Новосибирский государственный технический университет
Кафедра прикладной математики

Курсовой проект

Метод конечных элементов для эллиптических задач

ФПМИ, ПМ-24

Параскун И.

преподаватели

Рояк М. Э., д.т.н., профессор

Сивенкова А. П., ассистент

Новосибирск, 2024

Содержание

1 Постановка задачи

Построить МКЭ для уравнения эллиптического типа

$$-\nabla(\lambda\nabla u) + \gamma u = f,\ f = \frac{\partial g}{\partial x} \tag{1}$$

в декартовой системе координат

$$-\frac{\partial}{\partial x}(\lambda\frac{\partial u}{\partial x}) - \frac{\partial}{\partial y}(\lambda\frac{\partial u}{\partial y}) - \frac{\partial}{\partial z}(\lambda\frac{\partial u}{\partial z}) + \gamma u = f \tag{2}$$

с учётом следующих условий:

- Краевые условия всех типов:

$$u|_{S_1} = u_g \tag{3}$$

$$\lambda\frac{\partial u}{\partial n}\bigg|_{S_2} = \theta \tag{4}$$

$$\lambda\frac{\partial u}{\partial n}\bigg|_{S_3} + \beta(u|_{S_3} - u\beta) = 0 \tag{5}$$

- Базисные функции - линейные на прямоугольных параллелепипедах.
- Коэффициент диффузии - кусочно-постоянная функция.
- Матрицы в разреженном строчном формате.
- Решение СЛАУ с использованием ЛОС с ILU факторизацией.

2 Теоретическая часть

2.1 Вариационная постановка

Эквивалентная вариационная постановка в форме уравнения Галёркина для уравнения 1:

$$\begin{split}\int_\Omega \lambda\nabla u \cdot \nabla v_0 d\Omega + \int_\Omega \gamma u v_0 d\Omega + \int_{S_3} \beta u v_0 dS = \\ = \int_\Omega f v_0 d\Omega + \int_{S_2} \theta v_0 dS + \int_{S_3} \beta u_\beta v_0 dS, \quad \forall v_0 \in H_0^1\end{split} \tag{6}$$

Аппроксимация уравнения 6 на конечномерных подпространствах V_g^h и V_0^h получается заменой функций $u \in H_g^1$ и $v_0 \in H_0^1$ на функции $u^h \in V_g^h$ и $v_0^h \in V_0^h$ соответственно:

$$\begin{split}\int_\Omega \lambda\nabla u^h \cdot \nabla v_0^h d\Omega + \int_\Omega \gamma u^h v_0^h d\Omega + \int_{S_3} \beta u^h v_0^h dS = \\ = \int_\Omega f v_0^h d\Omega + \int_{S_2} \theta v_0^h dS + \int_{S_3} \beta u_\beta v_0^h dS, \quad \forall v_0^h \in V_0^h\end{split} \tag{7}$$

2.2 Конечноэлементная СЛАУ

Раскладывая функции u^h и v_0^h по базису, переходим к конечноэлементной СЛАУ

$$\begin{aligned} \sum_{j=1}^{n}\Big(\int_{\Omega}\lambda\nabla\psi_j\cdot\nabla\psi_i d\Omega+\int_{\Omega}\gamma\psi_j\psi_i d\Omega+\int_{S_3}\beta\psi_j\psi_i dS\Big)q_j = \\ =\int_{\Omega}f\psi_i d\Omega+\int_{S_2}\theta\psi_i dS+\int_{S_3}\beta u_\beta\psi_i dS, \quad i\in N_0 \end{aligned} \tag{8}$$

$$\sum_{j}(q_j\psi_j)\Big|_{S_1}=u_g \tag{9}$$

или в матричном виде

$$\mathbf{Aq}=\mathbf{b} \tag{10}$$

где компоненты матрицы $\mathbf{A}$ и вектора $\mathbf{b}$ определяются соотношениями

$$A_{ij}=\begin{cases}\int_{\Omega}\lambda\nabla\psi_j\cdot\nabla\psi_i d\Omega+\int_{\Omega}\gamma\psi_j\psi_i d\Omega+\int_{S_3}\beta\psi_j\psi_i dS & i\in N_0\\ \delta_{ij} & i\notin N_0\end{cases} \tag{11}$$

$$b_i=\begin{cases}\int_{\Omega}f\psi_i d\Omega+\int_{S_2}\theta\psi_i dS+\int_{S_3}\beta u_\beta\psi_i dS & i\in N_0\\ u_g(\mathbf{x}_i) & i\notin N_0\end{cases} \tag{12}$$

