

Курсовая работа по теме  
«Метод конечных элементов»

Дмитрий Джус

# Содержание

<b>1. Метод конечных элементов</b>	<b>4</b>
1.1. Введение . . . . .	4
1.2. Общая процедура МКЭ . . . . .	5
1.3. Стержневой элемент . . . . .	5
<b>2. Статическая задача</b>	<b>7</b>
2.1. Системы координат и нумерация . . . . .	7
2.2. Постановка задачи . . . . .	8
2.3. Преобразование уравнений равновесия . . . . .	9
2.4. Глобальная матрица жёсткости . . . . .	10
2.5. Решение . . . . .	11
<b>3. Динамическая задача</b>	<b>12</b>
3.1. Общий вид системы . . . . .	12
3.2. Собственные значения . . . . .	12
<b>4. Примеры и реализация метода</b>	<b>13</b>
4.1. Статическая задача . . . . .	16
4.2. Динамическая задача . . . . .	16
<b>5. Выводы и результаты</b>	<b>17</b>
<b>А. Информация о документе</b>	<b>18</b>
<b>Б. Список литературы</b>	<b>18</b>

## Предмет работы и цели

Настоящая курсовая работа посвящена методу конечных элементов. Дано вводное описание метода, рассмотрен стержневой конечный элемент. Изложены процедуры использования МКЭ для задач статики и динамики. Представлена компьютерная реализация метода на алгоритмическом языке.

Целью работы является знакомство с теоретическими основами МКЭ и способами реализации метода на практике.

# 1. Метод конечных элементов

## 1.1. Введение

Метод конечных элементов предназначен для решения задач прикладной физики, в том числе механики сплошных сред.

Суть МКЭ заключается в аппроксимации исследуемой непрерывной величины моделью, состоящей из набора *конечных элементов*, и последующем сведении исходной краевой задачи к системе алгебраических уравнений.

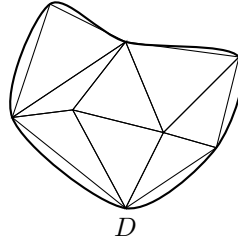


Рис. 1. Двумерная задача

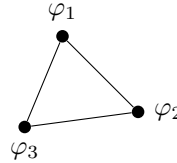


Рис. 2. Треугольный конечный элемент

На рис. 1 представлена двумерная область  $D$ , в которой решается некоторая краевая задача для функции  $\varphi(x, y)$ , которая в данном примере аппроксимирована треугольными конечными элементами. Изображённый на рис. 2 конечный элемент задаётся таким образом, что в нём выполняется соотношение

$$\varphi(x, y) = N_1(x, y)\varphi_1 + N_2(x, y)\varphi_2 + N_3(x, y)\varphi_3, \quad (1.1)$$

где  $\varphi_1, \varphi_2, \varphi_3$  — значения функции в узлах элемента, а  $N_1, N_2, N_3$  — непрерывные *функции формы*. В МКЭ в качестве неизвестных выступают значения  $\varphi_i$  в узлах конечных элементов, на которые разбита рассматриваемая область. Аналитический вид функций формы известен и выбирается исходя из требований к адекватности модели задаче.

Таким образом, МКЭ позволяет вычислить значение функции в любой точке рассматриваемой области  $D$  при известных узловых значениях. Этим он отличается от метода конечных разностей (см. [4]). Кроме того, МКЭ может применяться с нерегулярными сетками.

Существуют различные типы конечных элементов с разным количеством степеней свободы и свойствами. Конкретный тип элементов для задачи выбирается из различных соображений. В некоторых случаях (например, при рассмотрении ферм или трубопроводов), конструктивные элементы совпадают с конечными элементами. В общем случае конечные элементы выбираются так, чтобы обеспечить должную аппроксимацию, при обеспечении наименьшего уровня вычислительной сложности.

В дальнейшем более подробно будет рассмотрен стержневой элемент и его использование в МКЭ.

