

delt2yk = delt1yk+1 – delt1yk, k = 0, ..., n-2

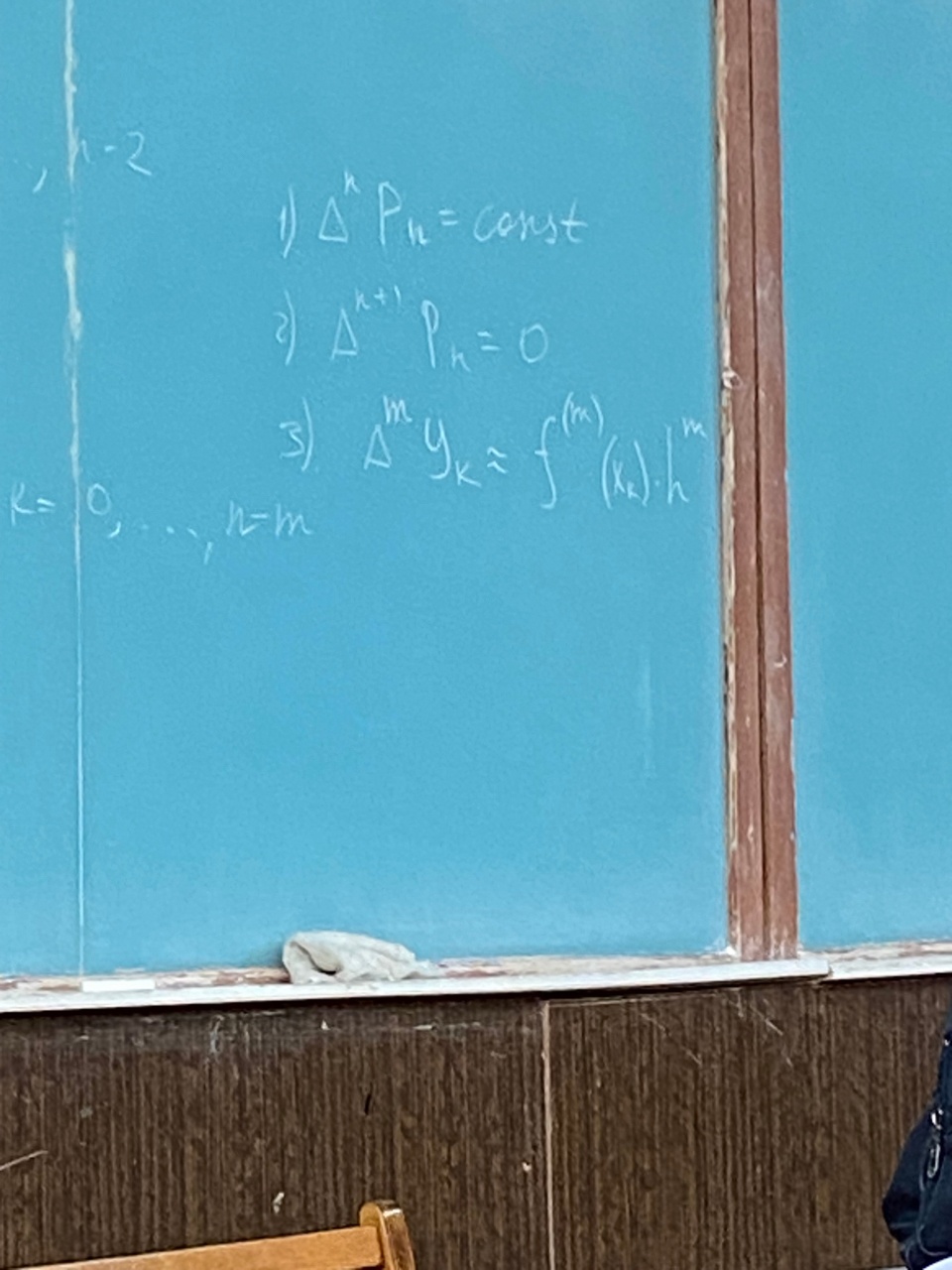
deltmyk = deltm+1yk+1 – deltm+1yk, k = 0, ..., n-2

Свойства конечной разности:

1. Конечная разность от полинома порядка n = const => n-1 порядок:

2.

