A1. Задача трёх кругов

Владимиров Алексей, БПИ235

24 ноября 2024 г.

1 Алгоритм

Так как в самой задаче, довольно подробно описано, каким образом реализовывать решение этой задачи, подробностей в этом параграфе не будет. Единственно что хотелось бы уточнить, то что для генерации точек был использован вихрь Мерсенна, а случайные значения брались из равномерного распределения.

2 Различные параметры работы алгоритма

Для того что бы провести экспериментальную проверку устойчивости алгоритма, фиксировались следующие параметры:

- Число итераций алгоритма: от 100, до 100100 генераций случайной точки.
- Начальная точка прямоугольника.
- Длина стороны прямоугольника.

Для автоматизации процесса генерации, вместо ручного подбора координат и размеров для каждого прямоугольника, были посчитаны 2 крайние позиции:

Стартовая прямоугольная область:

1. Левая нижняя координата по $x : x_0 = 0$
2. Левая нижняя координата по $y : y_0 = 0$
3. Размер обоих сторон: ≈ 3.117

Конечная прямоугольная область:

1. Левая нижняя координата по $x : x_0 = 0.88$
2. Левая нижняя координата по $y : y_0 = 0.88$
3. Размер стороны параллельной оси Ox сторон: ≈ 1.119
4. Размер стороны параллельной оси Oy сторон: ≈ 1.112

Далее был зафиксирован параметр отвечающий за число областей: 7. Соответственно все оставшиеся параметры для 5 областей между, рассчитывались путем подстановки $t \in [0; 7]$ в следующие выражения:

1. Левая нижняя координата по $x : x_0 \approx 0.12571t$
2. Левая нижняя координата по $y : y_0 \approx 0.12571t$
3. Размер стороны a_t параллельной оси Ox сторон: $\approx 3.117 - 0.28642t$
4. Размер стороны b_t параллельной оси Oy сторон: $\approx 3.117 - 0.28642t$

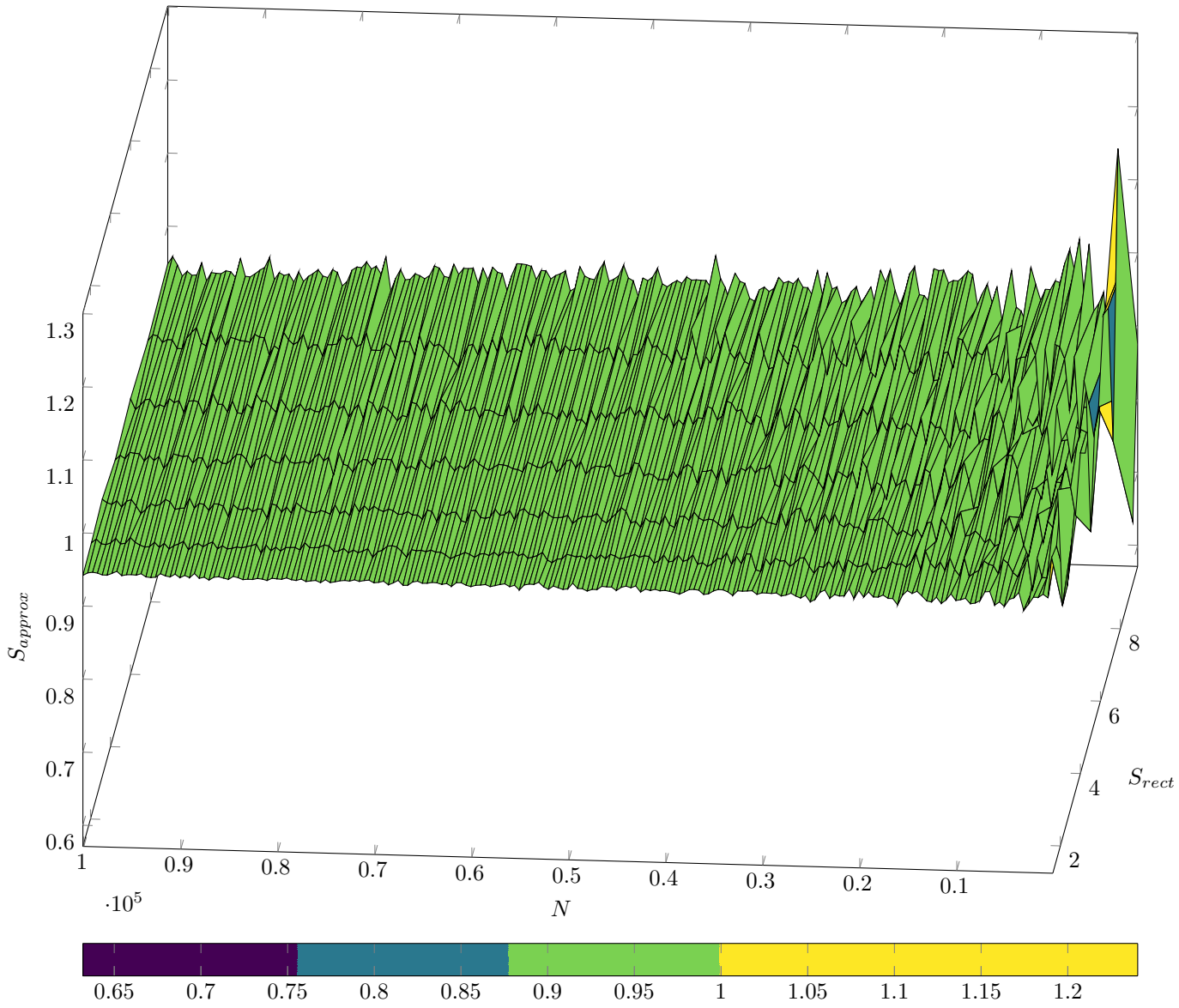
Таким образом удалось получить тестовые результаты алгоритма. Их можно представить в виде массива точек в пространстве $\mathbb{R}^3$:

$(S_{rect}; N; S_{approx})$ - где $S_{rect} = a_i \cdot b_i, N$ - число итераций алгоритма, S_{approx} - площадь искомой области

Тогда эти точки можно представить как кусочно-линейную поверхность. Для начала посмотрим на график зависимости S_{approx} от (S_{rect}, N). Обозначим данную поверхность как геометрическое представление функции:

$$S(S_{rect}, N) = S_{approx}$$

На данном моменте, важно отметить следующую деталь: так как мы линейно уменьшаем размеры обоих сторон прямоугольной области на каждой итерации, то полученную функцию можно строить как функцию от 2 аргументов, так как произведение $a_i \cdot b_i$ будет монотонно снижаться.



3 Анализ графика полученного ответа

Исходя из графика видна положительная корреляция с увеличением числа итераций, равно как и с уменьшением площади прямоугольника. Чем ближе мы подходим к левому нижнему углу поверхности, тем менее шумным становится график, а сама поверхность становится более гладкой, сконцентрированной на отметке по оси $z(S_{approx})$ около значения 9.44.

Это показывает, что приведенный нами алгоритм Монте-Карло сходится к точной площади фигуры которая составляет:

$$S_{real} = \frac{\pi}{4} + \frac{5\arcsin(0.8)}{4} - 1 \approx 0.944517185899\ldots$$

Однако, можно заметить, что наиболее дальние по направлении оси S_{rect} сегменты графика, образую более шумный рисунок вдоль всей оси N. Это, очевидно, объясняется тем, что в больших прямоугольных областях увеличивается число доступных вариантов, куда можно попасть случайной точкой. Поэтому для достижения одинаковой частоты, по сравнению с более компактными областями, требуется больше итераций алгоритма.

