied, guarantees that a given logarithmic density distribution $f(x, y)$ be rational. The condition is finally obtained in the current paper after eleven years have passed. The breakthrough was achieved in 2010 when I found a rational necessary condition for a given structure which is called the instinct equation [2]. A primitive form of the equation was almost obtained in the year of 2002. However, by the same year of 2002 after much painful experience of rational structure study I had changed the assumption of given logarithmic density distribution into the assumption of given orthogonal net of curves [3-5]. The study of the latter assumption suggested that there might not exist such orthogonal net of simple curves whose pattern is not axi-symmetric nor bilaterally symmetric yet corresponds to a spiral-shaped rational structure resembling galaxy arms. Many evidences suggest that spiral galaxy arms are not rational structure. Instead they are the disturbing waves to the rational body of galaxies. In order to minimize the disturbance, the perturbation waves propagate along the orthogonal proportion curves or the non-orthogonal proportion curves. Within the year of 2005 I successfully modeled all galaxy structures [6-8]. But submission of the central idea to over fifty scientific journals was all rejected. The initial horrible experience of rejection by those top astronomic journals made me frustrated. I was crazy about the beautiful concept of rational structure, and believed that Einstein field equation must suffer some defect. That is why I proposed the simple models of quantum gravity and flat universe in 2006 [9,10]. If my idea of rational galaxy structure is finally successful, people should put major credit to the editorial staff of Electronic Journal of Theoretical Physics. It is the only journal which finally accepted my central idea. Accordingly I resumed my rational galaxy study, and returned naturally to the assumption of given logarithmic density distribution. I obtained a rational necessary condition, the instinct equation in [2], under the assumption. The current paper presents the rational sufficient condition under the same assumption.

In Section 2, the necessary mathematical basis for rational structure study is achieved. I study the reversibility of the steps which led to the instinct equation in paper [2]. The factorization of the instinct equation is then given. The central result of the paper is the rational sufficient condition under the assumption of given logarithmic density distribution. Section 3 is dedicated to the application to barred spiral galaxies and Section 4 is the conclusion.

2 Instinct-Equation Factorization and Rational Sufficient Condition

2.1 Reversibility of the Steps Leading to Instinct Equation

Paper [2] is a breakthrough of rational-structure study because a rational necessary condition for a given density distribution was obtained. That is, if a given logarithmic density distribution $f(x,y)$ is rational then the slope of its orthogonal proportion curves must satisfy a cubic algebraic equation, called the instinct equation. The coefficients of the equation are known, which are combinations of the partial derivatives to the given logarithmic density. The slope is the tangent value of the azimuthal angle of the tangential line of the curves, i.e., $\tan\alpha$. Given a logarithmic density distribution $f(x,y)$, if we can prove that an orthogonal net of proportion curves exists which corresponds to the given density distribution then the density distribution must be rational. However, the existence of a smooth distribution of scalar angle

$$\alpha(x,y) \tag{3}$$

may not define a vectorial net of orthogonal curves. Therefore, a smooth root of the instinct equation can not guarantee that the given logarithmic density be a rational one. Now we review the mathematical steps in paper [2] that led to the instinct equation, and see which steps are reversible so that we can find a rational sufficient condition. The penultimate step which led to the instinct equation is a differential equation system, i.e., the formula (16) in the same paper,

$$\begin{cases} \alpha'_x & = f'_y \frac{(f''_{yy}-f''_{xx})\tan 2\alpha+2f''_{xy}}{(f'^2_y-f'^2_x)\tan 2\alpha+2f'_xf'_y} \\ & = f'_y \frac{\lambda\gamma+\omega}{\Lambda\gamma+\Omega} = f'_yF(\lambda,\omega,\Lambda,\Omega,\gamma), \\ \alpha'_y & = -f'_x \frac{(f''_{yy}-f''_{xx})\tan 2\alpha+2f''_{xy}}{(f'^2_y-f'^2_x)\tan 2\alpha+2f'_xf'_y} \\ & = -f'_x \frac{\lambda\gamma+\omega}{\Lambda\gamma+\Omega} = -f'_xF(\lambda,\omega,\Lambda,\Omega,\gamma) \end{cases} \tag{4}$$

where

$$\gamma = \tan 2\alpha, \tag{5}$$

and

$$\begin{array}{l} F(\lambda,\omega,\Lambda,\Omega,\gamma) = \frac{\lambda\gamma+\omega}{\Lambda\gamma+\Omega}, \\ \lambda = f''_{yy} - f''_{xx}, \quad \omega = 2f''_{xy}, \\ \Lambda = f'^2_y - f'^2_x, \quad \Omega = 2f'_xf'_y. \end{array} \tag{6}$$

Note that the equation system (4) is highly symmetric. I find out that all mathematical steps leading to the equation system (including the system itself) are completely reversible. Therefore, the equation system (4) is a rational sufficient condition for a given logarithmic density distribution $f(x,y)$. That is, if the equation system (4) has a smooth solution $\alpha(x,y)$ then we can prove that an orthogonal net of proportion curves exists which corresponds to the given density distribution. Now we prove this proposition. That is, we prove the existence of a net of proportion curves. Firstly, we define the curves based on the smooth function $\alpha(x,y)$. As explained in the first paragraph of the current Section, we should not imagine that $\alpha(x,y)$ itself can define a net of curves. For example, such differential equation as $dy/dx = s(x,y)$ may not be integrable. If it is integrable then $s(x,y)$ must be the slope of the net of integral curves: $s(x, y) = \tan \alpha$. Because the equation system (4) has a smooth solution $\alpha(x, y)$ and the logarithmic density $f(x, y)$ is a given smooth function, we can use them to define two functions,

$$\begin{aligned} u(x, y) &= f'_x \cos \alpha + f'_y \sin \alpha, \\ v(x, y) &= -f'_x \sin \alpha + f'_y \cos \alpha. \end{aligned} \tag{7}$$

The level curves of the two functions present two sets of parallel curves. They are given existent smooth curves. Now we need prove that the curves are indeed the required orthogonal net of proportion curves which corresponds to the given density distribution $f(x, y)$. To do so, we need prove the following results,

$$\begin{aligned} -u'_x \sin \alpha + u'_y \cos \alpha = 0, \\ v'_x \cos \alpha + v'_y \sin \alpha = 0. \end{aligned} \tag{8}$$

The proof is very simple because the partial derivatives of $\alpha(x, y)$ are given by the formula (4). Taking partial derivatives to the formulas (7) and substituting the result into the left-hand sides of the formulas (8), we will know that the equations (8) are true if the formulas (4) are substituted in the result. The formulas (8) indicate that the level curves of $u(x, y)$ are always perpendicular to the unit vector field $\mathbf{v} = (\cos \alpha, \sin \alpha)$ while the ones of $v(x, y)$ are always parallel to the vector. Therefore, the two sets of parallel curves are indeed orthogonal to each other and form an orthogonal net. Finally, we need to prove that these curves are proportion ones. That is, the variance rate of $f(x, y)$ along the normal direction of any curve (from the net) is constant along the curve. As indicated by the formulas (7), the variance rate of $f(x, y)$ along the vector direction is $u(x, y)$. Because the vector field is always perpendicular to the level curves of $u(x, y)$ and the variance rate of $f(x, y)$ along the normal direction of the level curves is $u(x, y)$ itself, the variance rate of $f(x, y)$ along the normal direction of the level curves is indeed constant along the curves. Similarly we can prove that the variance rate of $f(x, y)$ along the normal direction of the level curves of $v(x, y)$ is constant along the curves. This completes our proof that a smooth solution of the differential equation system (4) is a rational sufficient condition for a given logarithmic density distribution $f(x, y)$. However, the equation system may not be integrable to have a smooth solution. Therefore, it is not directly applicable to galaxy structure study. The instinct equation, however, is a cubic algebraic equation which is much easier for solution and deserves further discussion. In the following we discuss its factorization and seek a much simpler rational sufficient condition.

2.2 Instinct-Equation Factorization and Rational Orthogonal Condition

Differentiating the equations (4) to get α''_{xy}, α''_{yx} respectively, and using the equations themselves in the final results, we have the instinct equation by setting $\alpha''_{xy} - \alpha''_{yx} = 0$,

$$a(x, y)\gamma^3 + b(x, y)\gamma^2 + c(x, y)\gamma + d(x, y) = 0 \tag{9}$$

where $\gamma = \tan 2\alpha$ (see the formula (5)) and the coefficients are given by the formulas (19) through (22) in paper [2]. In fact, the cubic equation can be factorized, which has a quadratic factor and a linear factor. To do so, we review the details of the above-said differential process,

$$\alpha''_{xy} = f''_{yy}F + f'_y \sum_{\lambda=\lambda}^{\Omega} F'_\lambda \lambda'_y + f'_y F'_\gamma \gamma'_y, \tag{10}$$

$$\alpha''_{yx} = -f''_{xx}F - f'_x \sum_{\lambda=\lambda}^{\Omega} F'_\lambda \lambda'_x - f'_x F'_\gamma \gamma'_x. \tag{11}$$

Setting $\alpha''_{xy} - \alpha''_{yx} = 0$ gives

$$(f''_{xx} + f''_{yy})F + \sum_{\lambda=\lambda}^{\Omega}(f'_x\lambda'_x + f'_y\lambda'_y)F'_\lambda + (f'_x\gamma'_x + f'_y\gamma'_y)F'_\gamma = 0. \tag{12}$$

If the given logarithmic density distribution $f(x,y)$ is rational then the last term in (12) must be zero. That is,

$$f'_x\gamma'_x + f'_y\gamma'_y = \nabla f \cdot \nabla\gamma = 0. \tag{13}$$

In fact, as early as in 2001 I knew that a rational structure $f(x,y)$ must satisfy the simple equation,

$$\nabla f \cdot \nabla\alpha = 0 \tag{14}$$

(see the formula (10) in paper [3]) where α, as usual, is the azimuthal angle of the tangential line of the proportion curves. I call the above linear partial differential equation the rational orthogonal condition. It means that the gradient of $f(x,y)$ is everywhere perpendicular to the gradient of angle α. Given the logarithmic density distribution $f(x,y)$, we know that any combination (summation, multiplication) of the solutions of the equation (14) is still its solution because it is a linear equation of the unknown α. Furthermore, any composite function of a solution of the equation is still its solution. Therefore, $\nabla f \cdot \nabla\gamma$ must be zero because $\gamma = \tan 2\alpha$. We are left with a simpler equation,

$$(f''_{xx} + f''_{yy})F + \sum_{\lambda=\lambda}^{\Omega}(f'_x\lambda'_x + f'_y\lambda'_y)F'_\lambda = 0. \tag{15}$$

Substitution of the formulas (6) into the above equation leads to the quadratic instinct equation,

$$A(x,y)\gamma^2 + B(x,y)\gamma + C(x,y) = 0 \tag{16}$$

where

$$\begin{aligned} A = \; & -(f'^2_y + f'^2_x)(f''_{yy} - f''_{xx})^2 \\ & -(f'^2_y - f'^2_x)(f'_x f'''_{xxx} - f'_x f'''_{xyy} + f'_y f'''_{xxy} - f'_y f'''_{yyy}), \end{aligned} \tag{17}$$

$$\begin{aligned} B = \; & -2f'_x f'_y(f'_x f'''_{xxx} - f'_y f'''_{yyy}) - 2f'^3_x f'''_{xxy} + 2f'^3_y f'''_{xyy} \\ & -4(f'^2_y + f'^2_x)f''_{xy}(f''_{yy} - f''_{xx}), \end{aligned} \tag{18}$$

and

$$C = 4f'_x f'_y(f'_x f'''_{xxy} + f'_y f'''_{xyy}) - 4f''^2_{xy}(f'^2_x + f'^2_y). \tag{19}$$