3. Если ф y = f(x) бесконечно дифференцируема и сетка задана равномерно с шагом h, то конечная разность приближенно равна 2)



Разделенные разности первого порядка от функции f(x0, x1) = (f(x1) – f(x0)) /(x1 – x0) = (delt2 \* y0) / (x1 – x0)

f(xk,xk+1) = (delt1yk) / (xk+1 – xk), k=0, n-1

f(xk, xk+1, xk+2) = (f(xk+1, xk+2) – f(xk, xk+1)) / (xk+2 – xk) = [[?[[(delt2 \* yk+1) / (xk+2 – xk)]]?]]

mпорядка f(xk, xk+1, ... , xk+m) = (f(xk+1, ... , xk+m) – f(xk, ... , xk+m-1)) / (xk+m – xk)

Свойства:

1. Разделенная разность порядка n, примененная к полиному порядка n равна константе.

2. Разделенная разность порядка n+1 равна 0.

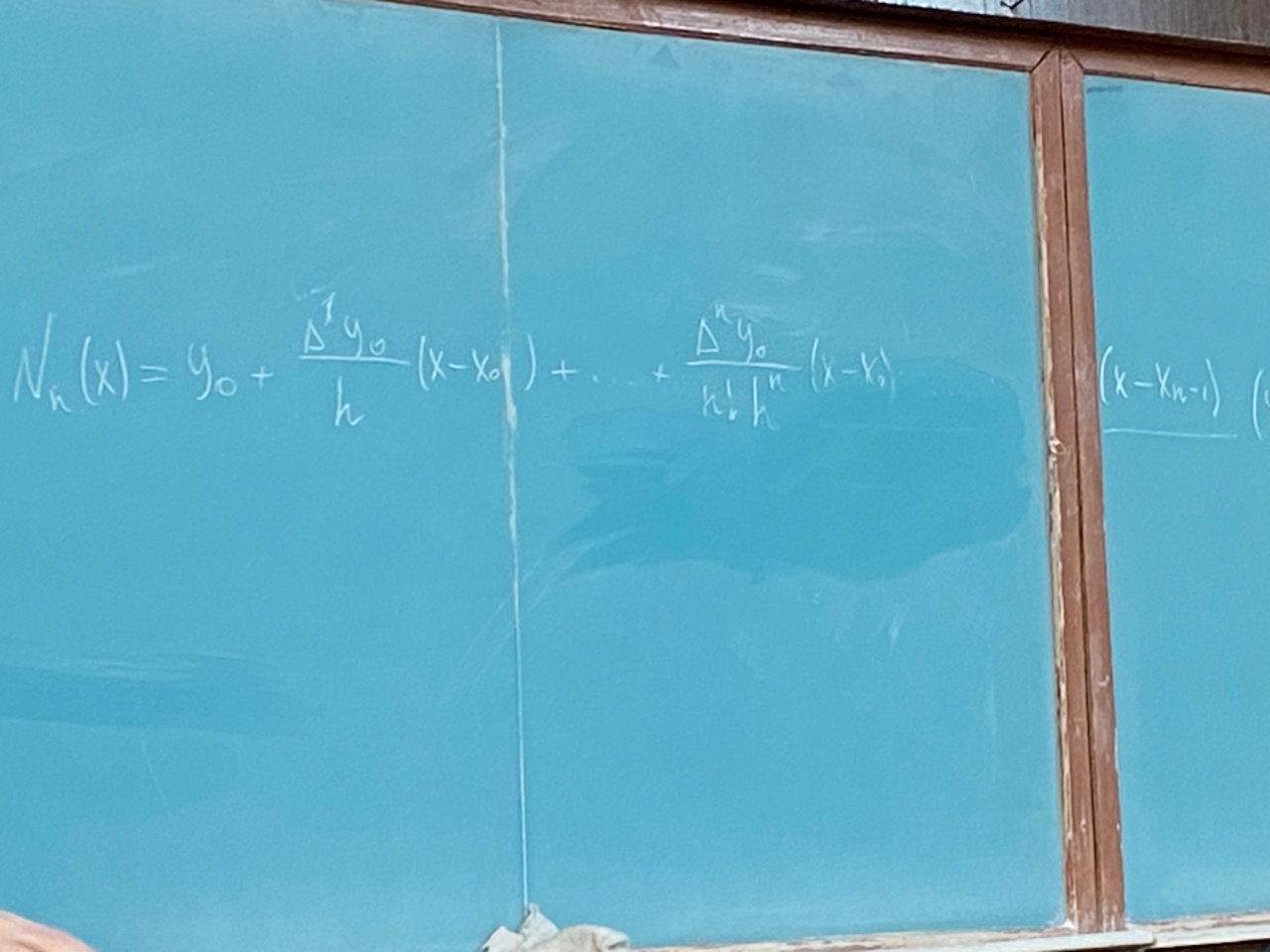
3. Если функция f бесконечно дифференцируема и сетка задана равномерно с шагом h, то разделенная разность порядка m равна конечной разности порядка m деленной на факториал и

Многочлен Ньютона для функции с порядком 1.