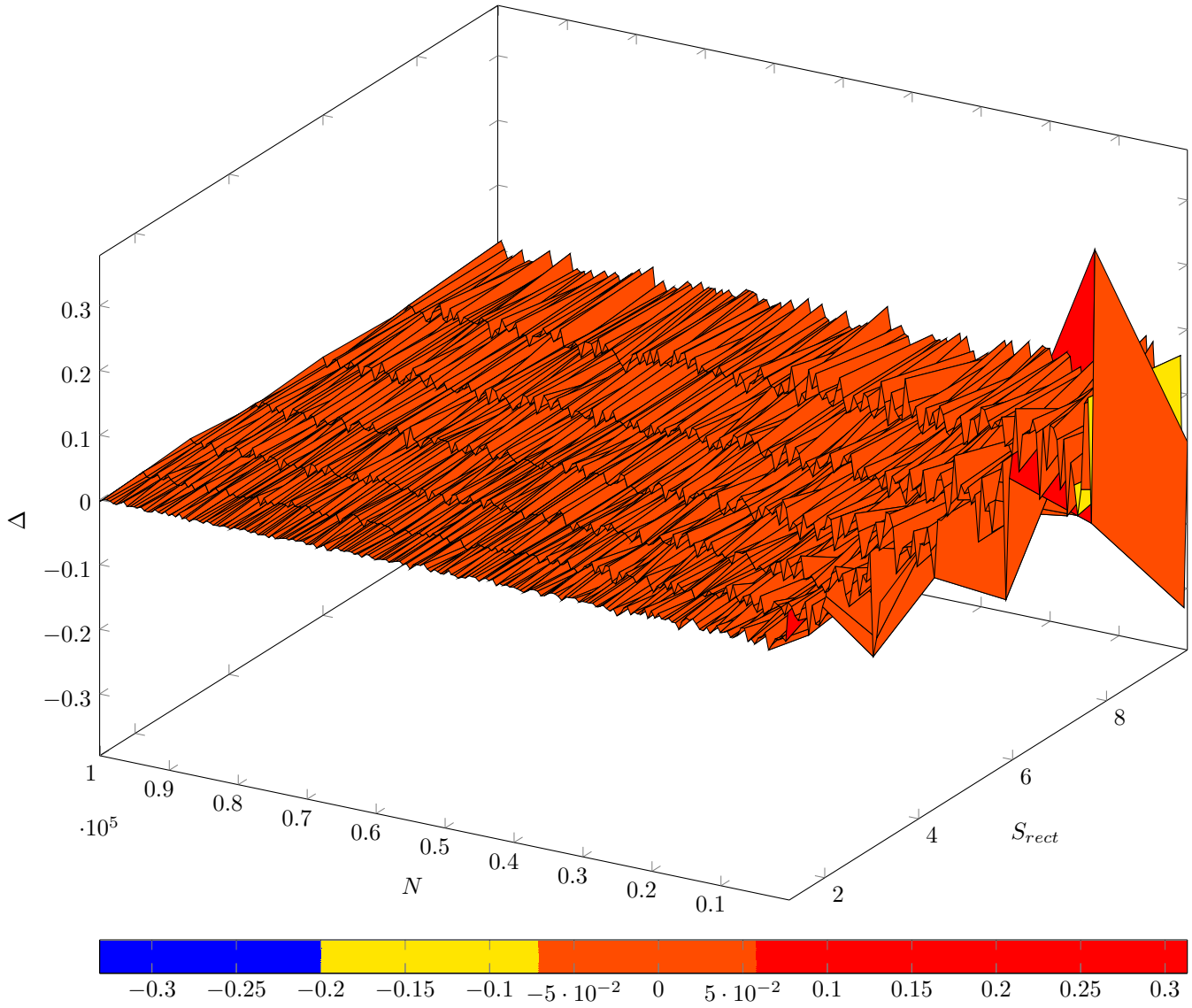
Это же подкрепляется и геометрической логикой. Чем ближе мы придвинем область в которой будут генерироваться точки, к нашей искомой области, тем меньше нужно будет итераций что бы сгенерировать ответ, с фиксированной точностью. Фактически, мы, задавая площадь прямоугольника, делаем «первую инициализацию» площади, и чем точнее мы сделаем первую инициализацию, тем лучше будет ответ.

4 Анализ графика относительного отклонения

Теперь можно задаться вопросом: а на сколько близко мы аппроксимировали нашу площадь? Для этого введем новую функцию:

$$\Delta(S_{rect}, N) = \frac{S(S_{rect}, N) - S_{real}}{S_{real}}$$

И измеряем ее значения на том же массиве данных.



Как и ожидалось, получился отнормированный по истинной площади график абсолютных значений(если сравнить оба графика, отличия заметны будут только в шкале оси z).

Из визуализации, легко понять что наибольший выброс в данных полученных в результате алгоритма отличается от корректного результата на 30%. А при приближении к оптимальным значениям выборки(движению вниз по оси S_{rect} и влево по оси N) ошибка будет менее 0.001

5 Ресурсы:

Ссылка на гитхаб репозиторий: `https://github.com/pepsiplusmilk/DSA_HSE_SE_SET3/tree/main/A1` Ссылка на посылку в контесте: `https://dsahse.contest.codeforces.com/group/NOflOR1Qt0/contest/565612/submission/292776613`