What is the relation between the quadratic and cubic instinct equations? In fact, the quadratic function is the quadratic factor of the cubic function. It is straightforward to show that the linear factor of the cubic function is

$$\Lambda\gamma + \Omega \tag{20}$$

which is the denominator of the function $F(\lambda, \omega, \Lambda, \Omega, \gamma)$ (see the formulas in (6)). Therefore, the required factorization is

$$a\gamma^3 + b\gamma^2 + c\gamma + d = -(A\gamma^2 + B\gamma + C)(\Lambda\gamma + \Omega). \tag{21}$$

2.3 Rational Sufficient Condition

The forward step from the differential equation (4) to the cubic algebraic equation (9) loses much information about the original assumption of rational structure. Therefore, the step is not reversible. For example, we can define the linear instinct equation to be

$$\Lambda\gamma + \Omega = 0. \tag{22}$$

Assume that it has a smooth root $\gamma = \tan 2\alpha$. The single assumption of smooth root can not guarantee that the logarithmic function $f(x, y)$ satisfies the rational orthogonal condition (14). If the following condition is satisfied,

$$(f''_{yy} - f''_{xx})f'_x f'_y - f''_{xy}(f'^2_y - f'^2_x) = 0, \ \text{ i. e., } \ \lambda\Omega - \omega\Lambda = 0, \tag{23}$$

then the rational orthogonal condition is satisfied. Even when the orthogonal condition is satisfied, we can not guarantee that α satisfies the equation system (4). What we know is that we can find a function $M(x, y)$ which satisfies the following equation system,

$$\begin{cases} M'_x = f'_y \frac{(f''_{yy} - f''_{xx}) \tan 2\alpha + 2f''_{xy}}{(f'^2_y - f'^2_x) \tan 2\alpha + 2f'_x f'_y}, \\ M'_y = -f'_x \frac{(f''_{yy} - f''_{xx}) \tan 2\alpha + 2f''_{xy}}{(f'^2_y - f'^2_x) \tan 2\alpha + 2f'_x f'_y} \end{cases} \tag{24}$$

which is different from the equation system (4). The equation systems (24) and (4) share the similar form but the function $M(x, y)$ may not be $\alpha(x, y)$ at all. However, if $M(x, y)$ approaches the value of $\alpha(x, y)$ at some boundary then $M(x, y)$ must be identical to $\alpha(x, y)$ and the given logarithmic function $f(x, y)$ is indeed a rational structure as shown in Section 2.1.

Similarly, the assumption that the quadratic instinct equation (16) has a smooth root $\gamma = \tan 2\alpha$ can not guarantee that the logarithmic function $f(x, y)$ satisfies the rational orthogonal condition (14). Assume the quadratic equation (16) has two roots,

$$\gamma_\pm = \frac{-B \pm \sqrt{\Delta}}{2A} \tag{25}$$

where Δ is the discriminant,

$$\Delta = B^2 - 4AC. \tag{26}$$

Then the rational orthogonal condition (13), $\nabla f \cdot \nabla \gamma_\pm = 0$, becomes

$$\nabla f \cdot \nabla \gamma_+ = \frac{-A}{\sqrt{\Delta}} \left((-B + \sqrt{\Delta}) \nabla f \cdot \nabla \frac{B}{A} + 2A \nabla f \cdot \nabla \frac{C}{A} \right) = 0, \tag{27}$$

$$\nabla f \cdot \nabla \gamma_- = \frac{A}{\sqrt{\Delta}} \left((-B - \sqrt{\Delta}) \nabla f \cdot \nabla \frac{B}{A} + 2A \nabla f \cdot \nabla \frac{C}{A} \right) = 0. \tag{28}$$

The proof of the above formula is very simple because we have the following general formula which holds for the roots of arbitrary quadratic equation $ak^2 + bk + c = 0$,

$$\nabla k_{\pm} = \frac{\mp a}{\sqrt{\Delta}} \left(\left(-b \pm \sqrt{\Delta} \right) \nabla \frac{b}{a} + 2a \nabla f \cdot \nabla \frac{c}{a} \right) \tag{29}$$

where $\Delta = b^2 - 4ac$.

Even when the rational orthogonal condition (27) or (28) is satisfied, we can not guarantee that α satisfies the equation system (4). However, if the given logarithmic function $f(x, y)$ approaches some rational structure at some boundary then the equation system (4) is satisfied and $f(x, y)$ is indeed a global rational structure. These consist of the rational sufficient condition for given logarithmic density distribution.

The formulas of the quadratic coefficients, (17) through (19), can be more simplified,

$$\begin{aligned} A &= (\nabla \cdot \nabla f)\lambda\Lambda + f'_x(\lambda'_x\Lambda - \lambda\Lambda'_x) + f'_y(\lambda'_y\Lambda - \lambda\Lambda'_y) \\ &= (\nabla \cdot \nabla f)\lambda\Lambda + \nabla f \cdot \bar{\nabla}(\lambda\Lambda) \\ &= \bar{\nabla} \cdot (\lambda\Lambda\nabla f) \end{aligned} \tag{30}$$

where I have introduced a new notation $\bar{\nabla}$ which is different from the common gradient ∇. The difference occurs only when applying to the products $\lambda\Lambda$, $\omega\Omega$, or $\lambda\Omega + \omega\Lambda$. Therefore,

$$\begin{aligned} B &= \bar{\nabla} \cdot ((\lambda\Omega + \omega\Lambda)\nabla f), \\ C &= \bar{\nabla} \cdot (\omega\Omega\nabla f). \end{aligned} \tag{31}$$

This is because Λ and Ω came originally from the linear denominators in the formulas (4).

The above results complete the necessary mathematical basis for the study of rational structure. The requirement of rationality is extremely strict. The slope of the proportion curves must satisfy either the linear instinct equation (22) or the quadratic instinct equation (16). The former rational structure is called the rational ground state, $|1>$, and the latter the rational excited state because the equation system and its solution have many similar aspects to the conventional discussion of quantum mechanics. If the "doubled" slope, $\tan 2\alpha$, of the proportion curves of the excited state is γ_+ (or γ_-), the state is called the excited plus state, $|2+>$, (or the excited minus state, $|2->$).

3 Application to Large Scale Galaxy Structure

3.1 Model of Galaxy structure

The above definition of flat rational structure can be generalized to three-dimensional material distribution. I found out that the stellar distribution of elliptical galaxies can be fitted with three-dimensional rational structure whose proportion surfaces can be expressed by the complex reciprocal function [8]. However, this paper deals with spiral galaxies only.

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. There are two kinds of spiral galaxies. A barred spiral galaxy has additional bar structure. Its body is a combination of exponential disk and bar structure. A spiral galaxy without bar structure is called a normal spiral galaxy. The exponential disk of spiral galaxies is a rational structure. It has infinite net of orthogonal curves. One net is composed of the curves of polar coordinates, i.e., all the circles centered at the galaxy center and all the radial lines starting at the same center. The variance rate of the logarithmic stellar density along the normal direction of the circles is denoted by $u(x, y)$, called the radial variance rate of $f(x, y)$. The level curves of $u(x, y)$ is all the circles. The variance rate along the normal direction of the radial lines is denoted by $v(x, y)$, called the axial variance rate of $f(x, y)$. The level curves of $v(x, y)$ is all the radial lines. Because exponential disks are axi-symmetric, the axial variance rate is identically zero. The other infinite nets of orthogonal curves of the exponential disk are all spirals. Because the logarithmic density of the disk is proportional to the value of the radius vector $\mathbf{r}$, it can be proved that these spirals are all equiangular ones. That is, the angle between the tangent direction of the spiral at a point and the polar radial line passing the point is constant along the spiral. Because the gradient value of the logarithmic disk density is constant throughout, the variance rate along the normal direction of equiangular spiral is constant too. Equiangular spirals are also known as golden spirals or logarithmic spirals. Coincidentally, astronomical observations show that the arms of any normal spiral galaxies (i.e, the body structure is the exponential disk itself) are all equiangular spirals. Because exponential disks are axi-symmetric, the corresponding coefficients of any instinct equation are all zeros. Therefore, exponential disks belong to both rational ground state and excited states. There is no need for further study on normal spiral galaxies in this paper.

Without regard to the central bulge, barred spiral galaxies are the superposition of exponential disks and bars. My research indicates that the bar of any barred spiral galaxy is a compound structure of two or three dual-handle structures. Each dual-handle structure is a rational structure whose logarithmic density is

$$f(x, y) = (b_2/3) \left(p(x, y) + b_1^2 x^2 / p(x, y)\right)^{3/2} \tag{32}$$

where b_1, b_2 are constant parameters, r is the polar radius, and

$$p(x, y) = \left(r^2 - b_1^2 + \sqrt{(r^2 - b_1^2)^2 + 4b_1^2 x^2}\right)/2. \tag{33}$$

The lower left corner of Figure 1 is the mass density distribution $\rho(x, y)$ of a dual-handle structure. In fact, it is one of the two dual-handles used to visually simulate the bar of galaxy NGC 3275. The orthogonal net of proportion curves of dual-handle structure is composed of all confocal ellipses and hyperbolas. The distance between the two foci is $2b_1$, known as the length of the dual-handles. The variance rate of $f(x, y)$ along the normal direction to the confocal ellipses is also called the 'radial' variance rate, denoted by $u(x, y)$, and the one along the normal direction to the confocal hyperbolas is also called 'axial' variance rate, denoted by $v(x, y)$. In fact, the normal direction of the ellipses do not generally point to the galaxy center but point to the galaxy bar. For visual purpose, we can draw the image of the function $u(x, y)$ with the image brightness at each point corresponding to the value of $u(x, y)$ and the level curves of the brightness must be all confocal ellipses. The lower right corner of Figure 1 is such image of the function $u(x, y)$ whose calculation, however, was obtained via a different route. With Maple software I verified that dual-handle structure is an excited minus state. Therefore, given the logarithmic density (32), the "doubled" slope $\tan 2\alpha$ of the proportion curves of the excited state is γ_- (see the formula (25), a root of the equation (16)). With the function $f(x,y)$ and the angle α being given, we can calculate $u(x,y)$ using the first formula in (7). The result is displayed in the lower right corner of Figure 1. Because angle α can not be defined along the central line connecting the two foci of ellipses, we see the error of the calculation along the line. This method of $u(x,y)$ calculation with the formula (7) is called the "blind variance-rate" because $f(x,y)$ may not be recovered by its nominal variance rates $u(x,y)$ and $v(x,y)$ if rational sufficient condition is not satisfied.

