

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра математического моделирования и управления

Василевский Андрей Игоревич

**Численный расчет энергетического спектра донора в сферически
симметричной квантовой точке**

Дипломная работа

Научные руководители:
ассистент кафедры ММУ,
Левчук Е.А
кандидат физ.-мат. наук,
доцент Макаренко Л. Ф.

Допущен к защите

«___» _____ 2018 г.

Зав. кафедрой математического
моделирования и управления,
кандидат физ.-мат. наук,
доцент В.И. Белько

Минск, 2018

РЕФЕРАТ

Дипломная работа, 57с., 21 рисунок, 2 таблицы, 1 приложение, 8 источников.

УРАВНЕНИЕ ШРЕДИНГЕРА, МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ, ЗАДАЧА НА СОБСТВЕННЫЕ ЗНАЧЕНИЯ, РАЗНОСТНАЯ СХЕМА, КВАНТОВАЯ ТОЧКА

Объект исследования – энергетические уровни донора в сферически симметричной квантовой точке.

Цель исследования – построение разностной схемы для моделирования состояний донора в сферически симметричной квантовой точке.

Результат: написана программа, реализующая метод конечных разностей для вычисления энергетических уровней донора в сферической квантовой точке, реализован интерфейс, проведено сравнение двух разностных схем, исследовано зависимость энергетического спектра от параметров.

РЭФЕРАТ

Дыпломная праца, 57с., 21 малюнак, 2 табліцы, 1 дадатак, 8 крыніц.

РАЎНАННЕ ШРОДЗІНГЕРА, МЕТАД КОНЦЫХ РОЗНАСЦЯЎ, ЗАДАЧА НА УЛАСНЫЯ ЗНАЧЭННІ, РОЗНАСНЫЯ СХЕМЫ, КВАНТАВАЯ КРОПКА

Аб'ект даследвання – энергетычныя ўзроўні донара ў сферычна сіметрычнай квантавай кропцы.

Мэта даследвання – пабудова рознаснай схемы для мадэлявання станаў донара ў сферычна сіметрычнай квантавай кропцы.

Вынікі: напісана праграма, якая рэалізуе метады канчатковых рознасцяў для вылічэння энергетычных узроўняў донора ў сферычна сіметрычнай квантавай кропцы, рэалізаваны інтэрфейс, праведзена параўнанне двух рознасных схем, даследавана залежнасць энергетычнага спектра ад параметраў.

ABSTRACT

Graduate work, 57 p., 21 pictures, 2 tables, 1 application, 8 sources.

SCHRODINGER EQUATION, FINITE DIFFERENCE METHOD,
EIGENVALUES PROBLEM, DIFFERENCE SCHEME, QUANTUM DOT

The object of research — energy levels of the donor in a spherically symmetric quantum dot.

Purpose — construction of a difference scheme for modeling the states of a donor in a spherically symmetric quantum dot.

The result — a program is developed that implements the finite difference method to calculate the donor energy levels in a spherical quantum dot, an interface is realized, two difference schemes are compared, and the energy spectrum is analyzed as a function of the parameters.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1 Постановка задачи.....	6
2 Численная процедура.....	10
2.1 Первый вариант разностной схемы.....	10
2.1.1 Разностная схема в случае нулевого магнитного квантового числа и симметрией относительно оси Z	12
2.1.2 Разностная схема в случае не нулевого магнитного квантового числа и антисимметрией относительно оси Z	14
2.1.3 Разностная схема в случае не нулевого магнитного квантового числа и симметрией относительно оси Z	16
2.2 Альтернативный вариант разностной схемы.....	18
2.2.1 Альтернативная разностная схема в случае нулевого магнитного квантового числа и симметрией относительно оси Z	19
2.2.2 Альтернативная разностная схема в случае не нулевого магнитного квантового числа и антисимметрией относительно оси Z	22
2.2.3 Альтернативная разностная схема в случае не нулевого магнитного квантового числа и симметрией относительно оси Z	25
3 Реализация на Python.....	29
3.1 Решение матричной задачи на собственные значения.....	29
3.2 Реализация графического интерфейса.....	31
4 Результаты численных экспериментов.....	34
4.1 Энергии в зависимости от радиуса квантовой точки.....	34
4.2 Энергии в зависимости от величины магнитного поля.....	38
4.3 Сравнение схем.....	41
4.3.1 Случай отсутствия донора.....	41
4.3.2 Случай присутствия донора.....	42
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ЛИТЕРАТУРЫ.....	46
ПРИЛОЖЕНИЕ А. Листинг модуля программы, реализующий разностную схему.....	47

ВВЕДЕНИЕ

Квантовая точка — фрагмент проводника или полупроводника, носители заряда (электроны) которого ограничены в пространстве по всем трём измерениям. Размер квантовой точки должен быть настолько мал, чтобы квантовые эффекты были существенными. Квантовые точки могут использоваться в транзисторах, фотоэлементах, светодиодах, лазерных диодах. Исследуются также возможности применения квантовых точек в качестве биомаркеров для визуализации в медицине и в качестве кубитов для квантовых вычислений [1].

Наличие примесей в квантовой точке может существенно изменять ее электронные и оптические свойства. В результате этого возникает необходимость изучения энергетического спектра донора в полупроводниковом нанокристалле. В частности, исследуется влияние размеров квантовой точки [2], а также внешнего электрического и магнитного полей [3] на состояния донора в сферически симметричной квантовой точке.

Согласно приближению эффективной массы, энергетический спектр в квантовой точке может быть описан с помощью стационарного уравнения Шредингера [4]. В общем случае стационарное уравнение Шредингера не имеет аналитического решения, поэтому зачастую используются приближенные методы. При этом, как правило, используются методы, основанные на вариационном принципе.

В данной работе для решения уравнения используется метод конечных разностей, который сводит дифференциальную задачу к матричной (алгебраической) задаче на собственные значения. Метод конечных разностей для решения стационарного уравнения Шредингера используется редко и недостаточно изучен. Цель данной работы – реализовать различные разностные схемы для расчета состояний донора в сферически симметричной квантовой точке, сравнить эти схемы между собой и исследовать зависимость энергетического спектра от размера квантовой точки и величины внешнего магнитного поля.

1 Постановка задачи

Рассмотрим однозарядную донорную примесь, находящуюся в центре сферически симметричной квантовой точки. На донорный электрон в квантовой точке действует однородное постоянное магнитное поле B , которое направлено вдоль оси Oz (Рисунок 1).

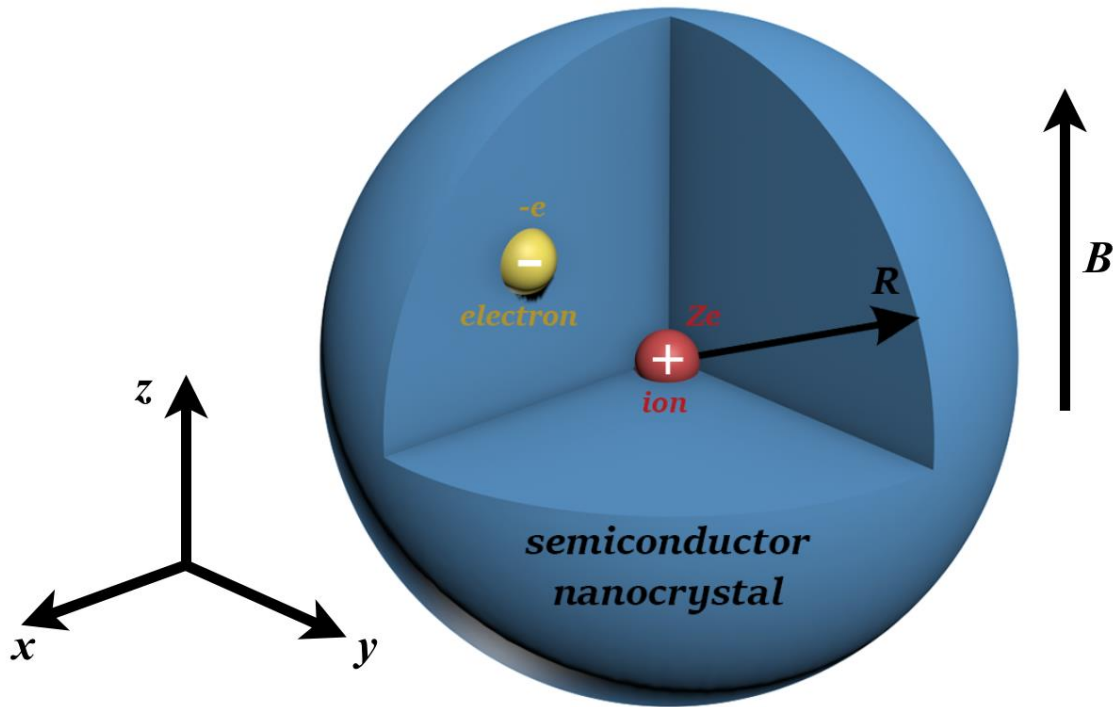


Рисунок 1 – Модель квантовой точки

Начало координат расположим на месте примеси и выразим длины и энергии в единицах измерения эффективного Боровского радиуса a^* :

$$a^* = \frac{4\pi\epsilon_0\epsilon_s\hbar^2}{m^*e^2},$$

и эффективного Ридберга Ry^* :

$$Ry^* = \frac{\hbar^2}{2m^*(a^*)^2} = \frac{1}{4\pi\epsilon_0\epsilon_s} \frac{e^2}{2a^*},$$

где \hbar – постоянная Планка, m^* – эффективная масса электрона, e – элементарный электрический заряд, ϵ_0 – диэлектрическая постоянная, ϵ_s – диэлектрическая проницаемость полупроводника. В этих единицах радиус точки обозначается как R .

Кроме того, безразмерный параметр $\gamma = (a^*/\lambda)^2$ показывает

напряженность магнитного поля, где $\lambda = \sqrt{\hbar/(eB)}$ - радиус Ландау.

Уравнение Шредингера для волновой функции Ψ и энергии электрона E в рамках приближения эффективной массы имеет вид [5]:

$$\hat{H}\Psi(r, \theta, \phi) = E\Psi(r, \theta, \phi), \quad (1)$$

где r, θ, ϕ сферические координаты:

$$\begin{cases} x = r \sin \theta \cos \phi \\ y = r \sin \theta \sin \phi \\ z = r \cos \theta \end{cases}.$$

Гамильтониан в уравнении (1) имеет следующий вид:

$$\hat{H} = -\nabla^2 - i\gamma \frac{\partial}{\partial \phi} + \frac{\gamma^2 r^2 \sin^2(\theta)}{4} - \frac{2}{r}, \quad (2)$$

где последнее слагаемое $(-2/r)$ представляет собой потенциальную энергию электрона в поле донорного центра. В выражении (2) $\nabla^2 = \Delta$ - оператор Лапласа, который в сферических координатах может быть записан как:

$$\Delta = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin(\theta)} \frac{\partial}{\partial \theta} \left(\sin(\theta) \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2(\theta)} \frac{\partial^2}{\partial \phi^2}.$$

Предположим, что электрон строго ограничен в пределах точки, тогда граничное условие $\Psi(R, \theta, \phi) = 0$ справедливо для всех значений θ и ϕ . Более того, благодаря осевой симметрии внешнего потенциала, собственные функции \hat{H} имеют вполне определенный угловой момент вдоль оси Oz . Тогда волновые функции электрона могут быть записаны как [5]:

$$\Psi^m(r, \theta, \phi) = \frac{e^{im\phi}}{\sqrt{2\pi}} \psi^m(r, \theta),$$

где m представляет собой целое число и $\psi_m(r, \theta)$ удовлетворяет уравнению:

$$\hat{H}_m \psi^m(r, \theta) = E^m \psi^m(r, \theta), \quad (3)$$

где \hat{H}_m в сферических координатах имеет вид:

$$\begin{aligned} \hat{H}_m = & -\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) - \frac{1}{r^2 \sin(\theta)} \frac{\partial}{\partial \theta} \left(\sin(\theta) \frac{\partial}{\partial \theta} \right) + \frac{m^2}{r^2 \sin^2(\theta)} + m\gamma \\ & + \frac{\gamma^2 r^2 \sin^2(\theta)}{4} - \frac{2}{r}. \end{aligned}$$

Кроме того, на границе квантовой точки $\psi^m(R, \theta) = 0$ для всех значений θ .

Так как $\hat{H}_{-m} - \hat{H}_m = 2m\gamma$, собственные значения E^{-m} и собственные функции $\psi^{-m}(r, \theta)$ могут быть найдены по формулам $E^{-m} = E^m - 2m\gamma$ и $\psi^{-m}(r, \theta) = \psi^m(r, \theta)$.

Условия на границе области $0 < r < R$, $0 < \theta < \frac{\pi}{2}$ будут различными в зависимости от вида симметрии волновой функции. Для описания симметрии функции относительно оси Oz введем величину $\sigma = \pm 1$, где $\psi^m(r, \pi - \theta) = \sigma \psi^m(r, \theta)$. Тогда $\sigma = 1$ будет соответствовать состояниям, симметричным относительно оси Oz , а $\sigma = -1$ – состояниям, антисимметричным относительно оси Oz . Для $\sigma = -1$ в точке $\theta = \pi/2$ волновая функция должна обращаться в ноль, для $\sigma = 1$ в точке $\theta = \pi/2$ производная по θ должна обращаться в ноль. Граничное условие на оси $\theta = 0$ определяется значением квантового числа m . При $m \neq 0$, $\psi^m(r, 0) = 0$, при $m = 0$ в ноль обращается производная по θ .

В данной работе будем рассматривать 3 случая: $m = 0$, $\sigma = 1$; $m \neq 0$, $\sigma = -1$ и $m \neq 0$, $\sigma = 1$.

В дальнейшем для заданных m и σ энергии и волновые функции будем обозначать E_0 , E_1 , E_2 и ψ_0 , ψ_1 , ψ_2 , где E_0 , ψ_0 соответствуют состоянию с минимальной энергией.

В итоге получаем следующую задачу:

$$\begin{aligned} & \left(-\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) - \frac{1}{r^2 \sin(\theta)} \frac{\partial}{\partial \theta} \left(\sin(\theta) \frac{\partial}{\partial \theta} \right) + \frac{m^2}{r^2 \sin^2(\theta)} + m\gamma \right. \\ & \quad \left. + \frac{\gamma^2 r^2 \sin^2(\theta)}{4} - \frac{2}{r} \right) \psi_m(r, \theta) = E_m \psi_m(r, \theta), \end{aligned} \quad (4)$$

$$0 < r < R, \quad 0 < \theta < \frac{\pi}{2}.$$

Задача дополняется граничными условиями, которые различаются от m , σ .

1) Для $m = 0$, $\sigma = 1$:

$$\psi^m(R, \theta) = 0, \quad 0 < \theta < \frac{\pi}{2}, \quad (5)$$

$$\frac{\partial \psi^m}{\partial \theta} \left(r, \frac{\pi}{2} \right) = 0, \quad 0 < r < R, \quad (6)$$

$$\frac{\partial \psi^m}{\partial \theta} (r, 0) = 0, \quad 0 < r < R, \quad (7)$$

$$|\psi^m(0, \theta)| < \infty, \quad 0 < \theta < \frac{\pi}{2}, \quad (8)$$

2) Для $m \neq 0$, $\sigma = -1$:

$$\psi^m(R, \theta) = 0, \quad 0 < \theta < \frac{\pi}{2}, \quad (9)$$

$$\psi^m\left(r, \frac{\pi}{2}\right) = 0, \quad 0 < r < R, \quad (10)$$

$$\psi^m(r, 0) = 0, \quad 0 < r < R, \quad (11)$$

$$|\psi^m(0, \theta)| < \infty, \quad 0 < \theta < \frac{\pi}{2}, \quad (12)$$

3) Для $m \neq 0$, $\sigma = 1$:

$$\psi^m(R, \theta) = 0, \quad 0 < \theta < \frac{\pi}{2}, \quad (13)$$

$$\frac{\partial \psi^m}{\partial \theta}\left(r, \frac{\pi}{2}\right) = 0, \quad 0 < r < R, \quad (14)$$

$$\psi^m(r, 0) = 0, \quad 0 < r < R, \quad (15)$$

$$|\psi^m(0, \theta)| < \infty, \quad 0 < \theta < \frac{\pi}{2}, \quad (16)$$

2 Численная процедура

Энергетические уровни доноров в квантовой точке будем вычислять на прямоугольной и равномерной сетке в расчетной области $0 \leq r \leq R$ и $0 \leq \theta \leq \pi/2$ с шагом $h = R/(M + 1)$ и $k = \pi/(2N + 2)$ соответственно, где R - безразмерный радиус квантовой точки, а M, N являются достаточно большими целыми числами. Следовательно, сетка имеет вид

$$\begin{aligned}\omega_r &= \{r_i = (i - 1)h, h = R/(M + 1), i = \overline{1, M + 2}\}, \\ \omega_\theta &= \{\theta_j = (j - 1)k, k = \pi/(2N + 2), j = \overline{1, N + 2}\}, \\ \omega &= \omega_r \times \omega_\theta.\end{aligned}$$

Значения волновой функции в узлах сетки будем обозначать:

$$G_{i,j} = \psi_m(r_i, \theta_j).$$

Задачи (4) – (13) будем решать с использованием двух вариантов разностных схем.