Преобразование в сумму интегралов по конечным элементам даёт более удобную форму записи:

$$G_{ij}=\int_{\Omega}\lambda\nabla\psi_i\cdot\nabla\psi_j d\Omega=\sum_k\int_{\Omega_k}\lambda\nabla\psi_i\cdot\nabla\psi_j d\Omega \tag{13}$$

$$M_{ij}=\int_{\Omega}\gamma\psi_i\psi_j d\Omega=\sum_k\int_{\Omega_k}\gamma\psi_i\psi_j d\Omega \tag{14}$$

$$M_{ij}^{S_3}=\int_{S_3}\beta\psi_i\psi_j dS=\sum_l\int_{S_3^l}\beta\psi_i\psi_j dS \tag{15}$$

$$b_i=\int_{\Omega}f\psi_i d\Omega=\sum_k\int_{\Omega_k}f\psi_i d\Omega \tag{16}$$

$$b_i^{S_2}=\int_{S_2}\theta\psi_i dS=\sum_p\int_{S_2^p}\theta\psi_i dS \tag{17}$$

$$b_i^{S_3}=\int_{S_3}\beta u_\beta\psi_i dS=\sum_l\int_{S_3^l}\beta u_\beta\psi_i dS \tag{18}$$

2.3 Трилинейные базисные функции

Трилинейные базисные функции на прямоугольном параллелепипеде Ω_{psr} строятся следующим образом:

$$X_1(x) = \frac{x_{p+1} - x}{h_x}, \qquad X_2(x) = \frac{x - x_p}{h_x}, \qquad h_x = x_{p+1} - x_p, \tag{19}$$
$$Y_1(y) = \frac{y_{s+1} - y}{h_y}, \qquad Y_2(y) = \frac{y - y_s}{h_y}, \qquad h_y = y_{s+1} - y_s, \tag{20}$$
$$Z_1(z) = \frac{z_{r+1} - z}{h_z}, \qquad Z_2(z) = \frac{z - z_r}{h_z}, \qquad h_z = z_{r+1} - z_r. \tag{21}$$

Локальные базисные функции представляются в виде произведения функций 19, 20 и 21:

$$\hat{\psi}_i = X_{\mu(i)} Y_{\nu(i)} Z_{\vartheta(i)} \tag{22}$$

где

$$\mu(i) = ((i-1)\ mod\ 2) + 1 \tag{23}$$
$$\nu(i) = \left(\left[\frac{i-1}{2}\right] mod\ 2\right) + 1 \tag{24}$$
$$\vartheta(i) = \left[\frac{i-1}{4}\right] + 1 \tag{25}$$

2.4 Элементы локальных матриц

Выражение для вычисления компонент локальных матриц на прямоугольном параллелепипеде Ω_{psr}:

$$\hat{G}_{ij} = \int_{x_p}^{x_{p+1}} \int_{y_s}^{y_{s+1}} \int_{z_r}^{z_{r+1}} \lambda(\frac{\partial \hat{\psi}_i}{\partial x}\frac{\partial \hat{\psi}_j}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial y}\frac{\partial \hat{\psi}_j}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial z}\frac{\partial \hat{\psi}_j}{\partial z}) dxdydz \tag{26}$$

$$\hat{M}_{ij} = \int_{x_p}^{x_{p+1}} \int_{y_s}^{y_{s+1}} \int_{z_r}^{z_{r+1}} \gamma \hat{\psi}_i \hat{\psi}_j dxdydz \tag{27}$$

$$\hat{M}_{ij}^{S_3} = \int_{\xi_1}^{\xi_2} \int_{\zeta_1}^{\zeta_2} \beta \psi_i \psi_j d\xi d\zeta \tag{28}$$

Учитывая вид базисных функций 22:

$$\hat{\mathbf{G}}^x = \frac{1}{h_x}\mathbf{G}^c \qquad \hat{\mathbf{M}}^x = h_x \mathbf{M}^c$$
$$\hat{\mathbf{G}}^y = \frac{1}{h_y}\mathbf{G}^c \qquad \hat{\mathbf{M}}^y = h_y \mathbf{M}^c$$
$$\hat{\mathbf{G}}^z = \frac{1}{h_z}\mathbf{G}^c \qquad \hat{\mathbf{M}}^z = h_z \mathbf{M}^c$$

где

$$\mathbf{G}^c = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \qquad \mathbf{M}^c = \frac{1}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \tag{29}$$