In this paper, we study the barred spiral galaxy NGC 3275 as an example. Through visual simulation, the bar is found to be the superposition of two dual-handle structures. The upper left corner of Figure 1 shows the simulated whole barred spiral galaxy (exponential disk plus galaxy bar). The fitting values of the parameters d_0, d_1 (exponential disk), b_0, b_1, b_2 (dual-handle structure) are presented in Table 1. The lower left corner of Figure 1 is the simulated shorter dual-handle structure. We denote the density of the structure by $\rho_1(x,y)$. The density of the longer dual-handle structure is denoted by $\rho_2(x,y)$ and the density of the simulated exponential disk is denoted by $\rho_0(x,y)$. Therefore, disregarding the central bulge, we have the visually simulated whole barred galaxy NGC 3275,

$$\rho(x,y) = \rho_0(x,y) + \rho_1(x,y) + \rho_2(x,y). \tag{34}$$

This is displayed in the upper-left corner of Figure 1. The gradient of the corresponding logarithmic density is

$$\nabla f(x,y) = \frac{\rho_0}{\rho}\nabla f_0(x,y) + \frac{\rho_1}{\rho}\nabla f_1(x,y) + \frac{\rho_2}{\rho}\nabla f_2(x,y) \tag{35}$$

where $f_0(x,y), f_1(x,y), f_2(x,y)$ are the logarithmic densities of the simulated exponential disk and dual-handle structures respectively. It indicates that the gradient of the compound logarithmic density is the weighted average of the gradients of the components' logarithmic densities. The weight is the percentage of the component's density over the whole density (not the whole logarithmic density, see the final formula on page 371 in [1]). That is, the body structure is a compound one of rational structures. I have not proved that the compound structure of rational components is still rational. I calculate the "blind variance-rate" $u(x,y)$ for the compound structure (34) as suggested in the previous paragraph, with the assumption that the whole galaxy NGC 3275 might be a rational structure in the ground state. The resulting $u(x,y)$ is displayed in the upper right corner of Figure 1, called the 'basket graph'. The level curves of the graph suggest what the global galaxy proportion curves might be (see also the sketches in Figure 2).

Why is there 'basket graph'? The answer is the following. In the area near galaxy center and the four foci, the density of the exponential disk, i.e. the weight factor in the first term in (35), is the greatest and makes the overwhelming contribution to the gradient (the left-hand side of (35)). However, the absolute values of the logarithmic densities of the dual-handle structures increase with radius cubically whereas the absolute value of the logarithmic density of exponential disk increases linearly. Therefore, the gradient values of the logarithmic densities of the dual-handle structures, i.e. the second factors in the second and third terms in (35), are the greatest and make the overwhelming contribution to the gradient at around the area of the 'basket' rim. However, far away from the galaxy center, the first factor in the disk term, i.e. the weight factor, makes larger impact, and the density

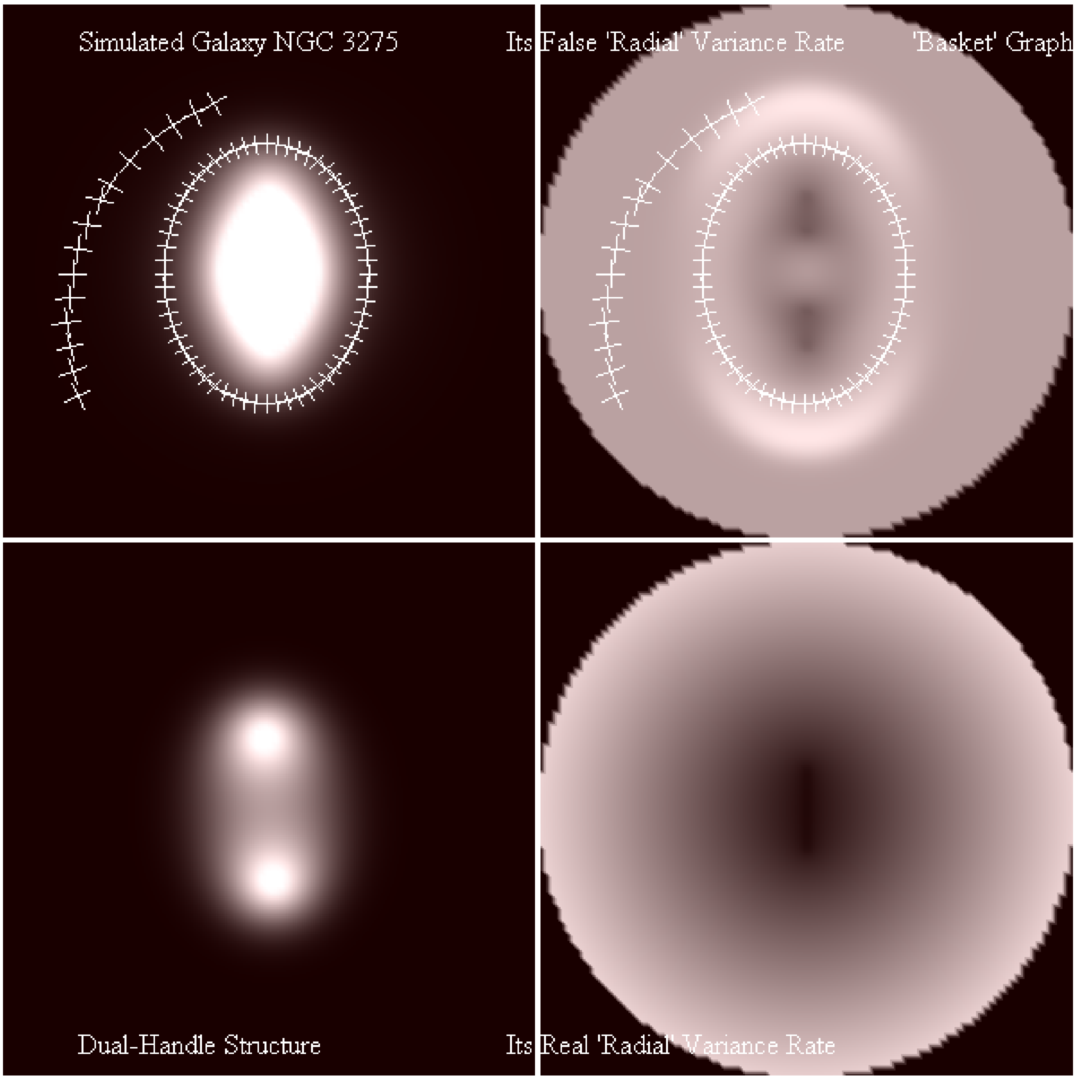

Figure 1: The upper left is the simulated barred spiral galaxy NGC 3275 with an exponential disk and two dual-handle structures. It is an image of density $\rho(x, y)$. The 'radial' variance rate, $u(x, y)$, of its logarithmic density is still under investigation. The false $u(x, y)$ based on rational ground state assumption is displayed in the upper right corner, called the 'basket graph' which suggests how galaxy global proportion curves might be (see the sketch in Figure 2). The 'crosses' on the graphs show the real position of galaxy arms and rings (see the images in [2]). The lower left corner is the simulated shorter dual-handle structure, an image of density $\rho_1(x, y)$. Its variance rate $u(x, y)$ is real and displayed in the lower right corner.

Table 1: The Reference Values of Parameter Fitting

galaxy	actual image size in arcsec	simulated side length	d_0	d_1	b_0	b_1	b_2	b_0	b_1	b_2
3275	110	22	4000	-1.6	134	1.76	-0.2	72	3.25	-0.095
4930	108	31	2200	-2.2	55	3.0	-0.08	56	5.73	-0.023
5921	185	7.4	5000	-10	144	0.31	-8.0	121	0.88	-6.0

of the exponential disk is the greatest which makes the overwhelming contribution to the gradient (the left-hand side of (35)).

3.2 Is the Compound Structure Still Rational?

However, is the compound structure which consists of an exponential disk and several dual-handle structures still rational? Fortunately, the rational sufficient condition presented in Section 2.3 can help us find the answer. Near the galaxy center or far away from the center, the exponential disk takes the overwhelming density value. Because every exponential disk is a rational structure, rational boundary condition is satisfied. Therefore, if a root of an instinct equation for the compound structure satisfies, correspondingly, the rational orthogonal condition (23), (27), or (28) then it must satisfy the equation system (4) and the compound structure is accordingly rational. I find out that the quantity $\lambda\Omega - \omega\Lambda$ does not approach zero in the bright area of galaxy NGC 3275 (the upper left corner of Figure 1). Therefore, the galaxy does not satisfy the rational orthogonal condition (23), and is not a rational structure corresponding to the ground state. Similarly, the quantity $(-B+\sqrt{\Delta})\nabla f\cdot(A\nabla B-B\nabla A)+2A\nabla f\cdot(A\nabla C-C\nabla A)$ does not approach zero. Therefore, the galaxy does not satisfy the rational orthogonal condition (27), and is not a rational structure corresponding to the excited plus state. The only chance is that the galaxy structure might be rational corresponding to an excited minus state, and satisfy the rational orthogonal condition (28):

$$S(x,y) = (-B-\sqrt{\Delta})\nabla f\cdot(A\nabla B - B\nabla A) + 2A\nabla f\cdot(A\nabla C - C\nabla A) = 0. \qquad (36)$$

I find out that $S(x, y)$ does approach zero in the bright area. The c++ program indicates that the absolute values of $S(x, y)$ are generally 'zero', which means that the values are less than 10^{-16} and the computer can not recognize them. However, there are some band-shaped regions especially around the 'basket' rim where $S(x, y)$ presents sharply non-zero values. I further find out that the band-shaped regions can shrink into lines or spots if the simulation of the spiral galaxy (see the upper-left corner of Figure 1) is refined. The hard part of the simulation of barred galaxy images is not the bar fitting. Given some graphic software, it is easy for us to visually simulate bars with dual-handle structure. The criterion is that the galaxy image after bar subtraction, must be an axi-symmetric disk. The hard job is galaxy disk simulation. Because of central bulges, disk simulation is done away from the bulges. Even though we have performed bar subtraction, the disks away from galaxy centers are generally disturbed by arms or rings. Therefore, disk simulation with exponential function is a sophisticated process, and Table 1 presents a preliminary result.

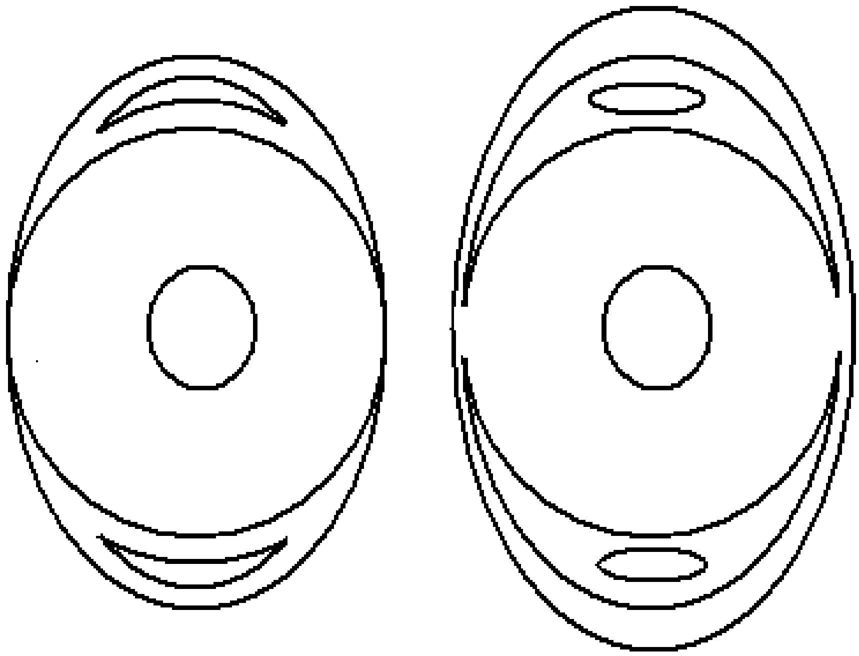

Figure 2: A sketch of the closed orthogonal proportion curves as suggested by the 'basket graphs' of barred galaxies. The sketch on the left corresponds to the barred galaxies whose images do not have apparent dual handles while the sketch on the right corresponds to the barred galaxies whose images do have apparent dual handles.