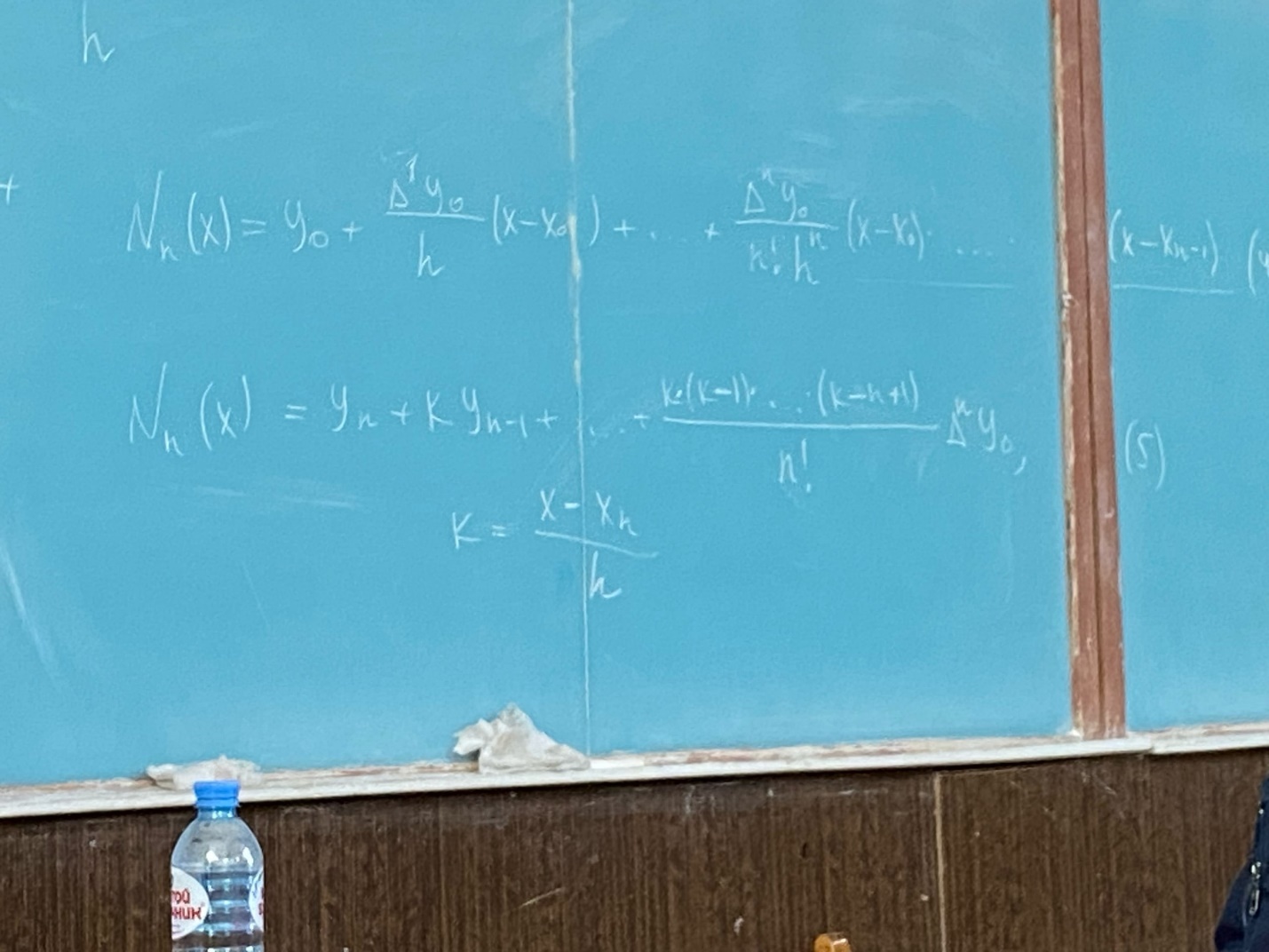
Nn(x) = f(x0) + f(x0, x1) \* (x-x0) + f(x0, x1, x2) ( x – x0) (x – x1) + ... + f(x0, x1, ... , xn) (x – x0) (x – x1 \* ... \* (x – xn+1) (3)

Пусть таблица 1 задана с постоянным шагом h и необходимо найти с помощью интерполяционного многочлена Ньютона приближенное значение ближе к началу таблицы, тогда производим интерполирование

Первая интерполяционная форма Ньютона и интерполирование – интерполирование вперед.



