Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра исследования операций

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Исследование и разработка методов построения рабочей области и прокладка траектории для планарных механизмов призматических двигателей

Выполнил:
Студент 4 курса,
Марков Игорь Сергеевич
412 группа

Научный руководитель:
Посыпкин Михаил Анатольевич

Москва —2019

Содержание

1 Введение

Параллельные кинематические механизмы, также называемые параллельными роботами манипуляторами, представляют собой замкнутые механизмы с достаточно хорошими показателями жесткости и точности. Они развиваются с начала 80-х годов прошлого века и прочно вошли во многие области человеческой деятельности. Несмотря на все свои плюсы, такие роботы обладают двумя недостатками: это, во-первых, ограниченная область рабочего пространства, а во-вторых - особые положения, при попадении механизма в которые возникают либо потеря некоторого количества степеней свобод, либо неуправляемая подвижность механизма. Исследование таких положений - достаточно важная задача, так как их необходимо учитывать при прокладке оптимальной траектории движения механизма.

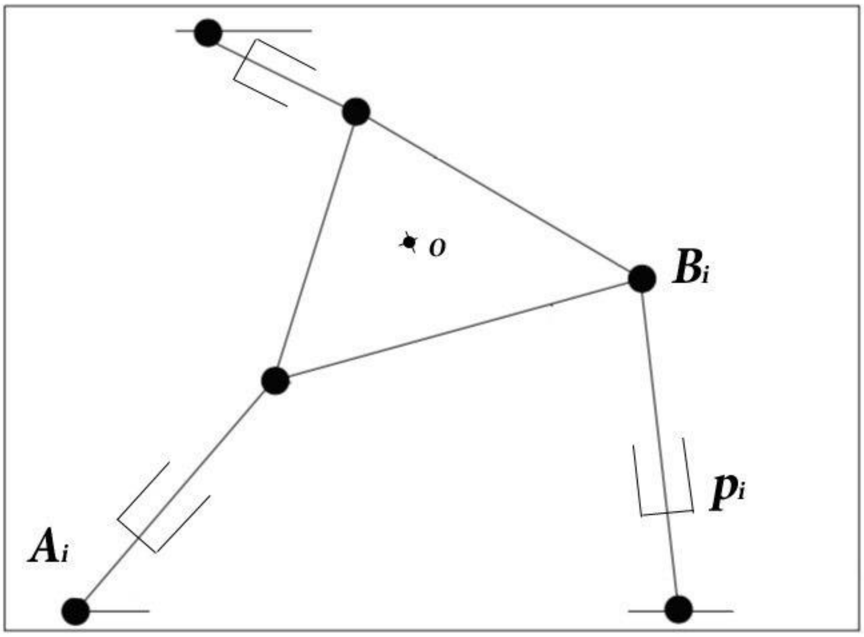


Рис. 1. 3-R$\underline{\text{P}}$R робот ПОДПРАВИТЬ ПОДПИСЬ

Плоскими параллеными кинематическими роботами называются механизмы, обладающие тремя степенями свободы - двумя поступательными и одной вращательной. В данной работе рассматривается 3-R$\underline{\text{P}}$R робот (см. рис. 1). Цифра 3 в его названии означает число кинематических цепей, буква R - вращательную кинематическую пару, буква P - поступательную кинематическую пару. Буква P выделена нижним подчеркиванием - это значит, что данная пара является приводной. Пары без подчеркивания - пассивные. Точки A_i фиксированны, звенья A_iB_i могут менять свой размер с помощью приводов p_i, изменяя при этом так же и координаты центра выходного звена **о**

В данной работе исследуются положения сингулярности робота 3-RPR. Строится его рабочая область и положения сингулярности на основе двух подходов - метода неравномерных покрытий с минимизацией функций ВОТ ТУТ ТАКОЕ СЕБЕ. ИСПРАВИТЬ И ПОЛУЧШЕ НАПИСАТЬ., а так же метода неравномерных покрытий с использованием интервального расширения. Производится сравнение этих двух подходов. Строится оптимальная траектория движения выходного звена механизма в обход точек сингулярности. Производится отрисовка полученных результатов в формате 2D.