В первом случае рассмотрим схему, использованную в [5]. Для её получения гамильтониан из уравнения (4) представляется в виде:

$$\hat{H}_m = -\frac{\partial^2}{\partial r^2} - \frac{2}{r} \frac{\partial}{\partial r} - \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} - \frac{\cot(\theta)}{r^2} \frac{\partial}{\partial \theta} + \frac{m^2}{r^2 \sin^2(\theta)} + m\gamma + \frac{\gamma^2 r^2 \sin^2(\theta)}{4} - \frac{2}{r}.$$

Затем в полученном выражении аппроксимируются производные. Во втором случае разностная схема строилась согласно методу, приведенному в [7] для уравнений с переменными коэффициентами.

2.1 Первый вариант разностной схемы

Вначале рассмотрим и опишем разностную схему, приведенную в статье [5]. Дискретизации второго порядка уравнения (4) приводят к следующему равенству:

$$\begin{aligned}-\frac{1}{h^2}(G_{i+1,j} - 2G_{i,j} + G_{i-1,j}) - \frac{1}{hr_i}(G_{i+1,j} - G_{i-1,j}) \\ - \frac{1}{k^2 r_i^2}(G_{i,j+1} - 2G_{i,j} + G_{i,j-1}) - \frac{\cot(\theta_i)}{2kr_i^2}(G_{i,j+1} - G_{i,j-1}) \\ + V_{i,j}G_{i,j} = E_m(M, N)G_{i,j},\end{aligned}\tag{17}$$

$$i = \overline{2, M + 1}, j = \overline{2, N + 1},$$

где

$$V_{i,j} = m\gamma + \frac{m^2}{r_i^2 \sin^2(\theta_j)} + \frac{\gamma^2}{4} r_i^2 \sin^2(\theta_j) - \frac{2}{r_i}, \quad (18)$$

$$i = \overline{2, M+1}, j = \overline{2, N+1},$$

$E_m(M, N)$ является конечно-разностной оценкой энергетического уровня E_m .

Граничные условия дискретизированы до второго порядка. Удержание электрона внутри квантовой точки, или граничное условие $\psi_m(R, \theta) = 0$, приводит к $G_{M+2,j} = 0$. Когда $\sigma = -1$ ($\sigma = 1$), функция (частная производная по θ) должна исчезать при $\theta = \pi/2$, другими словами $G_{i,N+2} = 0$ ($G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$) [8].

Кроме того, гладкость по полярной оси состояний $m \neq 0$ ($m = 0$) требует, чтобы функция (производная по θ) исчезла при $\theta = 0$, другими словами $G_{i,1} = 0$ ($G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$). Последнее граничное условие относится к $\psi_m(0, \theta)$. Поскольку каждый узел имеющий вид $(0, \theta_j)$ попадает в центр квантовой точки, то $G_{1,j} = G_{1,1}$. Более того, коэффициент при $G_{1,j}$ в уравнении (4) имеет вид:

$$\frac{\left(\frac{1}{h} - \frac{1}{r^2}\right)}{h} = 0. \quad (19)$$

Это означает, что $G_{1,1}$ можно опустить в конечно-разностных расчетах энергетических уровней.

Уравнение (4) и граничные условия (5-16) приводят в матричной задаче на собственные значения:

$$A\vec{G} = E_m(M, N)\vec{G}, \quad (20)$$

где \vec{G} является линейным разложением длины MN, которое содержит значения $G_{i,j}$ для $2 \leq i \leq M+1, 2 \leq j \leq N+1$. Приближенные значения энергетических уровней являются собственными значениями матрицы A (см. равенство (20)).

Раскрыв скобки и приведя подобные уравнения (17), мы получим следующее равенство:

$$\begin{aligned} G_{i+1,j} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \right) + G_{i-1,j} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) \\ + G_{i,j+1} \left(-\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \right) + G_{i,j-1} \left(-\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \right) \\ = E_m(M, N)G_{i,j}, \end{aligned} \quad (21)$$

$$i = \overline{2, M+1}, j = \overline{2, N+1}.$$

2.1.1 Разностная схема в случае нулевого магнитного квантового числа и симметрией относительно оси Z

Рассмотрим аппроксимацию задачи (4)-(8) ($m = 0, \sigma = 1$).

Равенство (21) используется для нахождения значений волновых функций во внутренних узлах.

Воспользуемся граничными условиями для получения схемы нахождения волновой функции в узлах, примыкающим к границе.

Согласно (19) считаем $G_{1,j} = 0$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$G_{3,j} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,j} \right) + G_{2,j+1} \left(-\frac{1}{k^2 r_2^2} - \frac{\cot(\theta_j)}{2kr_2^2} \right) + G_{2,j-1} \left(-\frac{1}{k^2 r_2^2} + \frac{\cot(\theta_j)}{2kr_2^2} \right) = E_m(M, N) G_{2,j},$$

$$j = \overline{2, N+1}.$$

Граничное условие (5) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$G_{M+1,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,j} \right) + G_{M,j} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,j+1} \left(-\frac{1}{k^2 r_{M+1}^2} - \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) + G_{M+1,j-1} \left(-\frac{1}{k^2 r_{M+1}^2} + \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) = E_m(M, N) G_{M+1,j},$$

$$j = \overline{2, N+1}.$$

Граничное условие (7) дает ограничение что $3G_{i,1} - 4G_{i,2} + G_{i,3} = 0$, это приводит к $G_{i,1} = \frac{4G_{i,2} - G_{i,3}}{3}$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$G_{i+1,2} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,2} \left(\frac{2}{h^2} - \frac{2}{3k^2 r_i^2} + \frac{2\cot(\theta_2)}{3kr_i^2} + V_{i,2} \right) + G_{i-1,2} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) + G_{i,3} \left(-\frac{2}{3k^2 r_i^2} - \frac{2\cot(\theta_2)}{3kr_i^2} \right) = E_m(M, N) G_{i,2},$$

$$i = \overline{2, M+1}.$$

Граничное условие (6) дает ограничение что $G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$, это приводит к $G_{i,N+2} = \frac{4G_{i,N+1} - G_{i,N}}{3}$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N + 1$), уравнение имеет вид:

$$\begin{aligned} G_{i+1,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,N+1} \left(\frac{2}{h^2} + \frac{2}{3k^2r_i^2} - \frac{2\cot(\theta_{N+1})}{3kr_i^2} + V_{i,N+1} \right) \\ + G_{i-1,N+1} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) + G_{i,N+2} \left(-\frac{1}{k^2r_i^2} - \frac{\cot(\theta_{N+1})}{2kr_i^2} \right) \\ + G_{i,N} \left(-\frac{2}{3k^2r_i^2} + \frac{2\cot(\theta_{N+1})}{3kr_i^2} \right) = E_m(M, N)G_{i,N+1}, \\ i = \overline{2, M+1}. \end{aligned}$$

При расчете угловых точек принимаются во внимание два граничных условия.

Согласно (19) считаем что $G_{1,2} = 0$, а также, граничное условие (7) дает ограничение $3G_{2,1} - 4G_{2,2} + G_{2,3} = 0$, это приводит к $G_{2,1} = \frac{4G_{2,2} - G_{2,3}}{3}$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2$, $j = 2$) уравнение имеет вид:

$$\begin{aligned} G_{3,2} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,2} \left(\frac{2}{h^2} + \frac{2}{3k^2r_2^2} + \frac{2\cot(\theta_2)}{3kr_2^2} + V_{2,j} \right) \\ + G_{2,3} \left(-\frac{2}{3k^2r_2^2} - \frac{2\cot(\theta_2)}{3kr_2^2} \right) = E_m(M, N)G_{2,2}. \end{aligned}$$

Согласно (19) считаем что $G_{1,N+1} = 0$, а также, граничное условие (6) дает ограничение $G_{2,N} + 4G_{2,N+1} + 3G_{2,N+2} = 0$, это приводит к $G_{2,N+2} = \frac{G_{2,N+1} - G_{2,N}}{3}$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2$, $j = N + 1$) уравнение имеет вид:

$$\begin{aligned} G_{3,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,N+1} \left(\frac{2}{h^2} + \frac{2}{3k^2r_2^2} - \frac{2\cot(\theta_{N+1})}{3kr_2^2} + V_{2,N+1} \right) \\ + G_{2,N} \left(-\frac{2}{3k^2r_2^2} + \frac{2\cot(\theta_{N+1})}{3kr_2^2} \right) = E_m(M, N)G_{2,N+1}. \end{aligned}$$

Граничные условия (5), (7) дают ограничения что $G_{M+2,2} = 0$, $3G_{M+1,1} - 4G_{M+1,2} + G_{M+1,3} = 0$, это приводит к $G_{M+1,1} = \frac{4G_{M+1,2} - G_{M+1,3}}{3}$, тогда в угловом узле при $r = R$, $\theta = 0$ ($i = M + 1$, $j = 2$) уравнение имеет вид:

$$G_{M+1,2} \left(\frac{2}{h^2} + \frac{2}{3k^2 r_{M+1}^2} + \frac{2 \cot(\theta_2)}{3k r_{M+1}^2} + V_{M+1,2} \right) + G_{M,2} \left(\frac{1}{h r_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,3} \left(-\frac{2}{3k^2 r_{M+1}^2} - \frac{2 \cot(\theta_2)}{3k r_{M+1}^2} \right) = E_m(M, N) G_{M+1,2}.$$

Граничные условия (5), (6) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N} - 4G_{M+1,N+1} + 3G_{M+1,N+2} = 0$, это приводит к $G_{M+1,N+2} = \frac{4G_{M+1,N+1} - G_{M+1,N}}{3}$, тогда в угловом узле при $r = R$, $\theta = \pi/2$ ($i = M + 1$, $j = N + 1$) уравнение имеет вид:

$$G_{M+1,N+1} \left(\frac{2}{h^2} + \frac{2}{3k^2 r_{M+1}^2} - \frac{2 \cot(\theta_{N+1})}{3k r_{M+1}^2} + V_{M+1,N+1} \right) + G_{M,N+1} \left(\frac{1}{h r_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,N} \left(-\frac{2}{3k^2 r_{M+1}^2} + \frac{2 \cot(\theta_{N+1})}{3k r_{M+1}^2} \right) = E_m(M, N) G_{M+1,N+1}$$

2.1.2 Разностная схема в случае не нулевого магнитного квантового числа и антисимметрией относительно оси Z

Рассмотрим аппроксимацию задачи (4), (9)-(12) (случай $m \neq 0$, $\sigma = -1$).

Для нахождения значений волновых функций во внутренних узлах, используется равенство (21) с $m \neq 0$.

Воспользуемся граничными условиями для получение схемы нахождения волновой функции в узлах, примыкающим к границе.

Согласно (19) считаем $G_{1,j} = 0$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$G_{3,j} \left(-\frac{1}{h^2} - \frac{1}{h r_2} \right) + G_{2,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,j} \right) + G_{2,j+1} \left(-\frac{1}{k^2 r_2^2} - \frac{\cot(\theta_j)}{2k r_2^2} \right) + G_{2,j-1} \left(-\frac{1}{k^2 r_2^2} + \frac{\cot(\theta_j)}{2k r_2^2} \right) = E_m(M, N) G_{2,j},$$

$$j = \overline{2, N+1}.$$

Граничное условие (9) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$\begin{aligned}
& G_{M+1,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,j} \right) + G_{M,j} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) \\
& + G_{M+1,j+1} \left(-\frac{1}{k^2 r_{M+1}^2} - \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) \\
& + G_{M+1,j-1} \left(-\frac{1}{k^2 r_{M+1}^2} + \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) = E_m(M, N) G_{M+1,j}, \\
& j = \overline{2, N+1}.
\end{aligned}$$

Граничное условие (11) дает ограничение что $G_{i,1} = 0$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,2} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,2} \right) + G_{i-1,2} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) \\
& + G_{i,3} \left(-\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_2)}{2kr_i^2} \right) = E_m(M, N) G_{i,2}, \\
& j = \overline{2, N+1}
\end{aligned}$$

Граничное условие (10) дает ограничение что $G_{i,N+2} = 0$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N+1$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,N+1} \left(\frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,N+1} \right) + G_{i-1,N+1} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) \\
& + G_{i,N} \left(-\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_{N+1})}{2kr_i^2} \right) = E_m(M, N) G_{i,N+1}, \\
& i = \overline{2, M+1}.
\end{aligned}$$

При расчете угловых точек принимаются во внимание два граничных условия.

Согласно (19) считаем что $G_{1,2} = 0$, а также, граничное условие (11) дает ограничение $G_{2,1} = 0$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2$, $j = 2$) уравнение имеет вид:

$$\begin{aligned}
& G_{3,2} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,j} \right) + G_{2,3} \left(-\frac{1}{k^2 r_2^2} - \frac{\cot(\theta_2)}{2kr_2^2} \right) \\
& = E_m(M, N) G_{2,2}
\end{aligned}$$

Согласно (19) считаем что $G_{1,N+1} = 0$, а также, граничное условие (10) дает ограничение $G_{2,N+2} = 0$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2$,

$j = N + 1$) уравнение имеет вид:

$$G_{3,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,N+1} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,N+1} \right) + G_{2,N} \left(-\frac{1}{k^2 r_2^2} + \frac{\cot(\theta_{N+1})}{2kr_2^2} \right) = E_m(M, N) G_{2,N+1}.$$

Граничные условия (9), (12) дают ограничения что $G_{M+2,2} = 0$, $G_{M+1,1} = 0$, тогда в угловом узле при $r = R, \theta = 0$ ($i = M + 1, j = 2$) уравнение имеет вид:

$$G_{M+1,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,2} \right) + G_{M,2} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,3} \left(-\frac{1}{k^2 r_{M+1}^2} - \frac{\cot(\theta_2)}{2kr_{M+1}^2} \right) = E_m(M, N) G_{M+1,2}.$$

Граничные условия (9), (10) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N+2} = 0$, тогда в угловом узле при $r = R, \theta = \pi/2$ ($i = M + 1, j = N + 1$) уравнение имеет вид:

$$G_{M+1,N+1} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,N+1} \right) + G_{M,N+1} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,N} \left(-\frac{1}{k^2 r_{M+1}^2} + \frac{\cot(\theta_{N+1})}{2kr_{M+1}^2} \right) = E_m(M, N) G_{M+1,N+1}.$$

2.1.3 Разностная схема в случае не нулевого магнитного квантового числа и симметрией относительно оси Z

Рассмотрим аппроксимацию задачи (4), (13)-(16) (случай $m \neq 0, \sigma = 1$).

Для нахождения значений волновых функций во внутренних узлах, используется равенство (21) с $m \neq 0$.

Воспользуемся граничными условиями для получение схемы нахождения волновой функции в узлах, примыкающим к границе.

Согласно (19) считаем $G_{1,j} = 0$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$G_{3,j} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,j} \right) + G_{2,j+1} \left(-\frac{1}{k^2 r_2^2} - \frac{\cot(\theta_j)}{2kr_2^2} \right) + G_{2,j-1} \left(-\frac{1}{k^2 r_2^2} + \frac{\cot(\theta_j)}{2kr_2^2} \right) = E_m(M, N) G_{2,j},$$

$$j = \overline{2, N+1}.$$

Граничное условие (13) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$\begin{aligned} G_{M+1,j} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,j} \right) + G_{M,j} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) \\ + G_{M+1,j+1} \left(-\frac{1}{k^2 r_{M+1}^2} - \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) \\ + G_{M+1,j-1} \left(-\frac{1}{k^2 r_{M+1}^2} + \frac{\cot(\theta_j)}{2kr_{M+1}^2} \right) = E_m(M, N) G_{M+1,j}, \\ j = \overline{2, N+1}. \end{aligned}$$

Граничное условие (15) дает ограничение что $G_{i,1} = 0$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$\begin{aligned} G_{i+1,2} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,2} \right) + G_{i-1,2} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) \\ + G_{i,3} \left(-\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_2)}{2kr_i^2} \right) = E_m(M, N) G_{i,2}, \\ j = \overline{2, N+1}. \end{aligned}$$

Граничное условие (14) дает ограничение что $G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$, это приводит к $G_{i,N+2} = \frac{4G_{i,N+1} - G_{i,N}}{3}$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N + 1$), уравнение имеет вид:

$$\begin{aligned} G_{i+1,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_i} \right) + G_{i,N+1} \left(\frac{2}{h^2} + \frac{2}{3k^2 r_i^2} - \frac{2 \cot(\theta_{N+1})}{3kr_i^2} + V_{i,N+1} \right) \\ + G_{i-1,N+1} \left(\frac{1}{hr_i} - \frac{1}{h^2} \right) + G_{i,N} \left(-\frac{2}{3k^2 r_i^2} + \frac{2 \cot(\theta_{N+1})}{3kr_i^2} \right) \\ = E_m(M, N) G_{i,N+1}, \\ i = \overline{2, M+1}. \end{aligned}$$

При расчете угловых точек принимаются во внимание два граничных условия.