## 1.2. Общая процедура МКЭ

Применение метода к решению задач состоит из нескольких этапов:

- 1) Определение геометрии задачи;
- 2) Выбор типа используемых конечных элементов и задание их свойств;
- 3) Построение подходящей сетки для заданной геометрии;
- 4) Приложение граничных условий к модели, в результате чего получается система алгебраических уравнений;
- 5) Решение полученной системы, благодаря чему определяются значения изучаемой величины в узлах сетки.

Полученное решение и известные свойства элементов используются для последующего анализа: вычисления интересующих параметров, производных от найденных значений.

Например, МКЭ может быть применён для изучения воздействия известных внешних сил на некоторую сплошную конструкцию. Смещение точки конструкции дискретизируется с помощью конечно-элементной сетки, после чего накладываются граничные условия закрепления и нагрузок. Решение получившейся системы уравнений даёт набор смещений в узлах конечных элементов, после чего деформации и напряжения в конструкции могут быть вычислены с учётом физических свойств введённых элементов.

## 1.3. Стержневой элемент

Стержень — простой конечный элемент, особенно удобный для моделирования двух- и трёхмерных ферм. Формальное определение этого конечного элемента основано на следующих утверждениях (см. [2]):

- 1) Стержень прямой;
- 2) Материал стержня подчиняется закону Гука;
- 3) Силы прилагаются только к концам стержня;
- 4) Стержень может передавать только осевое воздействие.

К характеристикам стержня относятся длина  $L$ , площадь осевого сечения  $A$ , модуль упругости  $E$ . Рассматриваются зависимости осевых перемещений  $u = u(x)$ , относительной деформации  $\varepsilon = \varepsilon(x)$ , напряжения  $\sigma = \sigma(x)$  от координаты точки на оси стержня.

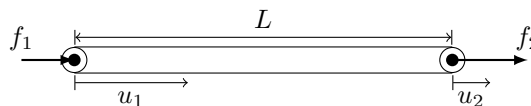


Рис. 3. Стержневой элемент

На рис. 3 изображён стержневой элемент длины  $L$  со смещениями в концах стержня  $u_1$  и  $u_2$ . К концам приложены силы  $f_1$  и  $f_2$ .

Считается, что перемещение  $u$  линейно изменяется вдоль оси стержня:

$$u(x) = \left(1 - \frac{x}{L}\right) u_1 + \left(\frac{x}{L}\right) u_2. \quad (1.2)$$

Таким образом, для стержня функции формы имеют вид:

$$\begin{aligned} N_1(x) &= 1 - \frac{x}{L}, \\ N_2(x) &= \frac{x}{L}. \end{aligned} \quad (1.3)$$

Из теории упругости известно, что  $\varepsilon = \frac{du}{dx}$  и  $\sigma = E\varepsilon$ . С учётом приближения (1.2) получаем:

$$\varepsilon = \frac{u_2 - u_1}{L}. \quad (1.4)$$

Соответствующая сила, действующая на стержень, выражается следующим образом:

$$P = A\sigma = AE\varepsilon = \frac{AE}{L}(u_2 - u_1).$$

Уравнение статического равновесия стержневого элемента:

$$\begin{cases} f_1 &= -\frac{AE}{L}(u_2 - u_1), \\ f_2 &= \frac{AE}{L}(u_2 - u_1). \end{cases} \quad (1.5)$$

В матричном виде оно может быть записано так:

$$\underbrace{\frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}_{[K]} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (1.6)$$

Матрица  $[K]$ , связывающая смещения узлов конечного элемента с приложенными к нему силами, называется *матрицей жёсткости* данного элемента.

Введём обозначение  $k = \frac{AE}{L}$ , с учётом которого матрица жёсткости стержня принимает вид:

$$[k] = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}. \quad (1.7)$$

## 2. Статическая задача

### 2.1. Системы координат и нумерация

В дальнейшем будем выделять две системы координат — *локальную*, оси которой ориентированы в соответствии с положением конечного элемента в пространстве, и *глобальную*, оси которой направлены одинаково для всех конечных элементов модели. На рис. ?? смещения узлов стержневого элемента в плоскости приведены в двух системах. Узловые смещения в глобальных координатах обозначаются большими буквами, в локальных — маленькими.