In fact, the second partial derivatives to the dual-handle structure (32) do not exist at its two foci. In the vicinity of the foci, the partial derivatives exist but their values tend to infinity (keeping the angle α finite). Because the computer can not accurately handle large numbers, all results near the foci present huge errors. Fortunately the instinct equations are homogeneous about their coefficients and the formulas of the coefficients are highly symmetric. Therefore, we can eliminate the common denominators which tend to zero, and have computers always handle normal values. The denominators generally contain the following term (see the formula (32)),

$$(r^2 - b_1^2)^2 + 4b_1^2 x^2. \tag{37}$$

Note also that the logarithmic density of the exponential disk has a singular point, i.e., the center point, at which there is no partial derivatives to the logarithmic density though the structure $\rho(x, y)$ is continuous throughout. Away from the singular area, the computer's calculation is normal and meets required precision. The procedure to eliminating all the singular factors is left for future study. Here comes the great galaxy conjecture.

Galaxy conjecture: Relatively independent galaxies are all rational structure, and barred galaxies are the rational ones in excited minus state. We look forward to astronomers or laymans who testify the conjecture by fitting more barred galaxy images.

3.3 The Evidences of Rational Galaxy Structure

The fact that spiral galaxies can be fitted with rational structure is a marvelous result. But the most marvelous one is the revelation of how gas and dust originate. Since there is no elementary complex function whose corresponding rational structure is linear-shaped like spiral arms, galaxy arms and rings must be the disturbance to the rational structures. 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 living structures need. 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.

The disturbing waves try to achieve the minimal disturbance and, as a result, follow the orthogonal or non-orthogonal proportion curves. Note that spiral arms are usually broken-shaped (not connecting end to end), composed of segments which follow orthogonal or non-orthogonal proportion curves [2]. Galaxy rings follow the closed orthogonal proportion curves, i.e., the level curves of $u(x, y)$. Spiral arms trace non-orthogonal proportion curves and cut through the orthogonal curves proportionally. Figures 1 shows the real position of the arm and ring on the graphs (see [2]). Note that the ring is located inside the 'basket'. This ring can be called the inner ring. There are other types of rings for barred galaxies: outer rings and nuclear rings [11]. The nuclear rings must be located in the central convex balls of the 'basket graphs', and they are approximately circular. Galaxy rings can not coincide with the bright 'basket rims'. This is because the brightness of the rims is usually uneven. As shown in Figure 2, the orthogonal curves around the rims are in fact not the closed ones surrounding the bars. Figures 1 shows that the ring is located immediately off the rims. However, arms can cross through the basket rims because the 'non-radial' variance rates of $f(x, y)$ are usually small. For example, the absolute value of the 'axial' variance rate $v(x, y)$ for the galaxy NGC 3275 is generally so small that it is buried in the numerical errors of ordinary computer calculation.

Dual-handle structures which are the components of barred galaxies, are the simple analytical solution of rational structure. The twelve evidences (coincidences given in [1]) of spiral galaxies as the compound rational structure are mostly based on the results of the component rational structure, i.e., the dual-handle structure. The above explanation of 'basket graph' justifies these evidences. Now let us review the twelve evidences.

(1) The component structure of any galaxy (except the arm) is either axi-symmetric or bilaterally symmetric. Theoretically, the examples of rational structure are very few, which are generally axi-symmetric. The only example found so far which is not axi-symmetric is the dual-handle structure, and it is bilaterally symmetric. (2) Gas and dust are closely related to arms and rings while arms and rings are the disturbance to rational structure. This indicates that gas and dust result from the disturbance to rational structure, which is a very promising explanation to gas and dust origin. (3) In normal spiral galaxies, exponential disks and equiangular arms are interrelated through the concept of rational structure. (4) Among all rational structures determined by the orthogonal curves expressed by elementary complex functions, there is only one structure which is not axi-symmetric: the dual-handle structure. Coincidentally, there are only two types of spiral galaxies: normal and barred. In addition, some barred spiral galaxies do present apparently the dual-handle structure. (5) In some galaxies (e.g. NGC1365) there exist two bars which are not aligned. This can be explained simply: the corresponding dual-handle structures are not aligned. (6) We know that the disk density of spiral galaxies decreases outwards exponentially, which is the numerical result observed over 80 years since the discovery of galaxies in the universe. Spiral galaxy disks are thus called exponential disks. We add the dual-handle structure to the exponential disk for them to be the model of barred spiral galaxies. If the density of dual-handle structure were comparable to or stronger than the exponential disk in the far distances from the galaxy center then our model would fail. That would suggest that the main structure of spiral galaxies were not the exponential disk, a result inconsistent with astronomical observation. The mathematical result is that the density distribution of dual-handle structure decreases outwards cubic-exponentially. That means the bar structure is so weak in the outer areas of spiral galaxies that it is ignored. (7) Astronomical observations show that arms of barred spiral galaxies surround the middle lines of their bars, and they are not equiangular. Generally there are two arms making approximately elliptical shapes with the long axes being parallel to the bar middle lines. The 'basket graphs' confirm the result. (8) Normal spiral galaxies do not have elliptical ring while barred spiral galaxies may have elliptical rings. These are confirmed by the models of both normal and barred spiral galaxies. (9) The image sample of nine barred galaxies can visually be simulated with dual-handle structures. (10) Elliptical galaxies can also be simulated with three-dimensional rational structure (see [8]). (11) The components of spiral galaxies, i.e. the exponential disk and dual-handle structure, all have the orthogonal proportion curves expressed by the complex exponential function. Elliptical galaxies have the orthogonal proportion surfaces expressed by the complex reciprocal function. These results are in line with the principle of simple truth. (12) 'Basket graphs' show that barred spiral galaxies may have nuclear rings and arms which are located in the central area of 'baskets', which is confirmed in galaxy

image analysis.

We can investigate 'basket graphs' deeply and find further evidences of rational galaxy structure. We take a preliminary overall look at 'basket graphs'. Outside the 'baskets', the orthogonal proportion curves return to the ones of exponential disks, and the disks have infinite nets of proportion curves which are generally equiangular. Barred galaxy images do show that the arms are much richer outside of the bars. The area near galaxy bars is cleaner and has less gas and dust. Arms can cross the 'baskets' but their curves can not be parallel to the 'basket' rims for the same reason for galaxy rings. Galaxy images show that some arms originate from around the endpoints of bars and the arms make a sharp turn near the endpoints. The turn near the endpoints is so sharp that the arms are approximately perpendicular to the 'basket' rims.

4 Conclusion and Future Work

This paper completes the mathematical basis for the study of rational structure, and presents the galaxy conjecture that barred galaxies are all rational in the excited minus state. The conjecture can be proved or disaproved by the straightforward testification that the quantity $S(x, y)$ is zero (see the formula (36)). We look forward to laymans or the experts in numerical calculation who answer the call. If the conjecture is proved, we will see the light pointing to the discovery of the secret of life and the universe. Otherwise, mankind would continue to seek truth in darkness in the forseeing future.

"What exactly is a galaxy? Surprising as it may sound, astronomers don't have an answer to this fundamental question." These are the opening words of a piece of January news (2011) in Science magazine written by Jon Cartwright. The news introduces a scientific paper [12] and an online survey launched by the paper's authors. Astronomers have long known that galaxy structure is very simple [13,14] but the two-body theories of gravity established by Newton and Einstein can not explain it. According to these theories, galaxy formation and evolution should be governed by six independent parameters. However, the astronomical observation shows that only one parameter is independent. My simple model of galaxy structure is based on galaxy images and the simple concept of rationality. Real galaxy structure may be even simpler. For example, the parameter in the third column of Table 1 is the simulated side-length of galaxy images, and the simulated values of other parameters are dependent on it. Galaxy structure may be so simple that it is scale invariant. That is, the simulated side-length may actually be arbitrary. This suggestion is left for future testification.

If galaxy structure is meaningful then it is unlikely to find a simple and consistent meaning that is different from the meaning of rationality explained in my papers. It is unlikely that galaxy structure has no meaning.

'Basket graph' will play an essential role in future study on barred galaxies. We humans live in a barred galaxy. What does the 'basket graph' of Milky Way look like? Do human beings live inside or outside the 'basket'? The mass distribution of galaxies, i.e., the stellar distribution, is represented by the long-wavelength galaxy images. However, life phenomena are more involved in the distribution of gas and dust. Therefore, the study of the correlation between the 'basket' structure and the galaxy pattern shown on short-wavelength images may reveal some surprising results.

References

[1] He J. (2010) Electr. Journ. Theo. Phys. 24, 361

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

[3] He J. (2003) Astrophys. Space Sci. 283, 301

[4] He J. (2004) Astrophys. Space Sci. 291, 163

[5] He J. and Yang X. (2006) Astrophys. Space Sci. 302, 7

[6] He J. (2005) PhD thesis, Department of Physics and Astronomy,
The University of Alabama. Publication Number: AAT 3210452; ISBN: 0542620561.

[7] He J. (2006) Astrophys. Space Sci. 305, 197

[8] He J. (2008) Astrophys. Space Sci. 313, 373

[9] He J. (2006) viXra:1101.0076, http://vixra.org/abs/1011.0057

[10] He J. (2006) viXra:1101.0077, http://vixra.org/abs/1102.0035

[11] Buta R. and Combes F. (1996) Fund. Cosmic Physics 17, 95

[12] Forbes D. and Kroupa P. (2011) Pub. Astron. Soc. Aus. 28, 77, arXiv:1101.3309

