| Gerb-BMSTU_01 | **Министерство 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 (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**ОТЧЕТ**

| **по лабораторной работе №** | 6 |
| --- | --- |



Построение и программная реализация алгоритмов численного дифференцирования

**Дисциплина:** Вычислительные алгоритмы

| Студент | ИУ7И - 46Б |  |  | Андрич К. |
| --- | --- | --- | --- | --- |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | В.М. Градов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

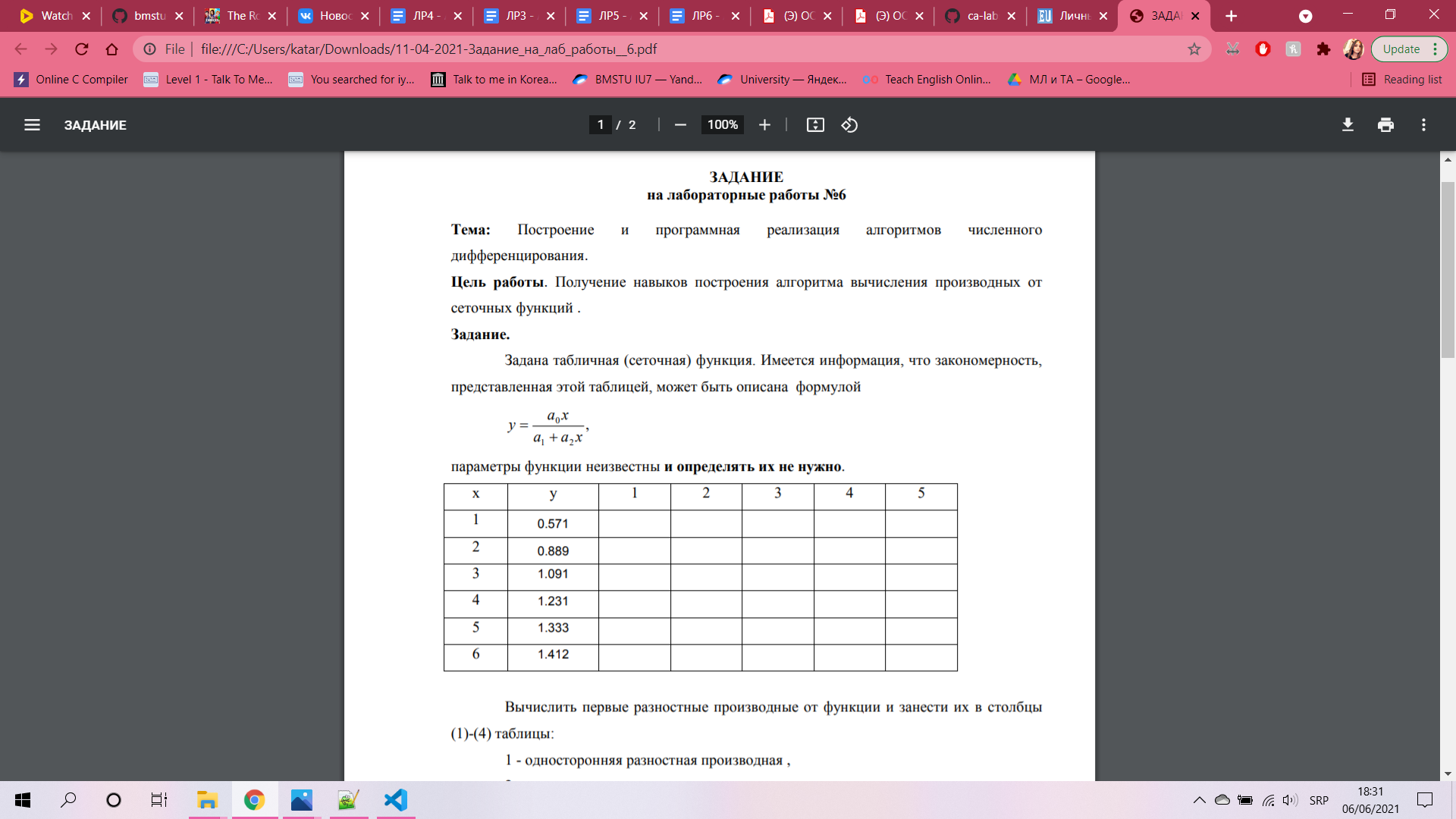
Москва, 2021

Цель работы

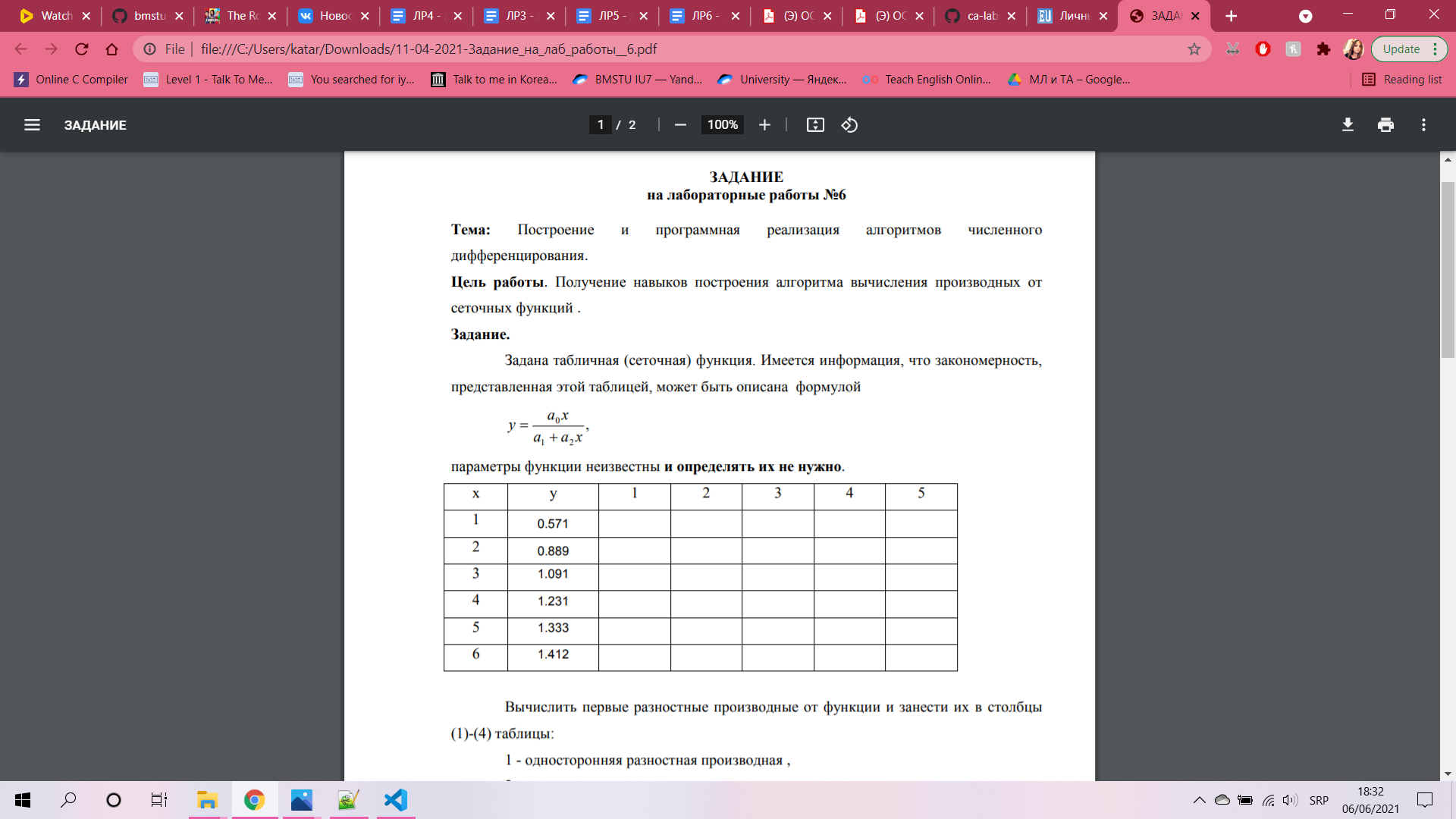
Получение навыков построения алгоритма вычисления производных от сеточных функций.