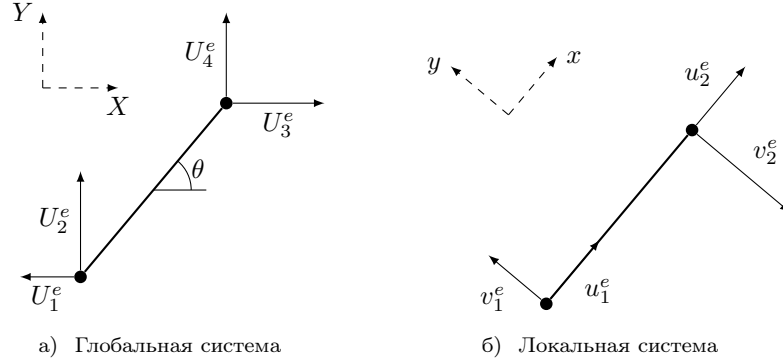


Рис. 4. Системы координат

Ориентация стержня на плоскости характеризуется углом  $\theta$  наклона элемента к глобальной оси  $X$ .

Связь локальных и глобальных координат выражается матричным соотношением:

$$\begin{bmatrix} u_1^e \\ v_1^e \\ u_2^e \\ v_2^e \end{bmatrix} = \underbrace{\begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \\ 0 & 0 & -\sin \theta & \cos \theta \end{bmatrix}}_{R_0} \begin{bmatrix} U_1^e \\ U_2^e \\ U_3^e \\ U_4^e \end{bmatrix} \quad (2.1)$$

На практике конечно-элементная сетка задаётся перечислением координат всех её узлов и перечислением элементов, соединяющих различные узлы сетки. Таким образом, каждый стержневой элемент определяется парой координат своих узлов  $(X_i, Y_i)$  и  $(X_j, Y_j)$ . Знание этих координат позволяет вычислить длину каждого узла:

$$L = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \quad (2.2)$$

Компоненты матрицы  $R_0$  для каждого элемента могут быть найдены из соотношений

$$\begin{aligned} \cos \theta &= \frac{X_j - X_i}{L}, \\ \sin \theta &= \frac{Y_j - Y_i}{L}. \end{aligned} \quad (2.3)$$

Итак, матрица жёсткости стержня (1.7) в глобальных координатах может быть определена при данных параметрах  $A, E$  и координатах узлов стержня.

Поскольку в модели обычно одновременно рассматривается множество конечных элементов, удобно ввести единую глобальную нумерацию для узловых смещений. В дальнейшем смещение узла  $\varphi_k$  в глобальном направлении  $X$  будем обозначать как  $U_{2k-1}$ , а в направлении  $Y$  —  $U_{2k}$ . Например, для стержня  $e_n$  с узлами  $\varphi_i$  и  $\varphi_j$  вектор смещений

$$U^{e[n]} = [U_1^e \quad U_2^e \quad U_3^e \quad U_4^e]$$

в глобальной нумерации будет иметь следующие компоненты:

$$[U_{2i-1} \quad U_{2i} \quad U_{2j-1} \quad U_{2j}]. \quad (2.4)$$

Таким образом, смещения  $N$  узлов описываются вектором длины  $2N$ . Аналогичные обозначения будем использовать для внешних сил, приложенных к узлам.

Для перехода от локальной нумерации к глобальной для каждого элемента  $e_n$  с узлами  $\varphi_i, \varphi_j$  введём *вектор связности*, описывающий глобальные индексы компонент узловых смещений данного элемента:

$$L^{[n]} = [2i-1 \quad 2i \quad 2j-1 \quad 2j] \quad (2.5)$$

Тогда можно видеть, что  $L_i^{[n]}$  есть индекс  $i$ -ой компоненты вектора  $U^{e[n]}$  в векторе  $U$ , то есть выполняется соотношение

$$U_{L_i^{[n]}} = U_i^{e[n]}. \quad (2.6)$$

Компоненты приложенных сил  $F_m$  используют аналогичную схему глобальной нумерации.

Надстрочным индексом  $e$  будем обозначать величины, использующие локальную нумерацию.

