



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 генери-ровать 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

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

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

$$\begin{aligned} u|_{S_1} &= u_g \\ \lambda \frac{\partial u}{\partial n} \Big|_{S_2} &= \theta \\ \lambda \frac{\partial u}{\partial n} \Big|_{S_3} + \beta(u|_{S_3} - u_\beta) &= 0 \end{aligned}$$

В декартовой системе координат  $x, y$  это уравнение может быть записано в виде

$$-\frac{\partial}{\partial x} \left( \lambda \frac{\partial u}{\partial x} \right) - \frac{\partial}{\partial y} \left( \lambda \frac{\partial u}{\partial y} \right) + \gamma u = f$$

## Конечноэлементная дискретизация

Так как для решения задачи используются линейные базисные функции, то на каждом конечном элементе  $\Omega_k$  - треугольнике эти функции будут совпадать с функциями  $L_1(x, y), L_2(x, y), L_3(x, y)$ , такими, что  $L_1(x, y)$  равна единице в вершине  $(x_1, y_1)$  и нулю во всех остальных вершинах,  $L_2(x, y)$  равна единице в вершине  $(x_2, y_2)$  и нулю во всех остальных вершинах,  $L_3(x, y)$  равна единице в вершине  $(x_3, y_3)$  и нулю во всех остальных вершинах. Любая линейная на  $\Omega_k$  функция представима в виде линейной комбинации этих базисных линейных функций, коэффициентами будут значения функции в каждой из вершин треугольника  $\Omega_k$ . Таким образом, на каждом конечном элементе нам понадобятся три узла – вершины треугольника.

$$\psi_1 = L_1(x, y)$$

$$\psi_2 = L_2(x, y)$$

$$\psi_3 = L_3(x, y)$$

Учитывая построение  $L$ -функций, получаем следующие соотношения:

$$\begin{cases} L_1 + L_2 + L_3 = 1 \\ L_1 x_1 + L_2 x_2 + L_3 x_3 = x \\ L_1 y_1 + L_2 y_2 + L_3 y_3 = y \end{cases}$$

Т.е. имеем систему:

$$\begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} \cdot \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} = \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$$

Отсюда находим коэффициенты линейных функций  $L_1(x, y), L_2(x, y), L_3(x, y)$

$$L_i = a_0^i + a_1^i x + a_2^i y, i = \overline{1, 3}$$

$$\begin{pmatrix} \alpha_0^1 & \alpha_1^1 & \alpha_2^1 \\ \alpha_0^2 & \alpha_1^2 & \alpha_2^2 \\ \alpha_0^3 & \alpha_1^3 & \alpha_2^3 \end{pmatrix} = D^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}^{-1}$$

## 2 Текст программы

## 3 Тестирование

## 4 Выводы

```

1  #include "argparse/argparse.hpp"           ///  $\frac{a+b}{2}$ 
2  #include "timer/cxxtimer.hpp"             ///  $\frac{a+b}{2}$ 
3  #include "LOS/LOS.hpp"                    ///  $\frac{a+b}{2}$ 
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 }

```