Задание

Задана табличная (сеточная) функция. Имеется информация, что закономерность, представленная этой таблицей, может быть описана формулой



параметры функции неизвестны и определять их не нужно.



Вычислить первые разностные производные от функции и занести их в столбцы (1)-(4) таблицы:

1 - односторонняя разностная производная,

2 - центральная разностная производная,

3 - 2-я формула Рунге с использованием односторонней производной,

4 - введены выравнивающие переменные.

В столбец 5 занести вторую разностную производную.

Код программы

В программе есть 4 файлов: 2 заголовочных файла (functions.h и errors.h) и 2 файла кода в СИ (main.c, functions.c)

errors.h

| #ifndef ERRORS\_H #define ERRORS\_H  #define OK 0 #define NO\_VALUE -1  #endif //ERRORS\_H |
| --- |

functions.h

| #ifndef FUNCTIONS\_H #define FUNCTIONS\_H  #include <stdio.h> #include <stdlib.h>  #define N 6  typedef struct {  float y1;  float y2;  float y3;  float y4;  float y5; } struct\_t;  float left\_diffder(float Y[N], int step, int inx); float center\_diffder(float Y[N], int step, int inx); float runge\_second(float Y[N], int step, int inx); float alignment\_vars(float Y[N], float X[N], int inx); float second\_diffder(float Y[N], int step, int inx); void solve(struct\_t results[N], float X[N], float Y[N]); void check\_print(float x); void print(struct\_t results[N], float Y[N]);  #endif //FUNCTIONS\_H |
| --- |

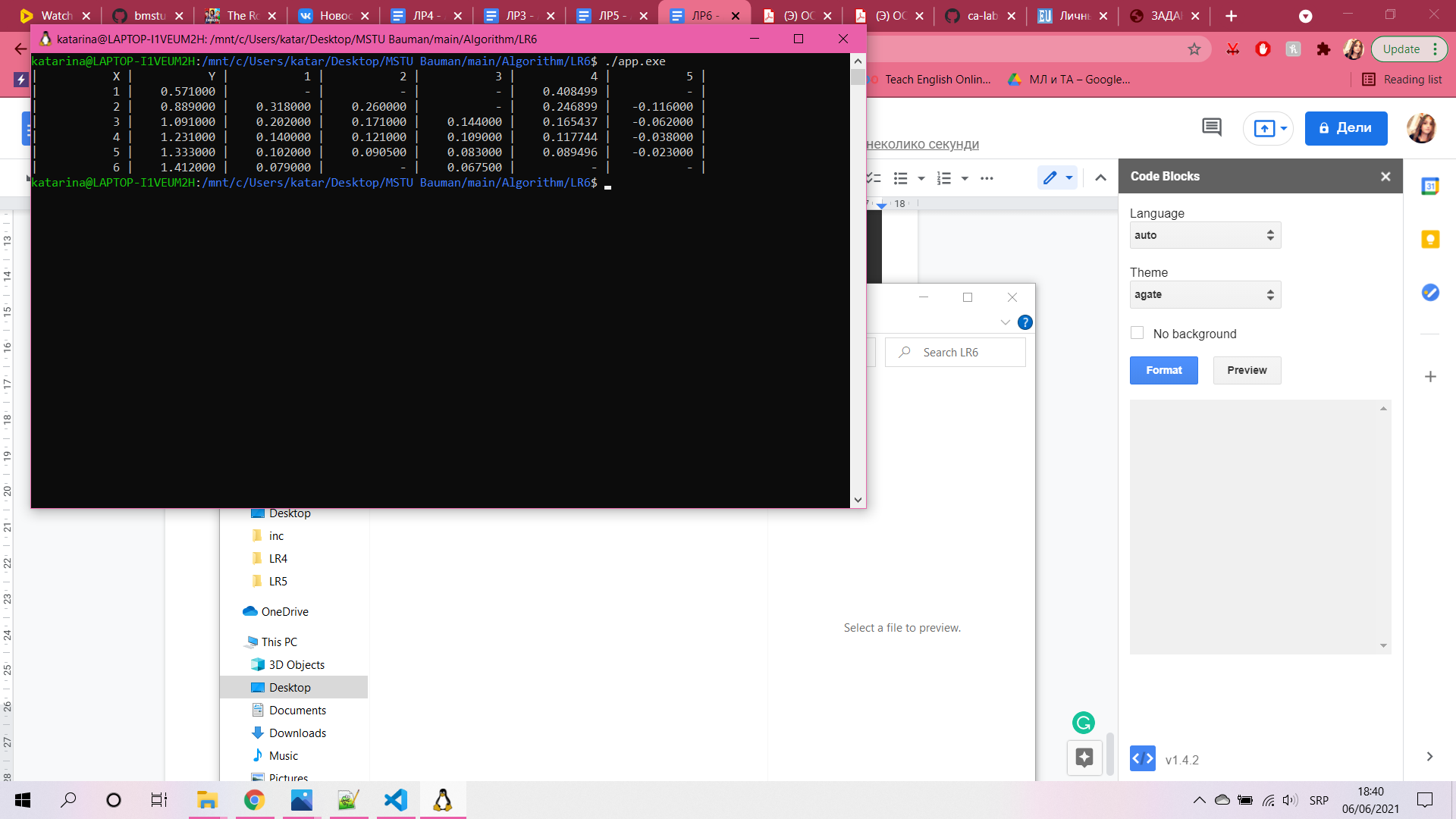
functions.c

| #include "functions.h" #include "errors.h"  float left\_diffder(float Y[N], int step, int inx) {  float result;  if (inx > 0 && inx < N)  {  result = (Y[inx] - Y[inx - 1]) / step;  return result;  }  else  return NO\_VALUE; }  float center\_diffder(float Y[N], int step, int inx) {  float result;  if (inx > 0 && inx < N - 1)  {  result = (Y[inx + 1] - Y[inx - 1]) / (2 \* step);  return result;  }  else  return NO\_VALUE; }  float runge\_second(float Y[N], int step, int inx) {  float result, y1, y2;  if (inx >= 2)  {  y1 = left\_diffder(Y, step, inx);  y2 = (Y[inx] - Y[inx - 2]) / (2 \* step);  result = 2 \* y1 - y2;  return result;  }  else  return NO\_VALUE; }  float alignment\_vars(float Y[N], float X[N], int inx) {  float result, d;  if (inx <= N - 2)  {  d = ((1 / Y[inx + 1]) - (1 / Y[inx])) / ((1 / X[inx + 1]) - (1 / X[inx]));  result = (d \* (Y[inx] \* Y[inx])) / (X[inx] \* X[inx]);  return result;  }  else   return NO\_VALUE; }  float second\_diffder(float Y[N], int step, int inx) {  float result;  if (inx > 0 && inx < N - 1)  {  result = (Y[inx - 1] - 2 \* Y[inx] + Y[inx + 1]) / (step \* step);  return result;  }  else  return NO\_VALUE; }  void solve(struct\_t results[N], float X[N], float Y[N]) {  int step = 1;  for (int i = 0; i < N; i++)  {  results[i].y1 = left\_diffder(Y, step, i);  results[i].y2 = center\_diffder(Y, step, i);  results[i].y3 = runge\_second(Y, step, i);  results[i].y4 = alignment\_vars(Y, X, i);  results[i].y5 = second\_diffder(Y, step, i);  } }  void check\_print(float x) {  if (x == NO\_VALUE)  printf("| %11s ", "-");   else  printf("| %11f ", x);  }  void print(struct\_t results[N], float Y[N]) {  printf("| %11s ", "X");  printf("| %11s ", "Y");  for (int i = 0; i < N - 1; i++)  printf("| %11d ", i + 1);  printf("|\n");  for (int i = 0; i < N; i++)  {  printf("| %11d ", i + 1);  printf("| %11f ", Y[i]);  check\_print(results[i].y1);  check\_print(results[i].y2);  check\_print(results[i].y3);  check\_print(results[i].y4);  check\_print(results[i].y5);  printf("|\n");  } } |
| --- |

