Задание A1

Программа вычисляет приближенную площадь алгоритмом Монте Карло, генерируя N рандомных точек в указанном диапазоне и проверяя попадание в заданную окрестность. В файл записываются вычисленные приближенные площади при различных N для широкой и узкой площадей, а также отклонение от точного значения площади.

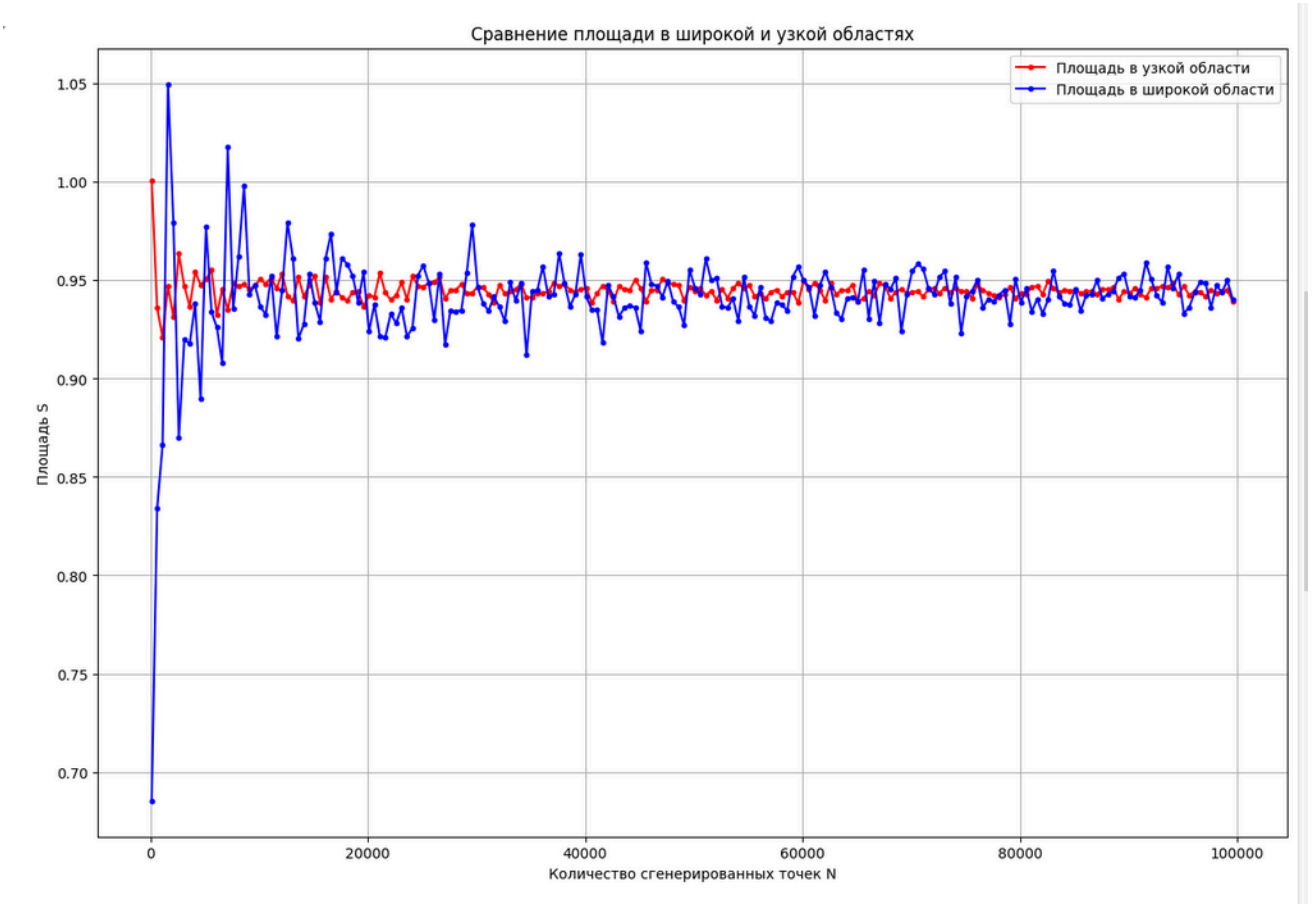
```cpp
#include <iostream>
#include <fstream>
#include <random>
#include <cmath>

int main() {
  std::ofstream out( s: "diffWide.txt");
  std::ofstream out2( s: "SquareWide.txt");
  const double s_real = (0.25 * M_PI + 1.25 * std::asin( x: 0.8) - 1);

  std::cout << s_real;
  double x1, y1, r1, x2, y2, r2, x3, y3, r3;
  x1 = 1; y1 = 1; r1 = 1;
  x2 = 1.5; y2 = 2; r2 = std::sqrt( x: 5)/2;
  x3 = 2; y3 = 1.5; r3 = std::sqrt( x: 5)/2;

  double bottom = std::min(y1 - r1, std::min(y2 - r2, y3 - r3));
  double top = std::max(y1 + r1, std::min(y2 + r2, y3 + r3));
  double left = std::min(x1 - r1, std::min(x2 - r2, x3 - r3));
  double right = std::max(x1 + r1, std::min(x2 + r2, x3 + r3));
  double S_rec = (right - left) * (top - bottom);

  std::random_device rand_dev;
  std::mt19937 generator( sd: rand_dev());
  std::uniform_real_distribution<> distr_x( a: left,  b: right);
  std::uniform_real_distribution<> distr_y( a: bottom,  b: top);

