SET3. Задача A1

Фролов-Буканов Виктор Дмитриевич БПИ-228

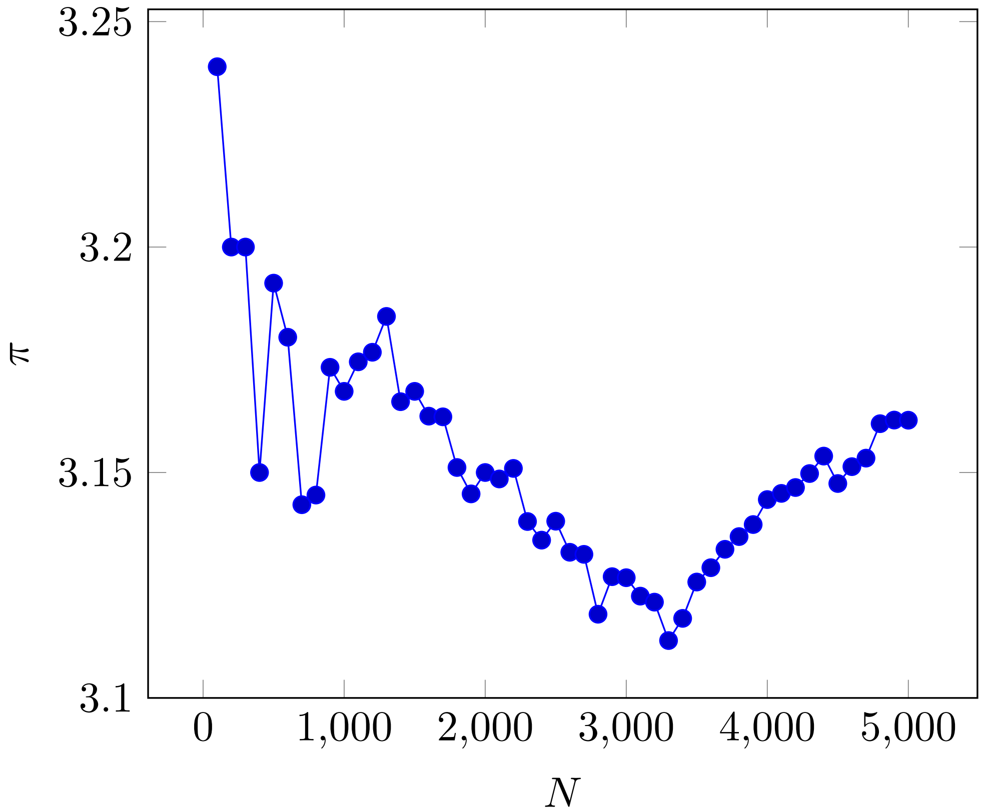
28 ноября 2023

1 Исходный код

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

std::ofstream fout_abs(R"(C:\Users\frolo\CLionProjects\
    assemblyTestProgram\abs_coordinates.txt)");
std::ofstream fout_rel(R"(C:\Users\frolo\CLionProjects\
    assemblyTestProgram\rel_coordinates.txt)");

std::pair<double, double> get_random_point(std::
    uniform_real_distribution<double> &unif,
                                           std::default_random_engine &
                                               re) {
  return std::make_pair(unif(re), unif(re));
}

double approximate_pi(int n) {
  double m = 0;
  std::uniform_real_distribution<double> unif(-1, 1);
  std::default_random_engine re; // NOLINT
  for (auto i = 0; i < n; ++i) {
    auto point = get_random_point(unif, re);
    auto x = point.first;
    auto y = point.second;
    if (x * x + y * y <= 1) m += 1;
  }

  auto app_pi = (4 * m) / n;
  fout_abs << "(" << n << ", " << app_pi << ") ";
  fout_rel << "(" << n << ", " << (std::abs(app_pi - M_PI) * 100 / M_PI
     ) << ") ";

  return app_pi;
}

int main() {

  for (auto i = 100; i <= 5000; i += 100) {
    std::cout << "N = " << i << ": " << approximate_pi(i) << '\n';
  }