Согласно (19) считаем что $G_{1,2} = 0$, а также, граничное условие (15) дает ограничение $G_{2,1} = 0$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2$, $j = 2$) уравнение имеет вид:

$$G_{3,2} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_2^2} + V_{2,j} \right) + G_{2,3} \left(-\frac{1}{k^2 r_2^2} - \frac{\cot(\theta_2)}{2kr_2^2} \right) = E_m(M, N)G_{2,2}.$$

Согласно (19) считаем что $G_{1,N+1} = 0$, а также, граничное условие (14) дает ограничение $G_{2,N} - 4G_{2,N+1} + 3G_{2,N+2} = 0$, приводит к $G_{2,N+2} = \frac{4G_{2,N+1} - G_{2,N}}{3}$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2$, $j = N + 1$) уравнение имеет вид:

$$G_{3,N+1} \left(-\frac{1}{h^2} - \frac{1}{hr_2} \right) + G_{2,N+1} \left(\frac{2}{h^2} + \frac{2}{3k^2 r_2^2} - \frac{2\cot(\theta_{N+1})}{3kr_2^2} + V_{2,N+1} \right) + G_{2,N} \left(-\frac{2}{3k^2 r_2^2} + \frac{2\cot(\theta_{N+1})}{3kr_2^2} \right) = E_m(M, N)G_{2,N+1}.$$

Граничные условия (13), (15) дают ограничения что $G_{M+2,2} = 0$, $G_{M+1,1} = 0$, тогда в угловом узле при $r = R$, $\theta = 0$ ($i = M + 1$, $j = 2$) уравнение имеет вид:

$$G_{M+1,2} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,2} \right) + G_{M,2} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,3} \left(-\frac{1}{k^2 r_{M+1}^2} - \frac{\cot(\theta_2)}{2kr_{M+1}^2} \right) = E_m(M, N)G_{M+1,2}.$$

Граничные условия (13), (14) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N} - 4G_{M+1,N+1} + 3G_{M+1,N+2} = 0$, это приводит к $G_{M+1,N+2} = \frac{4G_{M+1,N+1} - G_{M+1,N}}{3}$, тогда в угловом узле при $r = R$, $\theta = \pi/2$ ($i = M + 1$, $j = N + 1$) уравнение имеет вид:

$$G_{M+1,N+1} \left(\frac{2}{h^2} + \frac{2}{k^2 r_{M+1}^2} + V_{M+1,N+1} \right) + G_{M,N+1} \left(\frac{1}{hr_{M+1}} - \frac{1}{h^2} \right) + G_{M+1,N} \left(-\frac{2}{3k^2 r_{M+1}^2} + \frac{2\cot(\theta_{N+1})}{3kr_{M+1}^2} \right) = E_m(M, N)G_{M+1,N+1}.$$

2.2 Альтернативный вариант разностной схемы

Построим еще один вариант разностной схемы в случае $m = 0$, $\sigma = 1$. Для этого уравнение (4) заменим следующим разностным соотношением:

$$\begin{aligned}
& - \frac{r_{i+0,5}^2(G_{i+1,j} - G_{i,j}) - r_{i-0,5}^2(G_{i,j} - G_{i-1,j})}{r_i^2 h^2} \\
& - \frac{\sin(\theta_{j+0,5})(G_{i,j+1} - G_{i,j}) - \sin(\theta_{j-0,5})(G_{i,j} - G_{i,j-1})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j} G_{i,j} \quad (22) \\
& = E G_{i,j},
\end{aligned}$$

$$i = \overline{2, M+1}, j = \overline{2, N+1}.$$

Раскрыв скобки в выражении (22) и приведя подобные, мы получим следующее:

$$\begin{aligned}
& G_{i+1,j} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,j} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\
& + G_{i,j} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j} \right) \quad (23) \\
& + G_{i,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} \right) + G_{i,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2} \right) = E G_{i,j}, \\
& i = \overline{2, M+1}, j = \overline{2, N+1}.
\end{aligned}$$

2.2.1 Альтернативная разностная схема в случае нулевого магнитного квантового числа и симметрией относительно оси Z

Граничные условия аппроксимируем со вторым порядком:

$$\theta = 0: \quad 3G_{i,1} - 4G_{i,2} + G_{i,3} = 0, \quad (24)$$

$$\theta = \frac{\pi}{2}: \quad G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0, \quad (25)$$

$$r = 0: \quad 3G_{1,j} - 4G_{2,j} + G_{3,j} = 0, \quad (26)$$

$$r = R: \quad G_{M+2,j} = 0, \quad (27)$$

$$i = \overline{2, M+1}, j = \overline{2, N+1}.$$

Равенство (23) используется для нахождения значений волновых функций во внутренних узлах.

Граничное условие (27) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$\begin{aligned}
& G_{M,j} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,j} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} + V_{M+1,j} \right) \\
& + G_{M+1,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) + G_{M+1,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) \\
& = E G_{M+1,j}, \\
& j = \overline{2, N+1}.
\end{aligned}$$

Граничное условие (26) дает ограничение что $3G_{1,j} - 4G_{2,j} + G_{3,j} = 0$, это приводит к $G_{1,j} = \frac{4G_{2,j} - G_{3,j}}{3}$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$\begin{aligned}
& G_{3,j} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3r_2^2 h^2} \right) \\
& + G_{2,j} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} - \frac{4r_{1,5}^2}{3r_2^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} + V_{2,j} \right) \\
& + G_{2,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} \right) + G_{2,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_2^2 \sin(\theta_j) k^2} \right) = E G_{2,j}, \\
& j = \overline{2, N+1}.
\end{aligned}$$

Граничное условие (24) дает ограничение что $3G_{i,1} - 4G_{i,2} + G_{i,3} = 0$, это приводит к $G_{i,1} = \frac{4G_{i,2} - G_{i,3}}{3}$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,2} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,2} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\
& + G_{i,2} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_i^2 \sin(\theta_2) k^2} - \frac{4\sin(\theta_{1,5})}{3r_i^2 \sin(\theta_2) k^2} + V_{i,2} \right) \\
& + G_{i,3} \left(-\frac{\sin(\theta_{2,5})}{r_i^2 \sin(\theta_2) k^2} + \frac{\sin(\theta_{1,5})}{3r_i^2 \sin(\theta_2) k^2} \right) = E G_{i,2}, \\
& i = \overline{2, M+1}.
\end{aligned}$$

Граничное условие (25) дает ограничение что $G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$, это приводит к $G_{i,N+2} = \frac{4G_{i,N+1} - G_{i,N}}{3}$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N+1$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,N+1} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,N+1} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\
& + G_{i,N+1} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} - \frac{4 \sin(\theta_{N+1,5})}{3 r_i^2 \sin(\theta_j) k^2} \right. \\
& \left. + V_{i,N+1} \right) + G_{i,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} + \frac{\sin(\theta_{N+1,5})}{3 r_i^2 \sin(\theta_j) k^2} \right) = E G_{i,N+1}, \\
& i = \overline{2, M+1}.
\end{aligned}$$

При расчете угловых точек принимаются во внимание два граничных условия.

Граничные условия (26), (24) дают ограничения что $3G_{1,2} - 4G_{2,2} + G_{3,2} = 0$, $3G_{2,1} - 4G_{2,2} + G_{2,3} = 0$, это приводит к $G_{1,2} = \frac{4G_{2,2} - G_{3,2}}{3}$, $G_{2,1} = \frac{4G_{2,2} - G_{2,3}}{3}$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2$, $j = 2$) уравнение имеет вид:

$$\begin{aligned}
& G_{3,2} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3 r_2^2 h^2} \right) \\
& + G_{2,2} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_2^2 \sin(\theta_2) k^2} - \frac{4 r_{1,5}^2}{3 r_2^2 h^2} - \frac{4 \sin(\theta_{1,5})}{3 r_2^2 \sin(\theta_2) k^2} \right. \\
& \left. + V_{2,2} \right) + G_{2,3} \left(-\frac{\sin(\theta_{2,5})}{r_2^2 \sin(\theta_2) k^2} + \frac{\sin(\theta_{1,5})}{3 r_2^2 \sin(\theta_2) k^2} \right) = E G_{2,2}.
\end{aligned}$$

Граничные условия (26), (25) дают ограничения что $3G_{1,N+1} - 4G_{2,N+1} + G_{3,N+1} = 0$, $G_{2,N} - 4G_{2,N+1} + 3G_{2,N+2} = 0$, это приводит к $G_{1,N+1} = \frac{4G_{2,N+1} - G_{3,N+1}}{3}$, $G_{2,N+2} = \frac{4G_{2,N+1} - G_{2,N}}{3}$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2$, $j = N+1$) уравнение имеет вид:

$$\begin{aligned}
& G_{3,N+1} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3 r_2^2 h^2} \right) \\
& + G_{2,N+1} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} - \frac{4 r_{1,5}^2}{3 r_2^2 h^2} \right. \\
& \left. - \frac{4 \sin(\theta_{N+1,5})}{3 r_2^2 \sin(\theta_{N+1}) k^2} + V_{2,N+1} \right) \\
& + G_{2,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} - \frac{\sin(\theta_{N+1,5})}{3 r_2^2 \sin(\theta_{N+1}) k^2} \right) = E G_{2,N+1}.
\end{aligned}$$

Граничные условия (27), (24) дают ограничения что $G_{M+2,2} = 0$, $3G_{M+1,1} - 4G_{M+1,2} + G_{M+1,3} = 0$, это приводит к $G_{M+2,2} = 0$, $G_{M+1,1} = \frac{4G_{M+1,2} - G_{M+1,3}}{3}$,

тогда в угловом узле при $r = R, \theta = 0$ ($i = M + 1, j = 2$) уравнение имеет вид:

$$\begin{aligned}
& G_{M,2} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) \\
& + G_{M+1,2} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_{M+1}^2 \sin(\theta_2) k^2} - \frac{4 \sin(\theta_{1,5})}{3 r_{M+1}^2 \sin(\theta_2) k^2} \right. \\
& \left. + V_{M+1,2} \right) + G_{M+1,3} \left(-\frac{\sin(\theta_{2,5})}{r_{M+1}^2 \sin(\theta_2) k^2} + \frac{\sin(\theta_{1,5})}{3 r_{M+1}^2 \sin(\theta_2) k^2} \right) \\
& = E G_{M+1,2}.
\end{aligned}$$

Граничные условия (27), (25) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N} - 4G_{M+1,N+1} + 3G_{M+1,N+2} = 0$, это приводит к $G_{M+2,N+1} = 0$, $G_{M+1,N+2} = \frac{4G_{M+1,N+1} - G_{M+1,N}}{3}$, тогда в угловом узле при $r = R, \theta = \frac{\pi}{2}$ ($i = M + 1, j = N + 1$) уравнение имеет вид:

$$\begin{aligned}
& G_{M,N+1} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) \\
& + G_{M+1,N+1} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} \right. \\
& \left. - \frac{4 \sin(\theta_{N+1,5})}{3 r_{M+1}^2 \sin(\theta_{N+1}) k^2} + V_{M+1,N+1} \right) \\
& + G_{M+1,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} + \frac{\sin(\theta_{N+1,5})}{3 r_{M+1}^2 \sin(\theta_{N+1}) k^2} \right) \\
& = E G_{M+1,N+1}.
\end{aligned}$$

2.2.2 Альтернативная разностная схема в случае не нулевого магнитного квантового числа и антисимметрией относительно оси Z

Построим еще один вариант разностной схемы в случае $m \neq 0, \sigma = -1$. Граничные условия аппроксимируем со вторым порядком:

$$\theta = 0: G_{i,1} = 0, \quad (28)$$

$$\theta = \frac{\pi}{2}: G_{i,N+2} = 0, \quad (29)$$

$$r = 0: 3G_{1,j} - 4G_{2,j} + G_{3,j} = 0, \quad (30)$$

$$r = R: G_{M+2,j} = 0, \quad (31)$$

$$i = \overline{2, M+1}, j = \overline{2, N+1}.$$

Для нахождения значений волновых функций во внутренних узлах, используется равенство (23) с $m \neq 0$.

Условие конечности волновой функции при $\theta = 0$ заменим на равенство нулю первой производной по θ в этой точке.

Граничное условие (31) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$\begin{aligned} G_{M,j} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,j} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} + V_{M+1,j} \right) \\ + G_{M+1,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) + G_{M+1,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) \\ = E G_{M+1,j}, \\ j = \overline{2, N+1}. \end{aligned}$$

Граничное условие (30) дает ограничение что $3G_{1,j} - 4G_{2,j} + G_{3,j} = 0$, это приводит к $G_{1,j} = \frac{4G_{2,j} - G_{3,j}}{3}$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$\begin{aligned} G_{3,j} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3r_2^2 h^2} \right) \\ + G_{2,j} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} - \frac{4r_{1,5}^2}{3r_2^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} + V_{2,j} \right) \\ + G_{2,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} \right) + G_{2,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_2^2 \sin(\theta_j) k^2} \right) = E G_{2,j}, \\ j = \overline{2, N+1}. \end{aligned}$$

Граничное условие (28) дает ограничение что $G_{i,1} = 0$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$\begin{aligned} G_{i+1,2} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,2} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\ + G_{i,2} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_i^2 \sin(\theta_2) k^2} + V_{i,2} \right) \\ + G_{i,3} \left(-\frac{\sin(\theta_{2,5})}{r_i^2 \sin(\theta_2) k^2} \right) = E G_{i,2}, \\ i = \overline{2, M+1}. \end{aligned}$$

Граничное условие (29) дает ограничение что $G_{i,N+2} = 0$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N + 1$), уравнение имеет вид:

$$\begin{aligned} G_{i+1,N+1} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,N+1} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\ + G_{i,N+1} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} + V_{i,N+1} \right) \\ + G_{i,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} \right) = E G_{i,N+1}, \end{aligned}$$

$$i = \overline{2, M+1}.$$

При расчете угловых точек принимаются во внимание два граничных условия.

Граничные условия (30), (28) дают ограничения что $3G_{1,2} - 4G_{2,2} + G_{3,2} = 0$, $G_{2,1} = 0$, это приводит к $G_{1,2} = \frac{4G_{2,2} - G_{3,2}}{3}$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2, j = 2$) уравнение имеет вид:

$$\begin{aligned} G_{3,2} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3r_2^2 h^2} \right) \\ + G_{2,2} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_2^2 \sin(\theta_2) k^2} - \frac{4r_{1,5}^2}{3r_2^2 h^2} + V_{2,2} \right) \\ + G_{2,3} \left(-\frac{\sin(\theta_{2,5})}{r_2^2 \sin(\theta_2) k^2} \right) = E G_{2,2}. \end{aligned}$$

Граничные условия (30), (29) дают ограничения что $3G_{1,N+1} - 4G_{2,N+1} + G_{3,N+1} = 0$, $G_{2,N+2} = 0$, это приводит к $G_{1,N+1} = \frac{4G_{2,N+1} - G_{3,N+1}}{3}$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2, j = N + 1$) уравнение имеет вид:

$$\begin{aligned} G_{3,N+1} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3r_2^2 h^2} \right) \\ + G_{2,N+1} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} - \frac{4r_{1,5}^2}{3r_2^2 h^2} + V_{2,N+1} \right) \\ + G_{2,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} \right) = E G_{2,N+1}. \end{aligned}$$

Граничные условия (31), (28) дают ограничения что $G_{M+2,2} = 0$, $G_{M+1,1} = 0$, тогда в угловом узле при $r = R$, $\theta = 0$ ($i = M + 1, j = 2$) уравнение имеет вид:

$$G_{M,2} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,2} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_{M+1}^2 \sin(\theta_2) k^2} + V_{M+1,2} \right) + G_{M+1,3} \left(-\frac{\sin(\theta_{2,5})}{r_{M+1}^2 \sin(\theta_2) k^2} \right) = E G_{M+1,2}.$$

Граничные условия (31), (29) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N+2} = 0$, тогда в угловом узле при $r = R$, $\theta = \pi/2$ ($i = M + 1$, $j = N + 1$) уравнение имеет вид:

$$G_{M,N+1} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,N+1} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} + V_{M+1,N+1} \right) + G_{M+1,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} \right) = E G_{M+1,N+1}.$$

2.2.3 Альтернативная разностная схема в случае не нулевого магнитного квантового числа и симметрией относительно оси Z

Построим еще один вариант разностной схемы в случае $m \neq 0, \sigma = 1$. Граничные условия аппроксимируем со вторым порядком:

$$\theta = 0: G_{i,1} = 0, \quad (32)$$

$$\theta = \frac{\pi}{2}: G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0, \quad (33)$$

$$r = 0: 3G_{1,j} - 4G_{2,j} + G_{3,j} = 0, \quad (34)$$

$$r = R: G_{M+2,j} = 0, \quad (35)$$

$$i = \overline{2, M+1}, j = \overline{2, N+1}.$$

Для нахождения значений волновых функций во внутренних узлах, используется равенство (23) с $m \neq 0$.