main.c

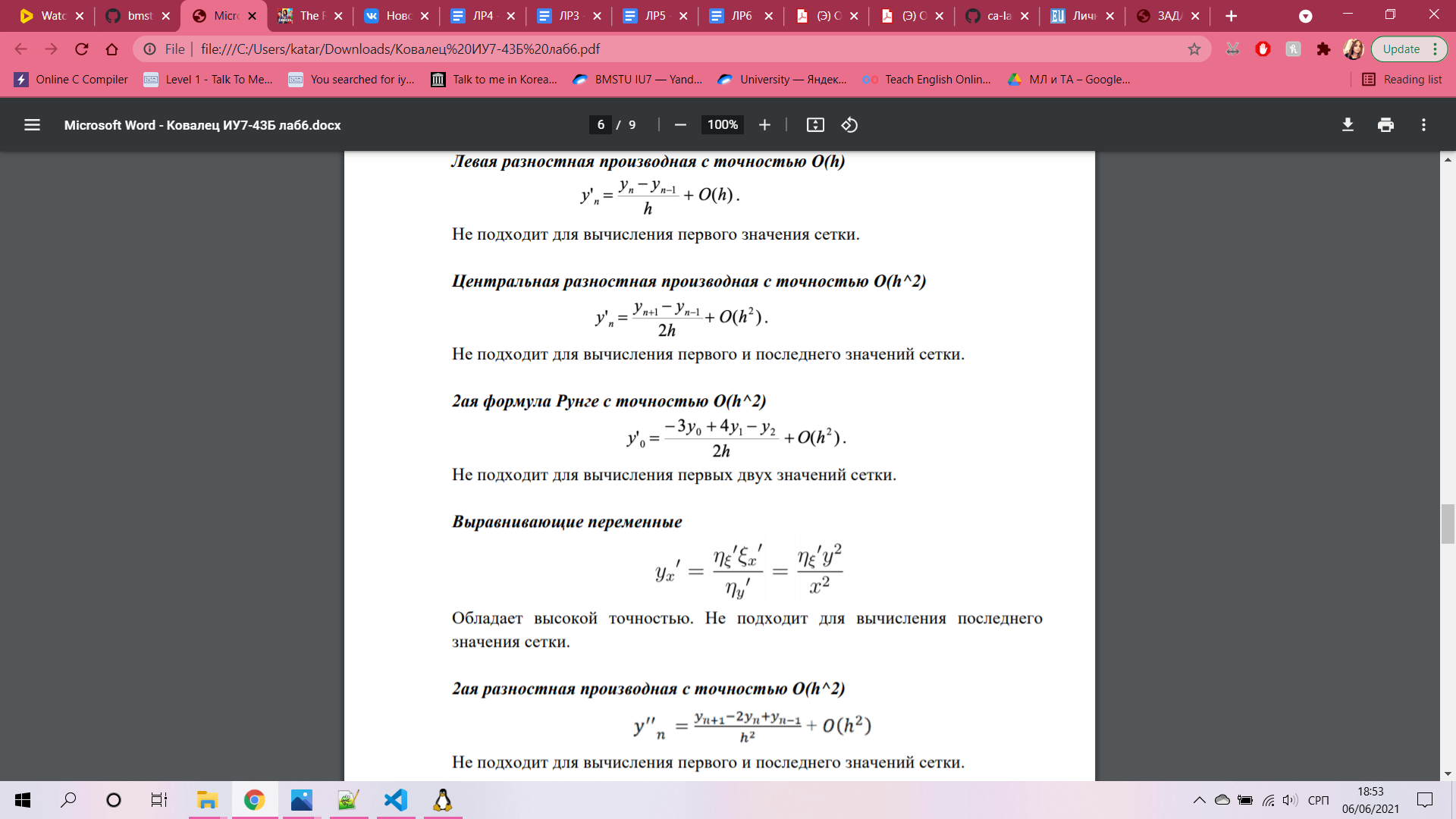
| #include "functions.h" #include "errors.h"  int main(void) {  float X[N] = {1.00, 2.00, 3.00, 4.00, 5.00, 6.00};  float Y[N] = {0.571, 0.889, 1.091, 1.231, 1.333, 1.412};  struct\_t results[N];  solve(results, X, Y);  print(results, Y);  return OK; } |
| --- |

Результаты работы



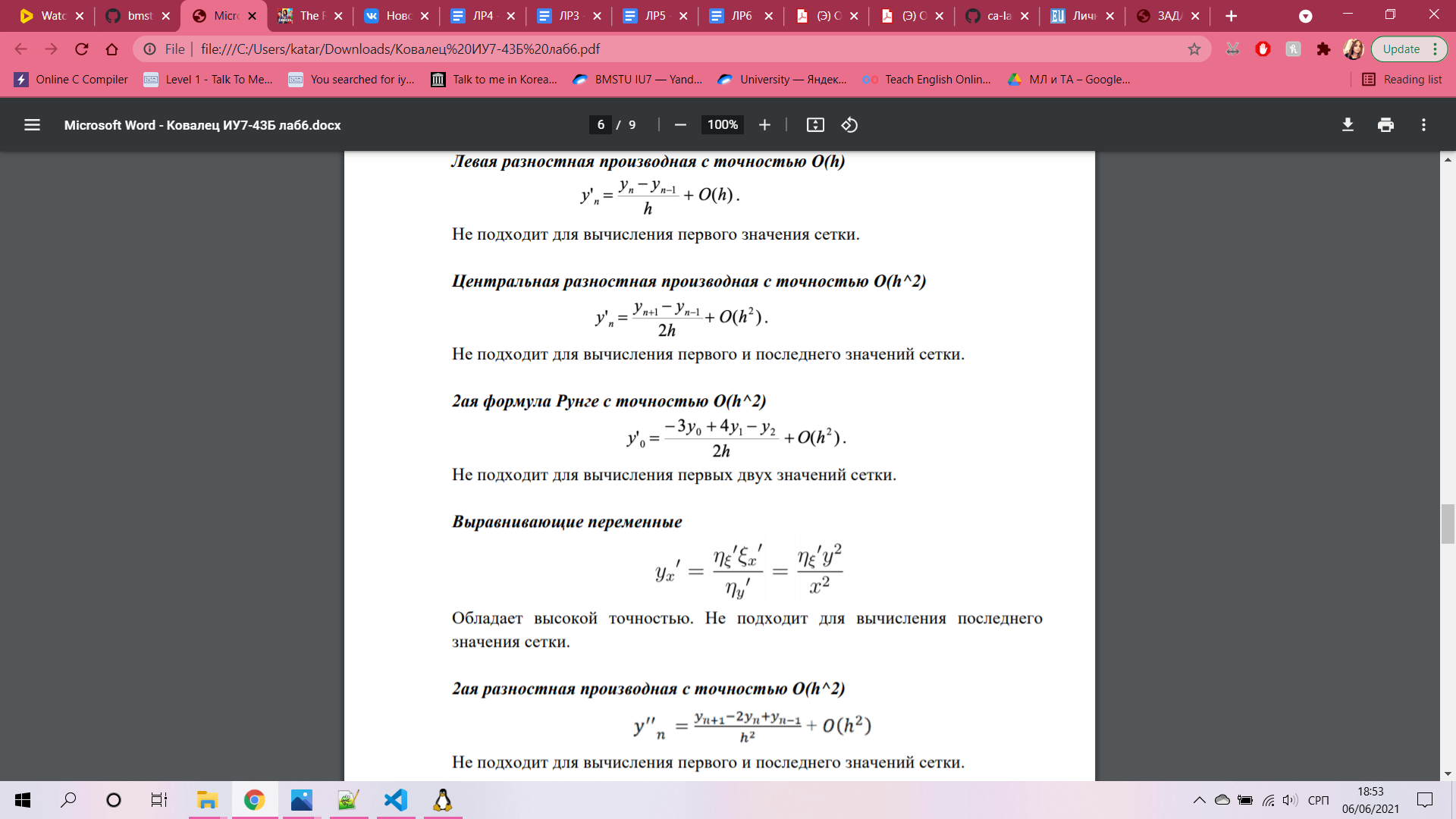
| X | Y | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.571 | - | - | - | 0.408499 | - |
| 2 | 0.889 | 0.318000 | 0.260000 | - | 0.2468899 | -0.11600 |
| 3 | 1.091 | 0.202000 | 0.171000 | 0.144000 | 0.165437 | -0.06200 |
| 4 | 1.231 | 0.140000 | 0.121000 | 0.109000. | 0.117744 | -0.038000 |
| 5 | 1.333 | 0.102000 | 0.090500 | 0.083000 | 0.089496 | -0.023000 |
| 6 | 1.412 | 0.079000 | - | 0.067500 | - | - |

