## Gerb-BMSTU_01Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение**

## высшего образования

**«Московский государственный технический университет имени Н.Э. Баумана**

## (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## Лабораторная работа № 6

**Тема** Построение и программная реализация алгоритмов численного дифференцирования

**Студент** Климов И.С.

**Группа** ИУ7-42Б

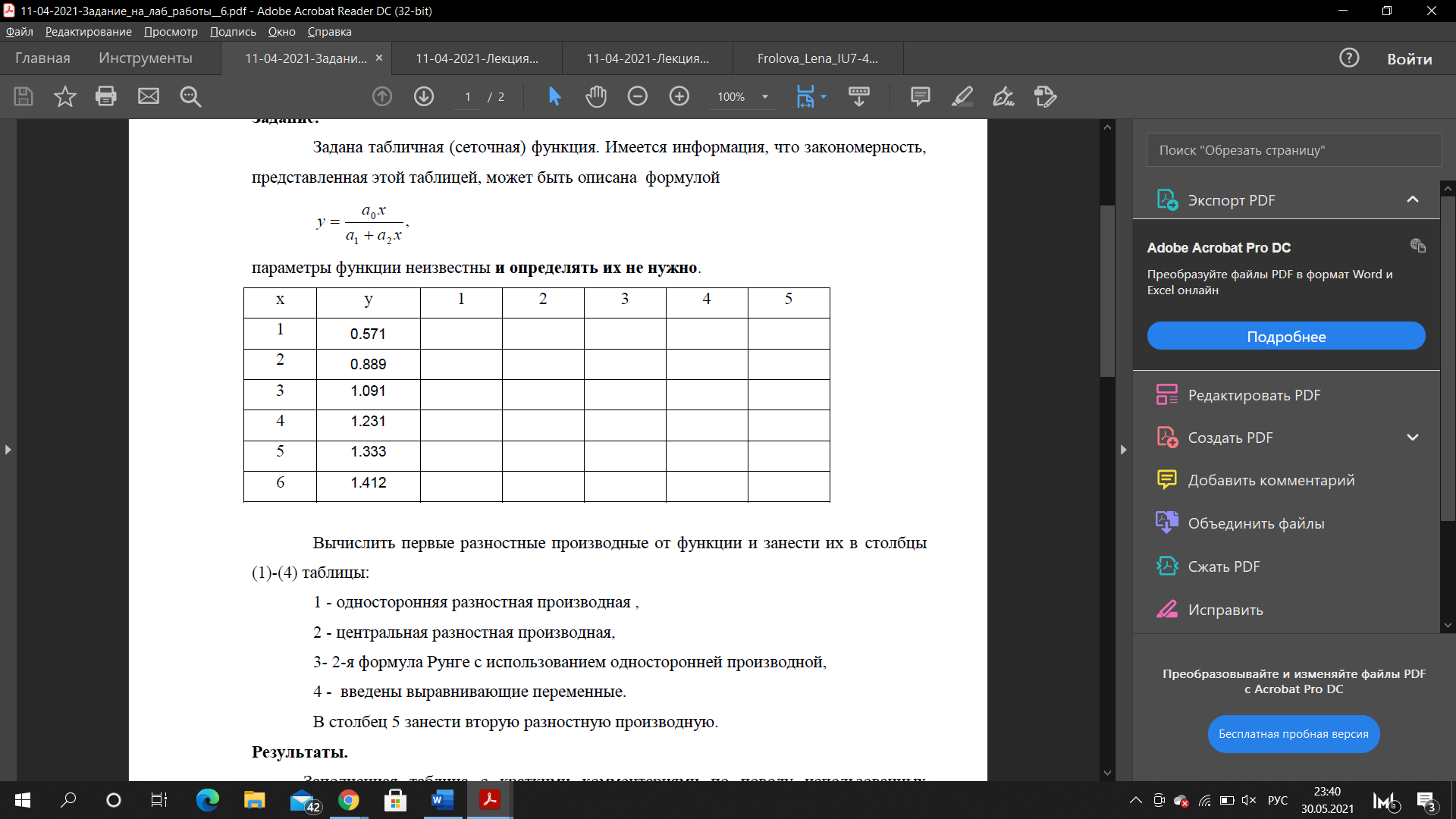
## Оценка (баллы)