Граничное условие (35) дает ограничение что $G_{M+2,j} = 0$, тогда в узлах, примыкающим к границе $r = R$ ($i = M + 1$), уравнение имеет вид:

$$\begin{aligned}
& G_{M,j} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,j} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} + V_{M+1,j} \right) \\
& + G_{M+1,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) + G_{M+1,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_{M+1}^2 \sin(\theta_j) k^2} \right) \\
& = E G_{M+1,j}, \\
& j = \overline{2, N+1}.
\end{aligned}$$

Граничное условие (34) дает ограничение что $3G_{1,j} - 4G_{2,j} + G_{3,j} = 0$, это приводит к $G_{1,j} = \frac{4G_{2,j} - G_{3,j}}{3}$, тогда в узлах, примыкающим к границе $r = 0$ ($i = 2$), уравнение имеет вид:

$$\begin{aligned}
& G_{3,j} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3r_2^2 h^2} \right) \\
& + G_{2,j} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} - \frac{4r_{1,5}^2}{3r_2^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} + V_{2,j} \right) \\
& + G_{2,j-1} \left(-\frac{\sin(\theta_{j-0,5})}{r_2^2 \sin(\theta_j) k^2} \right) + G_{2,j+1} \left(-\frac{\sin(\theta_{j+0,5})}{r_2^2 \sin(\theta_j) k^2} \right) = E G_{2,j}, \\
& j = \overline{2, N+1}.
\end{aligned}$$

Граничное условие (32) дает ограничение что $G_{i,1} = 0$, тогда в узлах, примыкающим к границе $\theta = 0$ ($j = 2$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,2} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,2} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\
& + G_{i,2} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_i^2 \sin(\theta_2) k^2} + V_{i,2} \right) \\
& + G_{i,3} \left(-\frac{\sin(\theta_{2,5})}{r_i^2 \sin(\theta_2) k^2} \right) = E G_{i,2}, \\
& i = \overline{2, M+1}.
\end{aligned}$$

Граничное условие (33) дает ограничение что $G_{i,N} - 4G_{i,N+1} + 3G_{i,N+2} = 0$, это приводит к $G_{i,N+2} = \frac{4G_{i,N+1} - G_{i,N}}{3}$, тогда в узлах, примыкающим к границе $\theta = \pi/2$ ($j = N+1$), уравнение имеет вид:

$$\begin{aligned}
& G_{i+1,N+1} \left(-\frac{r_{i+0,5}^2}{r_i^2 h^2} \right) + G_{i-1,N+1} \left(-\frac{r_{i-0,5}^2}{r_i^2 h^2} \right) \\
& + G_{i,N+1} \left(\frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} - \frac{4 \sin(\theta_{N+1,5})}{3 r_i^2 \sin(\theta_j) k^2} \right. \\
& \left. + V_{i,N+1} \right) + G_{i,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_i^2 \sin(\theta_{N+1}) k^2} + \frac{\sin(\theta_{N+1,5})}{3 r_i^2 \sin(\theta_j) k^2} \right) = E G_{i,N+1}, \\
& i = \overline{2, M+1}.
\end{aligned}$$

При расчете угловых точек принимаются во внимание два граничных условия.

Граничные условия (34), (32) дают ограничения что $3G_{1,2} - 4G_{2,2} + G_{3,2} = 0$, $G_{2,1} = 0$, это приводит к $G_{1,2} = \frac{4G_{2,2} - G_{3,2}}{3}$, тогда в угловом узле при $r = 0$, $\theta = 0$ ($i = 2, j = 2$) уравнение имеет вид:

$$\begin{aligned}
& G_{3,2} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3 r_2^2 h^2} \right) \\
& + G_{2,2} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_2^2 \sin(\theta_2) k^2} - \frac{4 r_{1,5}^2}{3 r_2^2 h^2} + V_{2,2} \right) \\
& + G_{2,3} \left(-\frac{\sin(\theta_{2,5})}{r_2^2 \sin(\theta_2) k^2} \right) = E G_{2,2}.
\end{aligned}$$

Граничные условия (34), (33) дают ограничения что $3G_{1,N+1} - 4G_{2,N+1} + G_{3,N+1} = 0$, $G_{2,N} - 4G_{2,N+1} + 3G_{2,N+2} = 0$, это приводит к $G_{1,N+1} = \frac{4G_{2,N+1} - G_{3,N+1}}{3}$, $G_{2,N+2} = \frac{4G_{2,N+1} - G_{2,N}}{3}$, тогда в угловом узле при $r = 0$, $\theta = \pi/2$ ($i = 2, j = N+1$) уравнение имеет вид:

$$\begin{aligned}
& G_{3,N+1} \left(-\frac{r_{2,5}^2}{r_2^2 h^2} + \frac{r_{1,5}^2}{3 r_2^2 h^2} \right) \\
& + G_{2,N+1} \left(\frac{r_{2,5}^2 + r_{1,5}^2}{r_2^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} - \frac{4 r_{1,5}^2}{3 r_2^2 h^2} \right. \\
& \left. - \frac{4 \sin(\theta_{N+1,5})}{3 r_2^2 \sin(\theta_{N+1}) k^2} + V_{2,N+1} \right) \\
& + G_{2,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_2^2 \sin(\theta_{N+1}) k^2} - \frac{\sin(\theta_{N+1,5})}{3 r_2^2 \sin(\theta_{N+1}) k^2} \right) = E G_{2,N+1}.
\end{aligned}$$

Граничные условия (35), (32) дают ограничения что $G_{M+2,2} = 0$, $G_{M+1,1} = 0$, тогда в угловом узле при $r = R$, $\theta = 0$ ($i = M+1, j = 2$)

уравнение имеет вид:

$$G_{M,2} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,2} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{2,5}) + \sin(\theta_{1,5})}{r_{M+1}^2 \sin(\theta_2) k^2} + V_{M+1,2} \right) + G_{M+1,3} \left(-\frac{\sin(\theta_{2,5})}{r_{M+1}^2 \sin(\theta_2) k^2} \right) = E G_{M+1,2}.$$

Граничные условия (35), (33) дают ограничения что $G_{M+2,N+1} = 0$, $G_{M+1,N} - 4G_{M+1,N+1} + 3G_{M+1,N+2} = 0$, это приводит к $G_{M+2,N+1} = 0$, $G_{M+1,N+2} = \frac{4G_{M+1,N+1} - G_{M+1,N}}{3}$, тогда в угловом узле при $r = R$, $\theta = \pi/2$ ($i = M + 1$, $j = N + 1$) уравнение имеет вид:

$$G_{M,N+1} \left(-\frac{r_{M+0,5}^2}{r_{M+1}^2 h^2} \right) + G_{M+1,N+1} \left(\frac{r_{M+1,5}^2 + r_{M+0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{N+1,5}) + \sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} - \frac{4\sin(\theta_{N+1,5})}{3r_{M+1}^2 \sin(\theta_{N+1}) k^2} + V_{M+1,N+1} \right) + G_{M+1,N} \left(-\frac{\sin(\theta_{N+0,5})}{r_{M+1}^2 \sin(\theta_{N+1}) k^2} + \frac{\sin(\theta_{N+1,5})}{3r_{M+1}^2 \sin(\theta_{N+1}) k^2} \right) = E G_{M+1,N+1}.$$

3 Реализация на Python

Разностные схемы были реализованы на языке Python. Текст программы приведен в приложении А.

3.1 Решение матричной задачи на собственные значения

Каждому элементу $G_{i,j}$ в матрице А (см. равенство (20)) соответствует строка, индекс которой находится по формуле:

$$index = (i - 2) * M + j - 2$$

Рассмотрим по одному случаю каждой из схем. Возьмем случай не нулевого квантового числа с антисимметрией относительно оси Z ($m \neq 0$, $\sigma = -1$). Применяя схемы нахождения точек из пунктов 2.1.1 и 2.2.1 получим следующие реализации заполнения матрицы А:

1) Схема из статьи [5]:

$$\begin{aligned} i = 2, j = 2 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index + 1] &= -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index + M] &= -\frac{1}{h^2} - \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} i = N+1, j = 2 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index + 1] &= -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index - M] &= -\frac{1}{h^2} + \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} i = 2, 2 < j < M+1 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index + 1] &= -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index - 1] &= -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index + M] &= -\frac{1}{h^2} - \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} i = N+1, 2 < j < M+1 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index - 1] &= -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index + 1] &= -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \end{aligned}$$

$$\begin{aligned} i = 2, j = M+1 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index - 1] &= -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index + M] &= -\frac{1}{h^2} - \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} i = N+1, j = M+1 \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index - 1] &= -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index - M] &= -\frac{1}{h^2} + \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} 2 < i < N+1, j = 2, \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index + 1] &= -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index - M] &= -\frac{1}{h^2} + \frac{1}{hr_i} \\ A[index, index + M] &= -\frac{1}{h^2} - \frac{1}{hr_i} \end{aligned}$$

$$\begin{aligned} 2 < i < N+1, j = M+1, \\ A[index, index] &= \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j} \\ A[index, index - 1] &= -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2} \\ A[index, index - M] &= -\frac{1}{h^2} + \frac{1}{hr_i} \end{aligned}$$

$$A[index, index - M] = -\frac{1}{h^2} + \frac{1}{hr_i}$$

$$A[index, index + M] = -\frac{1}{h^2} - \frac{1}{hr_i}$$

$$2 < i < N+1, 2 < j < M+1$$

$$A[index, index] = \frac{2}{h^2} + \frac{2}{k^2 r_i^2} + V_{i,j}$$

$$A[index, index - 1] = -\frac{1}{k^2 r_i^2} + \frac{\cot(\theta_j)}{2kr_i^2}$$

$$A[index, index + 1] = -\frac{1}{k^2 r_i^2} - \frac{\cot(\theta_j)}{2kr_i^2}$$

$$A[index, index - M] = -\frac{1}{h^2} + \frac{1}{hr_i}$$

$$A[index, index + M] = -\frac{1}{h^2} - \frac{1}{hr_i}$$

2) Альтернативная схема:

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} - \frac{4r_{i-0,5}^2}{3r_i^2 h^2} + V_{i,j}$$

$$A[index, index + 1] = -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index + M] = -\frac{r_{i+0,5}^2}{r_i^2 h^2} + \frac{r_{i-0,5}^2}{3r_i^2 h^2}$$

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_{M+1}^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j}$$

$$A[index, index + 1] = -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index - M] = -\frac{r_{i-0,5}^2}{r_i^2 h^2}$$

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} - \frac{4r_{i-0,5}^2}{3r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j}$$

$$A[index, index - 1] = -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index + 1] = -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index + M] = -\frac{r_{i+0,5}^2}{r_i^2 h^2} + \frac{r_{i-0,5}^2}{3r_i^2 h^2}$$

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} - \frac{4r_{i-0,5}^2}{3r_i^2 h^2} + V_{i,j}$$

$$A[index, index - 1] = -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index + M] = -\frac{r_{i+0,5}^2}{r_i^2 h^2} + \frac{r_{i-0,5}^2}{3r_i^2 h^2}$$

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j}$$

$$A[index, index - 1] = -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index - M] = -\frac{r_{i-0,5}^2}{r_i^2 h^2}$$

$$A[index, index] = \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j}$$

$$A[index, index + 1] = -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2}$$

$$A[index, index - M] = -\frac{r_{i-0,5}^2}{r_i^2 h^2}$$

$$A[index, index + M] = -\frac{r_{i+0,5}^2}{r_i^2 h^2}$$

$$\begin{aligned}
& i = N+1, 2 < j < M+1 & 2 < i < N+1, j = M+1 \\
A[index, index] &= \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j} & A[index, index] &= \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j} \\
A[index, index - 1] &= -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} & A[index, index - 1] &= -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} \\
A[index, index + 1] &= -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2} & A[index, index - M] &= -\frac{r_{i-0,5}^2}{r_i^2 h^2} \\
A[index, index - M] &= -\frac{r_{i-0,5}^2}{r_i^2 h^2} & A[index, index + M] &= -\frac{r_{i+0,5}^2}{r_i^2 h^2}
\end{aligned}$$

$$\begin{aligned}
& 2 < i < N+1, 2 < j < M+1 \\
A[index, index] &= \frac{r_{i+0,5}^2 + r_{i-0,5}^2}{r_i^2 h^2} + \frac{\sin(\theta_{j+0,5}) + \sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} + V_{i,j} \\
A[index, index - 1] &= -\frac{\sin(\theta_{j-0,5})}{r_i^2 \sin(\theta_j) k^2} \\
A[index, index + 1] &= -\frac{\sin(\theta_{j+0,5})}{r_i^2 \sin(\theta_j) k^2} \\
A[index, index - M] &= -\frac{r_{i-0,5}^2}{r_i^2 h^2} \\
A[index, index + M] &= -\frac{r_{i+0,5}^2}{r_i^2 h^2}
\end{aligned}$$

Матричная задача на собственные значения решалась с помощью функции `linalg.eigs()` из модуля `scipy.sparse`, который реализует работу с разреженными матрицами, математической библиотеки `scipy`, функция `linalg.eigs()` реализует метод Арнольди для разреженных матриц.

3.2 Реализация графического интерфейса

Для дипломной работы был реализован графический интерфейс с помощью обертки библиотек графического интерфейса пользователя — `wxPython`, основой которого является API `wxWidgets`. Для построения графиков использовалась библиотека `matplotlib`.

Данная программа позволяет решать задачу для стационарного уравнения Шредингера, используя введенные пользователем параметры, с последующим построением трехмерных графиков волновых функций, как в присутствии донора в квантовой точке, так и без донора (Рисунок 2). Также есть возможность построения двумерных графиков зависимости энергии от магнитного поля, радиуса квантовой точки и количества узлов сетки (Рисунок 3).