```

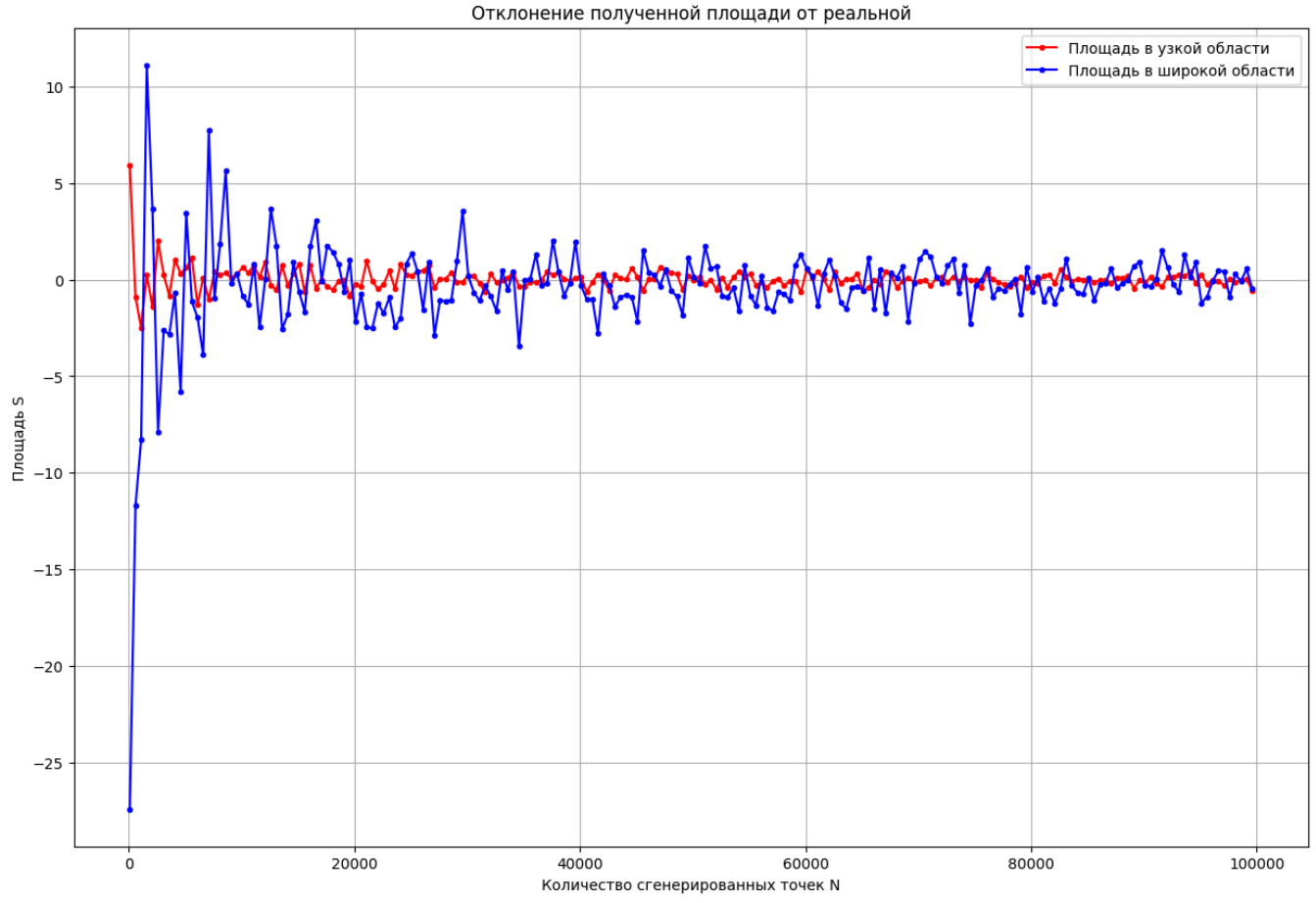
```
27
28      for (int j = 100; j <= 100000; j += 500) {
29        int N = j;
30        int M = 0;
31        for (int i = 0; i < j; ++i) {
32
33          double x = distr_x( &: generator);
34          double y = distr_y( &: generator);
35
36          if (pow( X: x1 - x,  Y: 2) + pow( X: y1 - y,  Y: 2) <= pow( X: r1,  Y: 2) &&
37              pow( X: x2 - x,  Y: 2) + pow( X: y2 - y,  Y: 2) <= pow( X: r2,  Y: 2) &&
38              pow( X: x3 - x,  Y: 2) + pow( X: y3 - y,  Y: 2) <= pow( X: r3,  Y: 2)) {
39            M += 1;
40          }
41        }
42        double S = (static_cast<double>(M) / N) * S_rec;
43        out << N << ", " << 100 * (S - s_real) / s_real << "\n";
44        out2 << N << ", " << S << "\n";
45      }
46
47      out = std::ofstream ( s: "diffNarrow.txt");
48      out2 = std::ofstream ( s: "SquareNarrow.txt");
49
50
51      bottom = 0.87;
52      top = 2.1;
53      left = 0.87;
54      right = 2;
55      S_rec = (right - left) * (top - bottom);
```

```
56
57      distr_x = std::uniform_real_distribution<>( a: left,  b: right);
58      distr_y = std::uniform_real_distribution<>( a: bottom,  b: top);
59
60      for (int j = 100; j <= 100000; j += 500) {
61          int N = j;
62          int M = 0;
63          for (int i = 0; i < j; ++i) {
64