**Преподаватель** Градов В.М.

Москва. 2021 г

**Цель работы**: Получение навыков построения алгоритма вычисления производных от сеточных функций.

1. **Исходные данные**



1. **Код программы**

**Листинг 1. main.py**

**import** **tkinter** **as** **tk**

x\_init = [1, 2, 3, 4, 5, 6]

y\_init = [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]

**def** left\_side(y, h, i):

**if** i > 0:

**return** (y[i] - y[i - 1]) / h

**else**:

**return** None

**def** center\_side(y, h, i):

**if** 0 < i < len(y) - 1:

**return** (y[i + 1] - y[i - 1]) / (2 \* h)

**else**:

**return** None

**def** runge(y, h, i):

**if** i > 1:

**return** 2 \* left\_side(y, h, i) - (y[i] - y[i - 2]) / (2 \* h)

**else**:

**return** None

**def** align\_vars(x, y, i):

**if** i < len(y) - 1:

tmp = (1 / y[i + 1] - 1 / y[i]) / (1 / x[i + 1] - 1 / x[i])

**return** tmp \* y[i] \*\* 2 / x[i] \*\* 2

**else**:

**return** None

**def** second(y, h, i):

**if** 0 < i < len(y) - 1:

**return** (y[i - 1] - 2 \* y[i] + y[i + 1]) / h \*\* 2

**else**:

**return** None

**def** get\_result():

result = [[x, y] **for** x, y **in** zip(x\_init, y\_init)]

length = len(x\_init)

methods = [left\_side, center\_side, runge, align\_vars, second]

**for** i **in** range(length):

**for** method **in** methods:

**if** method != align\_vars:

value = method(y\_init, x\_init[1] - x\_init[0], i)

**else**:

value = align\_vars(x\_init, y\_init, i)

result[i].append(value)

**return** result

**def** show\_table(table):

root = tk.Tk()

rows, columns = len(table), len(table[0])

names = ['x', 'y', '1', '2', '3', '4', '5']

**for** i **in** range(len(names)):

tk.Label(root, text=names[i], width=10, font='Times 16',

borderwidth=2, relief='ridge').grid(row=0, column=i)

**for** i **in** range(rows):

tk.Label(root, text=f'{table[i][0]}', width=10, font='Times 16',

borderwidth=2, relief='ridge').grid(row=i + 1, column=0)

**for** j **in** range(1, columns):

tk.Label(root, text=f'{table[i][j]:.3f}' **if** table[i][j] **else** '-', width=10, font='Times 16',

borderwidth=2, relief='ridge').grid(row=i + 1, column=j)

root.mainloop()