Для реализации программного кода используется Python3, а так же библиотеки matplotlib, scipy, numpy, pyinterval.

2 Прямая и обратная задачи кинематики

Рассмотрим робота 3-RPR еще раз (см. рис. 2):

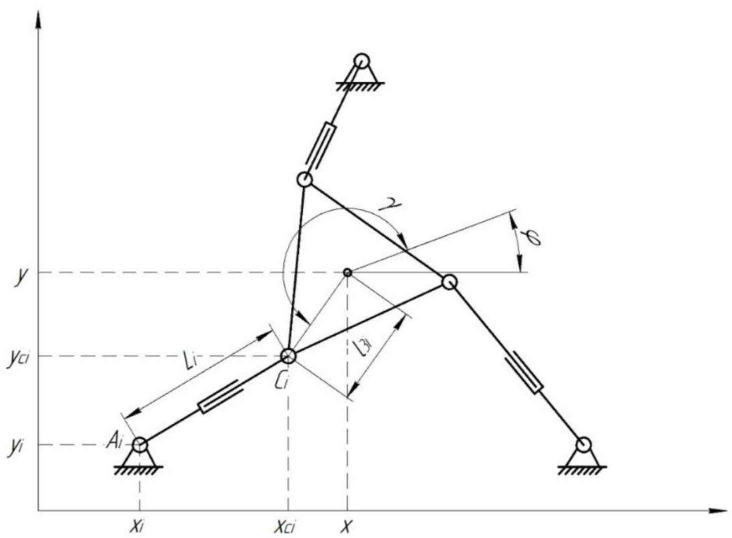


Рис. 2. ПОДПРАВИТЬ ПОДПИСЬ

x_Ci, y_Ci определяются следующим образом:

$$\begin{cases} x_{C_i} = x + l_{3i}cos(\psi + \gamma_i) \\ y_{C_i} = y + l_{3i}sin(\psi + \gamma_i) \end{cases} \tag{1}$$

По теореме Пифагора:

$$L_i{}^2 = (x_{C_i} - x_i)^2 - (y_{C_i} - y_i)^2, \tag{2}$$

где x_i и y_i - проекции точки A_i на оси x и y соответственно.

Подставляя 1 в 2 получим уравнение связи для 3-R$\underline{\text{P}}$R робота:

$$L_i{}^2 = (x + l_{3i}cos(\psi + \gamma_i) - x_i)^2 + (y + l_{3i}sin(\psi + \gamma_i) - y_i)^2, i = 1..3, \tag{3}$$

где L_i - обобщенная координата.

Прямая задача кинематики - задача о нахождении координат подвижной платформы L_1, L_2, L_3 манипулятора по заданным x, y, ψ.

Обратная задача кинематики - нахождение, наоборот, координат центра выходного звена по заданным L_1, L_2, L_3.

Решение обратной задачи очевидно. Чаще всего необходимо решить именно прямую задачу. Она может быть решена с помощью различных методов численного анализа, например - метода Ньютона.

3 Построение рабочей области

В реальном механизме 3-R$\underline{\text{P}}$R длины штанг L_i будут ограничены значениями $[l^{min}, l^{max}]$, где l^{min} - минимальная длина поступательной кинематической пары, l^{max} - максимальная длина поступательной кинематической пары. Эти ограничения приводят к ограничениям рабочей области манипулятора.

Так же имеет смысл задать ограничения на переменные $x, y\psi$. Без ограничения общности можно положить:
$\psi \in [0, 2\pi]$
Так же положим
$x \in [\max\limits_{i=1,2,3}(x_i - l_i^{max}) + \frac{l_{3i}}{2}, \min\limits_{i=1,2,3}(x_i + l_i^{max}) - \frac{l_{3i}}{2}]$
$y \in [\max\limits_{i=1,2,3}(y_i - l_i^{max}) + \frac{l_{3i}}{2}, \min\limits_{i=1,2,3}(y_i + l_i^{max}) - \frac{l_{3i}}{2}]$