*Левая разностная производная с точностью O(h)*



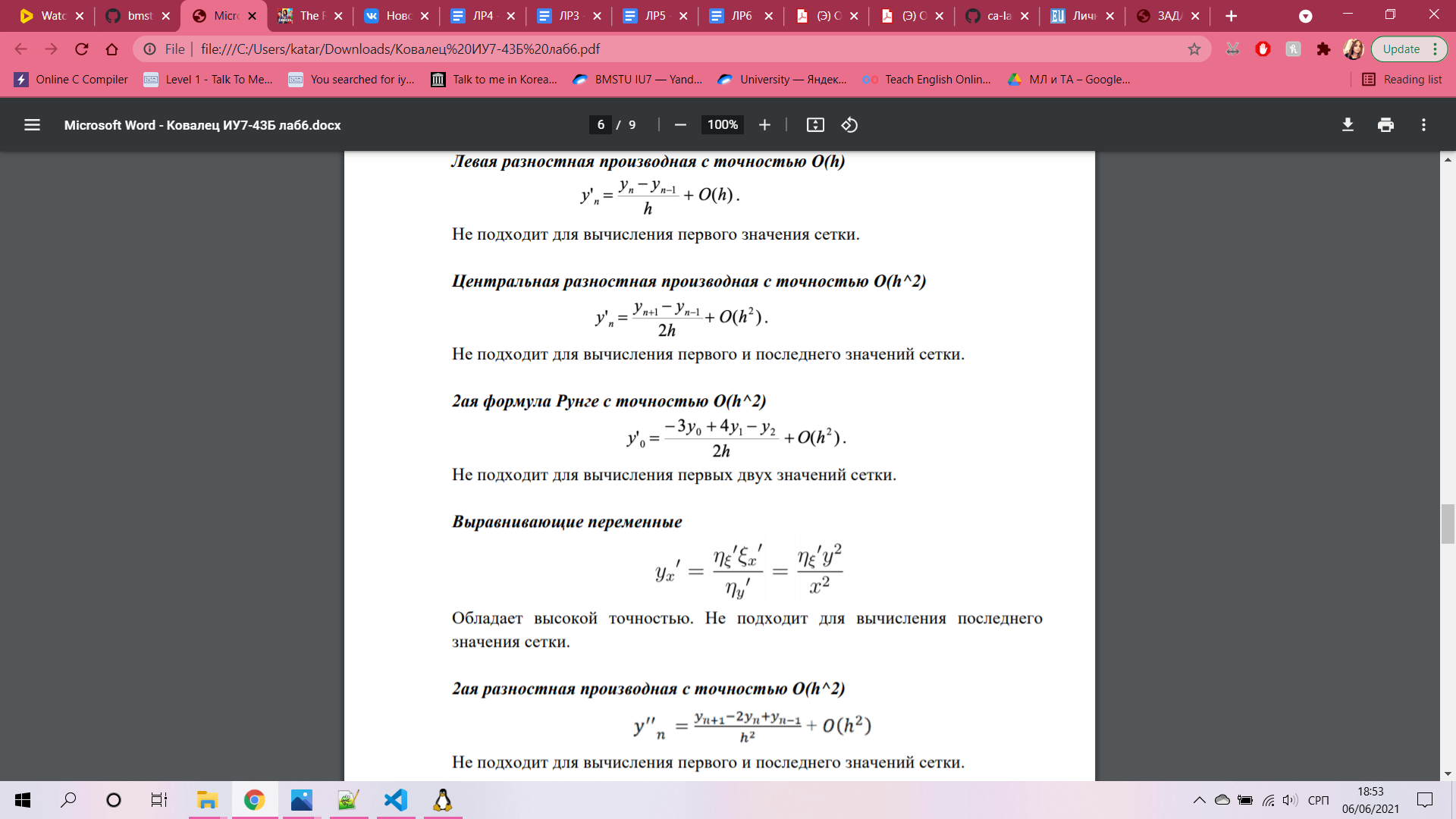
Не возможно вычислить первое значение

*Центральная разностная производная с точностью O(h2)*



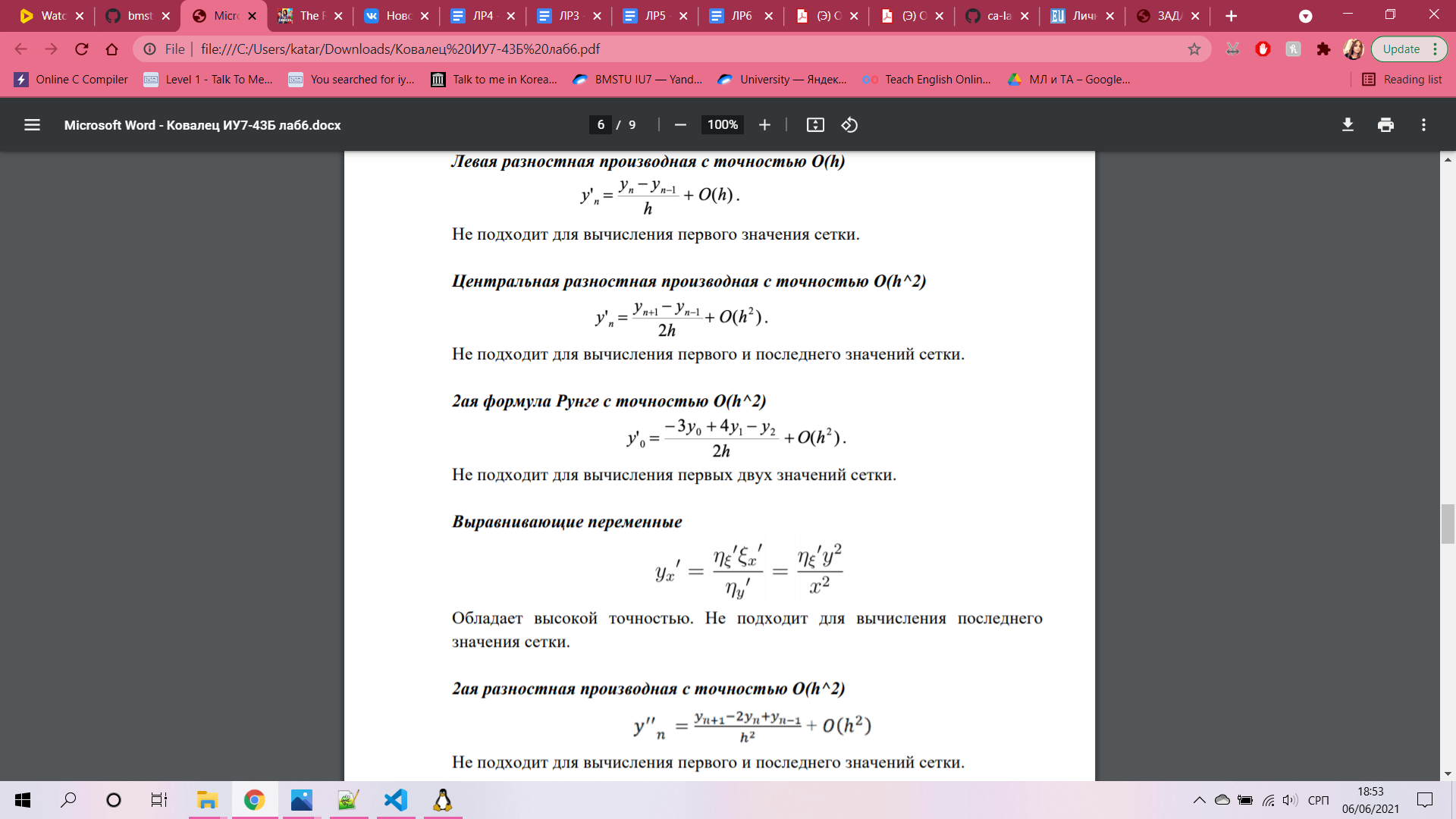
Не возможно вычислить первое и шестое значение

*2ая формула Рунге с точностью O(h2)*

**

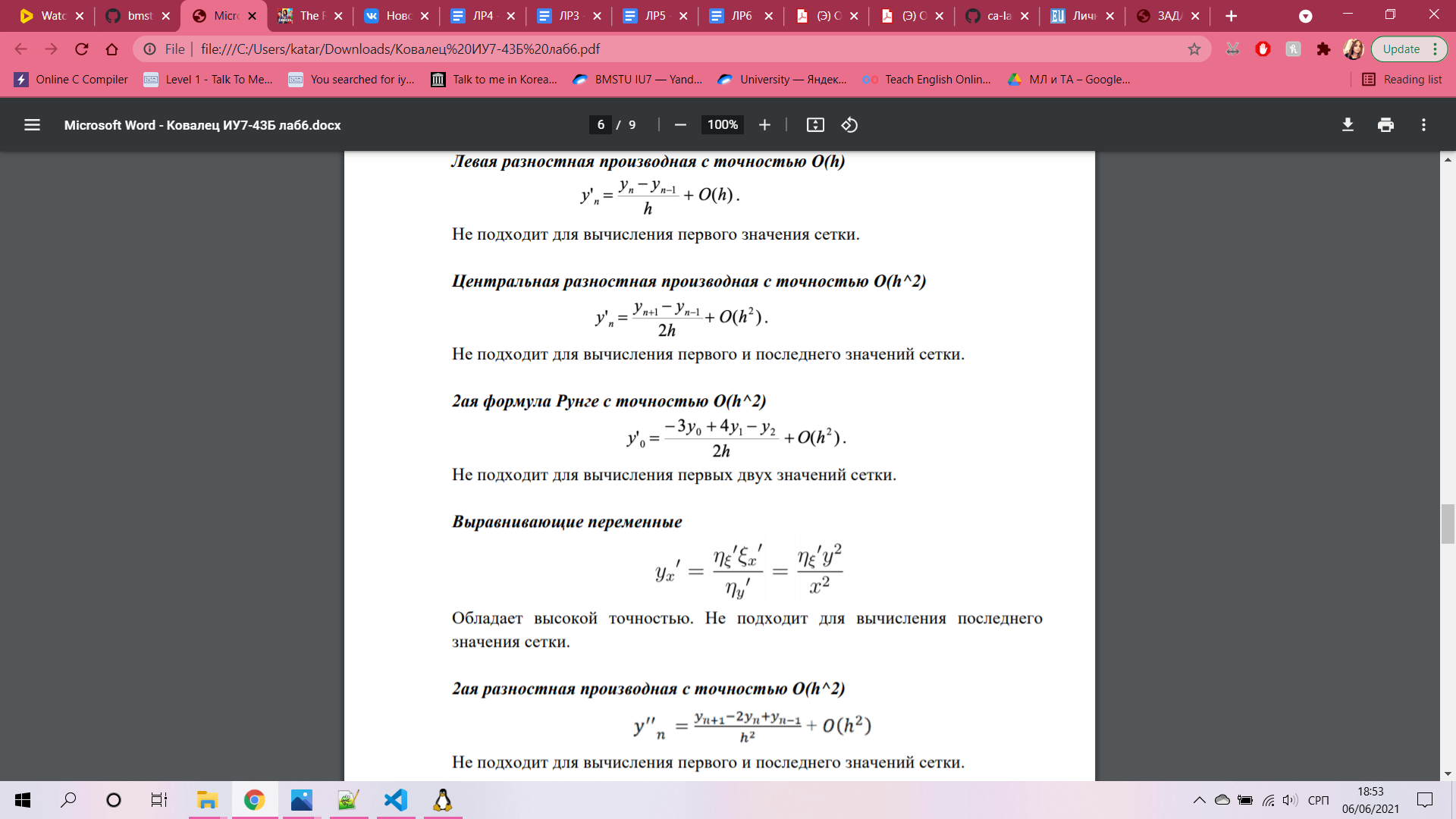
Не возможно вычислить первое и второе значение