Тогда

$$\hat{G}_{ij} = \hat{\lambda}(\hat{G}^x_{\mu(i)\mu(j)}\hat{M}^y_{\nu(i)\nu(j)}\hat{M}^z_{\vartheta(i)\vartheta(j)} + \hat{M}^x_{\mu(i)\mu(j)}\hat{G}^y_{\nu(i)\nu(j)}\hat{M}^z_{\vartheta(i)\vartheta(j)} + \hat{M}^x_{\mu(i)\mu(j)}\hat{M}^y_{\nu(i)\nu(j)}\hat{G}^z_{\vartheta(i)\vartheta(j)}) \tag{30}$$

$$\hat{M}_{ij} = \hat{\gamma}(\hat{M}^x_{\mu(i)\mu(j)}\hat{M}^y_{\nu(i)\nu(j)}\hat{M}^z_{\vartheta(i)\vartheta(j)}) \tag{31}$$

$$\hat{M}^{S_3}_{ij} = \hat{\beta}(\hat{M}^{\xi}_{\mu(i)\mu(j)}\hat{M}^{\zeta}_{\nu(i)\nu(j)}) \tag{32}$$

2.5 Элементы локальных векторов

Выражение для вычисления компонент локальных векторов на прямоугольном параллелепипеде Ω_{psr}:

$$\begin{aligned}\hat{b}_i &= \int_{x_p}^{x_{p+1}}\int_{y_s}^{y_{s+1}}\int_{z_r}^{z_{r+1}} f\hat{\psi}_i dxdydz \\ &= \sum_j \hat{q}^g_j \int_{x_p}^{x_{p+1}}\int_{y_s}^{y_{s+1}}\int_{z_r}^{z_{r+1}} \frac{\partial\hat{\psi}_j}{\partial x}\hat{\psi}_i dxdydz\end{aligned} \tag{33}$$

Учитывая вид базисных функций 22:

$$\begin{aligned}\hat{b}_i = \sum_j \hat{q}^g_j \int_{x_p}^{x_{p+1}} X_{\mu(i)}\frac{\partial X_{\mu(j)}}{\partial x}dx \int_{y_s}^{y_{s+1}} Y_{\nu(i)}Y_{\nu(j)}dy \int_{z_r}^{z_{r+1}} Z_{\vartheta(i)}Z_{\vartheta(j)}dz = \\ = \sum_j \hat{q}^g_j X_{\mu(i)\mu(j)}\hat{M}^y_{\nu(i)\nu(j)}\hat{M}^z_{\vartheta(i)\vartheta(j)}\end{aligned} \tag{34}$$

где

$$\mathbf{X} = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \tag{35}$$

Для правой части, заданной в явном виде:

$$\begin{aligned}\hat{b}_i = \sum_j \hat{f}_j \int_{x_p}^{x_{p+1}} X_{\mu(i)}Y_{\mu(j)}dx \int_{y_s}^{y_{s+1}} Y_{\nu(i)}Y_{\nu(j)}dy \int_{z_r}^{z_{r+1}} Z_{\vartheta(i)}Z_{\vartheta(j)}dz = \\ = \sum_j \hat{f}_j \hat{M}^x_{\mu(i)\mu(j)}\hat{M}^y_{\nu(i)\nu(j)}\hat{M}^z_{\vartheta(i)\vartheta(j)}\end{aligned} \tag{36}$$

Вклад краевых условий 17 и 18:

$$\hat{b}^{S_2}_i = \int_{\xi_1}^{\xi_2}\int_{\zeta_1}^{\zeta_2} \hat{\theta}\psi_i d\xi d\zeta \tag{37}$$

$$\hat{b}^{S_3}_i = \int_{\xi_1}^{\xi_2}\int_{\zeta_1}^{\zeta_2} \hat{\beta}\hat{u}_{\beta}\psi_i d\xi d\zeta \tag{38}$$

$$\hat{g}^{ip} = \sum_{i=1}^{8} g(\hat{x}_i, \hat{y}_i, \hat{z}_i)\hat{\psi}_i(x,y,z) = \sum_{i=1}^{8} \hat{q}_i^g \hat{\psi}_i(x,y,z) \tag{39}$$

$$\hat{\theta}^{ip} = \sum_{i=1}^{4} \hat{\theta}(\hat{x}_i, \hat{y}_i, \hat{z}_i)\hat{\psi}_i(x,y,z) = \sum_{i=1}^{4} \hat{\theta}_i \hat{\psi}_i(x,y,z) \tag{40}$$