## 2.2. Постановка задачи

Рассмотрим схему применения метода конечных элементов для статического анализа конструкции типа фермы, изображённой на рис. 5. Она состоит из двух конечных элементов  $e_1$  и  $e_2$ .

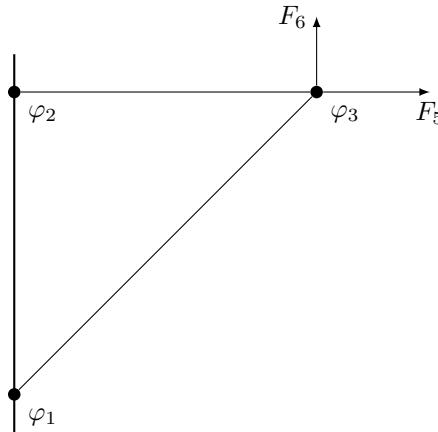


Рис. 5. Ферма из двух стержневых элементов

Узлы  $\varphi_1$  и  $\varphi_2$  жёстко закреплены. Узлы  $\varphi_3$  испытывает внешнюю нагрузку. Задача состоит в нахождении смещений и реакций в узлах модели.



Отметим, что в рассматриваемой модели смещения узлов вдоль локальной оси  $y$  ( $v_1^e, v_2^e$ ) не связаны с жёсткостью элемента, поскольку считается, что стержень передаёт только нагрузку, направленную вдоль его локальной оси  $x$ .

Применение метода конечных элементов к данной конструкции заключается в объединении условий равновесия (1.6) каждого элемента в общую систему уравнений, связывающую вектора узловых перемещений  $[U]$  и нагрузок  $[F]$ .

Рассмотрим порядок вывода системы уравнений. Пример расчёта с использованием МКЭ для представленной конструкции приведены в разделе 4.1.

Более общий вывод с использованием вариационных принципов дан в [1] и [3].

### 2.3. Преобразование уравнений равновесия

Для начала перепишем уравнение (1.6):

$$[k^e][u^e] = \begin{bmatrix} k^e & -k^e \\ -k^e & k^e \end{bmatrix} \begin{bmatrix} u_1^e \\ u_2^e \end{bmatrix} = \begin{bmatrix} f_1^e \\ f_2^e \end{bmatrix} = [f^e] \quad (2.7)$$

в глобальных координатах так, чтобы оно приняло следующий вид:

$$[K^e][U^e] = [K^e] \begin{bmatrix} U_1^e \\ U_2^e \\ U_3^e \\ U_4^e \end{bmatrix} = \begin{bmatrix} F_1^e \\ F_2^e \\ F_3^e \\ F_4^e \end{bmatrix} = [F^e]. \quad (2.8)$$

Для этого используем введённую в разделе 2.1 матрицу  $R_0$ , принимая во внимание отсутствие компонент  $v_1^e, v_2^e$  в уравнении равновесия. Определим матрицу  $R$  следующим образом:

$$[R] = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ 0 & 0 & \cos \theta & \sin \theta \end{bmatrix}, \quad (2.9)$$

так что с учётом (2.1) получаем:

$$[u^e] = [R][U^e]. \quad (2.10)$$

Подстановка (2.10) в (2.7) даёт

$$[k^e][R][U^e] = [f^e]$$

Поскольку локальные координаты узловых нагрузок  $f_1^e, f_2^e$  подчиняются соотношениям, подобным (2.10), переход к вектору  $[F^e]$  в правой части (2.7) может быть осуществлён с помощью умножения на  $[R]^T$ :

$$[R]^T[k^e][R][U^e] = [R]^T[f^e] = [F^e]. \quad (2.11)$$

Сопоставляя последнее уравнение с (2.8), заключаем, что

$$[K^e] = [R]^T[k^e][R]. \quad (2.12)$$

## 2.4. Глобальная матрица жёсткости

Дальнейшая задача состоит в объединении соотношений (2.8) для каждого элемента в общую систему уравнений

$$[K][U] = [F] \quad (2.13)$$

с использованием глобальной нумерации, введённой в разделе 2.1. Необходимо определить, каким образом элементы матриц  $[K^e]$  объединяются в матрицу  $[K]$ .