[13] Disney M. J. et al. (2008) Nature 455, 1082

[14] Kroupa P. *et al.* (2010) Astron. Astrophys. 523, A32

5 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 in the paper. 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 the output data file: orthocond. It displays the values of the quantity $(-B-\sqrt{\Delta})\nabla f\cdot\nabla(B/A)+2A\nabla f\cdot\nabla(C/A)$. But you will find the 'numbers', -1.#IND, in the output file which mean that some operations would generate large positive or negative numbers than could be stored in a 'double' space or some operations do not make mathematical sense, such as taking the square root of a negative number. We look forward to your solution.

```
#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 f0xxxx( long double x, long double y);
long double f0xxxy( long double x, long double y);
long double f0xxyy( long double x, long double y);
long double f0xyyy( long double x, long double y);
long double f0yyyy( 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 gixxxx( int i, long double x, long double y);
long double gixxxy( int i, long double x, long double y);
long double gixxyy( int i, long double x, long double y);
long double gixyyy( int i, long double x, long double y);
long double giyyyy( 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 fixxxx( int i, long double x, long double y);
long double fixxxy( int i, long double x, long double y);
long double fixxyy( int i, long double x, long double y);
long double fixyyy( int i, long double x, long double y);
long double fiyyyy( 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 fxxxx( long double x, long double y);
long double fxxxy( long double x, long double y);
long double fxxyy( long double x, long double y);
long double fxyyy( long double x, long double y);
long double fyyyy( long double x, long double y);
long double mu( long double p, long double q);
long double nu( long double p, long double q);
long double La( long double p, long double q);
long double Om( long double p, long double q);
long double mux( long double p, long double q);
long double muy( long double p, long double q);
long double muxx( long double p, long double q);
long double muxy( long double p, long double q);
long double muyy( long double p, long double q);
long double nux( long double p, long double q);
long double nuy( long double p, long double q);
long double nuxx( long double p, long double q);
long double nuxy( long double p, long double q);
long double nuyy( long double p, long double q);
long double Lax( long double p, long double q);
long double Lay( long double p, long double q);
long double Laxx( long double p, long double q);
long double Laxy( long double p, long double q);
long double Layy( long double p, long double q);
long double Omx( long double p, long double q);
long double Omy( long double p, long double q);
long double Omxx( long double p, long double q);
long double Omxy( long double p, long double q);
```

```
long double Omyy( long double p, long double q);
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 Ax( long double p, long double q);
long double Ay( long double p, long double q);
long double Bx( long double p, long double q);
long double By( long double p, long double q);
long double Cx( long double p, long double q);
long double Cy( long double p, long double q);
long double Orthm( long double p, long double q);
int main(){
ofstream of1("orthocond");
of1<<setw(16)<<setprecision (10)<< resetiosflags( ios::left )
<<setiosflags(ios::fixed — ios::showpoint);
const int M=100,N=100,hM=M/2,hN=N/2;
long double X,Y,DI,DJ;
int m,n;

s0=11;
//2*s0 is the fitted galaxy length on a side of the square image
d0=1500;d1=-1.6; //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;
of1¡¡Orthm(X,Y)¡¡" "; } // n of1¡¡endl;
} // m
return 0;
}

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 f0xxxx( long double p, long double q){
long double x=p, y=q;
```

```
return 3*d1*y*y*(4*x*x-y*y)/pow(x*x+y*y,3.5);
}
long double f0xxxy( long double p, long double q){
long double x=p, y=q;
return -3*d1*x*y*(2*x*x-3*y*y)/pow(x*x+y*y,3.5);
}
long double f0xxyy( long double p, long double q){
long double x=p, y=q;
return d1*(-11*x*x*y*y+2*x*x*x*x+2*y*y*y*y)/pow(x*x+y*y,3.5);
}
long double f0xyyy( long double p, long double q){
long double x=p, y=q;
return 3*d1*x*y*(3*x*x-2*y*y)/pow(x*x+y*y,3.5);
}
long double f0yyyy( long double p, long double q){
long double x=p, y=q;
return -3*d1*x*x*(x*x-4*y*y)/pow(x*x+y*y,3.5);
}
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 gixxxx( int j, long double p, long double q){ long double x=p, y=q;
int i=j;
return -15.*8*pow(x*x*x+x*y*y+x*b1[i]*b1[i],4)/
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],3.5)
+9.*8*pow(x*x*x+x*y*y+x*b1[i]*b1[i],2)*(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],2.5)
-6*pow(3*x*x+y*y+b1[i]*b1[i],2)/
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)
-48*(x*x*x+x*y*y+x*b1[i]*b1[i])*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],1.5)
+6./
```

```
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 gixxxy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -15.*8*pow(x*x*x+x*y*y+x*b1[i]*b1[i],3)*(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],3.5)
+9.*8*pow(x*x*x+x*y*y+x*b1[i]*b1[i],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],2.5)
+9.*4*(x*x*x+x*y*y+x*b1[i]*b1[i])*(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],2.5)
-12*(3*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)
-12*(x*x*x+x*y*y+x*b1[i]*b1[i])*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)
-12*(x*x*y+y*y*y-y*b1[i]*b1[i])*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],1.5);
}
long double gixxyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -15.*8*pow(x*x*x+x*y*y+x*b1[i]*b1[i],2)*pow(x*x*y+y*y*y-y*b1[i]*b1[i],2)/
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],3.5)
+6.*16*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x*y+y*y*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],2.5)
+12*pow(x*x*x+x*y*y+x*b1[i]*b1[i],2)*(x*x+3*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)
-16*x*x*y*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)
-8*(x*x*x+x*y*y+x*b1[i]*b1[i])*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],1.5)
+12*pow(x*x*y+y*y*y-y*b1[i]*b1[i],2)*(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],2.5)
-8*(x*x*y+y*y*y-y*b1[i]*b1[i])*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+3*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./
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 gixyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -15.*8*(x*x*x+x*y*y+x*b1[i]*b1[i])*pow(x*x*y+y*y*y-y*b1[i]*b1[i],3)/
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],3.5)
```

```
+9.*8*pow(x*x*y+y*y*y-y*b1[i]*b1[i],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],2.5)
+9.*4*(x*x*x+x*y*y+x*b1[i]*b1[i])*(x*x+3*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],2.5)
-12*(x*x*y+y*y*y-y*b1[i]*b1[i])*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],1.5)
-12*(x*x+3*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)
-12*(x*x*x+x*y*y+x*b1[i]*b1[i])*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);
}
long double giyyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return -15.*8*pow(x*x*y+y*y*y-y*b1[i]*b1[i],4)/
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],3.5)
+9.*8*pow(x*x*y+y*y*y-y*b1[i]*b1[i],2)*(x*x+3*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*pow(x*x+3*y*y-b1[i]*b1[i],2)/
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)
-48*(x*x*y+y*y*y-y*b1[i]*b1[i])*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)
+6./
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 fixxxx( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return (3./16.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),2.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),4)
-(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),2)*(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)
-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*pow(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)
-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2),2)
```

```
+(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2) )*(
gixxx(i,x,y)-6*b1[i]*b1[i]*gix(i,x,y)/pow(gi(i,x,y),2)
+12*b1[i]*b1[i]*x*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-6*b1[i]*b1[i]*x*gixx(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(gix(i,x,y),3)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*gix(i,x,y)*gixx(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxx(i,x,y)/pow(gi(i,x,y),2) )
+(1./2.)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*b2[i]*(
gixxxx(i,x,y)+24*b1[i]*b1[i]*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-12*b1[i]*b1[i]*gixx(i,x,y)/pow(gi(i,x,y),2)
-48*b1[i]*b1[i]*x*pow(gix(i,x,y),3)/pow(gi(i,x,y),4)
+48*b1[i]*b1[i]*x*gix(i,x,y)*gixx(i,x,y)/pow(gi(i,x,y),3)
-8*b1[i]*b1[i]*x*gixxx(i,x,y)/pow(gi(i,x,y),2)
+24*b1[i]*b1[i]*x*x*pow(gix(i,x,y),4)/pow(gi(i,x,y),5)
-36*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)*gixx(i,x,y)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*gixx(i,x,y)*gixx(i,x,y)/pow(gi(i,x,y),3)
+8*b1[i]*b1[i]*x*x*gix(i,x,y)*gixxx(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxxx(i,x,y)/pow(gi(i,x,y),2) );
}
long double fixxxy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return (3./16.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),2.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),3)*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )
-(3./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),2)*(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2) )
-(3./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),1)*(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2) )*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2) )*(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
```

```
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2) )*(
gixxy(i,x,y)-2*b1[i]*b1[i]*giy(i,x,y)/pow(gi(i,x,y),2)
+8*b1[i]*b1[i]*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-4*b1[i]*b1[i]*x*gixy(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)*giy(i,x,y)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*gix(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*gixx(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxy(i,x,y)/pow(gi(i,x,y),2) )
+(1./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gixxx(i,x,y)-6*b1[i]*b1[i]*gix(i,x,y)/pow(gi(i,x,y),2)
+12*b1[i]*b1[i]*x*pow(gix(i,x,y),2)/pow(gi(i,x,y),3)
-6*b1[i]*b1[i]*x*gixx(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(gix(i,x,y),3)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*gix(i,x,y)*gixx(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxx(i,x,y)/pow(gi(i,x,y),2) )*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )
+(1./2.)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*b2[i]*(
gixxxy(i,x,y)+12*b1[i]*b1[i]*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-6*b1[i]*b1[i]*gixy(i,x,y)/pow(gi(i,x,y),2)
-36*b1[i]*b1[i]*x*pow(gix(i,x,y),2)*giy(i,x,y)/pow(gi(i,x,y),4)
+24*b1[i]*b1[i]*x*gix(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+12*b1[i]*b1[i]*x*gixx(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-6*b1[i]*b1[i]*x*gixxy(i,x,y)/pow(gi(i,x,y),2)
+24*b1[i]*b1[i]*x*x*pow(gix(i,x,y),3)*giy(i,x,y)/pow(gi(i,x,y),5)
-18*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)*gixy(i,x,y)/pow(gi(i,x,y),4)
-18*b1[i]*b1[i]*x*x*gix(i,x,y)*gixx(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*gixy(i,x,y)*gixx(i,x,y)/pow(gi(i,x,y),3)
+6*b1[i]*b1[i]*x*x*gix(i,x,y)*gixxy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*gixxx(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxxy(i,x,y)/pow(gi(i,x,y),2) );
}
long double fixxyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return (3./16.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),2.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),2)*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),2)
-(1./2.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2) )*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )*(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2) )
-(1./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),2)*(
```