Итоговая система, описывающая рабочую область робота манипулятора 3-R$\underline{\text{P}}$R будет иметь следующий вид:

$$\begin{cases} {L_1}^2 = (x + l_{3_1} cos(\psi + \gamma_1) - x_1)^2 + (y + l_{3_1} sin(\psi + \gamma_1) - y_1)^2 \\ {L_2}^2 = (x + l_{3_2} cos(\psi + \gamma_2) - x_2)^2 + (y + l_{3_2} sin(\psi + \gamma_2) - y_2)^2 \\ {L_3}^2 = (x + l_{3_3} cos(\psi + \gamma_3) - x_3)^2 + (y + l_{3_3} sin(\psi + \gamma_3) - y_3)^2 \\ \psi \in [0, 2\pi] \\ x \in [\underset{i=1,2,3}{max}(x_i - l_i^{max}) + \frac{l_{3i}}{2}, \underset{i=1,2,3}{min}(x_i + l_i^{max}) - \frac{l_{3i}}{2}] \\ y \in [\underset{i=1,2,3}{max}(y_i - l_i^{max}) + \frac{l_{3i}}{2}, \underset{i=1,2,3}{min}(y_i + l_i^{max}) - \frac{l_{3i}}{2}] \end{cases} \tag{4}$$

Система 4 содержит в себе 3 уравнения и 6 переменных. На практике решение такой системы может быть затруднено или нереализуемо из-за вычислительной сложности, поэтому часто 3 переменные L_i исключают, переводя систему равенств в систему неравенств:

$$\begin{cases} (x + l_{3_1} cos(\psi + \gamma_1) - x_1)^2 + (y + l_{3_1} sin(\psi + \gamma_1) - y_1)^2 - l_1^{max2} \leq 0 \\ (x + l_{3_2} cos(\psi + \gamma_2) - x_2)^2 + (y + l_{3_2} sin(\psi + \gamma_2) - y_2)^2 - l_2^{max2} \leq 0 \\ (x + l_{3_3} cos(\psi + \gamma_3) - x_3)^2 + (y + l_{3_3} sin(\psi + \gamma_3) - y_3)^2 - l_3^{max2} \leq 0 \\ l_1^{min2} - (x + l_{3_1} cos(\psi + \gamma_1) - x_1)^2 + (y + l_{3_1} sin(\psi + \gamma_1) - y_1)^2 \leq 0 \\ l_2^{min2} - (x + l_{3_2} cos(\psi + \gamma_2) - x_2)^2 + (y + l_{3_2} sin(\psi + \gamma_2) - y_2)^2 \leq 0 \\ l_3^{min2} - (x + l_{3_3} cos(\psi + \gamma_3) - x_3)^2 + (y + l_{3_3} sin(\psi + \gamma_3) - y_3)^2 \leq 0 \\ \psi \in [0, 2\pi] \\ x \in [\underset{i=1,2,3}{max}(x_i - l_i^{max}) + \frac{l_{3i}}{2}, \underset{i=1,2,3}{min}(x_i + l_i^{max}) - \frac{l_{3i}}{2}] \\ y \in [\underset{i=1,2,3}{max}(y_i - l_i^{max}) + \frac{l_{3i}}{2}, \underset{i=1,2,3}{min}(y_i + l_i^{max}) - \frac{l_{3i}}{2}] \end{cases} \tag{5}$$

Так же на практике представляет интерес не все множество решений систем 4 или 5 в пространстве R^6, а его проекция на множество координат (x, y, ψ).

Рассмотрим способы решения указанных задач, основанных на апроксимации множества решений систем уравнений с помощью метода неравномерных покрытий. УКАЗАТЬ ССЫЛКУ Пусть дана некоторая система уравнений вида

$$\begin{cases} f_i(x) = 0, i = 1...m \\ x \in [a, b], \end{cases} \tag{6}$$

где f_i - функция, зависящая от x переменных, $x \in R^6$ а $[a, b]$ - n-мерный параллелепипед, являющийся ограничением для параметров x. Отображение f_i непрерывно. Пусть так же $X \in R$ - множевство всех решений системы 6. Тогда данная система будет эквивалентна системе 7:

$$\begin{cases} \phi_i(x) = 0, i = 1...m \\ x \in [a, b], \end{cases} \tag{7}$$

где $\phi_i(x)$ - свертка функций $g_i(x), i = 1, ..., m$. В качестве свертки можно взять любую норму вектора в пространстве R^m.