65              double x = distr_x( &: generator);
66              double y = distr_y( &: generator);
67
68              if (pow( X: x1 - x,  Y: 2) + pow( X: y1 - y,  Y: 2) <= pow( X: r1,  Y: 2) &&
69                  pow( X: x2 - x,  Y: 2) + pow( X: y2 - y,  Y: 2) <= pow( X: r2,  Y: 2) &&
70                  pow( X: x3 - x,  Y: 2) + pow( X: y3 - y,  Y: 2) <= pow( X: r3,  Y: 2)) {
71                  M += 1;
72              }
73          }
74          double S = (static_cast<double>(M) / N) * S_rec;
75          out << N << ", " << 100 * (S - s_real) / s_real << "\n";
76          out2 << N << ", " << S << "\n";
77      }
78  }
79
80
```

Графики

При маленьких N разброс значений больше, т.к меньше точность вычисления площади, причем в широкой области значения площади колеблются больше, из-за большего размаха погрешности. Графики отклонения и площади совпадают, т.к. изменение площади прямо пропорционально изменению погрешности.


Сравнение площади в широкой и узкой областях
Площадь в узкой области
Площадь в широкой области
Площадь S
Количество сгенерированных точек N


Отклонение полученной площади от реальной
Площадь в узкой области
Площадь в широкой области
Площадь S
Количество сгенерированных точек N

ID посылки на CodeForces: 293020152
Ссылка на публичный репозиторий:
https://github.com/AnechkaShv/SET3/new/main/A1