giyy(i,x,y)+2*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2))
+(1./2.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*pow(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2),2)
+(1./2.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2))*(
gixyy(i,x,y)+4*b1[i]*b1[i]*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-2*b1[i]*b1[i]*x*giyy(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)*gix(i,x,y)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*giy(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*giyy(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixyy(i,x,y)/pow(gi(i,x,y),2))
-(1./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),2)*(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2))
+(1./2.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gixxy(i,x,y)-2*b1[i]*b1[i]*giy(i,x,y)/pow(gi(i,x,y),2)
+8*b1[i]*b1[i]*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-4*b1[i]*b1[i]*x*gixy(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)*giy(i,x,y)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*gix(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*gixx(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxy(i,x,y)/pow(gi(i,x,y),2))*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2))
+(1./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gixx(i,x,y)+2*b1[i]*b1[i]/gi(i,x,y)-4*b1[i]*b1[i]*x*gix(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixx(i,x,y)/pow(gi(i,x,y),2))*(
giyy(i,x,y)+2*b1[i]*b1[i]*x*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2))
+(1./2.)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*b2[i]*(
gixxyy(i,x,y)+4*b1[i]*b1[i]*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-2*b1[i]*b1[i]*giyy(i,x,y)/pow(gi(i,x,y),2)
-24*b1[i]*b1[i]*x*pow(giy(i,x,y),2)*gix(i,x,y)/pow(gi(i,x,y),4)
+16*b1[i]*b1[i]*x*giy(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+8*b1[i]*b1[i]*x*giyy(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-4*b1[i]*b1[i]*x*gixyy(i,x,y)/pow(gi(i,x,y),2)
+24*b1[i]*b1[i]*x*x*pow(gix(i,x,y),2)*pow(giy(i,x,y),2)/pow(gi(i,x,y),5)
-24*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),4)
-6*b1[i]*b1[i]*x*x*gix(i,x,y)*gix(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*gixy(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)

```
+4*b1[i]*b1[i]*x*x*gix(i,x,y)*gixyy(i,x,y)/pow(gi(i,x,y),3)
-6*b1[i]*b1[i]*x*x*gixx(i,x,y)*pow(giy(i,x,y),2)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*gixxy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*gixx(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixxyy(i,x,y)/pow(gi(i,x,y),2) );
}
long double fixyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return (3./16.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),2.5))*b2[i]*(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2) )*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),3)
-(3./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),2)*(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2) )
-(3./8.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2),1)*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )*(
giyy(i,x,y)+2*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )*(
gixyy(i,x,y)+4*b1[i]*b1[i]*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-2*b1[i]*b1[i]*x*giyy(i,x,y)/pow(gi(i,x,y),2)
-6*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)*gix(i,x,y)/pow(gi(i,x,y),4)
+4*b1[i]*b1[i]*x*x*giy(i,x,y)*gixy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*giyy(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixyy(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
gixy(i,x,y)-2*b1[i]*b1[i]*x*giy(i,x,y)/pow(gi(i,x,y),2)
+2*b1[i]*b1[i]*x*x*gix(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixy(i,x,y)/pow(gi(i,x,y),2) )*(
giyy(i,x,y)+2*b1[i]*b1[i]*x*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2) )
+(1./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
giyyy(i,x,y)-6*b1[i]*b1[i]*x*x*pow(giy(i,x,y),3)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*giy(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyyy(i,x,y)/pow(gi(i,x,y),2) )*(
gix(i,x,y)+2*b1[i]*b1[i]*x/gi(i,x,y)-b1[i]*b1[i]*x*x*gix(i,x,y)/pow(gi(i,x,y),2) )
+(1./2.)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*b2[i]*(
gixyyy(i,x,y)-12*b1[i]*b1[i]*x*pow(giy(i,x,y),3)/pow(gi(i,x,y),4)
+12*b1[i]*b1[i]*x*giy(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
-2*b1[i]*b1[i]*x*giyyy(i,x,y)/pow(gi(i,x,y),2)
```

```
+24*b1[i]*b1[i]*x*x*pow(giy(i,x,y),3)*gix(i,x,y)/pow(gi(i,x,y),5)
-18*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)*gixy(i,x,y)/pow(gi(i,x,y),4)
-18*b1[i]*b1[i]*x*x*gix(i,x,y)*giyy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*giy(i,x,y)*gixyy(i,x,y)/pow(gi(i,x,y),3)
+6*b1[i]*b1[i]*x*x*gixy(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
+2*b1[i]*b1[i]*x*x*giyyy(i,x,y)*gix(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*gixyyy(i,x,y)/pow(gi(i,x,y),2) );
}
long double fiyyyy( int j, long double p, long double q){
long double x=p, y=q;
int i=j;
return (3./16.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),2.5))*b2[i]*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),4)
-(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),1.5))*b2[i]*pow(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2),2)*(
giyy(i,x,y)+2*b1[i]*b1[i]*x*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2) )
+(3./4.)*(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*pow(
giyy(i,x,y)+2*b1[i]*b1[i]*x*x*giy(i,x,y)*giy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyy(i,x,y)/pow(gi(i,x,y),2),2)
+(1./pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5))*b2[i]*(
giy(i,x,y)-b1[i]*b1[i]*x*x*giy(i,x,y)/pow(gi(i,x,y),2) )*(
giyyy(i,x,y)-6*b1[i]*b1[i]*x*x*pow(giy(i,x,y),3)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*giy(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyyy(i,x,y)/pow(gi(i,x,y),2) )
+(1./2.)*pow(gi(i,x,y)+b1[i]*b1[i]*x*x/gi(i,x,y),0.5)*b2[i]*(
giyyyy(i,x,y)+24*b1[i]*b1[i]*x*x*pow(giy(i,x,y),4)/pow(gi(i,x,y),5)
-36*b1[i]*b1[i]*x*x*pow(giy(i,x,y),2)*giyy(i,x,y)/pow(gi(i,x,y),4)
+6*b1[i]*b1[i]*x*x*giyy(i,x,y)*giyy(i,x,y)/pow(gi(i,x,y),3)
+8*b1[i]*b1[i]*x*x*giy(i,x,y)*giyyy(i,x,y)/pow(gi(i,x,y),3)
-b1[i]*b1[i]*x*x*giyyyy(i,x,y)/pow(gi(i,x,y),2) );
}
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 fxxxx( long double p, long double q){
long double x=p, y=q;
return (1/rh(x,y))*( r0(x,y)*(
f0xxxx(x,y)+4*f0xxx(x,y)*f0x(x,y)+3*f0xx(x,y)*f0xx(x,y)+6*f0xx(x,y)*pow(f0x(x,y),2)
+pow(f0x(x,y),4) )
+ri(1,x,y)*(
fixxxx(1,x,y)+4*fixxx(1,x,y)*fix(1,x,y)+3*fixx(1,x,y)*fixx(1,x,y)
+6*fixx(1,x,y)*pow(fix(1,x,y),2)+pow(fix(1,x,y),4) )
+ri(2,x,y)*(
fixxxx(2,x,y)+4*fixxx(2,x,y)*fix(2,x,y)+3*fixx(2,x,y)*fixx(2,x,y)
+6*fixx(2,x,y)*pow(fix(2,x,y),2)+pow(fix(2,x,y),4) ) )
-(4/pow(rh(x,y),2))*(
r0(x,y)*( f0xxx(x,y)+3*f0x(x,y)*f0xx(x,y)+pow(f0x(x,y),3) )
+ri(1,x,y)*( fixxx(1,x,y)+3*fix(1,x,y)*fixx(1,x,y)+pow(fix(1,x,y),3) )
+ri(2,x,y)*( fixxx(2,x,y)+3*fix(2,x,y)*fixx(2,x,y)+pow(fix(2,x,y),3) ) )*(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y) )
-(3/pow(rh(x,y),2))*pow(
r0(x,y)*( f0xx(x,y)+pow(f0x(x,y),2) )
+ri(1,x,y)*( fixx(1,x,y)+pow(fix(1,x,y),2) )
+ri(2,x,y)*( fixx(2,x,y)+pow(fix(2,x,y),2) ),2)
+(12/pow(rh(x,y),3))*(
r0(x,y)*( f0xx(x,y)+pow(f0x(x,y),2) )
+ri(1,x,y)*( fixx(1,x,y)+pow(fix(1,x,y),2) )
+ri(2,x,y)*( fixx(2,x,y)+pow(fix(2,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)
-(6/pow(rh(x,y),4))*pow(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y),4);
}
long double fxxxy( long double p, long double q){
long double x=p, y=q;
return (1/rh(x,y))*( r0(x,y)*(
f0xxxy(x,y)+f0xxx(x,y)*f0y(x,y)+3*f0x(x,y)*f0xxy(x,y)+3*(f0xx(x,y)+pow(f0x(x,y),2))*f0xy(x,y)
+3*f0x(x,y)*f0xx(x,y)*f0y(x,y)+pow(f0x(x,y),3)*f0y(x,y) )
+ri(1,x,y)*(
fixxxy(1,x,y)+fixxx(1,x,y)*fiy(1,x,y)+3*fix(1,x,y)*fixxy(1,x,y)+3*(fixx(1,x,y)+pow(fix(1,x,y),2)
)*fixy(1,x,y)+3*fix(1,x,y)*fixx(1,x,y)*fiy(1,x,y)+pow(fix(1,x,y),3)*fiy(1,x,y) )
+ri(2,x,y)*(
```

```
fixxxy(2,x,y)+fixxx(2,x,y)*fiy(2,x,y)+3*fix(2,x,y)*fixxy(2,x,y)+3*(fixx(2,x,y)+pow(fix(2,x,y),2)
)*fixy(2,x,y)+3*fix(2,x,y)*fixx(2,x,y)*fiy(2,x,y)+pow(fix(2,x,y),3)*fiy(2,x,y) ) )
-(1/pow(rh(x,y),2))*(
r0(x,y)*( f0xxx(x,y)+3*f0x(x,y)*f0xx(x,y)+pow(f0x(x,y),3) )
+ri(1,x,y)*( fixxx(1,x,y)+3*fix(1,x,y)*fixx(1,x,y)+pow(fix(1,x,y),3) )
+ri(2,x,y)*( fixxx(2,x,y)+3*fix(2,x,y)*fixx(2,x,y)+pow(fix(2,x,y),3) ) )*(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y) )
-(3/pow(rh(x,y),2))*(
r0(x,y)*( f0xxy(x,y)+f0y(x,y)*f0xx(x,y)+2*f0x(x,y)*f0xy(x,y)+pow(f0x(x,y),2)*f0y(x,y) )
+ri(1,x,y)*( fixxy(1,x,y)+fiy(1,x,y)*fixx(1,x,y)+2*fix(1,x,y)*fixy(1,x,y)
+pow(fix(1,x,y),2)*fiy(1,x,y) )
+ri(2,x,y)*( fixxy(2,x,y)+fiy(2,x,y)*fixx(2,x,y)+2*fix(2,x,y)*fixy(2,x,y)
+pow(fix(2,x,y),2)*fiy(2,x,y) ) )*(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y) )
-(3/pow(rh(x,y),2))*(
r0(x,y)*( f0xx(x,y)+pow(f0x(x,y),2) )
+ri(1,x,y)*( fixx(1,x,y)+pow(fix(1,x,y),2) )
+ri(2,x,y)*( fixx(2,x,y)+pow(fix(2,x,y),2) ) )*(
r0(x,y)*( f0xy(x,y)+f0x(x,y)*f0y(x,y) )
+ri(1,x,y)*( fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y) )
+ri(2,x,y)*( fixy(2,x,y)+fix(2,x,y)*fiy(2,x,y) ) )
+(6/pow(rh(x,y),3))*(
r0(x,y)*( f0xx(x,y)+pow(f0x(x,y),2) )
+ri(1,x,y)*( fixx(1,x,y)+pow(fix(1,x,y),2) )
+ri(2,x,y)*( fixx(2,x,y)+pow(fix(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) )*(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y) )
+(6/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)*( f0xy(x,y)+f0x(x,y)*f0y(x,y) )
+ri(1,x,y)*( fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y) )
+ri(2,x,y)*( fixy(2,x,y)+fix(2,x,y)*fiy(2,x,y) ) )
-(6/pow(rh(x,y),4))*pow(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y),3)*(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y) );
}
long double fxxyy( long double p, long double q){
long double x=p, y=q;
return (1/rh(x,y))*( r0(x,y)*( 2*f0xxy(x,y)*f0y(x,y)+f0xx(x,y)*f0yy(x,y)
+2*f0x(x,y)*f0xyy(x,y)+pow(f0x(x,y),2)*f0yy(x,y) )
+ri(1,x,y)*( 2*fixxy(1,x,y)*fiy(1,x,y)+fixx(1,x,y)*fiyy(1,x,y)
+2*fix(1,x,y)*fixyy(1,x,y)+pow(fix(1,x,y),2)*fiyy(1,x,y) )
+ri(2,x,y)*( 2*fixxy(2,x,y)*fiy(2,x,y)+fixx(2,x,y)*fiyy(2,x,y)
+2*fix(2,x,y)*fixyy(2,x,y)+pow(fix(2,x,y),2)*fiyy(2,x,y) )
+r0(x,y)*pow(f0y(x,y),2)*f0xx(x,y)+ri(1,x,y)*pow(fiy(1,x,y),2)*fixx(1,x,y)
```