Воспользуемся определением произведения матриц и запишем выражение для  $m$ -го элемента вектора  $[F]$  из (2.13):

$$F_m = \sum_{i=1}^{2N} K_{mi} U_i.$$

Формально заменим индексы  $m \rightarrow L_m^{[n]}, i \rightarrow L_i^{[n]}$ :

$$F_{L_m^{[n]}} = \sum_{i=1}^{2N} K_{L_m^{[n]}, L_i^{[n]}} U_{L_i^{[n]}}. \quad (2.14)$$

В то же время, из (2.8) имеем:

$$F_m^{e[n]} = \sum_{i=1}^4 K_{mi}^{e[n]} U_i^{e[n]}, \quad (2.15)$$

где с учётом (2.6) перейдём к глобальным индексам:

$$\begin{aligned} F_m^{e[n]} &\rightarrow F_{L_m^{[n]}}, \\ U_i^{e[n]} &\rightarrow U_{L_i^{[n]}}. \end{aligned}$$

Но поскольку  $\forall n, l: U_{L^{[n]}} \rightarrow U^{e[n]}$  не является инъекцией (за счёт соединений элементов индекс одного и того же узла может встречаться в векторах связности разных элементов), для правомерности перехода необходимо учесть все конечные элементы, в результате чего получается основное соотношение, определяющее элементы глобальной матрицы жёсткости:

$$K_{L_m^{[n]}, L_i^{[n]}} = \sum_n K_{mi}^{e[n]} \quad (2.16)$$

Последнее равенство, вообще говоря, не позволяет прямым образом определить произвольный элемент матрицы  $K$ . На практике алгоритм построения  $K$  реализуется итерационно с помощью перебора всех конечных элементов модели с постепенным заполнением  $K$ .

## 2.5. Решение

После составления системы (2.13) необходимо приложить граничные условия начальных узловых смещений, определяющие  $2M$  из компонент вектора  $[U]$ , и внешние нагрузки, задающие  $2N - 2M$  координат  $[F]$ .

В результате переупорядочивания строк система уравнений принимает вид:

$$\begin{bmatrix} K_{cc} & K_{ca} \\ K_{ac} & K_{aa} \end{bmatrix} \begin{bmatrix} U_c \\ U_a \end{bmatrix} = \begin{bmatrix} F_c \\ F_a \end{bmatrix},$$

где  $[U_c]$  — вектор начальных смещений, а  $[F_a]$  — вектор внешних нагрузок.

Вектор неизвестных смещений  $[U_a]$  может быть найден из соотношения

$$[U_a] = [K_{aa}]^{-1}([F_a] - [K_{ac}][U_c]) \quad (2.17)$$

, которое при распространённом в статике случае нулевого вектора  $[U_c]$  имеет форму

$$[U_a] = [K_{aa}]^{-1}[F_a] \quad (2.18)$$

После определения  $[U_a]$  вектор реакций в конструкции может быть найден как  $[F_c] = [K_{cc}][U_c] + [K_{ca}][U_a]$ .

При известных узловых смещениях, деформации и напряжения определяются с помощью соотношения (1.4).

### 3. Динамическая задача

При рассмотрении *динамических* задач во внимание принимают инерционные эффекты, возникающие в результате наличия массы у элементов.

#### 3.1. Общий вид системы

Рассмотрим свободные колебания системы, изображённой на рис. 6.

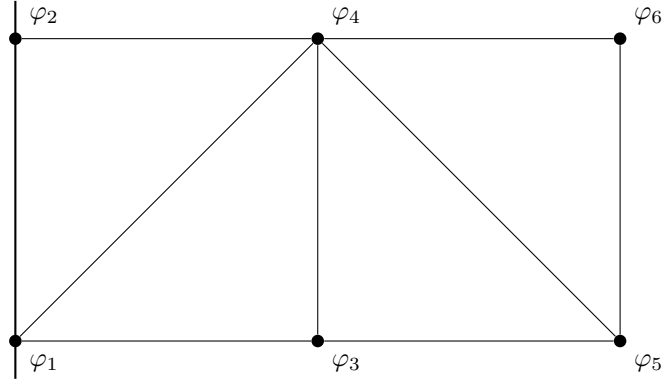


Рис. 6. Ферма из восьми стержневых элементов

Учитывая в соответствии с принципом Даламбера инерцию для записи условий равновесия элементов системы, можно получить систему вида

$$[M][\ddot{U}] + [K][U] = [0], \quad (3.1)$$

которая представляет собой системы обыкновенных дифференциальных уравнений второго порядка.