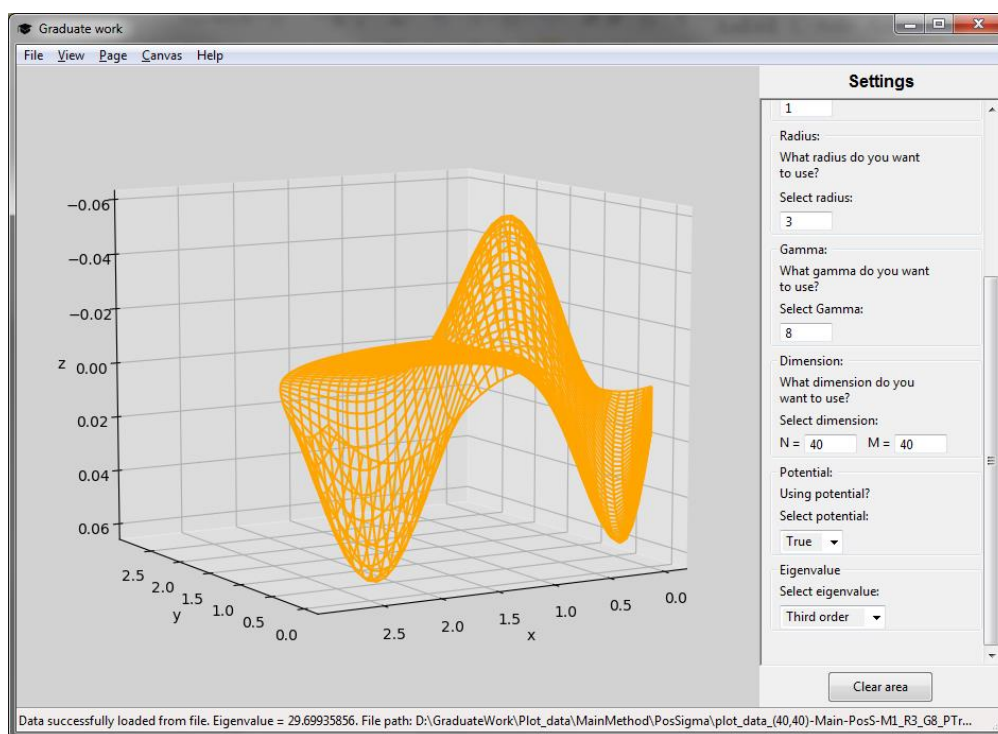


Рисунок 2 – Скриншот графического интерфейса для расчета состояния электрона, введения параметров, вывода графика соответствующей волновой функции

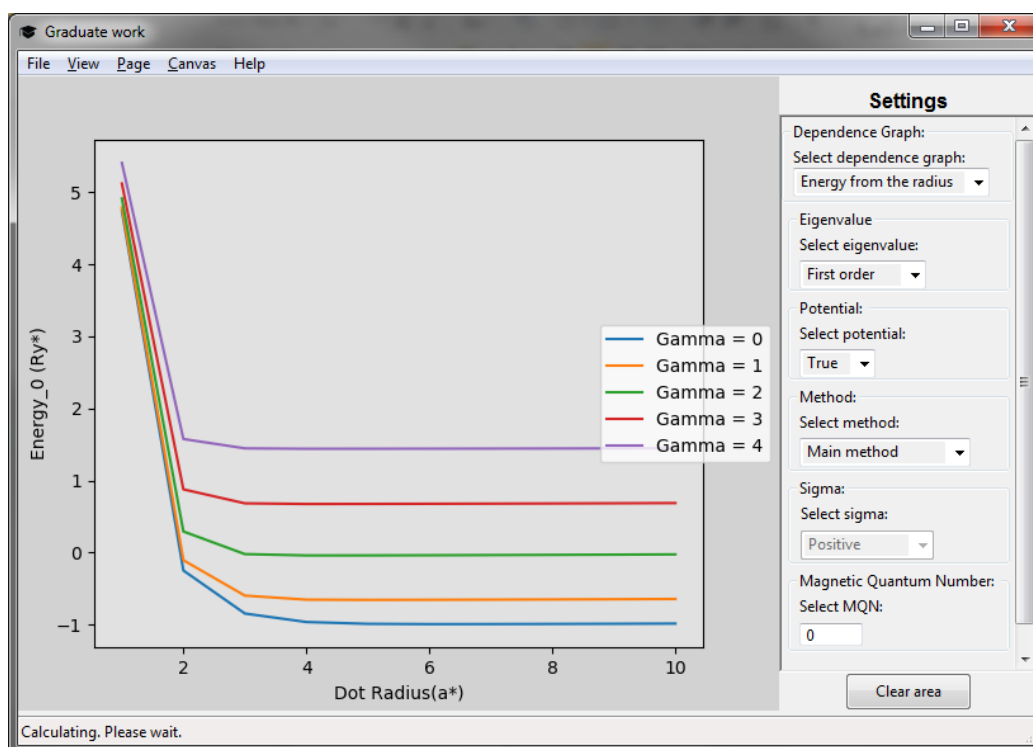


Рисунок 3 – Скриншот графического интерфейса для вывода результатов в виде графиков зависимостей энергии от параметров системы

В программе есть возможности сохранения результатов вычислений в файле формата JSON, их последующее открытие, а так же создание скриншотов

полученных графиков. Данный функционал доступен из меню представленного на Рисунке 4, выделенного красной областью.

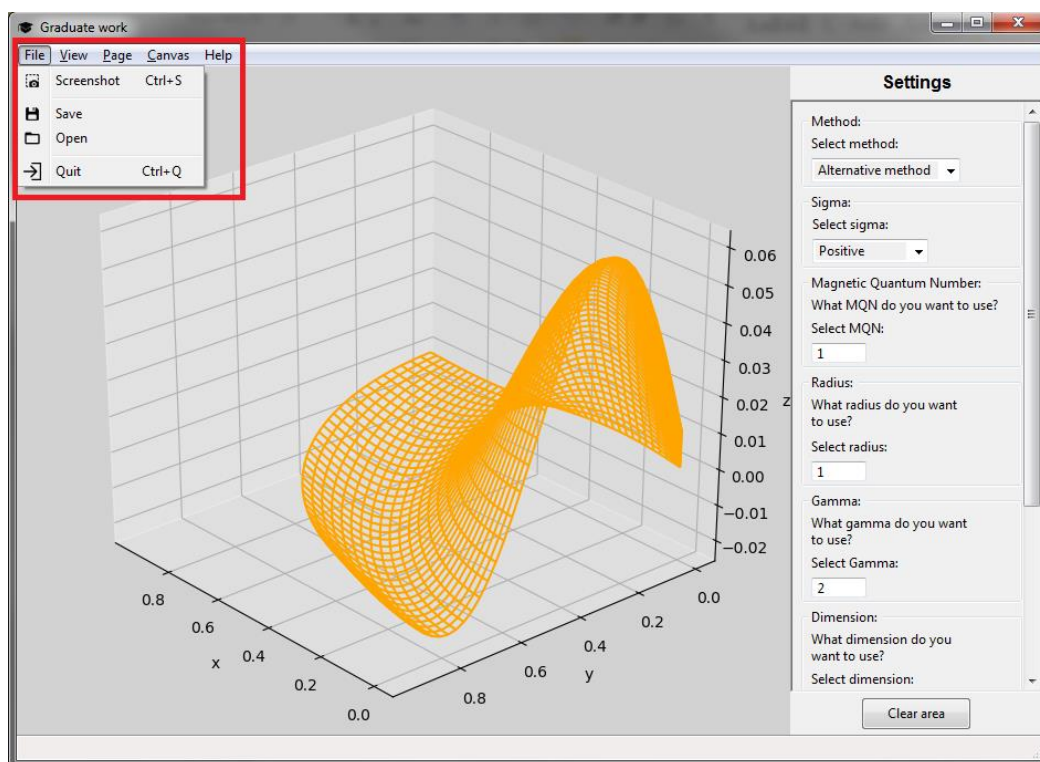


Рисунок 4 – Фрагмент меню для работы с выходными данными программы

4 Результаты численных экспериментов

В дальнейших вычислениях параметры отвечающие за количество узлов по θ и ρ , M и N соответственно, будут полагаться равными.

4.1 Энергии в зависимости от радиуса квантовой точки

На основании результатов численных расчетов были построены графики зависимости энергии от размера квантовой точки (Рисунок 5 - 6). Сравнив графики на рисунках 5-6 соответствующие основному состоянию и первому возбужденному состоянию, соответственно, видно, что энергия уменьшается с увеличением радиуса квантовой точки. Возбужденные состояния более чувствительны к размеру квантовой точки по сравнению с основным состоянием. Это связано с большей областью локализации волновой функции в возбужденном состоянии (см. Рисунок 7 - 9).

Из рисунков 5-6 можно сделать вывод, что уже при относительно небольших размерах квантовой точки (начиная с $R = 6a^*$) систему можно рассматривать как изолированный донор в полупроводнике. Начиная с этого радиуса квантовой точки энергия основного состояния близка к энергии водородоподобного атома, что позволяет использовать такую задачу как тестовую.

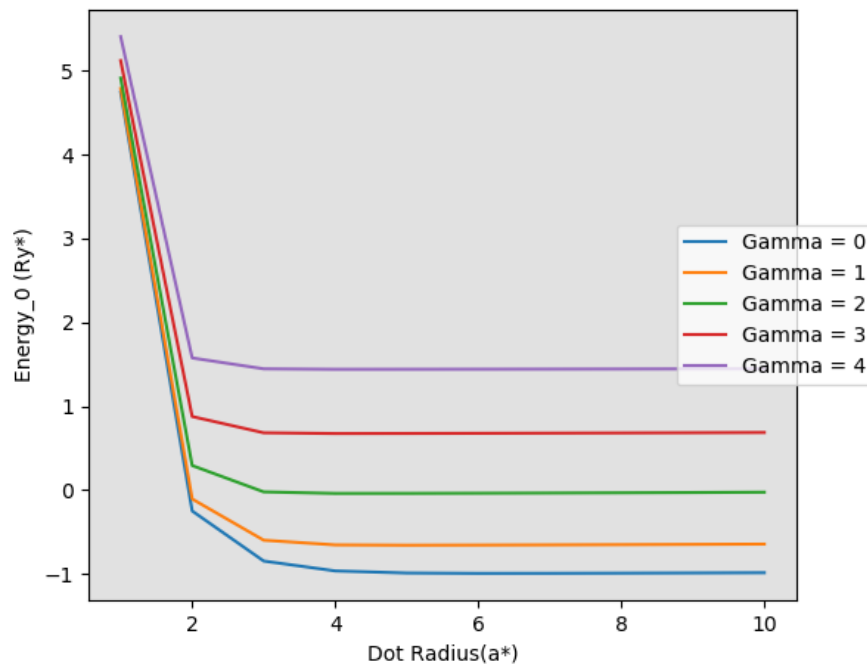


Рисунок 5 – График зависимости E_0 от радиуса квантовой точки для различных величин магнитного поля (Γ) при $\sigma = 1$, $m = 0$, в присутствии донора

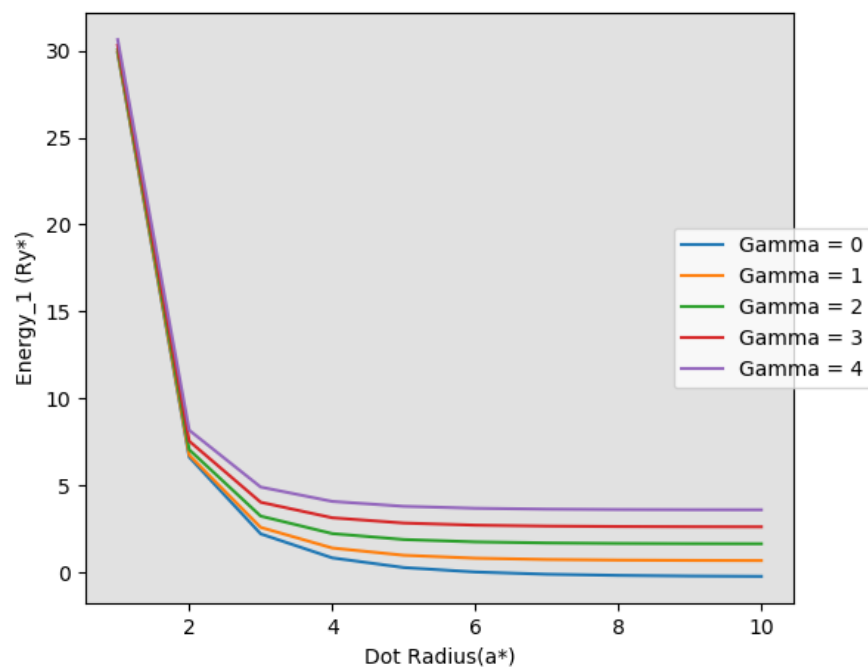


Рисунок 6 – График зависимости E_1 от радиуса квантовой точки для различных величин магнитного поля (Γ) при $\sigma = 1$, $m = 0$, в присутствии донора

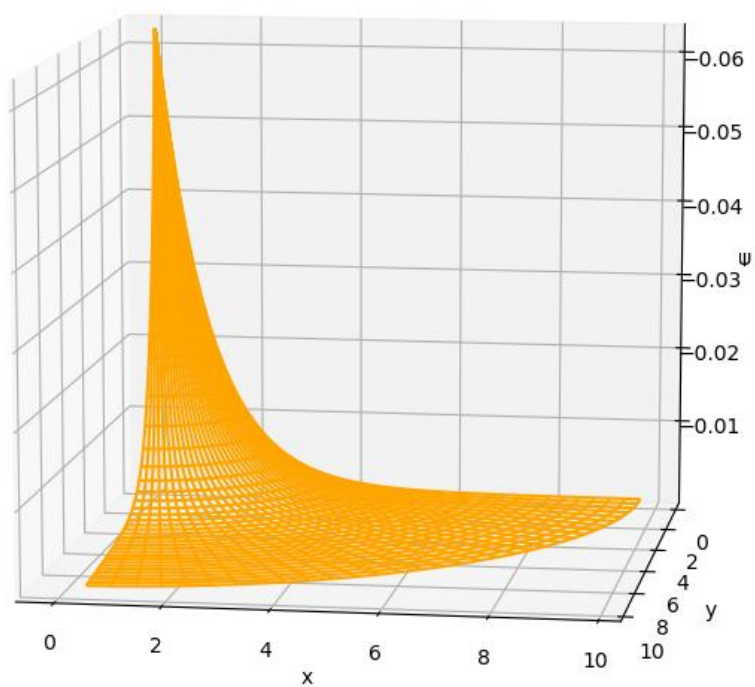


Рисунок 7 – График волновой функции E_0 , альтернативный метод с $m = 0$, $\sigma = 1$, $R = 10$, $\gamma = 0$, в присутствии донора

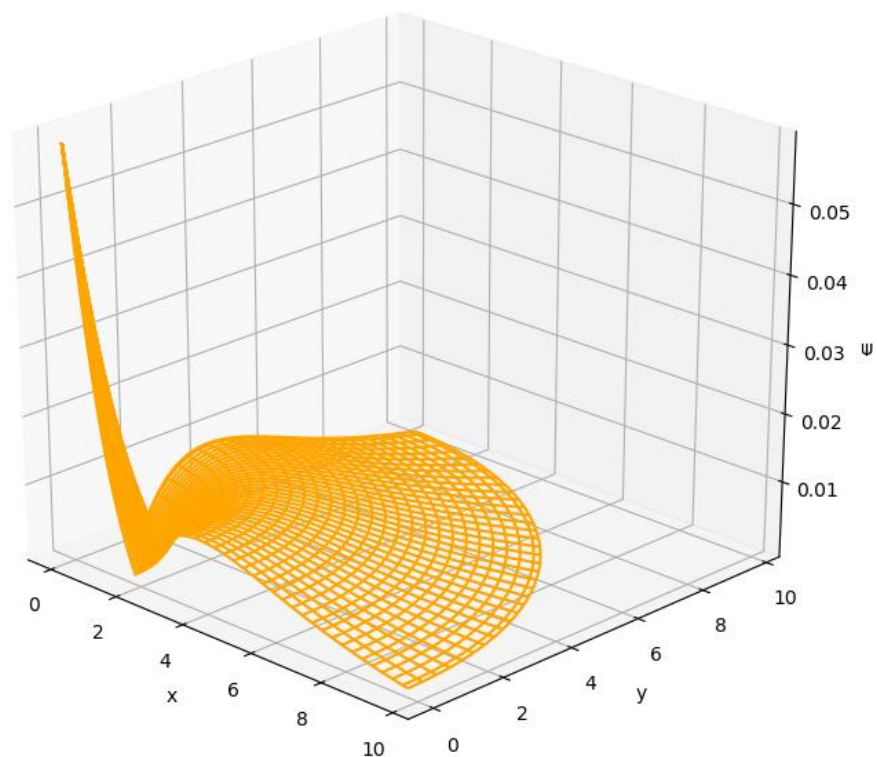


Рисунок 8 - График модуля волновой E_1 , альтернативный метод с $m = 0$, $\sigma = 1$, $R = 10$, $\gamma = 0$, в присутствии донора.

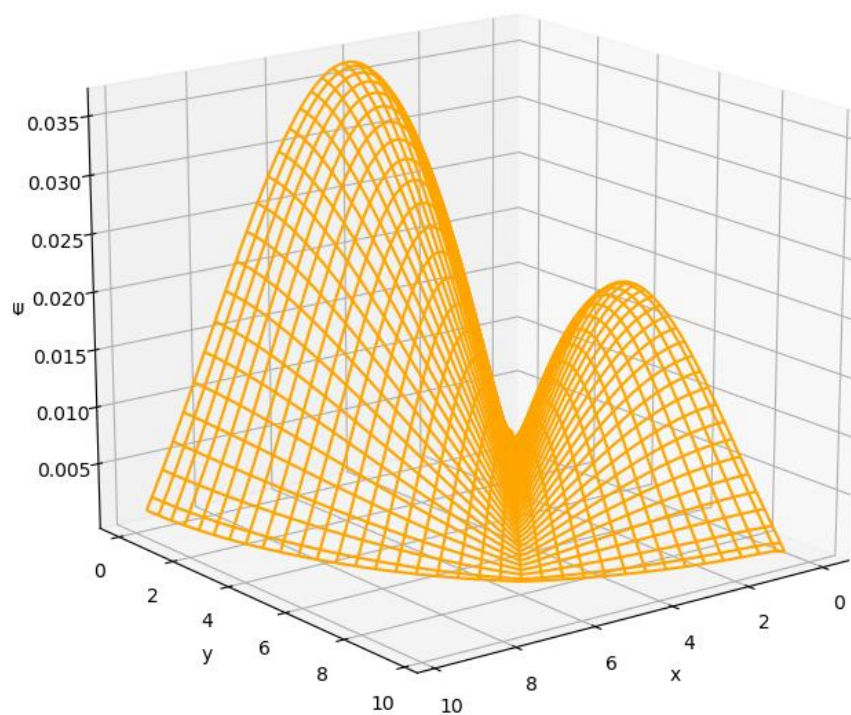


Рисунок 9 – График модуля волновой функции E_2 , альтернативный метод с $m = 0$, $\sigma = 1$, $R = 10$, $\gamma = 0$, в присутствии донора

Одна из важных характеристик для практического применения рассматриваемой системы является энергия перехода. Энергия перехода – это разность между энергиями двух различных состояний, например из состояния $m = 1, \sigma = 1$ в состояние $m = 0, \sigma = 1$. На рисунках 10 – 11 приведены графики зависимостей энергий перехода из возбужденных состояний с $m = \pm 1$ в основное состояние. На рисунках 10-11 можно заметить, что энергии перехода существенно уменьшаются при увеличении радиуса квантовой точки.