*Выравнивающие переменные*



Не возможно вычислить шестое значение

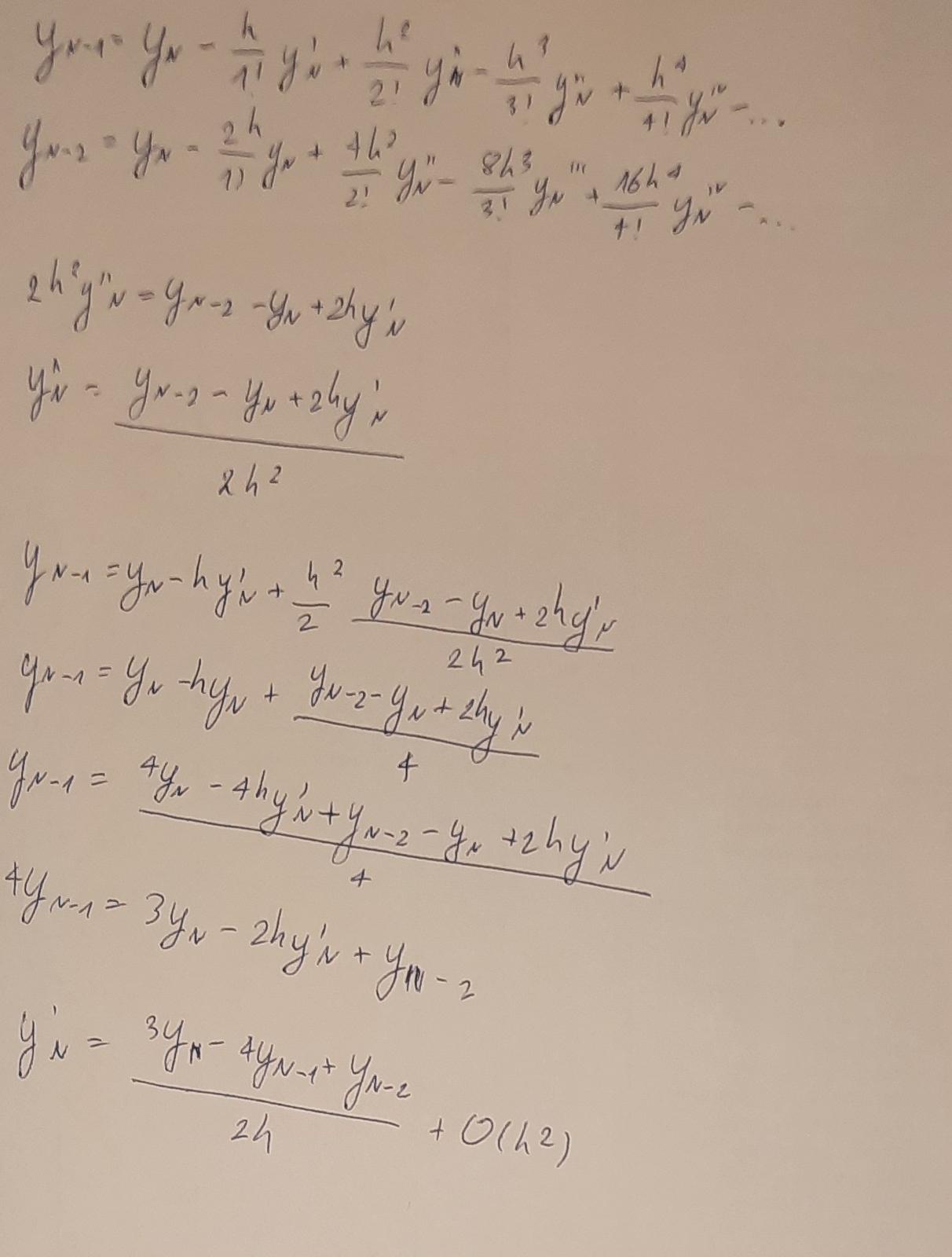
*2ая разностная производная с точностью O(h2)*

**

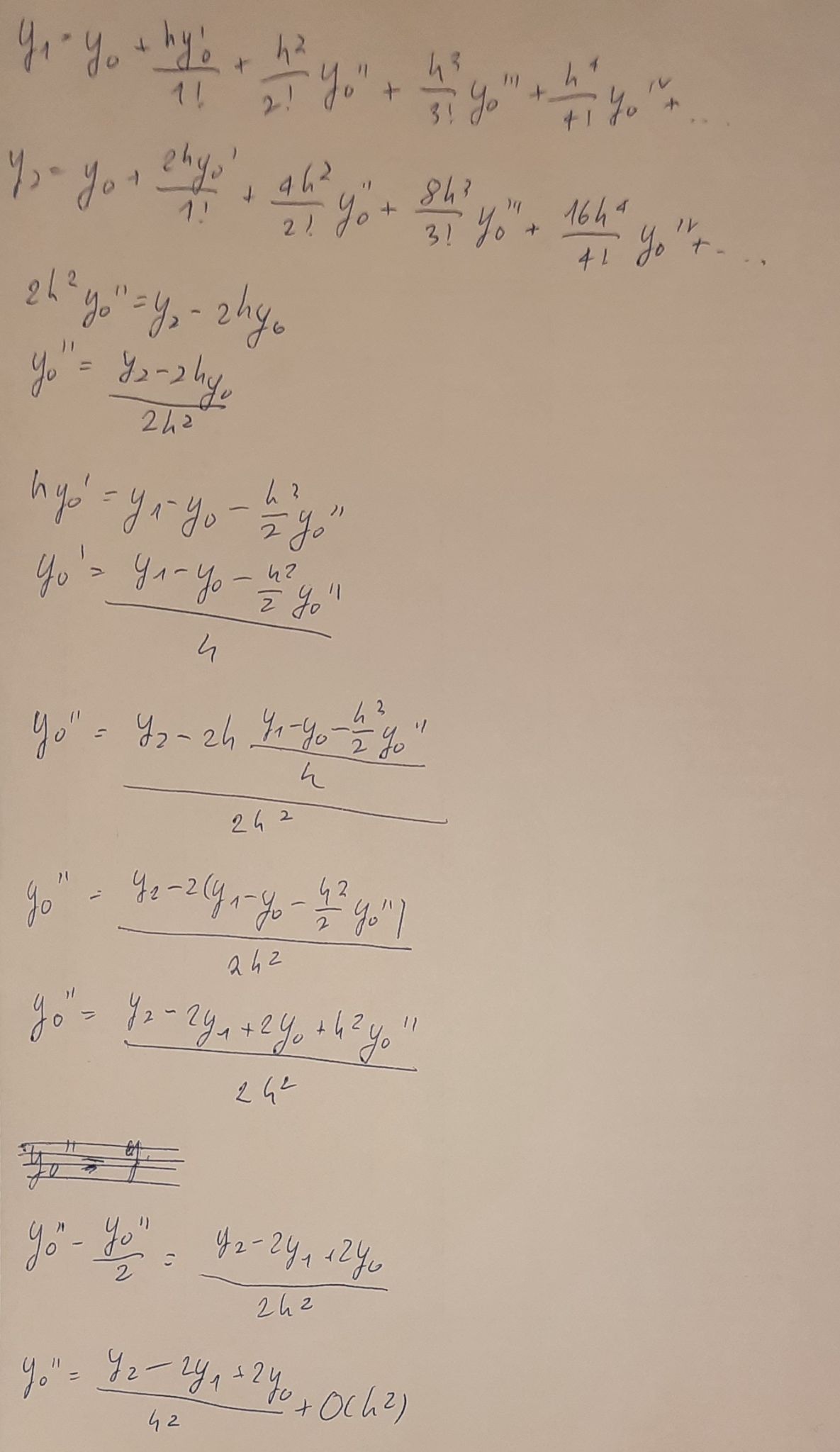
Не возможно вычислить первое и шестое значение