Для нахождения апроксимации множества X используется следующий алгоритм:

Пусть параллелепипед L - исходный параллелепипед, покрывающий всю область определения функции ϕ. Выберем некую точность δ. Будем делить в цикле L вдоль наибольшей диагонали на два паралледепипеда, пока все параллелепипеды не станут диаметром меньше δ. Рассматривая очередной параллелепипед, будем проверять, принимает ли на нем функция ϕ нулевое значение. Если не принимает, будем отбрасывать данный параллелепипед.

Для проверки параллелепипида будем использовать следующий подход:
Пусть $m = \underset{x \in P}{min} \phi(x)$, $M = \underset{x \in P}{max} \phi(x)$, $P \in L$ - рассматриваемый параллилепипед. Если выполняется $m > 0$ или $M < 0$, то пересечение P и X пусто, и, стало быть, мы должны отбросить P.

Псевдокод данного алгоритма приведен ниже.
L = [a, b]
A
while L != $\emptyset$:
take $P \in L$
L = L / P
if !(m > 0 or M < 0) then:
if diameter(P) < δ then:
A = A ∞ P
else then:
$P_{1,2}$ = P / 2
end

Утверждение 1. Множество A, построенное в результате данного алгоритма, содержит все множество X и расстояние по Хаусдорфу между множествами A и X не превосходит δ:

$$X \subseteq A \tag{8}$$

$$h(A, X) \leq \delta \tag{9}$$

вставить ссылку на материалы по рабочей области - статья для шк сем

Доказательство. Для любого параллелепипеда P выполняется неравенство $m \leq 0 \leq M$. При построении области A выкидывался каждый параллелепипед, заведомо не содержащий ни одной точки из X, следовательно, выражение 8 справедливо. Докажем теперь утверждение 9: множество P компактно, а функция $\phi(x)$ - непрерывна. Следовательно, найдется точка $x_0 \in P$ такая, что $\phi(x_0) = 0$, $x_0 \in P$ Диаметр P ограничен δ, а утверждение 6 справедливо - истинно, утверждение 9 справедливо.

Утверждение 2. Множество A, построенное в результате вычисления данного алгоритма, целиком содержит в себе множество X.
Доказывается это утверждение аналл=огично утверждению 1. ТУТ СОСЛАТЬСЯ НА [СТАТЬЮ ПОСЫПКИНА ПРО ПОСТРОЕНИЕ РАБОЧЕЙ ОБЛАСТИ]

Алгоритм, основанный на неравенствах, строится по аналогии с алгоритмом, описанным выше.

Результаты выполнения этой программы при различных приближениях δ (время округлено до 4х знаков после запятой):

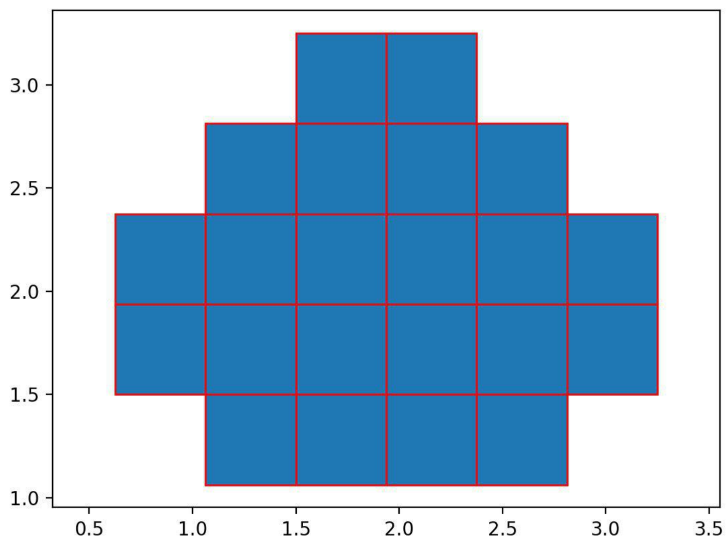


Рис. 3 EPSILON = 0.5
количество прямоугольников = 22
время вычисления площади - 0.5917 секунды.

