



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Новосибирский государственный технический университет»

НГТУ



НЭТИ

Кафедра прикладной математики

Курсовой проект
по дисциплине «Численные методы»



ФПМИ

ГРУППА

ПМ-92

ВАРИАНТ

21

СТУДЕНТ

ГЛУШКО ВЛАДИСЛАВ

ПРЕПОДАВАТЕЛЬ

СОЛОВЕЙЧИК ЮРИЙ ГРИГОРЬЕВИЧ

Новосибирск

1 Условие задачи

Формулировка задачи

МКЭ для двумерной краевой задачи для эллиптического уравнения в декартовой системе координат. Базисные функции линейные на треугольниках. Краевые условия всех ти-пов. Коэффициент разложить по линейным базисным функциям. Матрицу СЛАУ генерировать в разреженном строчном формате. Для решения СЛАУ использовать МСГ или ЛОС с неполной факторизацией.

Постановка задачи

Эллиптическая краевая задача для функции u определяется дифференциальным уравнением

$$-div(\lambda \text{gradu}) + \gamma u = f$$

заданным в некоторой области Ω с границей $S = S_1 \cup S_2 \cup S_3$ и краевыми условиями

2 Текст программы

3 Тестирование

4 Выводы

```
1  #include "argparse/argparse.hpp"
2  #include "timer/cxxtimer.hpp"
3  #include "LOS/LOS.hpp"
4  #include "FEM.hpp"
5
6  int main(int argc, char* argv[]) {
7      using namespace ::Log;
8      using ::std::chrono::milliseconds;
9
10     argparse::ArgumentParser program("FEM", "1.0.0");
11     program.add_argument("-i", "--input").required().help("path to input files");
12     program.add_argument("-o", "--output").required().help("path to output files");
13
14     try {
15         program.parse_args(argc, argv);
16
17         cxxtimer::Timer timer(true);
18         FEM fem (program.get<std::string>("-i"));
19         fem.writeFile(program.get<std::string>("-o"), 1E-14, 10000);
20         LOS<double> l(program.get<std::string>("-o"));
21         l.solve(Cond::HOLLESKY, true);
22         timer.stop();
23     }
```

```

24         l.printX();
25
26         std::cout << '\n' << "Milliseconds: "
27             << timer.count<milliseconds>() << '\n';
28         fem.printAll();
29         fem.printSparse();
30     }
31     catch(const std::runtime_error& err) {
32         Logger::append(getLog("argc != 3 (FEM -i input -o output)"));
33         std::cerr << err.what();
34         std::cerr << program;
35         std::exit(1);
36     }
37     return 0;
38 }

```