# OOPython

## Задача 8. Метод конечных элементов: решение ОДУ

### Мотивация

Метод конечных элементов и метод конечных объемов являются наиболее используемыми методами для решения промышленных задач (в связи со сложной геометрией расчетной 3D-области последних).

### Элементы теории

#### Обозначения

* – функционал, – значение функционала на функции
* – функция, – значение функции на числе .

**Постановка задачи:**

В **Задаче 7** мы обсуждали применение метода МКЭ к аппроксимации *аналитически заданной* функции ; аппроксимант мы искали в виде кусочно-линейной функции , которую возможно разложить по базису из функций-«шляпок» :

С учетом того, что решение на границах отрезка принимает нулевые значения, коэффициенты при функциях-«полушляпках» должны быть равны нулю:

Коэффициенты разложения по базису мы искали исходя из манипуляций со значением функционала-ошибки на функции . Рассматривались следующие методы нахождения коэффициентов:

|  |  |
| --- | --- |
| Название метода | Формула |
| Интерполяция |  |
| Регрессия |  |
| Наименьших квадратов |  |
| -проекция |  |

Другими словами, – пространство кусочно-линейных функций, принимающих нулевые значения на границах отрезка.

В cлучае с ОДУ мы не можем пользоваться функционалом т.к. нам неизвестно точное решение . Однако, вместо этого мы можем рассматривать *косвенную* меру ошибки – функционал-невязку :

Можно заметить, что на точном решении значение функционала-невязки (свойство, также свойственное функционалу-ошибке ).

Все методы нахождения коэффициентов в случае решения ОДУ находятся из тех же принципов/манипуляций, что и для аппроксимации функций, только вместо в формулы выше формально подставляют значение функционала-невязки :

|  |  |  |
| --- | --- | --- |
| Название метода | Формула | Название аналога метода |
| Коллокация |  | Интерполяция |
| Наименьших квадратов (дискретный) |  | Регрессия |
| Наименьших квадратов |  | Наименьших квадратов |
| **Галеркина** |  | -проекция. |

При этом результаты применений метода Галеркина и наименьших квадратов **не совпадают**, в отличие от задачи аппроксимации функций. Далее рассмотрим подробно метод Галеркина.

### Метод Галеркина

#### Получение СЛАУ

Исходная формула метода:

Преобразуем формулу с помощью интегрирования по частям:

*Примечание:* в случае кусочно-линейной функции мы не можем применять формулу интегрирования по частям, но в случае кусочно-полиномиальной степени это уместно. Поэтому представим, что мы работаем с последней.

Получаем:

Введем обозначения:

* билинейная форма:
* линейная форма:

и получим финальный вид записи:

Т.к. любую кусочно-линейную функцию можно разложить по базису , то равенство (6) эквивалентно выполнению условий (доказательство см. в теоретическом материале для **Задачи 7**):

Подставим в предыдущее равенство:

После раскрытия скобок получаем:

Эта система равенств представляет собой СЛАУ:

По историческим причинам матрица именуется **матрицей жесткости,** а вектор правой части– **вектором нагрузки.** Найдя решение , подставим его в исходное разложение (3) – получим приближенное решение .