+ri(2,x,y)*pow(fiy(2,x,y),2)*fixx(2,x,y)
+r0(x,y)*f0xxyy(x,y)+ri(1,x,y)*fixxyy(1,x,y)+ri(2,x,y)*fixxyy(2,x,y)
+2*r0(x,y)*pow(f0xy(x,y),2)+2*ri(1,x,y)*pow(fixy(1,x,y),2)+2*ri(2,x,y)*pow(fixy(2,x,y),2)
+r0(x,y)*pow(f0x(x,y),2)*pow(f0y(x,y),2)+ri(1,x,y)*pow(fix(1,x,y),2)*pow(fiy(1,x,y),2)
+ri(2,x,y)*pow(fix(2,x,y),2)*pow(fiy(2,x,y),2)
+4*r0(x,y)*f0x(x,y)*f0y(x,y)*f0xy(x,y)+4*ri(1,x,y)*fix(1,x,y)*fiy(1,x,y)*fixy(1,x,y)
+4*ri(2,x,y)*fix(2,x,y)*fiy(2,x,y)*fixy(2,x,y))
-(2/pow(rh(x,y),2))*(
r0(x,y)*(f0xxy(x,y)+f0xx(x,y)*f0y(x,y)+2*f0x(x,y)*f0xy(x,y)+pow(f0x(x,y),2)*f0y(x,y))
+ri(1,x,y)*(fixxy(1,x,y)+fixx(1,x,y)*fiy(1,x,y)+2*fix(1,x,y)*fixy(1,x,y)
+pow(fix(1,x,y),2)*fiy(1,x,y))
+ri(2,x,y)*(fixxy(2,x,y)+fixx(2,x,y)*fiy(2,x,y)+2*fix(2,x,y)*fixy(2,x,y)
+pow(fix(2,x,y),2)*fiy(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/pow(rh(x,y),2))*(
r0(x,y)*(f0xx(x,y)+pow(f0x(x,y),2))
+ri(1,x,y)*(fixx(1,x,y)+pow(fix(1,x,y),2))
+ri(2,x,y)*(fixx(2,x,y)+pow(fix(2,x,y),2)))*(
r0(x,y)*(f0yy(x,y)+pow(f0y(x,y),2))
+ri(1,x,y)*(fiyy(1,x,y)+pow(fiy(1,x,y),2))
+ri(2,x,y)*(fiyy(2,x,y)+pow(fiy(2,x,y),2)))
-(2/pow(rh(x,y),2))*pow(
r0(x,y)*(f0xy(x,y)+f0x(x,y)*f0y(x,y))
+ri(1,x,y)*(fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y))
+ri(2,x,y)*(fixy(2,x,y)+fix(2,x,y)*fiy(2,x,y)),2)
-(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)*(f0xyy(x,y)+2*f0xy(x,y)*f0y(x,y)+f0x(x,y)*f0yy(x,y)+f0x(x,y)*pow(f0y(x,y),2))
+ri(1,x,y)*(fixyy(1,x,y)+2*fixy(1,x,y)*fiy(1,x,y)+fix(1,x,y)*fiyy(1,x,y)
+fix(1,x,y)*pow(fiy(1,x,y),2))
+ri(2,x,y)*(fixyy(2,x,y)+2*fixy(2,x,y)*fiy(2,x,y)+fix(2,x,y)*fiyy(2,x,y)
+fix(2,x,y)*pow(fiy(2,x,y),2)))
+(2/pow(rh(x,y),3))*(
r0(x,y)*(f0xx(x,y)+pow(f0x(x,y),2))
+ri(1,x,y)*(fixx(1,x,y)+pow(fix(1,x,y),2))
+ri(2,x,y)*(fixx(2,x,y)+pow(fix(2,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)
+(8/pow(rh(x,y),3))*(
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)*(f0xy(x,y)+f0x(x,y)*f0y(x,y))
+ri(1,x,y)*(fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y))
+ri(2,x,y)*(fixy(2,x,y)+fix(2,x,y)*fiy(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))
+(2/pow(rh(x,y),3))*(
r0(x,y)*(f0yy(x,y)+pow(f0y(x,y),2))
+ri(1,x,y)*(fiyy(1,x,y)+pow(fiy(1,x,y),2))

```
+ri(2,x,y)*( fiyy(2,x,y)+pow(fiy(2,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)
-(6/pow(rh(x,y),4))*pow(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,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);
}
long double fxyyy( long double p, long double q){
long double x=p, y=q;
return (1/rh(x,y))*( r0(x,y)*(
f0xyyy(x,y)+f0yyy(x,y)*f0x(x,y)+3*f0y(x,y)*f0xyy(x,y)+3*(f0yy(x,y)+pow(f0y(x,y),2))*f0xy(x,y)
+3*f0x(x,y)*f0yy(x,y)*f0y(x,y)+pow(f0y(x,y),3)*f0x(x,y) )
+ri(1,x,y)*(
fixyyy(1,x,y)+fiyyy(1,x,y)*fix(1,x,y)+3*fiy(1,x,y)*fixyy(1,x,y)+3*(fiyy(1,x,y)+pow(fiy(1,x,y),2)
)*fixy(1,x,y)+3*fix(1,x,y)*fiyy(1,x,y)*fiy(1,x,y)+pow(fiy(1,x,y),3)*fix(1,x,y) )
+ri(2,x,y)*(
fixyyy(2,x,y)+fiyyy(2,x,y)*fix(2,x,y)+3*fiy(2,x,y)*fixyy(2,x,y)+3*(fiyy(2,x,y)+pow(fiy(2,x,y),2)
)*fixy(2,x,y)+3*fix(2,x,y)*fiyy(2,x,y)*fiy(2,x,y)+pow(fiy(2,x,y),3)*fix(2,x,y) ) )
-(1/pow(rh(x,y),2))*(
r0(x,y)*( f0yyy(x,y)+3*f0y(x,y)*f0yy(x,y)+pow(f0y(x,y),3) )
+ri(1,x,y)*( fiyyy(1,x,y)+3*fiy(1,x,y)*fiyy(1,x,y)+pow(fiy(1,x,y),3) )
+ri(2,x,y)*( fiyyy(2,x,y)+3*fiy(2,x,y)*fiyy(2,x,y)+pow(fiy(2,x,y),3) ) )*(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y) )
-(3/pow(rh(x,y),2))*(
r0(x,y)*( f0xyy(x,y)+f0x(x,y)*f0yy(x,y)+2*f0y(x,y)*f0xy(x,y)+pow(f0y(x,y),2)*f0x(x,y) )
+ri(1,x,y)*( fixyy(1,x,y)+fix(1,x,y)*fiyy(1,x,y)+2*fiy(1,x,y)*fixy(1,x,y)
+pow(fiy(1,x,y),2)*fix(1,x,y) )
+ri(2,x,y)*( fixyy(2,x,y)+fix(2,x,y)*fiyy(2,x,y)+2*fiy(2,x,y)*fixy(2,x,y)
+pow(fiy(2,x,y),2)*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) )
-(3/pow(rh(x,y),2))*(
r0(x,y)*( f0yy(x,y)+pow(f0y(x,y),2) )
+ri(1,x,y)*( fiyy(1,x,y)+pow(fiy(1,x,y),2) )
+ri(2,x,y)*( fiyy(2,x,y)+pow(fiy(2,x,y),2) ) )*(
r0(x,y)*( f0xy(x,y)+f0x(x,y)*f0y(x,y) )
+ri(1,x,y)*( fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y) )
+ri(2,x,y)*( fixy(2,x,y)+fix(2,x,y)*fiy(2,x,y) ) )
+(6/pow(rh(x,y),3))*(
r0(x,y)*( f0yy(x,y)+pow(f0y(x,y),2) )
+ri(1,x,y)*( fiyy(1,x,y)+pow(fiy(1,x,y),2) )
+ri(2,x,y)*( fiyy(2,x,y)+pow(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) )*(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y) )
+(6/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)*( f0xy(x,y)+f0x(x,y)*f0y(x,y) )
```

```
+ri(1,x,y)*( fixy(1,x,y)+fix(1,x,y)*fiy(1,x,y) )
+ri(2,x,y)*( fixy(2,x,y)+fix(2,x,y)*fiy(2,x,y) ) )
-(6/pow(rh(x,y),4))*pow(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y),3)*(
r0(x,y)*f0x(x,y)+ri(1,x,y)*fix(1,x,y)+ri(2,x,y)*fix(2,x,y) );
}
long double fyyyy( long double p, long double q){
long double x=p, y=q;
return (1/rh(x,y))*( r0(x,y)*(
f0yyyy(x,y)+4*f0yyy(x,y)*f0y(x,y)+3*f0yy(x,y)*f0yy(x,y)+6*f0yy(x,y)*pow(f0y(x,y),2)
+pow(f0y(x,y),4) )
+ri(1,x,y)*(
fiyyyy(1,x,y)+4*fiyyy(1,x,y)*fiy(1,x,y)+3*fiyy(1,x,y)*fiyy(1,x,y)
+6*fiyy(1,x,y)*pow(fiy(1,x,y),2)+pow(fiy(1,x,y),4) )
+ri(2,x,y)*(
fiyyyy(2,x,y)+4*fiyyy(2,x,y)*fiy(2,x,y)+3*fiyy(2,x,y)*fiyy(2,x,y)
+6*fiyy(2,x,y)*pow(fiy(2,x,y),2)+pow(fiy(2,x,y),4) ) )
-(4/pow(rh(x,y),2))*(
r0(x,y)*( f0yyy(x,y)+3*f0y(x,y)*f0yy(x,y)+pow(f0y(x,y),3) )
+ri(1,x,y)*( fiyyy(1,x,y)+3*fiy(1,x,y)*fiyy(1,x,y)+pow(fiy(1,x,y),3) )
+ri(2,x,y)*( fiyyy(2,x,y)+3*fiy(2,x,y)*fiyy(2,x,y)+pow(fiy(2,x,y),3) ) )*(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y) )
-(3/pow(rh(x,y),2))*pow(
r0(x,y)*( f0yy(x,y)+pow(f0y(x,y),2) )
+ri(1,x,y)*( fiyy(1,x,y)+pow(fiy(1,x,y),2) )
+ri(2,x,y)*( fiyy(2,x,y)+pow(fiy(2,x,y),2) ),2)
+(12/pow(rh(x,y),3))*(
r0(x,y)*( f0yy(x,y)+pow(f0y(x,y),2) )
+ri(1,x,y)*( fiyy(1,x,y)+pow(fiy(1,x,y),2) )
+ri(2,x,y)*( fiyy(2,x,y)+pow(fiy(2,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)
-(6/pow(rh(x,y),4))*pow(
r0(x,y)*f0y(x,y)+ri(1,x,y)*fiy(1,x,y)+ri(2,x,y)*fiy(2,x,y),4);
}
long double mu( long double p, long double q){
long double x=p, y=q;
return fyy(x,y)-fxx(x,y);
}
long double nu( long double p, long double q){
long double x=p, y=q;
return 2*fxy(x,y);
}
long double La( long double p, long double q){
long double x=p, y=q;
return fy(x,y)*fy(x,y)-fx(x,y)*fx(x,y);
```

```
}
long double Om( long double p, long double q){
long double x=p, y=q;
return 2*fx(x,y)*fy(x,y);
}
long double mux( long double p, long double q){
long double x=p, y=q;
return fxyy(x,y)-fxxx(x,y);
}