$$\hat{u}_\beta^{ip} = \sum_{i=1}^{4} \hat{u}_\beta(\hat{x}_i, \hat{y}_i, \hat{z}_i)\hat{\psi}_i(x,y,z) = \sum_{i=1}^{4} \hat{u}_{\beta i} \hat{\psi}_i(x,y,z) \tag{41}$$

$$\begin{aligned} \hat{b}_i^{S_2} = \hat{\theta}_1 \hat{M}^{\xi}_{\mu(i)\mu(1)} \hat{M}^{\chi}_{\nu(i)\nu(1)} + \hat{\theta}_2 \hat{M}^{\xi}_{\mu(i)\mu(2)} \hat{M}^{\chi}_{\nu(i)\nu(2)} + \\ + \hat{\theta}_3 \hat{M}^{\xi}_{\mu(i)\mu(3)} \hat{M}^{\chi}_{\nu(i)\nu(3)} + \hat{\theta}_4 \hat{M}^{\xi}_{\mu(i)\mu(4)} \hat{M}^{\chi}_{\nu(i)\nu(4)} \end{aligned} \tag{42}$$

$$\begin{aligned} \hat{b}_i^{S_3} = \hat{\beta}(\hat{u}_{\beta 1} \hat{M}^{\xi}_{\mu(i)\mu(1)} \hat{M}^{\chi}_{\nu(i)\nu(1)} + \hat{u}_{\beta 2} \hat{M}^{\xi}_{\mu(i)\mu(2)} \hat{M}^{\chi}_{\nu(i)\nu(2)} + \\ + \hat{u}_{\beta 3} \hat{M}^{\xi}_{\mu(i)\mu(3)} \hat{M}^{\chi}_{\nu(i)\nu(3)} + \hat{u}_{\beta 4} \hat{M}^{\xi}_{\mu(i)\mu(4)} \hat{M}^{\chi}_{\nu(i)\nu(4)}) \end{aligned} \tag{43}$$

3 Описание разработанных программ

3.1 Узел

```
// Узел
struct vtx {
  // Координата по X
  double x;

  // Координата по Y
  double y;

  // Координата по Z
  double z;
};

// Подпрограмма выделения памяти под узел
struct vtx* vtx_new();

// Подпрограмма чтения узла из строки
struct vtx* vtx_get(const char* buf);
```

3.2 Конечный элемент

```
// Прямоугольный параллелепипед
struct hex {
  // Список из глобальных номеров узлов, составляющих данный элемент
  int vtx[8];

  // Параметры
  struct {
    // Функция правой части
```

```
    double (*f)(struct vtx*);

    double lam;
    double gam;
  } pps;

  // Локальные матрица и вектор правой части
  struct {
    // Локальная матрица
    struct mtx* m;

    // Локальная правая часть
    struct vec* b;
  } loc;
};

// Подпрограмма выделения памяти под элемент
struct hex* hex_new();

// Подпрограмма чтения элемента из строки
//  fun - список глобальных функций
struct hex* hex_get(const char* buf, double (**fun)(struct vtx*));

// Подпрограмма вычисления локальной матрицы и вектора правой части
//  v - глобальный список узлов
int hex_evo(struct hex* h, struct vtx** v);

// Подпрограмма занесения локальной матрицы
// и вектора правой части в глобальную СЛАУ
```

3.3 Грань

```
// Тип краевых условий
//  DIR - краевые условия первого типа (Дирихле)
//  NEU - краевые условия второго типа (Неймана)
//  ROB - краевые условия третьего типа (Робена)
enum type { DIR, NEU, ROB };

// Краевые условия
struct cnd {
  // Тип
  enum type type;

  // Параментры
  union pps {
    // Параметры для условий первого рода
    struct {
      double (*tmp)(struct vtx* v);
```

```
    } dir;

    // Параметры для условий второго рода
    struct {
      double (*tta)(struct vtx* v);
    } neu;

    // Параметры для условий третьего рода
    struct {
      double (*tmp)(struct vtx* v);
      double bet;
    } rob;
  } pps;
};

// Грань
struct fce {
  // Список из глобальных номеров узлов, составляющих данную грань
  int vtx[4];

  // Краевые условия
  struct cnd cnd;

  // Локальные матрица и / или вектор правой части
  union loc {
    // Локальный вектор правой части для условий второго рода
    struct {
      struct vec* b;
    } neu;

    // Локальные матрица и вектор правой части для условий третьего рода
    struct {
      struct vec* b;
      struct mtx* m;
    } rob;
  } loc;
};

// Подпрограмма выделения памяти под грань
struct fce* fce_new(enum type type);

// Подпрограмма чтения грани из строки
//  fun - список глобальных функций
struct fce* fce_get(const char* buf, double (**fun)(struct vtx*));

// Подпрограмма вычисления локальной матрицы и вектора правой части
//  v - глобальный список узлов
int fce_evo(struct fce* f, struct vtx** v);
```

```
// Подпрограмма занесения локальной матрицы
// и вектора правой части в глобальную СЛАУ
//  a - матрица глобальной СЛАУ в строчно-столбцовом формате
//  b - вектор правой части глобальной СЛАУ
int fce_mov(struct fce* f, struct mtx_csj* a, struct vec* b);
```