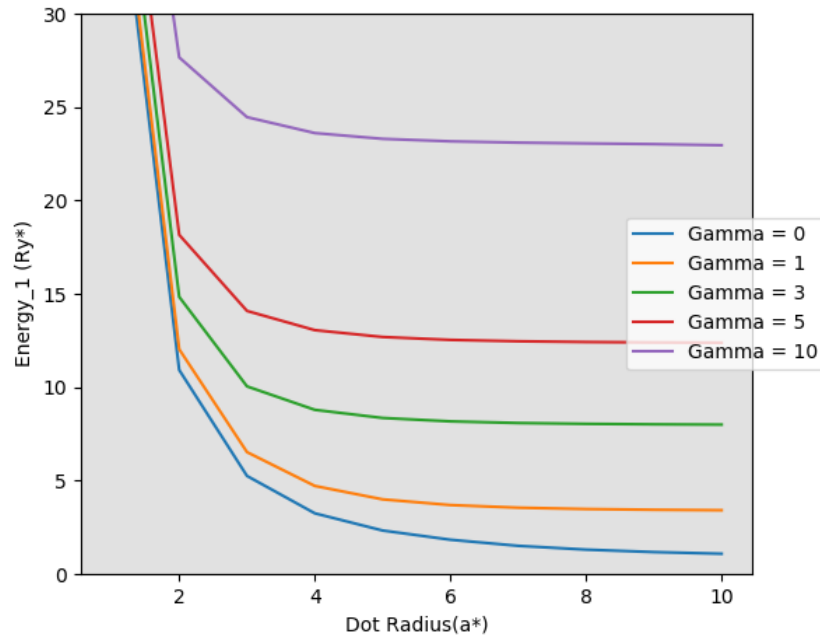


Рисунок 10 – График зависимости энергии перехода от радиуса квантовой точки для различных величин магнитного поля (Γ) при $\sigma = 1, m = 1$, в присутствии донора

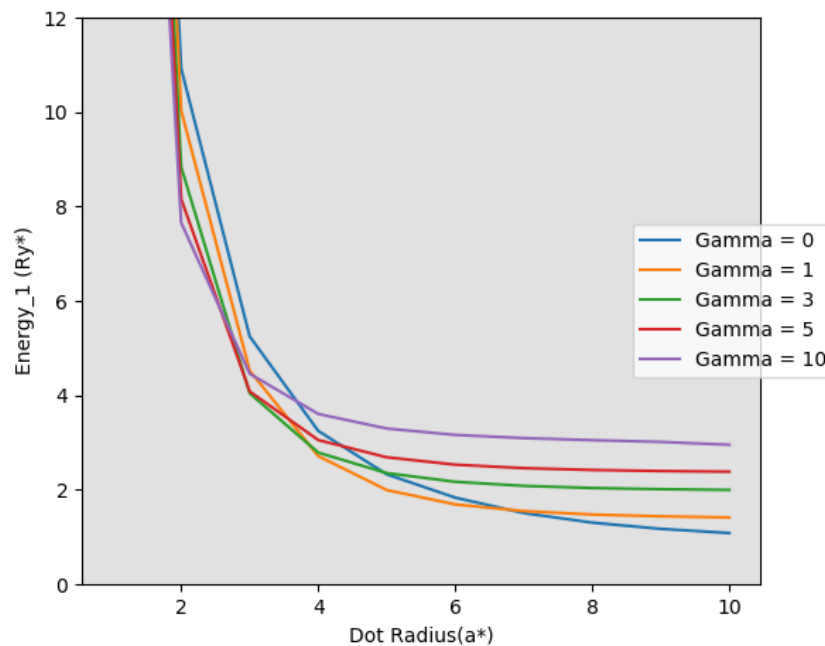


Рисунок 11 – График зависимости энергии перехода от радиуса квантовой точки для различных величин магнитного поля (Γ) при $\sigma = 1, m = -1$, в присутствии донора

4.2 Энергии в зависимости от величины магнитного поля

Исследуем влияние величин магнитного поля на энергии квантовых точек. На рисунках 12-13 приведен график зависимости энергии основного состояния от магнитного поля при различных размерах квантовой точки. Заметим, что при увеличении величины магнитного поля энергия квантовой точки существенно увеличивается.

Также магнитное поле можно использовать для изменения области локализации электрона. Увеличение напряженности магнитного поля, направленного по оси Oz, заставляет волновую функцию электрона сжиматься в плоскости Oxy, что означает уменьшению области локализации электрона внутри квантовой точки (см. Рисунки 14-15).

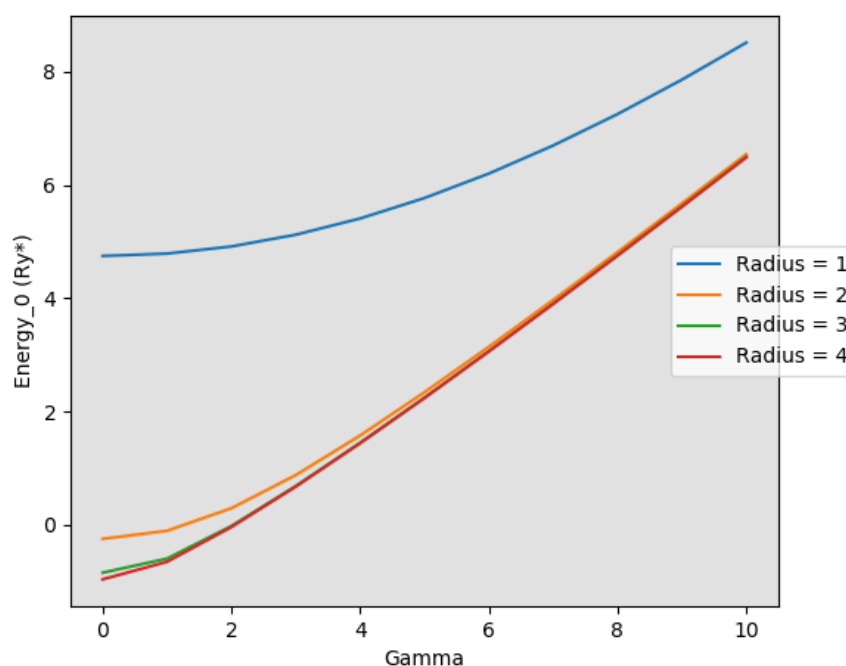


Рисунок 12 – График зависимости E_0 от величины магнитного поля (Γ) для различных радиусов квантовых точек при $\sigma = 1$, $m = 0$, в присутствии донора

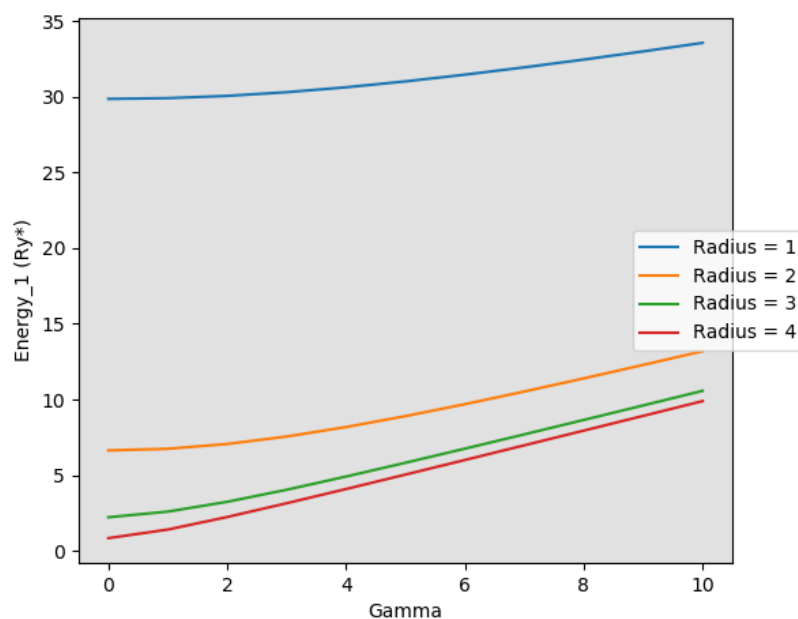


Рисунок 13 – График зависимости E_1 от величины магнитного поля (Gamma) для различных радиусов квантовых точек при $\sigma = 1$, $m = 0$, в присутствии донора

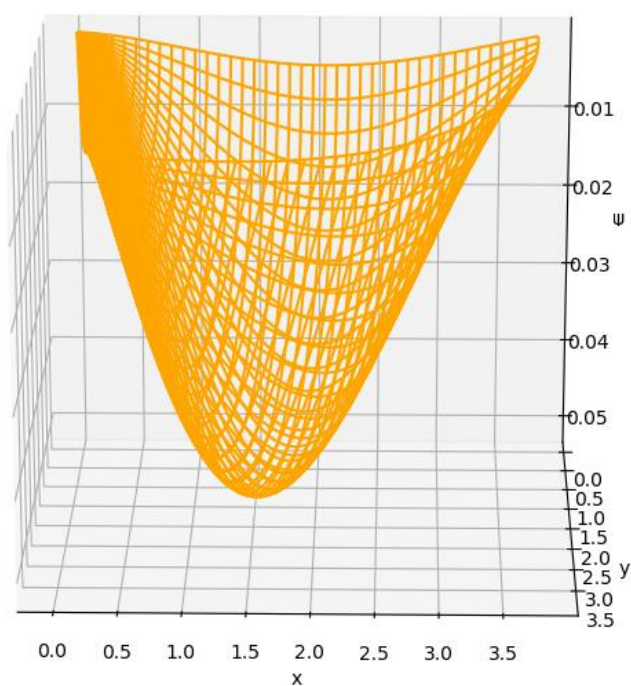


Рисунок 14 – График волновой функции E_0 , альтернативный метод с $m = 1$, $\sigma = -1$, $R = 4$, $\gamma = 0$, в присутствии донора

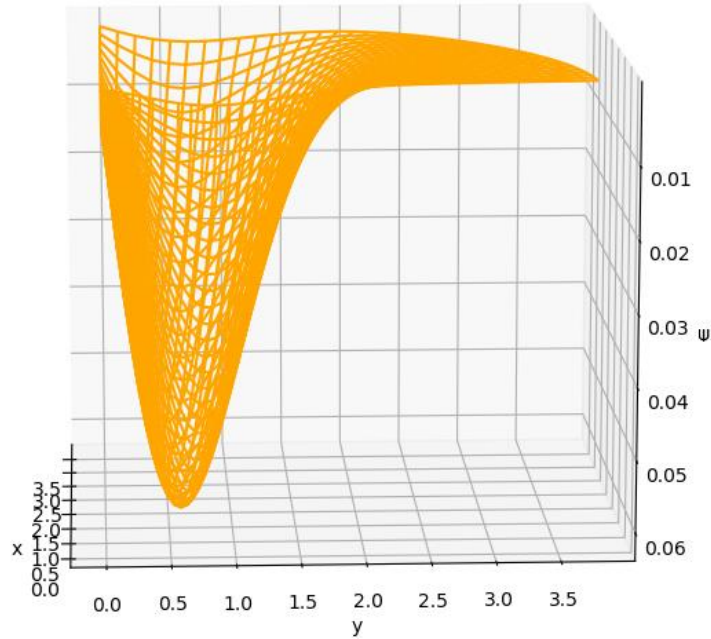


Рисунок 15 - График волновой функции E_0 , альтернативный метод с $m = 1$, $\sigma = -1$, $R = 4$, $\gamma = 4$, в присутствии донора

Еще одним эффектом влияния магнитного поля на энергетический спектр, является расщепление электронных уровней в магнитном поле. В частности на рисунке 16 показано такое расщепление для состояний E_0 , $m=\pm 1$, и E_1 , $m=0$.

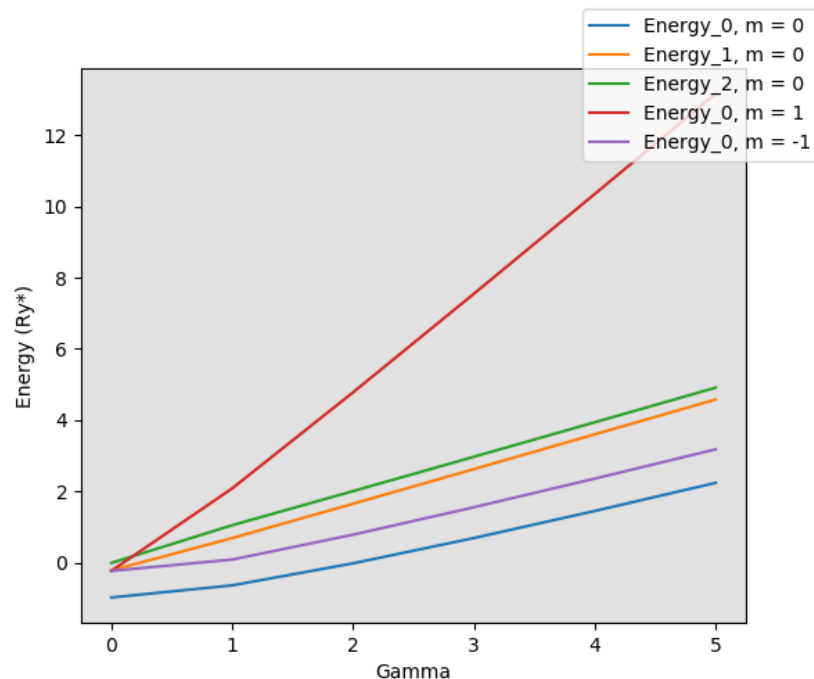


Рисунок 16 – График зависимости различных энергии от величины магнитного поля (Gamma) для различных радиусов квантовых точек, $\sigma = 1$, $R = 10$, в присутствии донора

4.3 Сравнение схем

4.3.1 Случай отсутствия донора

Для оценки погрешности рассмотрим случай, когда донор отсутствует. В этом случае система представляет собой сферически симметричный потенциальный ящик с бесконечными стенками [6]. Для этой системы известно точное аналитическое решение. В частности, энергия основного состояния в такой системе равна π^2/R^2 . В таком случае сравним полученные нами результаты с точным аналитическим решением. В таблице 1 приведены абсолютные и относительные погрешности для основного состояния в случае отсутствия донора.

График зависимости (Рисунок 17) определяет, что метод из статьи [5] дает оценку энергии снизу, альтернативный же метод дает оценку сверху.

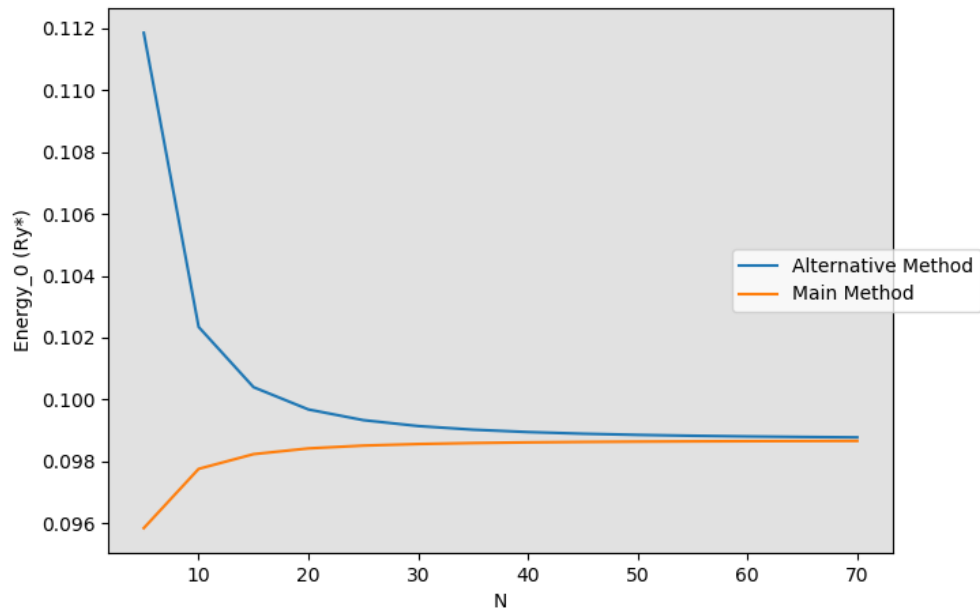


Рисунок 17 – График зависимости E_0 от размера сетки при $\sigma = 1$, $\gamma = 0$, $m = 0$ R = 10 и отсутствии донора

С помощью метода экстраполяции (метод ускорения сходимости решения разностной задачи) можно получить приближенное значение энергии при $N \rightarrow \infty$. На рисунке 18 приведен график зависимость энергии от шагов сетки. На графике видно, что зависимость является квадратичной, что позволяет представить приближенное значение энергии в виде $\widetilde{E}_0 = E_0 + C/N^2$. Полученные при этом значения E_0 для метода из статьи [5] и альтернативного

метода равны 0.09869, 0.0987 соответственно. Абсолютные погрешности разностных схем не превосходит 10^{-5} .