**def** main():

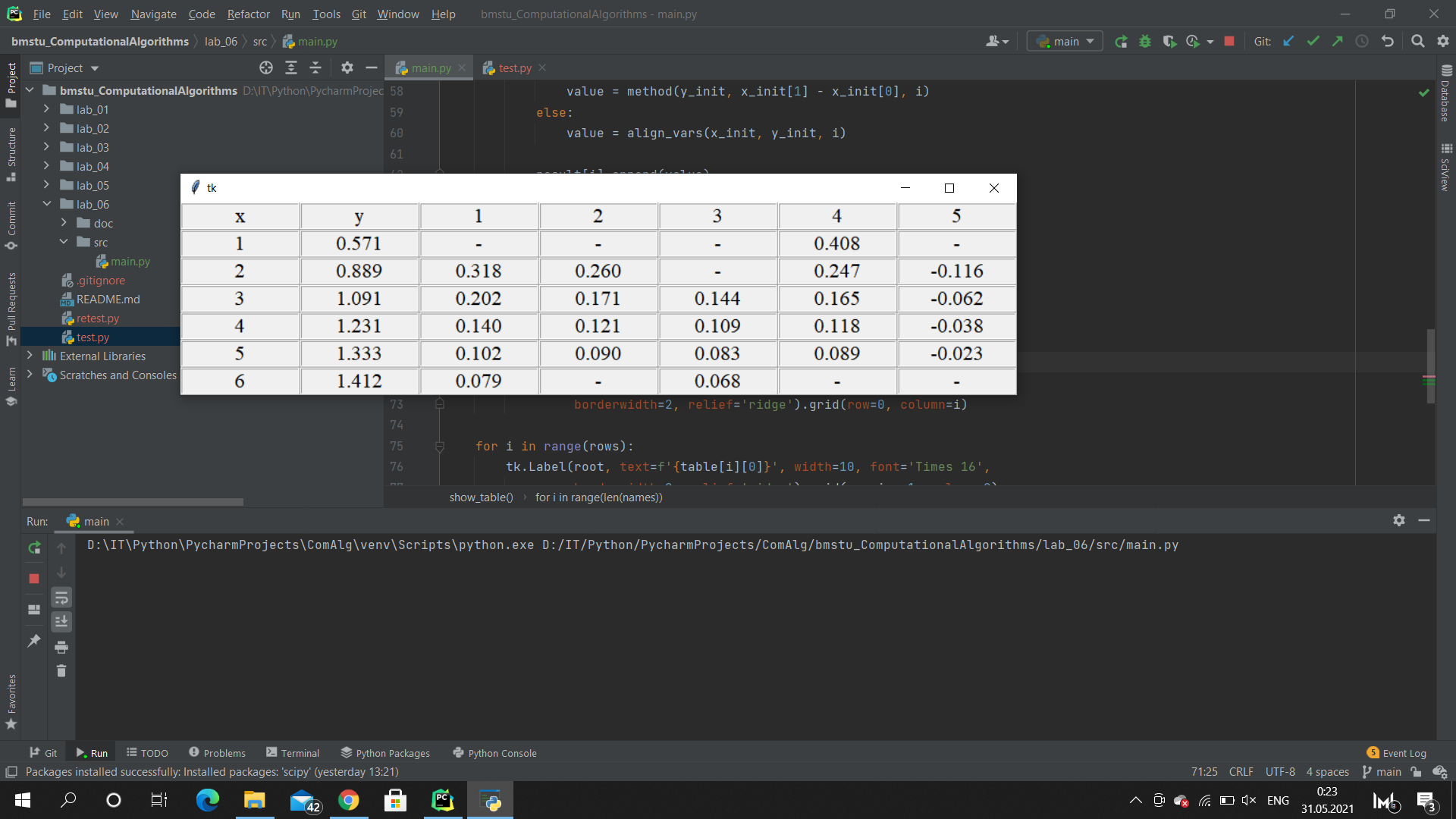
table = get\_result()

show\_table(table)

**if** \_\_name\_\_ == '\_\_main\_\_':

main()

1. **Результаты работы**



1. Первый столбец – левосторонняя формула (точность – O(h))

Получается путем разложения функции в ряд Тейлора для точки, в которой хотим найти производную. Затем выражаем значение через этот ряд для следующей точки и выражаем первую производную от него.

1. Второй столбец – центральная формула (точность – O(h2)).