3.4 Структура алгоритма

```
// Структура алгоритма
struct fem {
  int vs; // Количество вершин
  int hs; // Количество элементов
  int fs; // Количество граней

  // Список вершин
  struct vtx** vtx;

  // Список элементов
  struct hex** hex;

  // Список граней
  struct fce** fce;

  // Параметры
  struct {
    // Список функций
    double (**fun)(struct vtx*);
  } pps;

  // Матрица СЛАУ
  struct mtx_csj* a;

  // Вектор правой части СЛАУ
  struct vec* b;
};

// Подпрограмма выделения памяти
struct fem* fem_new(double (**fun)(struct vtx*));

// Подпрограмма инициализации из файла
int fem_get(FILE* obj, struct fem* fem);

// Подпрограмма вычисления локальных матриц и векторов
// конечных элементов и граней с заданными краевыми
// условиями
int fem_evo(struct fem* fem);
```

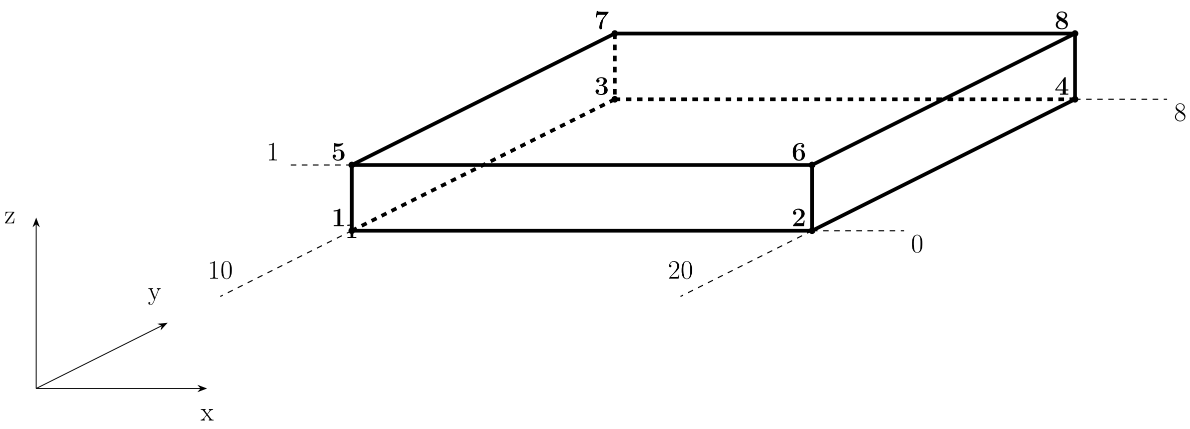
```
// Подпрограмма сборки СЛАУ
int fem_asm(struct fem* fem);

// Подпрограмма решения СЛАУ
//  q - результирующий вектор
int fem_slv(struct fem* fem, struct vec* q);
```

4 Описание тестирования программ

4.1 Тестирование на одном конечном элементе

4.1.1 Описание задачи

Простейшая расчетная область для тестирования трилинейных элементов выглядит следующим образом:



где

$$
\begin{aligned}
S_2^1 &= [1, 3, 5, 7] \\
S_2^2 &= [3, 4, 7, 8] \\
S_2^3 &= [5, 6, 7, 8] \\
S_3^1 &= [2, 4, 6, 8] \\
S_3^2 &= [1, 2, 5, 6] \\
S_3^3 &= [1, 2, 3, 4]
\end{aligned}
$$