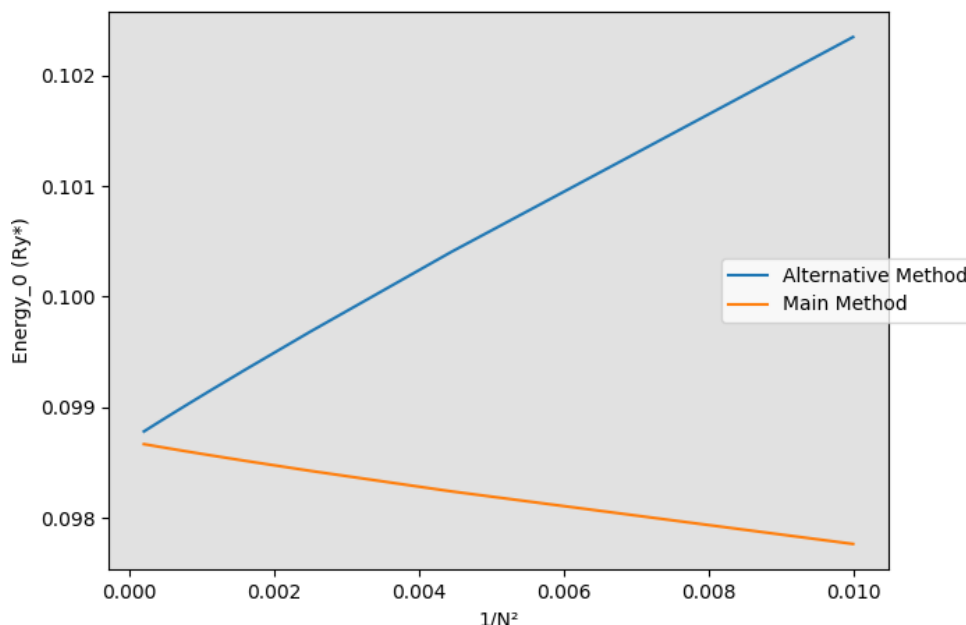


Рисунок 18 – График зависимости E_0 от величины $1/N^2$ при $\sigma = 1$, $\gamma = 0$, $m = 0$ R = 10 и отсутствии донора

Таблица 1 – Абсолютные и относительные погрешности в случае отсутствия донора при $N=M=40$

Основное состояние (E_0)			
Метод из статьи [5]		Альтернативный метод	
Abs =	0.000029208696858268324	Abs =	0.00008572487941402307
Rel =	0.00029603358378023706	Rel =	0.000867820857806428

4.3.2 Случай присутствия донора

В отличие от случая без донора, в присутствии донора оба метода дают оценку для энергии сверху (см. Рисунок 19-20). Также заметим, что из-за наличия особенности в точке расположения донора зависимость погрешности энергии от шага сетки не является квадратичной (см. Рисунок 21).

В разделе 4.1 отмечалось, что при радиусе большем чем $6a^*$ энергия приближается к энергии водородоподобного атома, для которого известны точные значения энергий. В трех низших состояниях эти энергии равны -1, -1/4, -1/9 [5]. Это позволяет сравнить точные значения с результатами полученными нами. В таблице 2 приведены абсолютные и относительные погрешности для 3 состояний в присутствии донора.

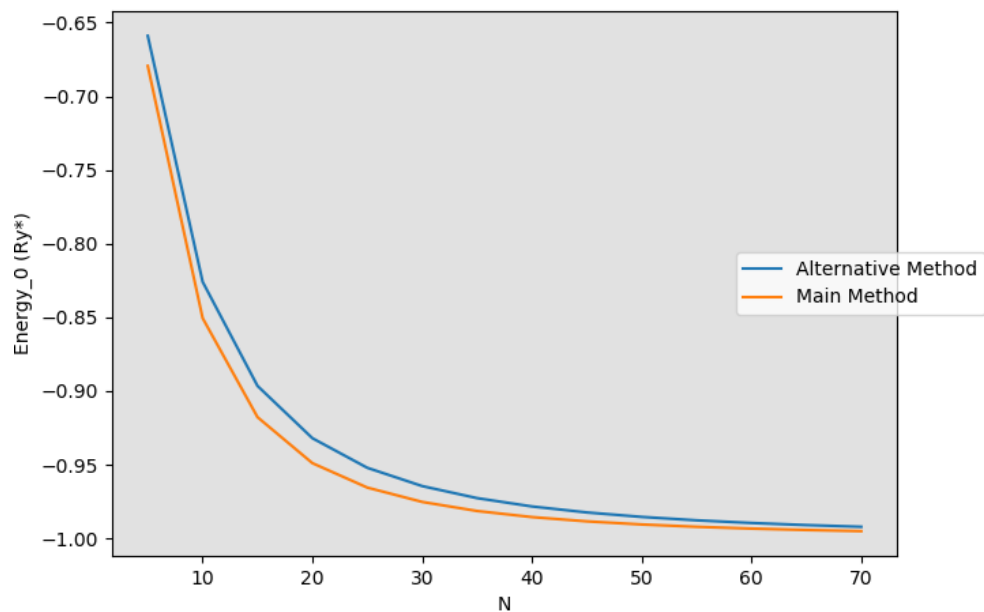


Рисунок 19 - График зависимости E_0 от размера сетки при $\sigma = 1$, $\gamma = 0$, $m = 0$ $R = 10$, в присутствии донора

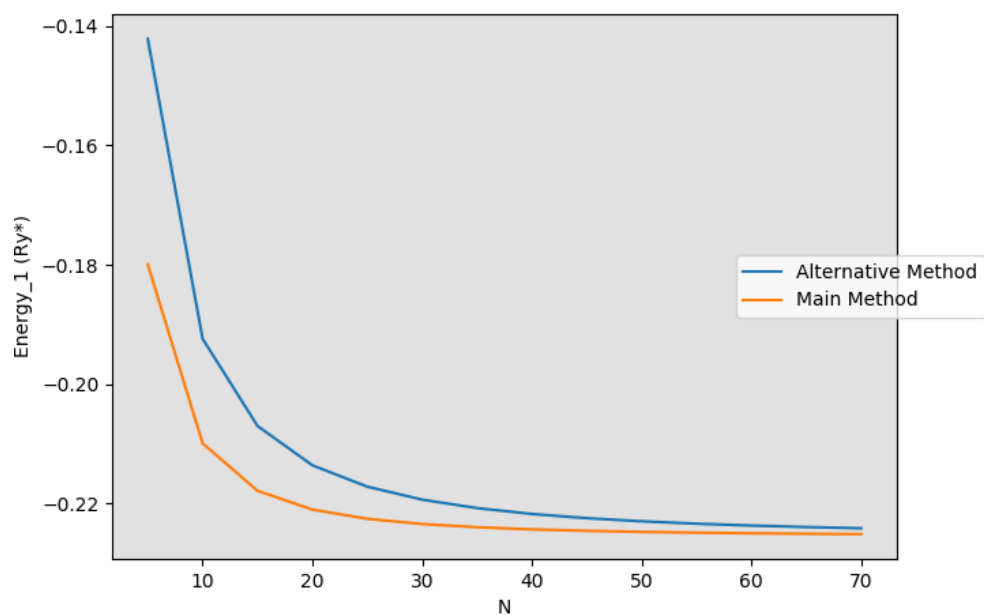


Рисунок 20 - График зависимости E_1 от размера сетки при $\sigma = 1$, $\gamma = 0$, $m=0$ $R = 10$, в присутствии донора

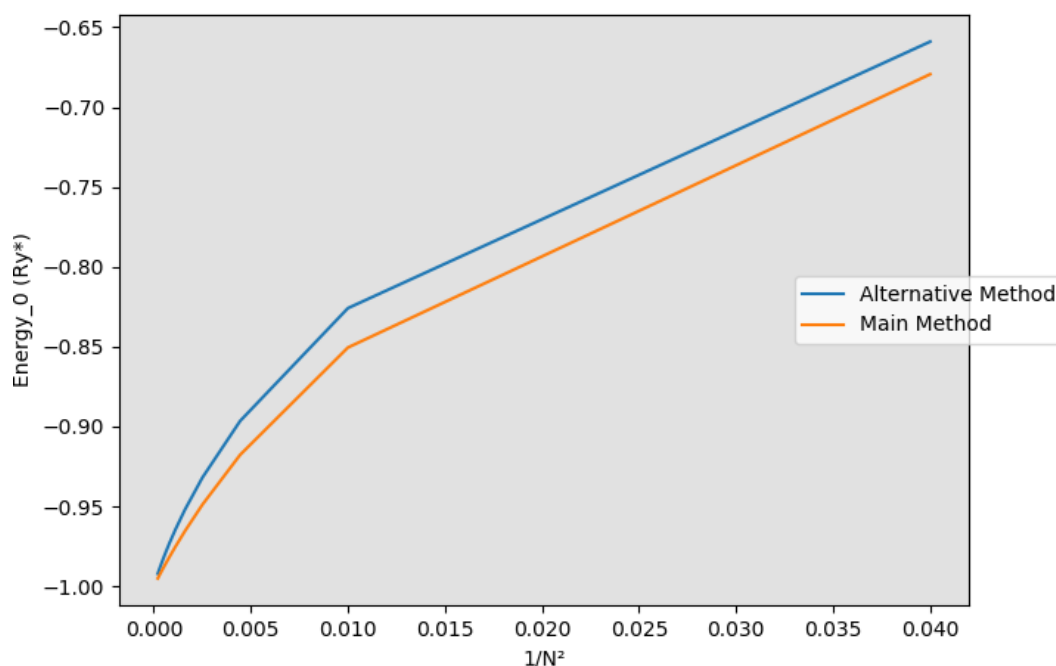


Рисунок 21 – График зависимости E_0 от величины $1/N^2$ при $m = 0, \gamma = 0, r = 10$, в присутствии донора

Таблица 2 – Абсолютные и относительные погрешности в случае присутствия донора при $M=N=40$

Основное состояние (E_0)			
Метод из статьи [5]		Альтернативный метод	
Abs =	0.0049122993526588	Abs =	0.007859748624066576
Rel =	0.004936549159901352	Rel =	0.00792201366003084
Первое возбужденное состояние (E_1)			
Метод из статьи [5]		Альтернативный метод	
Abs =	0.024787192475118708	Abs =	0.025773259896790773
Rel =	0.11006120276876458	Rel =	0.11494284707045918
Второе возбужденное состояние (E_2)			
Метод из статьи [5]		Альтернативный метод	
Abs =	0.09671921330337042	Abs =	0.09703873411779726
Rel =	6.720393279290104	Rel =	6.89568892049317

ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен конечно-разностный расчет уровней энергии донора в сферически симметричной квантовой точке. В ходе выполнения дипломной работы предложена разностная схема решения задачи для стационарного уравнения Шредингера. Было разработано приложение под Windows реализующее две разностные схемы: предложенную в работе и рассмотренную в статье [5]. С помощью разработанного приложения рассчитаны зависимости энергий основного и возбужденных состояний от величины магнитного поля, от размера квантовой точки и от количества узлов сетки. Получены абсолютные и относительные погрешности энергий системы при наличии донора и при его отсутствии.

СПИСОК ЛИТЕРАТУРЫ

1. Level Anticrossing of Impurity States in Semiconductor Nanocrystals / A. S. Baimuratov [et. al.] // Scientific reports. – 2014. – Vol. 4. – P 6917.
2. Chuu, D. S. Hydrogenic impurity states in quantum dots and quantum wires / Chuu, D. S., Hsiao, C. M., Mei, W. N. // Phys. Rev. B. – 1992. – Vol. 46, P. 3898–3905.
3. Aktas, S. The binding energy of hydrogenic impurity in multilayered spherical quantum dot / Aktas, S. Boz, F. K. // Physica E. – 2008. – Vol. 40, P. 753–758.
4. Ярив, А. Введение в теорию и приложения квантовой механики / А. Ярив. – М.: Мир, 1984. – 360 с.
5. Souza, G.V.B. Finite-difference calculation of donor energy levels in a spherical quantum dot subject to a magnetic field / G.V.B. de Souza, A. Bruno-Alfonso // Physica E. – 2015. – Vol. 66. – P. 128-132.
6. Флюгге З. Задачи по квантовой механике / Флюгге З. – М.: Мир, 1974. – 339 с.
7. Самарский, А.А. Теория разностных схем / А.А. Самарский. – М.: Наука, 1989. – 616 с.
8. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М.: Физматлит: Лаборатория базовых знаний, 2006. – 630 с.

ПРИЛОЖЕНИЕ А

Листинг модуля программы, реализующий разностную схему

```
import numpy as np
from numpy import cos, tan, sin
from scipy.sparse import csr_matrix, linalg
from wx.lib.pubsub import pub

def bubble_sort(val, vector):
    index = val.size - 1
    while index >= 0:
        for j in range(index):
            if val[j] > val[j+1]:
                val[j], val[j+1] = val[j+1], val[j]
                for i in range(int(vector.size/val.size)):
                    vector[i, j], vector[i, j+1] = vector[i, j+1], vector[i, j]
        index -= 1
    return val, vector

def index_transform(i, j, m_dimension):
    return (i - 2)*m_dimension + j - 2

def V_ij(i, j, mass, gamma, h, k, potential):
    if 'True' in str(potential):
        return mass*gamma + mass**2/((r_i(i, h))**2*(sin(o_j(j, k)))**2) +
        gamma**2*(r_i(i,h))**2*(sin(o_j(j,k)))**2/4 - 2/r_i(i,h)
    else:
        return 0

def r_i(i,h):
    return (i-1)*h

def o_j(j, k):
    return (j-1)*k

def cotang(item):
    return 1/tan(item)

def cart2pol(x, y):
    rho, theta = np.meshgrid(x, y)
    x = rho*cos(theta)
    y = rho*sin(theta)
    return x, y

def Calculate_loop(method, mass, radius, n_dimension, m_dimension, gamma,
potential, sigma, start_time):
    mass = float(mass)
    R = float(radius)
    gamma = float(gamma)
    N = n_dimension
    M = m_dimension
    h = R / (N + 1.)
    k = np.pi / (2 * (M + 1.))
    b = csr_matrix((N*M, N*M))

    if method == 'Main method':
        if not int(mass) == 0:
            if sigma == 'Positive':
                main_method_nonzero_MQN_PosSigma(b, mass, N, M, h, k, gamma,
potential)
            else:
```

```

        main_method_nonzero_MQN_NegSigma(b, mass, N, M, h, k, gamma,
potential)
    else:
        main_method_zero_MQN_PosSigma(b, mass, N, M, h, k, gamma, potential)
else:
    if not int(mass) == 0:
        if sigma == 'Positive':
            alternative_method_nonzero_MQN_PosSigma(b, mass, N, M, h, k,
gamma, potential)
        else:
            alternative_method_nonzero_MQN_NegSigma(b, mass, N, M, h, k,
gamma, potential)
        else:
            alternative_method_zero_MQN_PosSigma(b, mass, N, M, h, k, gamma,
potential)

    val, vector = linalg.eigs(b, k=3, which='SR')
    val, vector = bubble_sort(val.real, vector.real)
    sendingMessage(val=np.ndarray.tolist(val),
vector=np.ndarray.tolist(vector.transpose()), time=start_time)

def sendingMessage(val, vector, time):
    pub.sendMessage('calculatedTransfer', val=val, vector=vector, time=time)

def main_method_nonzero_MQN_NegSigma(b, mass, N, M, h, k, gamma, potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, M)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == 2 and i != 2 and i != N + 1:
                # j = 2, 2 < i < N + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == M + 1 and i != 2 and i != N + 1:
                # j = M + 1, 2 < i < N + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == 2 and j != 2 and j != M + 1:
                # i = 2, 2 < j < M + 1
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))