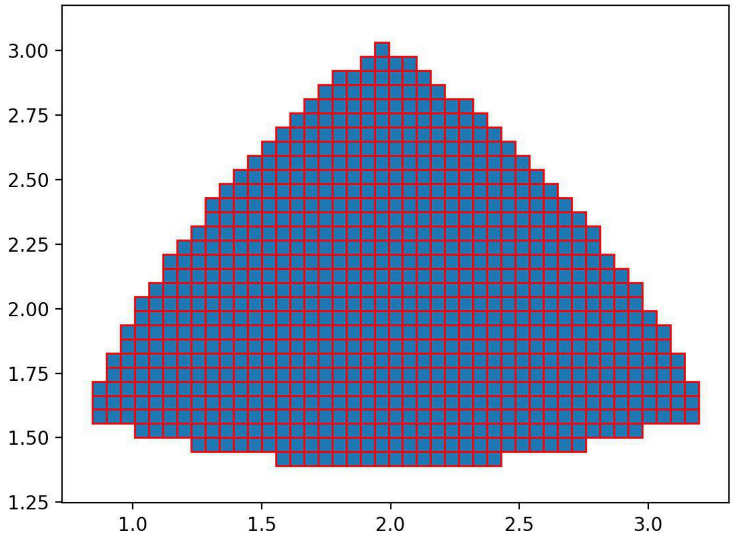
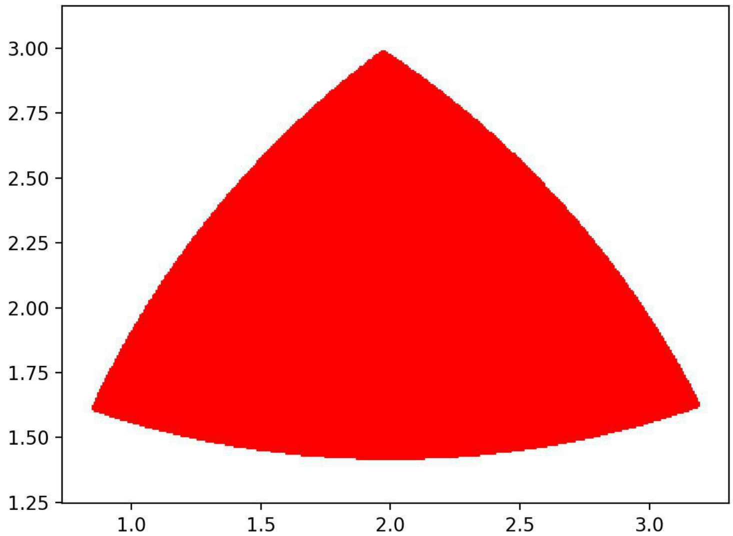


Рис. 4 EPSILON = 0.1
количество прямоугольников = 801
время вычисления площади - 9.2592 секунды.



]

Рис. 5 EPSILON = 0.01

количество прямоугольников = 47183
время вычисления площади - 9 минут 38.1353 секунд.

Далее мы рассмотрим второй подход к построению рабочей области и сравним эти два подхода.

4 Интервальный анализ

Интервальная арифметика

Введем интервальную арифметику. Пусть интервал - объект, оперирующий двумя вещественными числами (a, b), причем a и b могут достигать бесконечности, а так же $a \leq b$. $[a, b] = \{x \in R | a \leq x \leq b\}$.

Определим базовые операции, производящиеся над интервалами:

1. Сложение: $[a, b] + [c, d] = [a + c, b + d]$
2. Вычитание: $[a, b] - [c, d] = [a - d, b - c]$
3. Умножение:
   $[a, b] * [c, d] = [min(a * c, a * d, b * c, b * d), max(a * c, a * d, b * c, b * d)]$
4. Деление: $[a, b] \, / \, [c, d] = [min(a/c, a/d, b/c, b/d), max(a/c, a/d, b/c, b/d)]$

Операция деления определена только тогда, когда интервал, являющийся делителем, не содержит в себе нуля. Так же стоит отметить, что интервалы, у которых начало и конец, часто рассматриваются как обычные вещественные числа. Для них операции определены стандартным образом. Также для интервалов можно ввести операции пересечения, объеденения, отрицания и тд.