Значения коэффициентов λ и γ выбраны в виде:

$$\lambda = 5 \qquad \gamma = 0.4 \tag{44}$$

Решение - полином вида

$$u = 5 + 0.2x + y + 30z + 0.5xy + xz + 10yz + xyz \tag{45}$$

Исходя из этого

$$f = 0.4(5 + 0.2x + y + 30z + 0.5xy + xz + 10yz + xyz) \tag{46}$$

$$g = 2x + 0.04x^2 + 0.4xy + 12xz + 0.1x^2y + 0.2x^2z + 4xyz + 0.2x^2yz \tag{47}$$

$$\theta|_{S_2^1} = -1 - 2.5y - 5z - 5yz \tag{48}$$

$$\theta|_{S_2^2} = 5 + 2.5x + 50z + 5xz \tag{49}$$

$$\theta|_{S_2^3} = 150 + 5x + 50y + 5xy \tag{50}$$

При условии, что $\beta = 10$:

$$u_\beta|_{S_3^1} = 9.1 + 11.25y + 50.5z + 30.5yz \tag{51}$$

$$u_\beta|_{S_3^2} = 4.5 - 0.05x + 25z + 0.5xz \tag{52}$$

$$u_\beta|_{S_3^3} = -10 - 0.3x - 4y \tag{53}$$

4.1.2 Подпрограмма тестирования

```
class SolidStateEquationSingleElementTest : public testing::Test {
 public:
  static double f(struct vtx* v) {
    double x = v->x;
    double y = v->y;
    double z = v->z;

    return 0.4 * (5 + 0.2 * x + y + 30 * z + 0.5 * x * y + x * z + 10 * y * z +
                  x * y * z);
  }

  static double g(struct vtx* v) {
    double x = v->x;
    double y = v->y;
    double z = v->z;

    return 2 * x + 0.04 * x * x + 0.4 * x * y + 12 * x * z + 0.1 * x * x * y +
           0.2 * x * x * z + 4 * x * y * z + 0.2 * x * x * y * z;
  }

  static double tta1(struct vtx* v) {
    double y = v->y;
    double z = v->z;

    return -1 - 2.5 * y - 5 * z - 5 * y * z;
  }

  static double tta2(struct vtx* v) {
    double x = v->x;
    double z = v->z;
```

```
    return 5 + 2.5 * x + 50 * z + 5 * x * z;
  }

  static double tta3(struct vtx* v) {
    double x = v->x;
    double y = v->y;

    return 150 + 5 * x + 50 * y + 5 * x * y;
  }

  static double tmp1(struct vtx* v) {
    double y = v->y;
    double z = v->z;

    return 9.1 + 11.25 * y + 50.5 * z + 30.5 * y * z;
  }

  static double tmp2(struct vtx* v) {
    double x = v->x;
    double z = v->z;

    return 4.5 - 0.05 * x + 25 * z + 0.5 * x * z;
  }

  static double tmp3(struct vtx* v) {
    double x = v->x;
    double y = v->y;

    return -10 - 0.3 * x - 4 * y;
  }

  double (*fun[7])(struct vtx*) = {&f,    &tta1, &tta2, &tta3,
                                          &tmp1, &tmp2, &tmp3};
};

TEST_F(SolidStateEquationSingleElementTest, GeneralTest) {
  struct fem* fem = fem_new(fun);

  fem_get(Env::obj, fem);
  fem_evo(fem);
  fem_asm(fem);

  struct vec* q = vec_new(fem->vs);

  fem_slv(fem, q);

  mtx_csj_put(&Env::pkt.mtx, fem->a);
  vec_put(Env::pkt.pkt.f, fem->b);
```