$[M]$  — матрица масс системы, которая, подобно матрице жёсткости, формируется из элементных матриц масс.

Для стержневого элемента с плотностью  $\rho$  матрица масс имеет вид:

$$[m^e] = \frac{\rho AL}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix} \quad (3.2)$$

#### 3.2. Собственные значения

Решение (3.1) имеет вид

$$U_i(t) = A_i \sin(\omega t + \varphi),$$

где  $A_i$  — амплитуда колебаний, а  $\omega$  — их частота. Подстановка решения в (3.1) приводит к уравнению

$$-\omega^2[M] + [K] = 0,$$

которое, как известно, имеет нетривиальное решение только в случае нулевого детерминанта:

$$\det([K] - \omega^2[M]) = 0 \quad (3.3)$$

Соотношение (3.3) представляет собой *частотное уравнение* системы, позволяющее определить *собственные частоты*  $\omega_i$  динамической системы.

Анализ собственных частот важен при исследовании возможной реакции системы на внешнее воздействие (вынуждающие колебания, которые при резонансе с собственными частотами могут привести к разрушению системы).

В разделе 4.2 представлены численные результаты расчёта собственных частот фермы с рис. 6.

## 4. Примеры и реализация метода

В ходе работы был создан прототип реализации МКЭ на языке Python.

При реализации метода конечных элементов необходимо учитывать, что получающаяся глобальная матрица жёсткости обладает сильной разреженностью и может быть плохо обусловлена.

Программа имеет следующую структуру:

```
<*>≡
<расчёты>
<определения>

if __name__ == "__main__":
    import doctest
    print doctest.testmod()

Начнём с описания созданных инструментов.

<определения>≡
import math
from scipy.sparse import lil_matrix, csr_matrix
from scipy.sparse.linalg import gmres
from numpy.linalg import eigvals

<класс для узла>
<класс для стержня>
<основные процедуры МКЭ>
```

Класс узла двумерной сетки обеспечивает хранение координат узла, его глобального индекса, а также определение расстояния для других узлов.

```
<класс для узла>≡
class Node2D:
    def __init__(self, x, y, idx=None):
        """
        'idx' is 1-based global node index.
        """
        self.x = x
        self.y = y
        self.idx = idx

    def dist_to(self, node):
        return math.sqrt((self.x - node.x) ** 2 + \
                        (self.y - node.y) ** 2)
```

Двумерный стержень реализован в соответствии с материалом раздела 1.3. Обеспечивается вычисление матриц жёсткости и масс в глобальных координатах.

*⟨класс для стержня⟩*≡

```
class Bar2D:
    def __init__(self, node1, node2, area, elasticity, density=0):
        self.node1 = node1
        self.node2 = node2
        self.length = node1.dist_to(node2)
        self.cosine = (node2.x - node1.x) / self.length
        self.sine = (node2.y - node1.y) / self.length

        self.area = area
        self.elasticity = elasticity
        self.density = density

        self.stiffness = self._stiffness_matrix()
        self.mass = self._mass_matrix()
        self.connectivity = [2 * node1.idx - 1, 2 * node1.idx, \
                             2 * node2.idx - 1, 2 * node2.idx]

    def _make_rotation(self):
        r = lil_matrix((4, 4))
        r[0, 0] = r[2, 2] = self.cosine
        r[0, 1] = r[2, 3] = self.sine
        return r
```

*⟨матрица жёсткости⟩*

*⟨матрица масс⟩*

Определение матрицы жёсткости в глобальных координатах выполняется по формулам (1.7) и (2.12).