Свойства интервалов Интервалы обладают следующими свойствами:

1. Коммутативность
2. Ассоциативность
3. Дистрибутивность в ослабленном виде: $A(X + Y) \subset AX + AY$

Интервальное расширение Пусть дана функция $f(x)$. Интервальным расширением функции $f(x)$ называется функция $f(x)_{[I]}$, если:

1. Функция $f(x)_{[I]}$ монотонна по включению. Это значит, что, если $x \subseteq y$, то $f(x)_{[I]} \subseteq f(y)$

2. Значения функции интервального расширения $f(x)_{[I]}$ от интервала [a, b], где a = b, совпадает со значением аргумента функции $f(a)$.

Вообще говоря, интервальное расширение функции $f(x)$ есть ее внешняя оценка на брусе x.

Получение интервального расширения Существует множество подходов к получению интервального расширения. Самый простой - метод так называемого естественного интервального расширения, в котором в функции вместо текущих аргументов подстанавливаются интервалы изменения этих аргументов. Далее все операции производятся по правилам интервальной арифметики.
Вообще говоря, использование этого расширения при оценки области значения функции $f(x)$ часто приводит к довольно грубым результатам, а так же является избыточной из-за добавления "лишних"точек в процессе вычисления с использованием интервальной арифметики.

Поэтому обычно применяют более сложные методы интервального расширения. Так, например, широко распространена среднезначная форма интервального расширения. Пусть a - некоторое константное значение, тогда искомое расширение функции $f(x)$ на брусе x будет выглядеть как
$f(x)_{[I]} = f(a) + f^{'}(x)(x - a)$.

Так же используется наклонная форма интервального расширения. Представим функцию $f(x)$ в виде $f(x) = f(x_0) + f_s(x_0, x)(x - x_0)$, где $f_s(x_0, x)$ мы будем называть наклоном функции $f(x)$ между x_0 и x. Тогда внешнюю оценку области значений функции $f(x)$ можно записать следующим образом:
$f_{slope}(x, x_0) = f(x_0) + f_s(x_0, x)(x - x_0)$.

Для большинства функций на больших областях определения неплохо работает естественное интервальное расширение, для "узких"областей определения стоит пользоваться среднезначной формой интервального расширения. Для более точных оценок можно использовать пересечения нескольких расширений.
Подробнее о вопросе интервальных расширений можно прочитать в [ВСТАВИТЬ ССЫЛКУ НА ИСТОЧНИК ПАНОВ ОЦЕНКА ОБЛАСТИ ЗНАЧЕНИЯ ФУНКЦИЙ МЕТОДАМИ ИНТЕРВАЛЬНОГО АНАЛИЗА].

Рассмотрим построение рабочей области робота манипулятора 3-R$\underline{\text{P}}$R с помощью интервального анализа. Мы будем использовать метод неравномерных покрытий, оперируя при этом интервальными значениями. На этапе проверки вхождения точек из X в множество P нам следует проверять, входит ли ноль в интервал $\underset{x \in P}{f(x)_{[I]}}$. Если это не так, отбрасываем параллелепипед P. В конце работы алгоритма мы получим внешнюю границу рабочей области механизма. Рассмотрим выполнение алгоритма при различных приближениях δ:

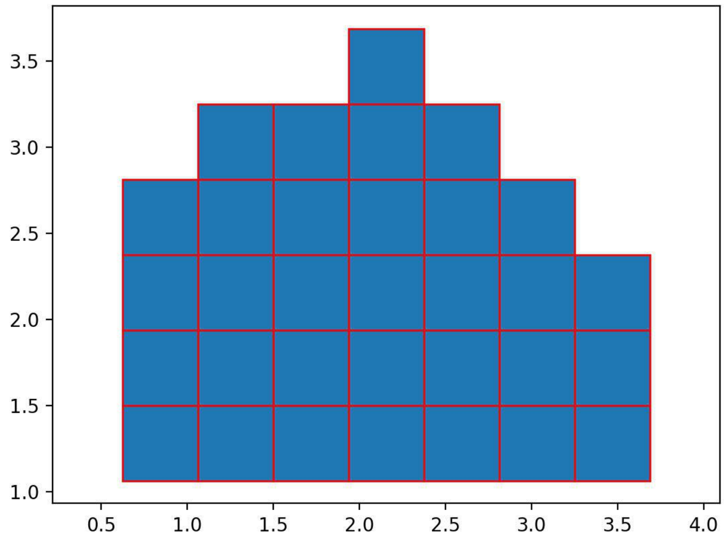


Рис. 4 EPSILON $=0.5$
количество прямоугольников $=32$
время вычисления площади - 0.1267 секунды.