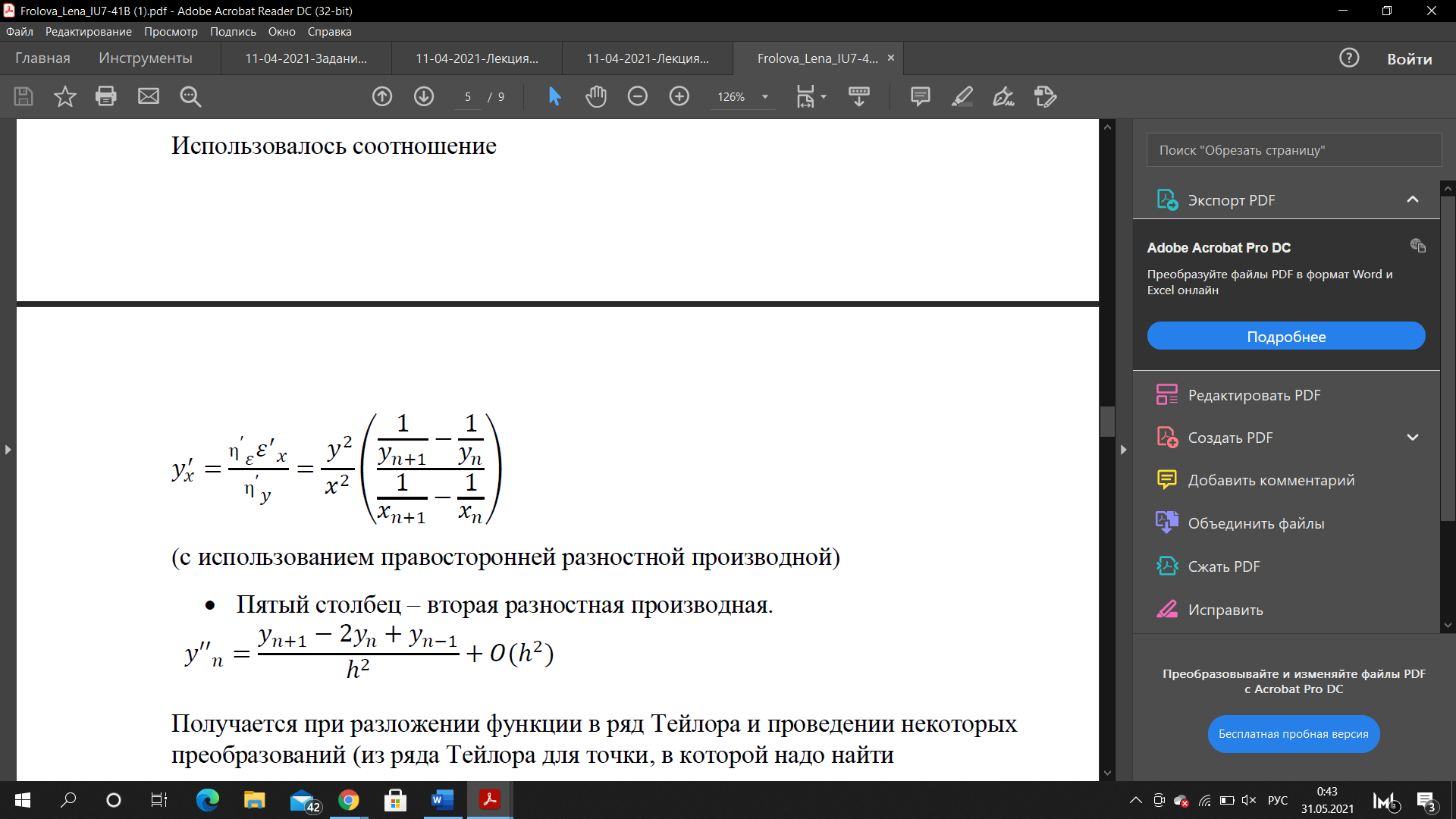
Получается при *fn+1 – fn-1*, где f – ряд Тейлора, построенный для точки, в которой требуется найти производную.

1. Третий столбец – вторая формула Рунге (точность – O(h2)).

Для вывода используется левосторонняя формула.

1. Четвертый столбец – применение выравнивающих переменных (невозможно точно оценить погрешность, так как параметры неизвестны).

Использовалось следующее соотношение:



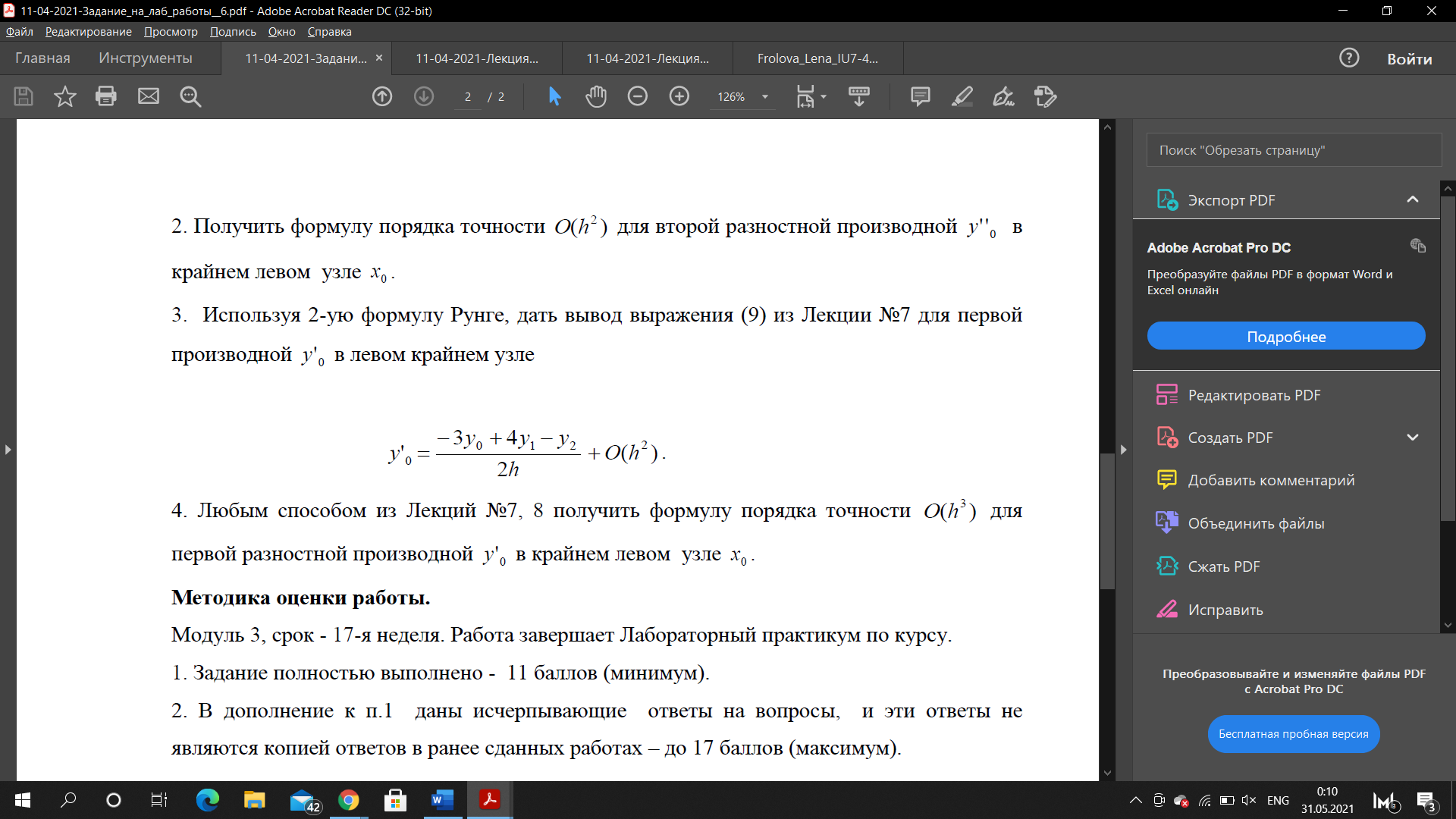
1. Пятый столбец – вторая разностная производная (точность – O(h2)).

Получается путем разложения функции в ряд Тейлора. При этом проводятся преобразования: из ряда Тейлора для точки, в которой надо найти производную, выражаем следующую и предыдущую и складываем получившиеся значения, выразив y``.

1. **Вопросы при защите лабораторной работы**
2. **Получить формулу порядка точности O(h2) для первой разностной производной y`n в крайнем правом узле xn.**

*m = 2*

1. **Получить формулу порядка точности O(h2) для второй разностной производной y``0 в крайнем правом узле x0.**
2. **Используя 2-ую формулу Рунге, дать вывод выражения для первой производной y`0 в левом крайнем узле.**



1. **Любым способом из Лекций №7, 8 получить формулу порядка точности O(h3) для первой разностной производной y`0 в крайнем левом узле x0.**

*p = 2*