*⟨матрица жёсткости⟩*≡

```
def _stiffness_matrix(self):
    k = self.area * self.elasticity / self.length
    m = lil_matrix((4, 4))
    m[0, 0] = m[2, 2] = k
    m[0, 2] = m[2, 0] = -k

    r = self._make_rotation()

    K = csr_matrix(r).T * csr_matrix(m) * csr_matrix(r)
    return K
```

Матрица масс вычисляется согласно формуле (3.2).

*⟨матрица масс⟩*≡

```
def _mass_matrix(self):
    m = self.density * self.area * self.length / 6
    M = lil_matrix((4, 4))
    M[0, 0] = M[1, 1] = M[2, 2] = M[3, 3] = m * 2
    M[2, 0] = M[0, 2] = M[3, 1] = M[1, 3] = m
    return M
```

После задания нужных классов для элементов и узлов, можно реализовать главные этапы МКЭ. Программа должна уметь строить глобальные матрицы жёсткости и массы, решать статические задачи и находить собственные частоты динамической системы.

*⟨основные процедуры МКЭ⟩*≡  
*⟨генерация глобальных матриц⟩*  
*⟨решение системы⟩*  
*⟨поиск собственных частот⟩*

Процедура **assemble** осуществляет генерацию глобальных матриц  $[K]$  и  $[M]$  по списку элементов и количеству узлов с учётом соотношения (2.16).

*⟨генерация глобальных матриц⟩*≡  

```
def assemble(elements, nodes):
    K = lil_matrix((nodes * 2, nodes * 2))
    M = lil_matrix((nodes * 2, nodes * 2))

    for e in elements:
        L = e.connectivity
        Ke = e.stiffness
        Me = e.mass
        for i in xrange(4):
            for j in xrange(4):
                K[L[i] - 1, L[j] - 1] += Ke[i, j]
                M[L[i] - 1, L[j] - 1] += Me[i, j]

    return K, M
```

Функция **solve\_fem** по матрице жёсткости, количеству зафиксированных узлов и вектору внешних нагрузок определяет вектор неизвестных смещений  $[U_a]$  в соответствии с (2.18).

*⟨решение системы⟩*≡  

```
def solve_fem(stiffness, displacements, forces):
    reduced = stiffness[2 * displacements:, 2 * displacements:]
    return gmres(csr_matrix(reduced), forces)
```

**find\_frequencies** определяет собственные частоты систем с заданными матрицами жёсткости и массы при условии жёсткого закрепления указанного количества узлов.

*⟨поиск собственных частот⟩*≡  

```
def find_frequencies(mass, stiffness, constrained):
    l = 2 * constrained
    S = mass[l:, l:].todense().getI() * stiffness[l:, l:]
    return map(math.sqrt, eigvals(S))
```

Теперь, когда все нужные структуры и алгоритмы описаны, можно переходить к практической части

*⟨расчёты⟩*≡  

```
"""
    ⟨расчёт статической задачи⟩
    ⟨расчёт динамической задачи⟩
    """
```

## 4.1. Статическая задача

Вернёмся к рассмотрению простейшей конструкции из двух элементов с рис. 5 из раздела 2.2.

```
<расчёт статической задачи>≡  
<установка простой сетки и конечных элементов>  
<получение глобальных матриц>  
<применение граничных условий>
```

Определим узлы  $\varphi_1$ ,  $\varphi_2$  и  $\varphi_3$  с координатами  $(0,0)$ ,  $(0,40)$  и  $(40,40)$ , соответственно. Также зададим стержни с параметрами  $A = 1.5$ ,  $E = 10^7$ .

```
<установка простой сетки и конечных элементов>≡  
»> n1 = Node2D(0, 0, 1)  
»> n2 = Node2D(0, 40, 2)  
»> n3 = Node2D(40, 40, 3)  
»> e1 = Bar2D(n1, n3, area=1.5, elasticity=1e7)  
»> e2 = Bar2D(n2, n3, 1.5, 1e7)
```

```
<получение глобальных матриц>≡  
»> K, M = assemble([e1, e2], 3)
```