long double muy( long double p, long double q){
long double x=p, y=q;
return fyyy(x,y)-fxxy(x,y);
}
long double muxx( long double p, long double q){
long double x=p, y=q;
return fxxyy(x,y)-fxxxx(x,y);
}
long double muxy( long double p, long double q)
long double x=p, y=q;
return fxyyy(x,y)-fxxxy(x,y);
}
long double muyy( long double p, long double q){
long double x=p, y=q;
return fyyyy(x,y)-fxxyy(x,y);
}
long double nux( long double p, long double q){
long double x=p, y=q;
return 2*fxxy(x,y);
}
long double nuy( long double p, long double q){
long double x=p, y=q;
return 2*fxyy(x,y);
}
long double nuxx( long double p, long double q){
long double x=p, y=q;
return 2*fxxxy(x,y);
}
long double nuxy( long double p, long double q){
long double x=p, y=q;
return 2*fxxyy(x,y);
}
long double nuyy( long double p, long double q){
long double x=p, y=q;
return 2*fxyyy(x,y);
}
```

```
long double Lax( long double p, long double q){
long double x=p, y=q;
return 2*(fy(x,y)*fxy(x,y)-fx(x,y)*fxx(x,y));
}
long double Lay( long double p, long double q){
long double x=p, y=q;
return 2*(fy(x,y)*fyy(x,y)-fx(x,y)*fxy(x,y));
}
long double Laxx( long double p, long double q){
long double x=p, y=q;
return 2*(fxy(x,y)*fxy(x,y)+fy(x,y)*fxxy(x,y)-fxx(x,y)*fxx(x,y)-fx(x,y)*fxxx(x,y));
}
long double Laxy( long double p, long double q){
long double x=p, y=q;
return 2*(fyy(x,y)*fxy(x,y)+fy(x,y)*fxyy(x,y)-fxy(x,y)*fxx(x,y)-fx(x,y)*fxxy(x,y));
}
long double Layy( long double p, long double q)
long double x=p, y=q;
return 2*(fyy(x,y)*fyy(x,y)+fy(x,y)*fyyy(x,y)-fxy(x,y)*fxy(x,y)-fx(x,y)*fxyy(x,y));
}
long double Omx( long double p, long double q)
long double x=p, y=q;
return 2*(fxx(x,y)*fy(x,y)+fx(x,y)*fxy(x,y));
}
long double Omy( long double p, long double q){
long double x=p, y=q;
return 2*(fxy(x,y)*fy(x,y)+fx(x,y)*fyy(x,y));
}
long double Omxx( long double p, long double q)
long double x=p, y=q;
return 2*(fxxx(x,y)*fy(x,y)+fxx(x,y)*fxy(x,y)+fxx(x,y)*fxy(x,y)+fx(x,y)*fxxy(x,y));
}
long double Omxy( long double p, long double q)
long double x=p, y=q;
return 2*(fxxy(x,y)*fy(x,y)+fxx(x,y)*fyy(x,y)+fxy(x,y)*fxy(x,y)+fx(x,y)*fxyy(x,y));
}
long double Omyy( long double p, long double q){
long double x=p, y=q;
return 2*(fxyy(x,y)*fy(x,y)+fxy(x,y)*fyy(x,y)+fxy(x,y)*fyy(x,y)+fx(x,y)*fyyy(x,y));
}
long double cA( long double p, long double q){
long double x=p, y=q;
return (fxx(x,y)+fyy(x,y))*mu(x,y)*La(x,y)
+fx(x,y)*(mux(x,y)*La(x,y)-mu(x,y)*Lax(x,y))
+fy(x,y)*(muy(x,y)*La(x,y)-mu(x,y)*Lay(x,y));
```

```
}
long double cB( long double p, long double q)
long double x=p, y=q;
return (fxx(x,y)+fyy(x,y))*(mu(x,y)*Om(x,y)+nu(x,y)*La(x,y))
+fx(x,y)*(mux(x,y)*Om(x,y)+nux(x,y)*La(x,y)-mu(x,y)*Omx(x,y)-nu(x,y)*Lax(x,y))
+fy(x,y)*(muy(x,y)*Om(x,y)+nuy(x,y)*La(x,y)-mu(x,y)*Omy(x,y)-nu(x,y)*Lay(x,y));
}
long double cC( long double p, long double q){
long double x=p, y=q;
return (fxx(x,y)+fyy(x,y))*nu(x,y)*Om(x,y)
+fx(x,y)*(nux(x,y)*Om(x,y)-nu(x,y)*Omx(x,y))
+fy(x,y)*(nuy(x,y)*Om(x,y)-nu(x,y)*Omy(x,y));
}
long double Ax( long double p, long double q){
long double x=p, y=q;
return (fxxx(x,y)+fxyy(x,y))*mu(x,y)*La(x,y)
+(fxx(x,y)+fyy(x,y))*(mux(x,y)*La(x,y)+mu(x,y)*Lax(x,y))
+fxx(x,y)*(mux(x,y)*La(x,y)-mu(x,y)*Lax(x,y))
+fx(x,y)*(muxx(x,y)*La(x,y)-mu(x,y)*Laxx(x,y))
+fxy(x,y)*(muy(x,y)*La(x,y)-mu(x,y)*Lay(x,y))
+fy(x,y)*(muxy(x,y)*La(x,y)+muy(x,y)*Lax(x,y)-mux(x,y)*Lay(x,y)-mu(x,y)*Laxy(x,y));
}
long double Ay( long double p, long double q){
long double x=p, y=q;
return (fxxy(x,y)+fyyy(x,y))*mu(x,y)*La(x,y)
+(fxx(x,y)+fyy(x,y))*(muy(x,y)*La(x,y)+mu(x,y)*Lay(x,y))
+fxy(x,y)*(mux(x,y)*La(x,y)-mu(x,y)*Lax(x,y))
+fx(x,y)*(muxy(x,y)*La(x,y)+mux(x,y)*Lay(x,y)-muy(x,y)*Lax(x,y)-mu(x,y)*Laxy(x,y))
+fyy(x,y)*(muy(x,y)*La(x,y)-mu(x,y)*Lay(x,y))
+fy(x,y)*(muyy(x,y)*La(x,y)-mu(x,y)*Layy(x,y));
}
long double Bx( long double p, long double q){
long double x=p, y=q;
return (fxxx(x,y)+fxyy(x,y))*(mu(x,y)*Om(x,y)+nu(x,y)*La(x,y))
+(fxx(x,y)+fyy(x,y))*(mux(x,y)*Om(x,y)+mu(x,y)*Omx(x,y)+nux(x,y)*La(x,y)+nu(x,y)*Lax(x,y))
+fxx(x,y)*(mux(x,y)*Om(x,y)+nux(x,y)*La(x,y)-mu(x,y)*Omx(x,y)-nu(x,y)*Lax(x,y))
+fx(x,y)*(muxx(x,y)*Om(x,y)+nuxx(x,y)*La(x,y)-mu(x,y)*Omxx(x,y)-nu(x,y)*Laxx(x,y))
+fxy(x,y)*(muy(x,y)*Om(x,y)+nuy(x,y)*La(x,y)-mu(x,y)*Omy(x,y)-nu(x,y)*Lay(x,y))
+fy(x,y)*(muxy(x,y)*Om(x,y)+muy(x,y)*Omx(x,y)+nuxy(x,y)*La(x,y)+nuy(x,y)*Lax(x,y)
-mux(x,y)*Omy(x,y)-mu(x,y)*Omxy(x,y)-nux(x,y)*Lay(x,y)-nu(x,y)*Laxy(x,y));
}
long double By( long double p, long double q){
long double x=p, y=q;
return (fxxy(x,y)+fyyy(x,y))*(mu(x,y)*Om(x,y)+nu(x,y)*La(x,y))
+(fxx(x,y)+fyy(x,y))*(muy(x,y)*Om(x,y)+mu(x,y)*Omy(x,y)+nuy(x,y)*La(x,y)+nu(x,y)*Lay(x,y))
```

```
+fxy(x,y)*(mux(x,y)*Om(x,y)+nux(x,y)*La(x,y)-mu(x,y)*Omx(x,y)-nu(x,y)*Lax(x,y))
+fx(x,y)*(muxy(x,y)*Om(x,y)+mux(x,y)*Omy(x,y)+nuxy(x,y)*La(x,y)+nux(x,y)*Lay(x,y)
-muy(x,y)*Omx(x,y)-mu(x,y)*Omxy(x,y)-nuy(x,y)*Lax(x,y)-nu(x,y)*Laxy(x,y))
+fyy(x,y)*(muy(x,y)*Om(x,y)+nuy(x,y)*La(x,y)-mu(x,y)*Omy(x,y)-nu(x,y)*Lay(x,y))
+fy(x,y)*(muyy(x,y)*Om(x,y)+nuyy(x,y)*La(x,y)-mu(x,y)*Omyy(x,y)-nu(x,y)*Layy(x,y));


long double Cx( long double p, long double q){
long double x=p, y=q;
return (fxxx(x,y)+fxyy(x,y))*nu(x,y)*Om(x,y)
+(fxx(x,y)+fyy(x,y))*(nux(x,y)*Om(x,y)+nu(x,y)*Omx(x,y))
+fxx(x,y)*(nux(x,y)*Om(x,y)-nu(x,y)*Omx(x,y))
+fx(x,y)*(nuxx(x,y)*Om(x,y)-nu(x,y)*Omxx(x,y))
+fxy(x,y)*(nuy(x,y)*Om(x,y)-nu(x,y)*Omy(x,y))
+fy(x,y)*(nuxy(x,y)*Om(x,y)+nuy(x,y)*Omx(x,y)-nux(x,y)*Omy(x,y)-nu(x,y)*Omxy(x,y));
}
long double Cy( long double p, long double q){
long double x=p, y=q;
return (fxxy(x,y)+fyyy(x,y))*nu(x,y)*Om(x,y)
+(fxx(x,y)+fyy(x,y))*(nuy(x,y)*Om(x,y)+nu(x,y)*Omy(x,y))
+fxy(x,y)*(nux(x,y)*Om(x,y)-nu(x,y)*Omx(x,y))
+fx(x,y)*(nuxy(x,y)*Om(x,y)+nux(x,y)*Omy(x,y)-nuy(x,y)*Omx(x,y)-nu(x,y)*Omxy(x,y))
+fyy(x,y)*(nuy(x,y)*Om(x,y)-nu(x,y)*Omy(x,y))
+fy(x,y)*(nuyy(x,y)*Om(x,y)-nu(x,y)*Omyy(x,y));
}
long double Orthm( long double p, long double q){
long double x=p, y=q, delt;
delt=cB(x,y)*cB(x,y)-4*cA(x,y)*cC(x,y);
if(delt¿=0){
return ( -cB(x,y)-sqrt(cB(x,y)*cB(x,y)-4*cA(x,y)*cC(x,y)) )*(
fx(x,y)*(cB(x,y)*Ax(x,y)-cA(x,y)*Bx(x,y))+fy(x,y)*(cB(x,y)*Ay(x,y)-cA(x,y)*By(x,y)) )
+2*cA(x,y)*(
fx(x,y)*(cC(x,y)*Ax(x,y)-cA(x,y)*Cx(x,y))+fy(x,y)*(cC(x,y)*Ay(x,y)-cA(x,y)*Cy(x,y)) );
}else{
return (fyy(x,y)-fxx(x,y))*fx(x,y)*fy(x,y)-fxy(x,y)*(fy(x,y)*fy(x,y)-fx(x,y)*fx(x,y));
}
}
```