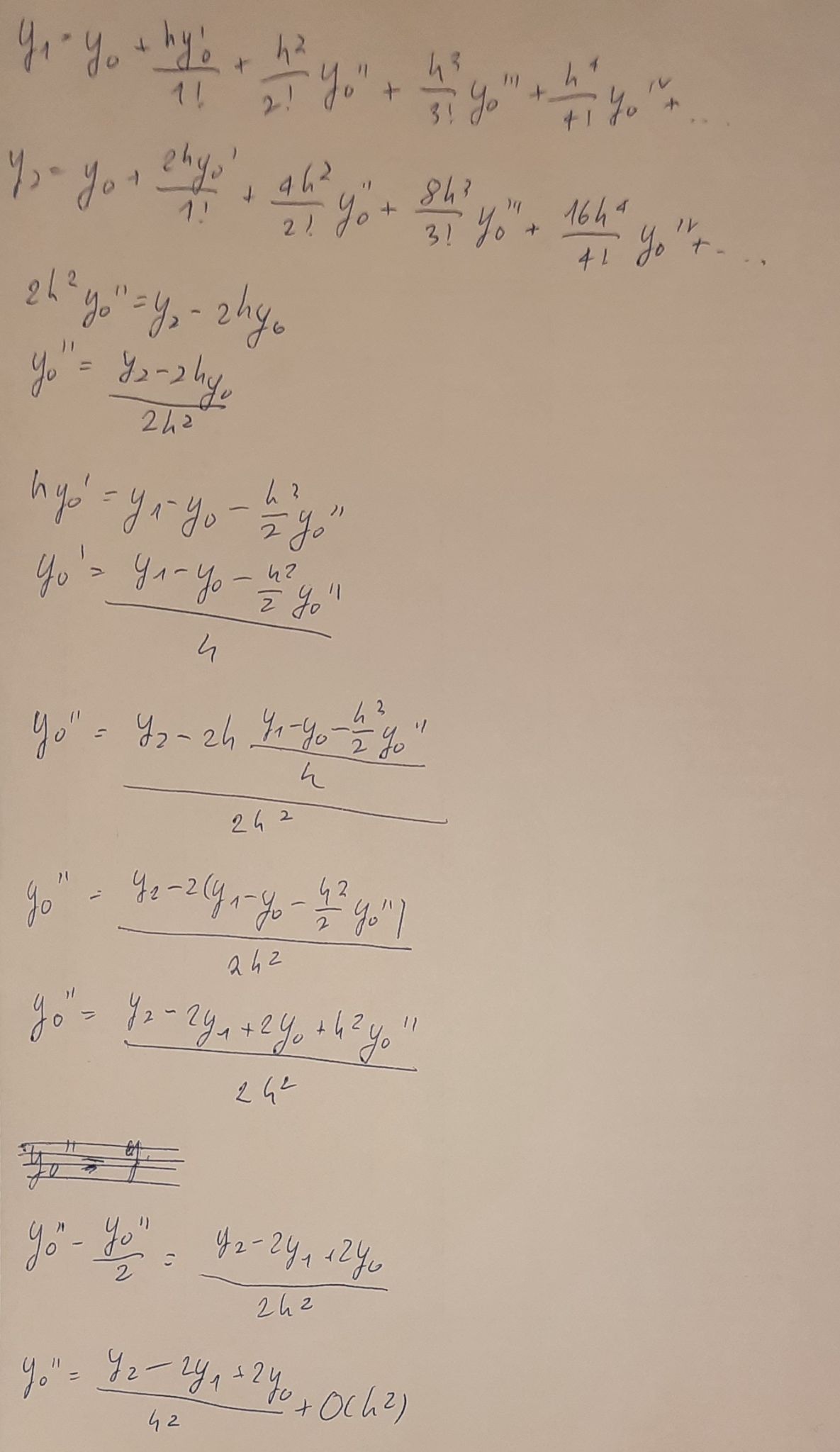
Вопросы при защите лабораторной работы

1. *Получить формулу порядка точности O(h2) для первой разностной производной N y' в крайнем правом узле xN.*