Фиксируя  $\varphi_1$  и  $\varphi_2$ , а также прилагая к  $\varphi_3$  внешнюю нагрузку с компонентами  $F_5 = 500$  и  $F_6 = 300$ , получаем вектор смещений для незафиксированного узла  $\varphi_3$ . Полученный результат совпадает с модельным решением, представленным в [2].

```
<применение граничных условий>≡  
»> solve_fem(K, 2, [500, 300])  
(array([ 0.00053333,  0.00172941]), 0)
```

## 4.2. Динамическая задача

Рассмотрим ферму из восьми элементов с рис. 6 из раздела 3.1.

```
<расчёт динамической задачи>≡  
<подготовка элементов и матриц>  
<поиск собственных значений>
```



Равномерно распределим шесть узлов с  $\varphi_1$  по node6 и зададим стержни с такими же показателями площади сечения и эластичности, как в статическом случае, и с плотностью  $\rho = 2.6 \times 10^{-5}$ , после чего создадим глобальные матрицы  $[K]$  и  $[M]$ .

```
⟨подготовка элементов и матриц⟩≡
>> n1 = Node2D(0, 0, 1)
>> n2 = Node2D(0, 40, 2)
>> n3 = Node2D(40, 0, 3)
>> n4 = Node2D(40, 40, 4)
>> n5 = Node2D(80, 0, 5)
>> n6 = Node2D(80, 40, 6)
>> e1 = Bar2D(n1, n3, area=1.5, elasticity=1e7, density=26e-5)
>> e2 = Bar2D(n1, n4, 1.5, 1e7, 26e-5)
>> e3 = Bar2D(n2, n4, 1.5, 1e7, 26e-5)
>> e4 = Bar2D(n3, n4, 1.5, 1e7, 26e-5)
>> e5 = Bar2D(n3, n5, 1.5, 1e7, 26e-5)
>> e6 = Bar2D(n5, n4, 1.5, 1e7, 26e-5)
>> e7 = Bar2D(n4, n6, 1.5, 1e7, 26e-5)
>> e8 = Bar2D(n5, n6, 1.5, 1e7, 26e-5)
>> K, M = assemble([e1, e2, e3, e4, e5, e6, e7, e8], 6)
```

Зафиксируем два первых узла и определим собственные частоты полученной системы.

```
⟨поиск собственных значений⟩≡
>> modes = find_frequencies(M, K, 2)
>> print '\n'.join(map(str, 1))
8968.85109574
8601.70375634
7683.47004786
6553.28996781
4467.53230784
3292.15742497
2094.23963882
844.63027114
```

## 5. Выводы и результаты

В результате работы осуществлено близкое знакомство с методом конечных элементов, разработан прототип компьютерной реализации метода для решения статических и динамических задач.

Заложены основы для дальнейшей работы в данном поле.

Среди направлений дальнейшего совершенствования реализации метода необходимо выделить:

- Исследование эффективных методов представления разреженных матриц в памяти компьютера;
- Разработка алгоритмов, хорошо работающих с большими разреженными матрицами;
- Использование параллельных вычислений при реализации (например, для перемножения матриц).

## А. Информация о документе

Данный документ был подготовлен с использованием  $\text{\LaTeX}$ . Программа из раздела 4 написана на Python с использованием SciPy и NumPy. Код представлен с использованием `noweb`. Иллюстрации были созданы с помощью инструментов PGF и TikZ.

Представленная работа выполнена в рамках программы седьмого семестра обучения по специальности «Вычислительная математика и математическая физика» в МГТУ им. Н. Э. Баумана.

Дата компиляции настоящего документа: 18 января 2011 г.

## Б. Список литературы

- [1] BATHE KLAUS-JÜRGEN. Finite Element Procedures. — Prentice-Hall, 1996.
- [2] HUTTON DAVID. Fundamentals of finite element analysis. — NY: McGraw-Hill, 2004.
- [3] ZIENKIEWICZ O. C., TAYLOR R. L. The Finite Element Method. — Butterworth-Heinemann, 2000. — Vol. 1.
- [4] ВОРОЖЦОВ Е. В. Разностные методы решения задач механики сплошных сред. — Новосибирск: Изд-во НГТУ, 1998. — 86 с.