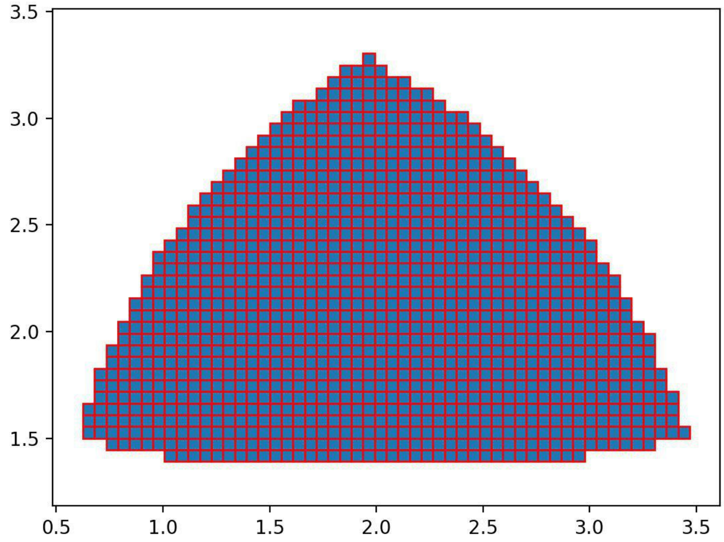
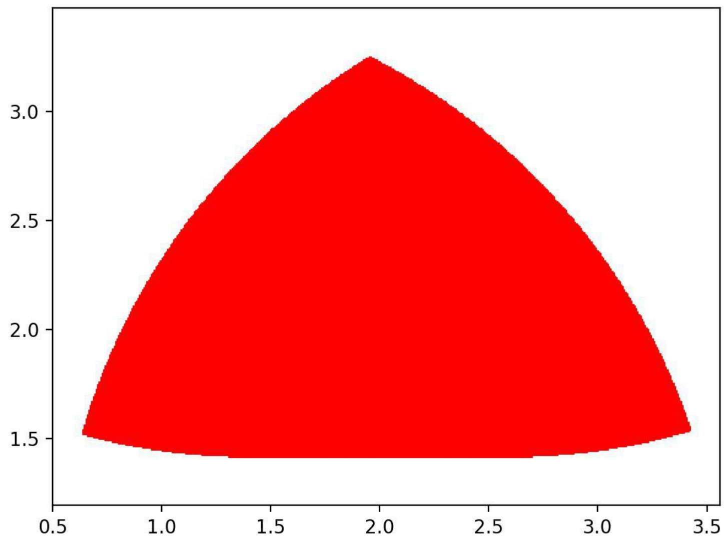


Рис. 5 EPSILON = 0.1
количество прямоугольников = 1171
время вычисления площади - 2.6512 секунды.



]

Рис. 6 EPSILON = 0.01
количество прямоугольников = 70110
время вычисления площади - 2 минуты 8.5468 секунды.

Отсюда видно, что интервальный подход значительно быстрее: при точности $\delta = 0.5$ вычисление у интервального подхода занимает 0.1267 секунды, а у подхода с минимизацией - 0.5917. При $\delta = 0.1$ вычисление происходит за 2.6512 и 9.2592 секунды, а при $\delta = 0.01$ - 2 минуты 8.5468 секунды и 9 минут 38.1353 секунд соответственно. В тоже время, интервальный подход показывает меньшую точность. Точность можно улучшить, пользуясь методами из [ССЫЛКА НА ТУ ЖЕ КНИГУ ИЛИ НОВОЕ НАЙТИ ПРО ИНТЕРВАЛЫ].

5 Положения сингулярности