  return 0;
}
```

Тут написаны 2 функции. Первая генерирует рандомную точку в квадрате с вершинами (1, 1), (1, -1), (-1, 1), (-1, -1). Вторая функция использует эту функцию генерации рандомной точки для апроксимации числи пи методом Монте-Карло, описанным в условии. Также функция записывает в отдельный файл каждое полученное значение числа пи в паре с числом N и в отдельный файл записывает относительное отклонение также в паре с N (эти данные необходимы для построения графиков)

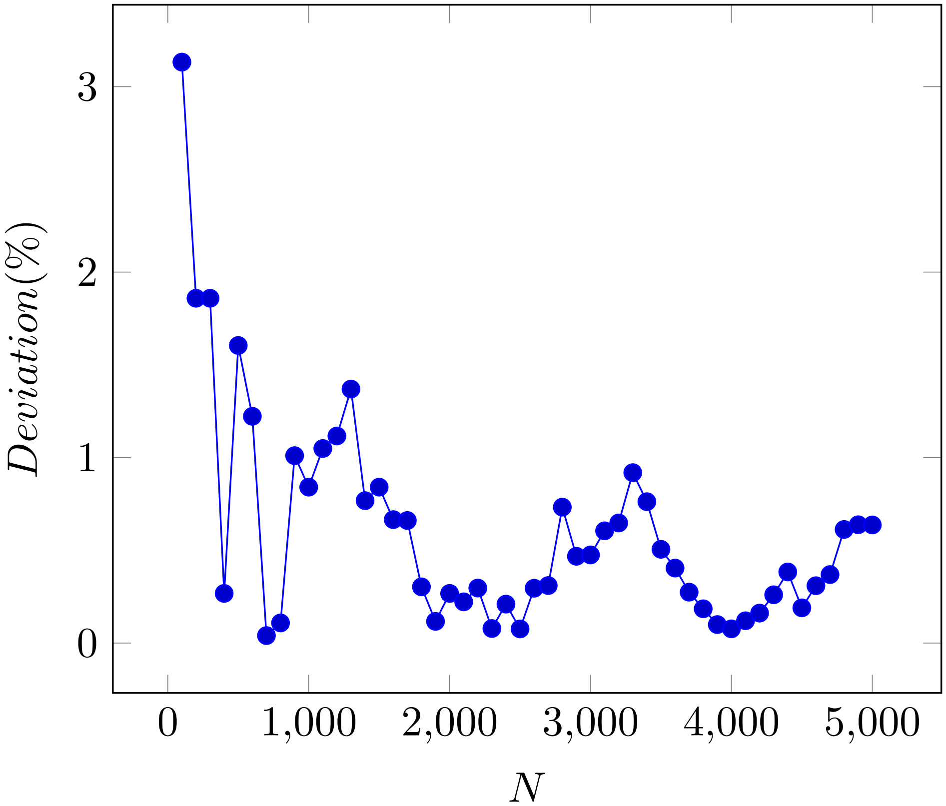
2 График №1 (приближенное значение числа π, полученного в разработанной программе, в зависимости от общего числа точек N)



3 Исходные данные для построение графика №1

```
(100, 3.24) (200, 3.2) (300, 3.2) (400, 3.15) (500, 3.192) (600, 3.18)
   (700, 3.14286) (800, 3.145) (900, 3.17333) (1000, 3.168) (1100,
   3.17455) (1200, 3.17667) (1300, 3.18462) (1400, 3.16571) (1500,
   3.168) (1600, 3.1625) (1700, 3.16235) (1800, 3.15111) (1900,
   3.14526) (2000, 3.15) (2100, 3.14857) (2200, 3.15091) (2300,
   3.13913) (2400, 3.135) (2500, 3.1392) (2600, 3.13231) (2700,
   3.13185) (2800, 3.11857) (2900, 3.1269) (3000, 3.12667) (3100,
   3.12258) (3200, 3.12125) (3300, 3.11273) (3400, 3.11765) (3500,
   3.12571) (3600, 3.12889) (3700, 3.13297) (3800, 3.13579) (3900,
   3.13846) (4000, 3.144) (4100, 3.14537) (4200, 3.14667) (4300,
   3.14977) (4400, 3.15364) (4500, 3.14756) (4600, 3.1513) (4700,
   3.15319) (4800, 3.16083) (4900, 3.16163) (5000, 3.1616)
```

4 График №2 (относительное отклонение (в %) приближенного значения числа π от точного в зависимости от общего числа точек N)



5 Исходные данные для построение графика №2

```
(100, 3.1324) (200, 1.85916) (300, 1.85916) (400, 0.267614) (500,
    1.60452) (600, 1.22254) (700, 0.0402499) (800, 0.108459) (900,
    1.01034) (1000, 0.840572) (1100, 1.04892) (1200, 1.11644) (1300,
    1.36946) (1400, 0.767815) (1500, 0.840572) (1600, 0.665502) (1700,
    0.66082) (1800, 0.302982) (1900, 0.116836) (2000, 0.267614) (2100,
    0.222141) (2200, 0.296551) (2300, 0.0783749) (2400, 0.209851) (2500,
     0.0761605) (2600, 0.295549) (2700, 0.310059) (2800, 0.732788)
    (2900, 0.467791) (3000, 0.475109) (3100, 0.605171) (3200, 0.647527)
    (3300, 0.918814) (3400, 0.762212) (3500, 0.505424) (3600, 0.404373)
    (3700, 0.274373) (3800, 0.184721) (3900, 0.0996665) (4000,
    0.0766282) (4100, 0.120105) (4200, 0.161511) (4300, 0.260212) (4400,
     0.383363) (4500, 0.189805) (4600, 0.309133) (4700, 0.369202) (4800,
     0.61245) (4900, 0.637893) (5000, 0.636854)
```

6 Выводы о проделанной работе

Как можно заметить из графиков, наилучшая апроксимация числа π достигается при значении 700, остальные, также хорошие апроксимации достигаются при значении 800, а также примерно 2500 и 4000. Примечательно, что с увеличением N апроксимация не стремится к идеалу, что с одной стороны странно, а с другой стороны это вполне может быть, так как результаты эксперимента напрямую зависят от генератора рандомных точек, и если взять другой генератор, то мы можем как улучшить текущий результат, так и ухудшить. Так или иначе, мы получили вполне адекватную апроксимацию числа π. У нас нет явных выбросов, и самая большая относительная погрешность, которой мы достигли, составляет примерно 3%