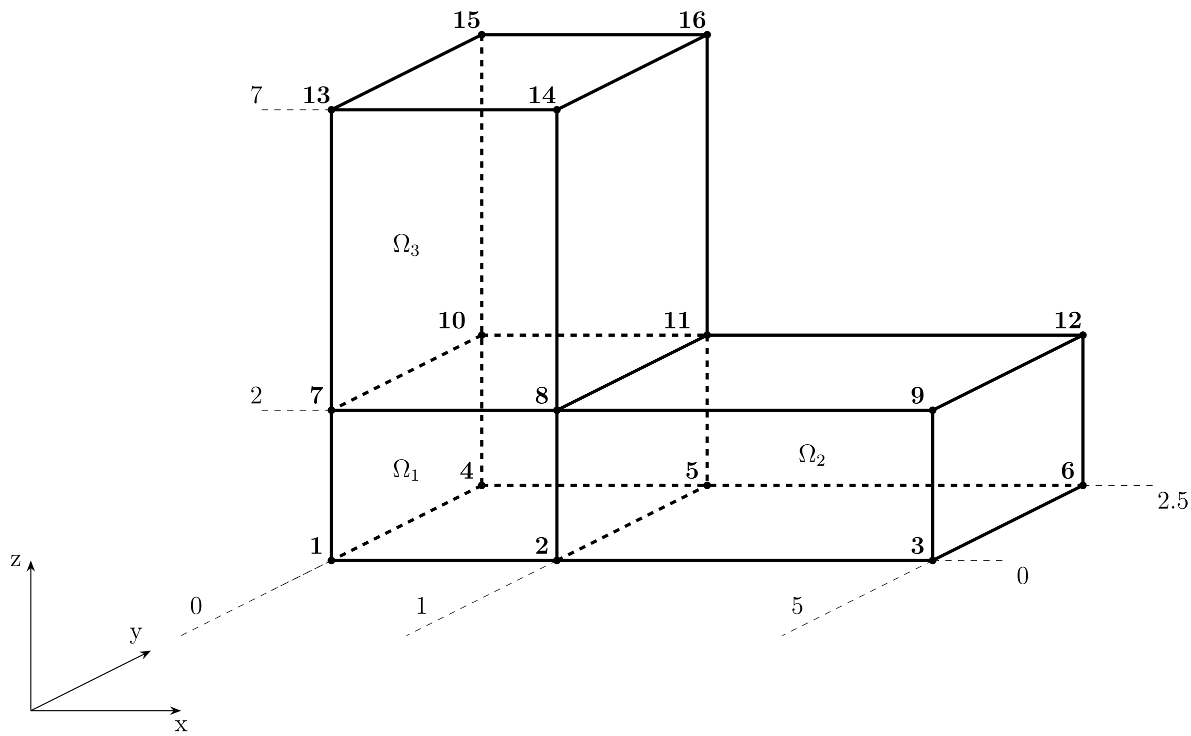
```
  vec_put(Env::pkt.pkt.x, q);

  fem_cls(fem);
  vec_cls(q);
}
```

4.1.3 Результат

```
7.0000000e+00
9.0000000e+00
5.5000000e+01
9.7000000e+01
4.7000000e+01
5.9000000e+01
2.5500000e+02
3.8700000e+02
```

4.2 Тестирование на сетке с разрывом коэффициентов

4.2.1 Описание задачи



Значения коэффициентов λ и γ выбраны в виде:

$$\lambda = \begin{cases} 1, & \Omega^1 \\ 2, & \Omega^2 \\ \frac{2}{3}, & \Omega^3 \end{cases} \qquad \gamma = \begin{cases} 4, & \Omega^1 \\ 0, & \Omega^2 \\ 1, & \Omega^3 \end{cases}$$

Решение - полином вида

$$u = \begin{cases} 2x(yz+z+y+1), & \Omega^1 \\ 1+x+y+z+xy+xz+yz+xyz, & \Omega^2 \\ 3xz(y+1), & \Omega^3 \end{cases}$$

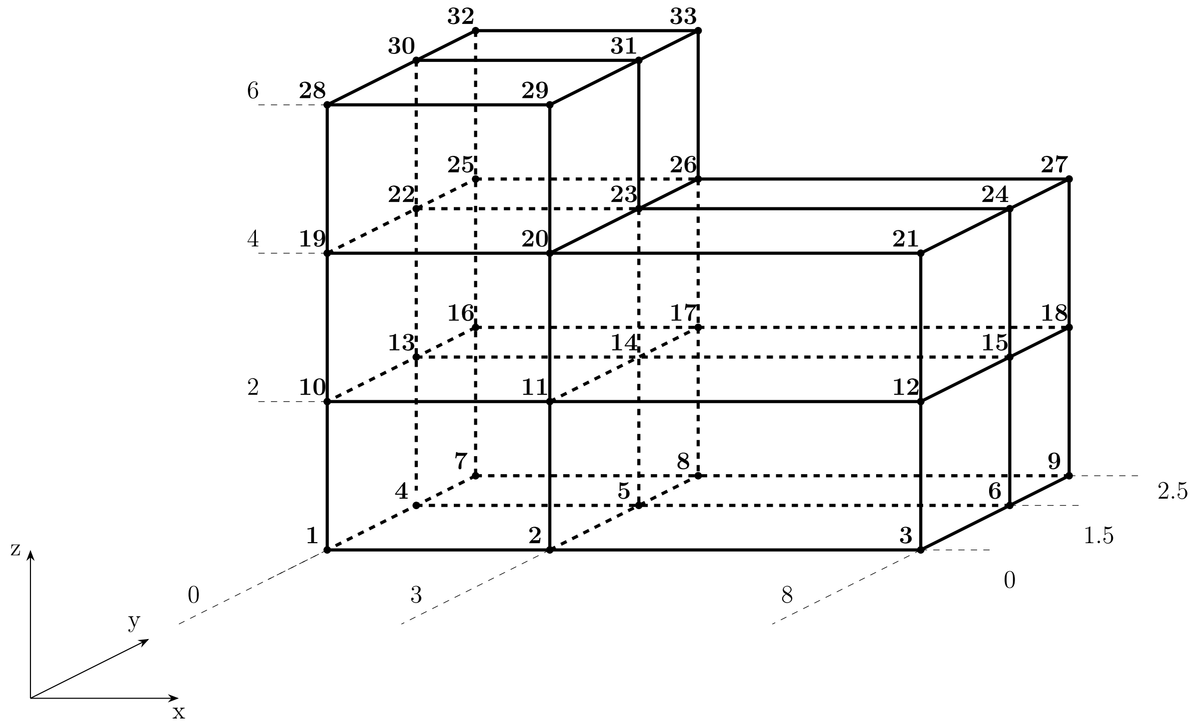
При условии, что $\beta = 2$:

$$\begin{array}{ll}
S_1 = [1,4,7,10,13,15] & u|_{S_1} = 0 \\
S_2^1 = [1,2,7,8] & \theta|_{S_2^1} = -2x(z+1) \\
S_2^2 = [7,8,13,14] & \theta|_{S_2^2} = -2xz \\
S_2^3 = [2,3,8,9] & \theta|_{S_2^3} = -2(1+x+z+xz) \\
S_2^4 = [3,6,9,12] & \theta|_{S_2^4} = 2(1+y+z+yz) \\
S_2^5 = [8,9,11,12] & \theta|_{S_2^5} = 2(1+x+y+xy) \\
S_2^6 = [8,11,14,16] & \theta|_{S_2^6} = 2z(1+y) \\
S_2^7 = [13,14,15,16] & \theta|_{S_2^7} = 2x(1+y) \\
S_2^8 = [1,2,4,5] & \theta|_{S_2^8} = -2x(1+y) \\
S_2^9 = [2,3,5,6] & \theta|_{S_2^9} = -2(1+x+y+xy) \\
S_3^1 = [4,5,10,11] & u_\beta|_{S_3^1} = 8x(z+1) \\
S_3^2 = [10,11,15,16] & u_\beta|_{S_3^2} = 11.5xz \\
S_3^3 = [5,6,11,12] & u_\beta|_{S_3^3} = 4.5(1+x+z+xz)
\end{array}$$

$$f = \begin{cases} 8x(yz+z+y+1), & \Omega^1 \\ 0, & \Omega^2 \\ 3xz(y+1), & \Omega^3 \end{cases}$$

4.3 Тестирование на сетке с внутренним узлом

4.3.1 Описание задачи



где

$$S_1^1 = [1, 2, 28, 29]$$
$$S_1^2 = [2, 3, 20, 21]$$
$$S_1^3 = [7, 8, 32, 33]$$
$$S_1^4 = [1, 3, 7, 9]$$
$$S_1^4 = [8, 9, 26, 27]$$
$$S_2^1 = [1, 7, 28, 32]$$
$$S_2^2 = [3, 9, 21, 27]$$
$$S_3^1 = [28, 29, 32, 33]$$
$$S_3^2 = [20, 26, 29, 33]$$
$$S_3^3 = [20, 21, 26, 27]$$

Значения коэффициентов λ и γ выбраны в виде:

$$\lambda = 5 \qquad \gamma = 0.4 \tag{54}$$

Решение - полином вида

$$u = 5 + 0.2x + y + 30z + 0.5xy + xz + 10yz + xyz \tag{55}$$

Исходя из этого

$$f = 0.4(5 + 0.2x + y + 30z + 0.5xy + xz + 10yz + xyz) \tag{56}$$

$$g = 2x + 0.04x^2 + 0.4xy + 12xz + 0.1x^2y + 0.2x^2z + 4xyz + 0.2x^2yz \tag{57}$$

$$u|_{S_1} = 0 \tag{58}$$

$$\theta|_{S_2^1} = -1 - 2.5y - 5z - 5yz \tag{59}$$

$$\theta|_{S_2^2} = 1 + 2.5y + 5z + 5yz \tag{60}$$

При условии, что $\beta = 10$:

$$u_\beta|_{S_3^1} = 7xy + 6.7x + 66y + 200 \tag{61}$$

$$u_\beta|_{S_3^2} = 13.5yz + 2.75y + 33.5z + 5.7 \tag{62}$$

$$u_\beta|_{S_3^3} = 5xy + 4.7x + 46y + 140 \tag{63}$$

5 Проведенные исследования и выводы

6 Тексты основных модулей программ