```



```

        elif i == N + 1 and j != 2 and j != M + 1:
            # i = N+1, 2 < j < M+1
            b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
            b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
            cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
            b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
            V_ij(i, j, mass, gamma, h, k, potential)
            b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
            cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
            elif i == 2 and j == 2:
                # i = 2, j = 2
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == 2 and j == M + 1:
                # i = 2, j = M+1
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == N + 1 and j == 2:
                # i = N + 1, j = 2
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
            elif i == N + 1 and j == M + 1:
                # i = N + 1, j = M + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)

def main_method_zero_MQN_PosSigma(b, mass, N, M, h, k, gamma, potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, N)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M+1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == 2 and i != 2 and i != N + 1:
                # j = 2, 2 < i < N+1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
                4 / (3 * k ** 2 * (r_i(i, h)) ** 2) + (4 * cotang(o_j(j, k))) / (3 * 2 * k * (r_i(i,
                h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3 * k ** 2 * (r_i(i, h)) **
                2) - cotang(o_j(j, k)) / (3 * 2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == M + 1 and i != 2 and i != N + 1:

```

```

# j = M+1, 2 < i < N+1
b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
b[index, index - 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) + cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) - (4*cotang(o_j(j, k))) / (2*3 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
elif i == 2 and j != 2 and j != M + 1:
# i = 2, 2 < j < M+1
b[index, index - 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
elif i == N + 1 and j != 2 and j != M + 1:
# i = N+1, 2 < j < M+1
b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
b[index, index - 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
elif i == 2 and j == 2:
# i = 2, j = 2
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) + (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) - cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
elif i == 2 and j == M + 1:
# i = 2, j = M+1
b[index, index - 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) + cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) - (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
elif i == N + 1 and j == 2:
# i = N + 1, j = 2
b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) + (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
b[index, index + 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) - cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
elif i == N + 1 and j == M + 1:
# i = N + 1, j = M + 1
b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
b[index, index - 1] = - 1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) + cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) - (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)

```

```

def main_method_nonzero_MQN_PosSigma(b, mass, N, M, h, k, gamma, potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, N)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == 2 and i != 2 and i != N + 1:
                # j = 2, 2 < i < N + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif j == M + 1 and i != 2 and i != N + 1:
                # j = M + 1, 2 < i < N + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3 * k ** 2 * (r_i(i, h)) **
                2) + cotang(o_j(j, k)) / (3 * 2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
                4 / (3 * k ** 2 * (r_i(i, h)) ** 2) - (4 * cotang(o_j(j, k))) / (3 * 2 * k * (r_i(i,
                h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == 2 and j != 2 and j != M + 1:
                # i = 2, 2 < j < M + 1
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == N + 1 and j != 2 and j != M + 1:
                # i = N + 1, 2 < j < M + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
            elif i == 2 and j == 2:
                # i = 2, j = 2
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
                V_ij(i, j, mass, gamma, h, k, potential)
                b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
                b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
            elif i == 2 and j == M + 1:
                # i = 2, j = M + 1
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
                cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3 * k ** 2 * (r_i(i, h)) **
                2) + cotang(o_j(j, k)) / (3 * 2 * k * (r_i(i, h)) ** 2)

```

```

        b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) - (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
        b[index, index + M] = -1 / h ** 2 - 1 / (h * r_i(i, h))
        elif i == N + 1 and j == 2:
            # i = N + 1, j = 2
            b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
            b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) +
V_ij(i, j, mass, gamma, h, k, potential)
            b[index, index + 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) -
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2)
            elif i == N + 1 and j == M + 1:
                # i = N + 1, j = M + 1
                b[index, index - M] = -1 / h ** 2 + 1 / (h * r_i(i, h))
                b[index, index - 1] = -1 / (k ** 2 * (r_i(i, h)) ** 2) +
cotang(o_j(j, k)) / (2 * k * (r_i(i, h)) ** 2) + 1 / (3*k ** 2 * (r_i(i, h)) **
2) + cotang(o_j(j, k)) / (3*2 * k * (r_i(i, h)) ** 2)
                b[index, index] = 2 / h ** 2 + 2 / (k ** 2 * (r_i(i, h)) ** 2) -
4 / (3*k ** 2 * (r_i(i, h)) ** 2) - (4*cotang(o_j(j, k))) / (3*2 * k * (r_i(i,
h)) ** 2) + V_ij(i, j, mass, gamma, h, k, potential)

def alternative_method_nonzero_MQN_NegSigma(b, mass, N, M, h, k, gamma,
potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, N)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M + 1
                b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
                b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                elif j == 2 and i != 2 and i != N + 1:
                    # j = 2, 2 < i < N + 1
                    b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
                    b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                    b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                    b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                    elif j == M + 1 and i != 2 and i != N + 1:
                        # j = M + 1, 2 < i < N + 1
                        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
                        b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *

```

```

sin(o_j(j, k)) * k ** 2)
    elif i == 2 and j != 2 and j != M + 1:
        # i = 2, 2 < j < M+1
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4*r_i(i + 0.5, h) ** 2) /
(3*r_i(i, h) ** 2 * h ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + r_i(i + 0.5, h) ** 2 / (3*r_i(i, h) ** 2 * h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == N + 1 and j != 2 and j != M + 1:
        # i = N+1, 2 < j < M+1
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == 2 and j == 2:
        # i = 2, j = 2
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4*r_i(i - 0.5, h) ** 2) /
(3*r_i(i, h) ** 2 * h ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
        b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + r_i(i - 0.5, h) ** 2 / (3*r_i(i, h) ** 2 * h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == 2 and j == M + 1:
        # i = 2, j = M+1
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4*r_i(i + 0.5, h) ** 2) /
(3*r_i(i, h) ** 2 * h ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + r_i(i + 0.5, h) ** 2 / (3*r_i(i, h) ** 2 * h ** 2)
        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == N + 1 and j == 2:
        # i = N + 1, j = 2
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h,
k, potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == N + 1 and j == M + 1:
        # i = N + 1, j = M + 1
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)

```

```

        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)

def alternative_method_zero_MQN_PosSigma(b, mass, N, M, h, k, gamma, potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, N)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M + 1
                b[index, index] = (r_i(i+0.5, h)**2+r_i(i-
0.5, h)**2)/(r_i(i, h)**2*h**2) + (sin(o_j(j+0.5, k))+sin(o_j(j-
0.5, k)))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) + V_ij(i, j, mass, gamma, h,
k, potential)
                b[index, index + M] = - r_i(i+0.5, h)**2/(r_i(i, h)**2*h**2)
                b[index, index - M] = - r_i(i-0.5, h)**2/(r_i(i, h)**2*h**2)
                b[index, index + 1] = -
sin(o_j(j+0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2)
                b[index, index - 1] = - sin(o_j(j-
0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2)
                elif j == 2 and i != 2 and i != N + 1:
                    # j = 2, 2 < i < N + 1
                    b[index, index] = (r_i(i+0.5, h)**2+r_i(i-
0.5, h)**2)/(r_i(i, h)**2*h**2) + (sin(o_j(j+0.5, k))+sin(o_j(j-
0.5, k)))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) - (4*sin(o_j(j-
0.5, k)))/(3*r_i(i, h)**2*sin(o_j(j, k))*k**2) + V_ij(i, j, mass, gamma, h,
k, potential)
                    b[index, index + M] = - r_i(i+0.5, h)**2/(r_i(i, h)**2*h**2)
                    b[index, index - M] = - r_i(i-0.5, h)**2/(r_i(i, h)**2*h**2)
                    b[index, index + 1] = -
sin(o_j(j+0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) + sin(o_j(j-
0.5, k))/(3*r_i(i, h)**2*sin(o_j(j, k))*k**2)
                    elif j == M + 1 and i != 2 and i != N + 1:
                        # j = M + 1, 2 < i < N + 1
                        b[index, index] = (r_i(i+0.5, h)**2+r_i(i-
0.5, h)**2)/(r_i(i, h)**2*h**2) + (sin(o_j(j+0.5, k))+sin(o_j(j-
0.5, k)))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) -
(4*sin(o_j(j+0.5, k)))/(3*r_i(i, h)**2*sin(o_j(j, k))*k**2) + V_ij(i, j, mass,
gamma, h, k, potential)
                        b[index, index + M] = - r_i(i+0.5, h)**2/(r_i(i, h)**2*h**2)
                        b[index, index - M] = - r_i(i-0.5, h)**2/(r_i(i, h)**2*h**2)
                        b[index, index - 1] = - sin(o_j(j-
0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) +
sin(o_j(j+0.5, k))/(3*r_i(i, h)**2*sin(o_j(j, k))*k**2)
                        elif i == 2 and j != 2 and j != M + 1:
                            # i = 2, 2 < j < M + 1
                            b[index, index] = (r_i(i+0.5, h)**2+r_i(i-
0.5, h)**2)/(r_i(i, h)**2*h**2) + (sin(o_j(j+0.5, k))+sin(o_j(j-
0.5, k)))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) - (4*r_i(i-
0.5, h)**2)/(3*r_i(i, h)**2*h**2) + V_ij(i, j, mass, gamma, h, k, potential)
                            b[index, index + M] = - r_i(i+0.5, h)**2/(r_i(i, h)**2*h**2) +
r_i(i-0.5, h)**2/(3*r_i(i, h)**2*h**2)
                            b[index, index + 1] = -
sin(o_j(j+0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2)
                            b[index, index - 1] = - sin(o_j(j-
0.5, k))/(r_i(i, h)**2*sin(o_j(j, k))*k**2)
                            elif i == N + 1 and j != 2 and j != M + 1:
                                # i = N + 1, 2 < j < M + 1
                                b[index, index] = (r_i(i+0.5, h)**2+r_i(i-
0.5, h)**2)/(r_i(i, h)**2*h**2) + (sin(o_j(j+0.5, k))+sin(o_j(j-
0.5, k)))/(r_i(i, h)**2*sin(o_j(j, k))*k**2) + V_ij(i, j, mass, gamma, h,
k, potential)
                                b[index, index - M] = - r_i(i-0.5, h)**2/(r_i(i, h)**2*h**2)
                                b[index, index + 1] = -

```

```

sin(o_j(j+0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2)
    b[index, index - 1] = - sin(o_j(j-
0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2)
    elif i == 2 and j == 2:
        # i = 2, j = 2
        b[index, index] = (r_i(i+0.5,h)**2+r_i(i-
0.5,h)**2)/(r_i(i,h)**2*h**2) + (sin(o_j(j+0.5,k))+sin(o_j(j-
0.5,k)))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) - (4*r_i(i-
0.5,h)**2)/(3*r_i(i,h)**2*h**2) - (4*sin(o_j(j-
0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2) + V_ij(i, j, mass, gamma, h,
k,potential)
        b[index, index + M] = - r_i(i+0.5,h)**2/(r_i(i,h)**2*h**2) +
(r_i(i-0.5,h)**2)/(3*r_i(i,h)**2*h**2)
        b[index, index + 1] = -
sin(o_j(j+0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) + (sin(o_j(j-
0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2)
        elif i == 2 and j == M + 1:
            # i = 2, j = M+1
            b[index, index] = (r_i(i+0.5,h)**2+r_i(i-
0.5,h)**2)/(r_i(i,h)**2*h**2) + (sin(o_j(j+0.5,k))+sin(o_j(j-
0.5,k)))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) - (4*r_i(i-
0.5,h)**2)/(3*r_i(i,h)**2*h**2) -
(4*sin(o_j(j+0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2) + V_ij(i, j, mass,
gamma, h, k,potential)
            b[index, index + M] = - r_i(i+0.5,h)**2/(r_i(i,h)**2*h**2) +
(r_i(i-0.5,h)**2)/(3*r_i(i,h)**2*h**2)
            b[index, index - 1] = - sin(o_j(j-
0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) +
(sin(o_j(j+0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2)
            elif i == N + 1 and j == 2:
                # i = N + 1, j = 2
                b[index, index] = (r_i(i+0.5,h)**2+r_i(i-
0.5,h)**2)/(r_i(i,h)**2*h**2) + (sin(o_j(j+0.5,k))+sin(o_j(j-
0.5,k)))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) - (4*sin(o_j(j-
0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2) + V_ij(i, j, mass, gamma, h,
k,potential)
                b[index, index - M] = - r_i(i-0.5,h)**2/(r_i(i,h)**2*h**2)
                b[index, index + 1] = -
sin(o_j(j+0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) + sin(o_j(j-
0.5,k))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2)
                elif i == N + 1 and j == M + 1:
                    # i = N + 1, j = M + 1
                    b[index, index] = (r_i(i+0.5,h)**2+r_i(i-
0.5,h)**2)/(r_i(i,h)**2*h**2) + (sin(o_j(j+0.5,k))+sin(o_j(j-
0.5,k)))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) -
(4*sin(o_j(j+0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2) + V_ij(i, j, mass,
gamma, h, k,potential)
                    b[index, index - M] = - r_i(i-0.5,h)**2/(r_i(i,h)**2*h**2)
                    b[index, index - 1] = - sin(o_j(j-
0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) +
(sin(o_j(j+0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2)

def alternative_method_nonzero_MQN_PosSigma(b, mass, N, M, h, k, gamma,
potential):
    for i in range(2, N + 2):
        for j in range(2, M + 2):
            index = index_transform(i, j, N)
            if i != 2 and i != N + 1 and j != 2 and j != M + 1:
                # 2 < i < N + 1, 2 < j < M+1
                b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)

```

```

        b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
        elif j == 2 and i != 2 and i != N + 1:
            # j = 2, 2 < i < N+1
            b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
            b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
            b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
            b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
            elif j == M + 1 and i != 2 and i != N + 1:
                # j = M+1, 2 < i < N+1
                b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4 * sin(o_j(j + 0.5, k))) / (3 *
r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
                b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2) + sin(o_j(j + 0.5, k)) / (3 * r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                elif i == 2 and j != 2 and j != M + 1:
                    # i = 2, 2 < j < M+1
                    b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4 * r_i(i - 0.5, h) ** 2) / (3 *
r_i(i, h) ** 2 * h ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
                    b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + r_i(i - 0.5, h) ** 2 / (3 * r_i(i, h) ** 2 * h ** 2)
                    b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                    b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                    elif i == N + 1 and j != 2 and j != M + 1:
                        # i = N+1, 2 < j < M+1
                        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k,
potential)
                        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
                        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
                        elif i == 2 and j == 2:
                            # i = 2, j = 2
                            b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4 * r_i(i - 0.5, h) ** 2) /

```



```

(3*r_i(i, h) ** 2 * h ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
    b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + r_i(i - 0.5, h) ** 2 / (3*r_i(i, h) ** 2 * h ** 2)
    b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == 2 and j == M + 1:
        # i = 2, j = M+1
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) - (4 * r_i(i - 0.5, h) ** 2) / (3 *
r_i(i, h) ** 2 * h ** 2) - (4 * sin(o_j(j + 0.5, k))) / (3 * r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h, k, potential)
        b[index, index + M] = - r_i(i + 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2) + (r_i(i - 0.5, h) ** 2) / (3 * r_i(i, h) ** 2 * h ** 2)
        b[index, index - 1] = - sin(o_j(j - 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2) + (sin(o_j(j + 0.5, k))) / (3 * r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == N + 1 and j == 2:
        # i = N + 1, j = 2
        b[index, index] = (r_i(i + 0.5, h) ** 2 + r_i(i - 0.5, h) ** 2)
/ (r_i(i, h) ** 2 * h ** 2) + (sin(o_j(j + 0.5, k)) + sin(o_j(j - 0.5, k))) /
(r_i(i, h) ** 2 * sin(o_j(j, k)) * k ** 2) + V_ij(i, j, mass, gamma, h,
k, potential)
        b[index, index - M] = - r_i(i - 0.5, h) ** 2 / (r_i(i, h) ** 2 *
h ** 2)
        b[index, index + 1] = - sin(o_j(j + 0.5, k)) / (r_i(i, h) ** 2 *
sin(o_j(j, k)) * k ** 2)
    elif i == N + 1 and j == M + 1:
        # i = N + 1, j = M + 1
        b[index, index] = (r_i(i+0.5,h)**2+r_i(i-
0.5,h)**2)/(r_i(i,h)**2*h**2) + (sin(o_j(j+0.5,k))+sin(o_j(j-
0.5,k)))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) -
(4*sin(o_j(j+0.5,k)))/(3*r_i(i,h)**2*sin(o_j(j,k))*k**2) + V_ij(i, j, mass,
gamma, h, k, potential)
        b[index, index - M] = - r_i(i-0.5,h)**2/(r_i(i,h)**2*h**2)
        b[index, index - 1] = - sin(o_j(j-
0.5,k))/(r_i(i,h)**2*sin(o_j(j,k))*k**2) +
(sin(o_j(j+0.5,k)))/(3*r_i(i,h)*sin(o_j(j,k))*k**2)

def x_y_arrays_creating(radius, n_dimension, m_dimension, quarterNumber=1):
    R = float(radius)
    N = n_dimension
    M = m_dimension

    h = R / (N + 1.)
    k = np.pi / (2 * M + 2.)

    x = np.zeros(N)
    for i in range(N):
        x[i] = (i) * h
    M = m_dimension*quarterNumber
    y = np.zeros(M)
    for i in range(M):
        y[i] = (i) * k

    x, y = cart2pol(x, y)
    return x, y,

```