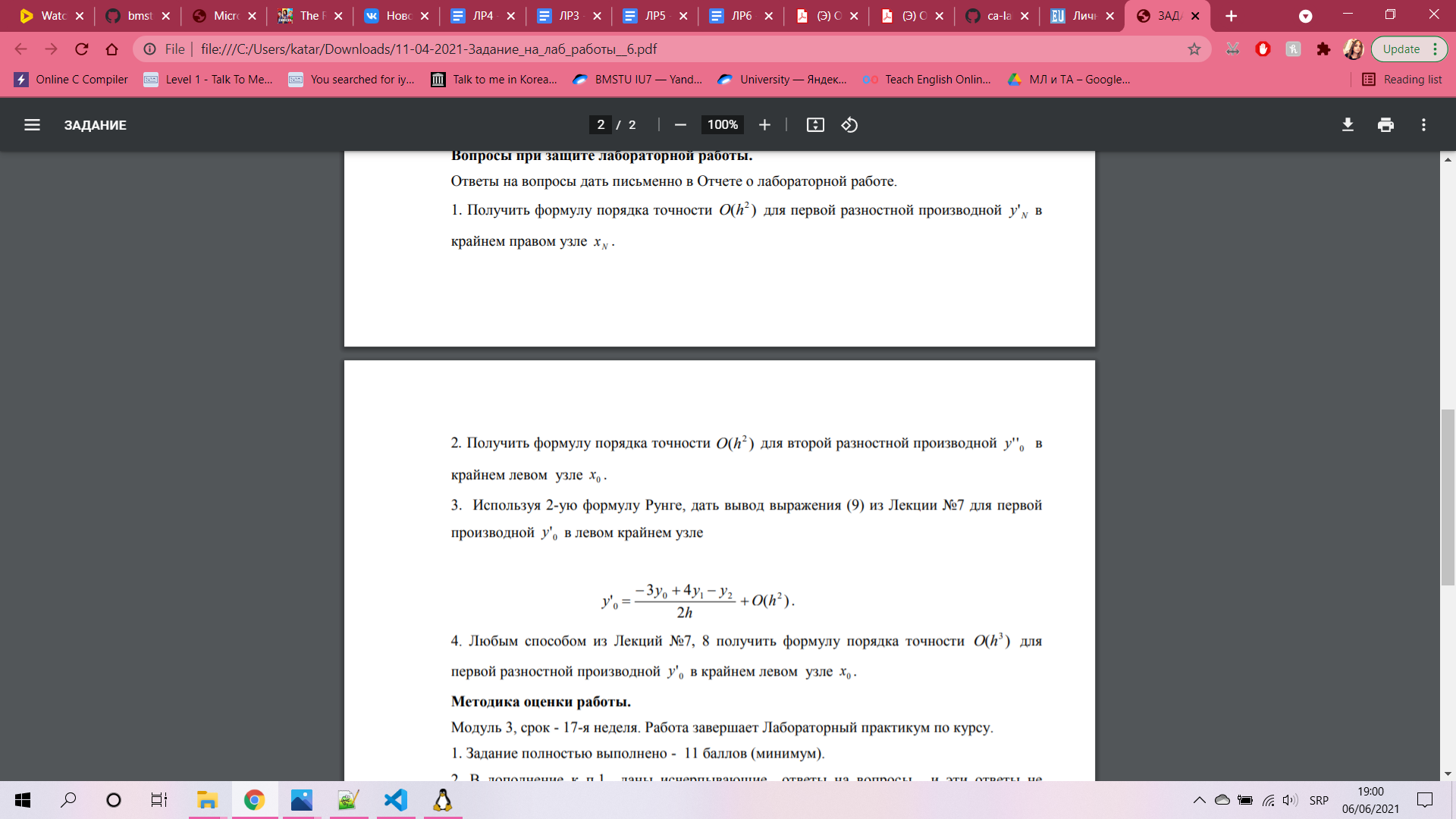
**

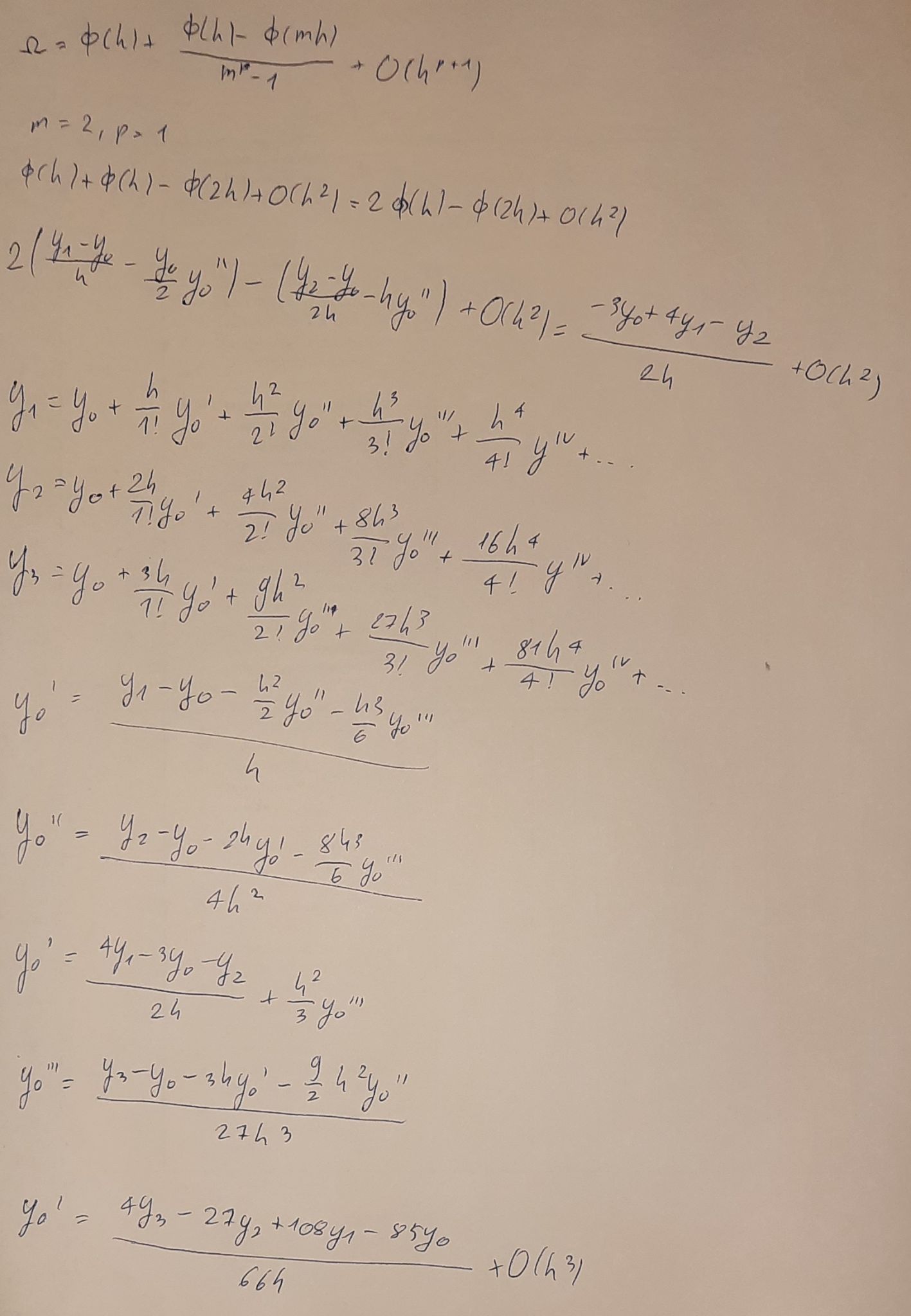
1. *Получить формулу порядка точности O(h2) для второй разностной производной y0'' в крайнем левом узле x0.*

**

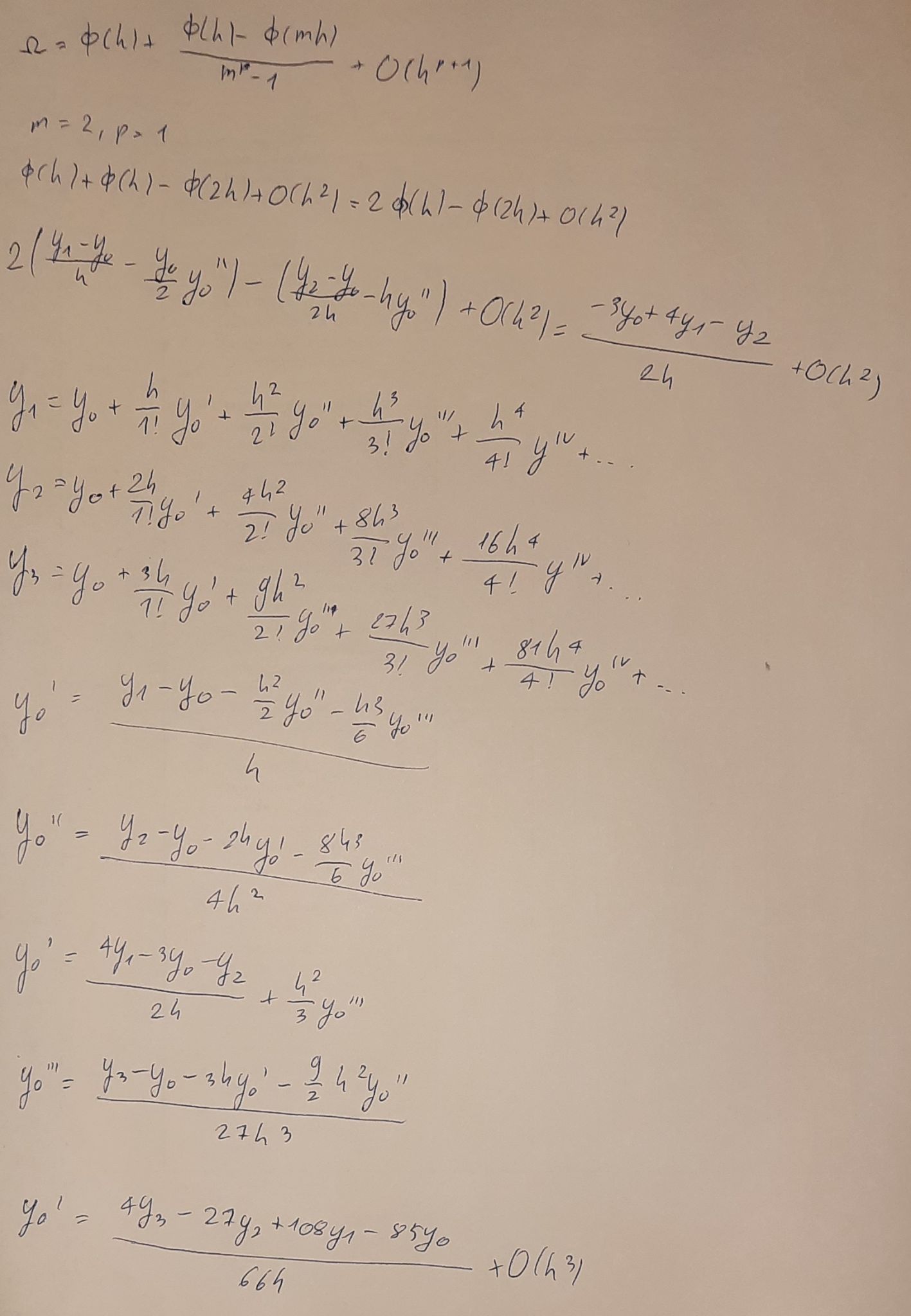
**

1. *Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции №7 для первой производной y0' в левом крайнем узле*

**

**

1. *Любым способом из Лекций №7, 8 получить формулу порядка точности O(h3) для первой разностной производной y0' в крайнем левом узле x0.*

**