(4)



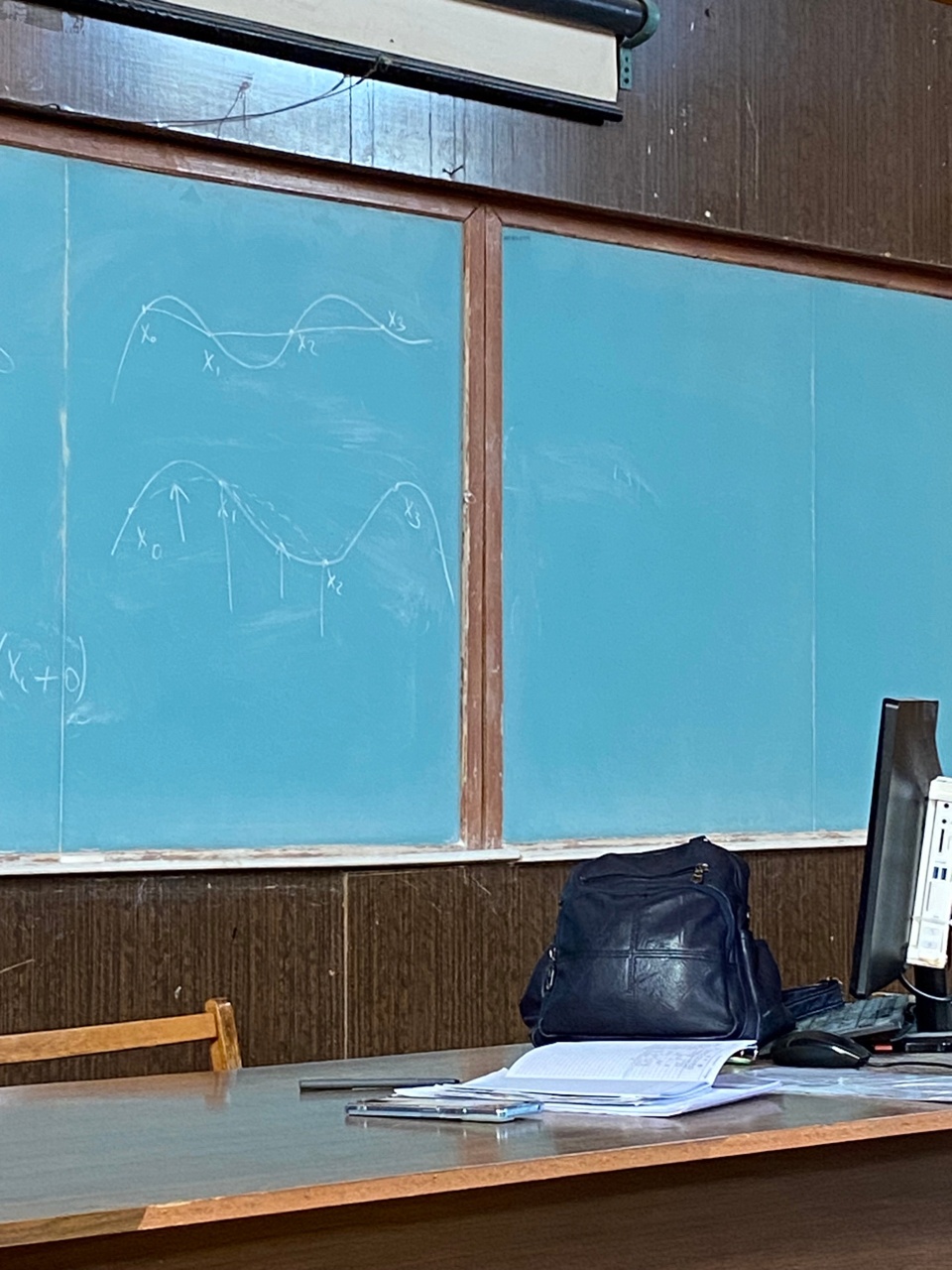
Формула (5) – второй интерполяционный многочлен (формула) Ньютона [интерполирования назад]

Сплайн-интерполяция

Сплайн – кусочное приближение функции.

Сплайном порядка m называется функция, являющаяся многочленом степени m [xi-1, xi], i = 1, ... , n и принимающая заданное значение Ф(Xi) = Yi

Кроме того, сплайн имеет прямые производные порядка m-1 и еще дополнительно Фm-1(x0) = Фm-1(xn) = 0



Рассмотрим сплайн первого порядка:

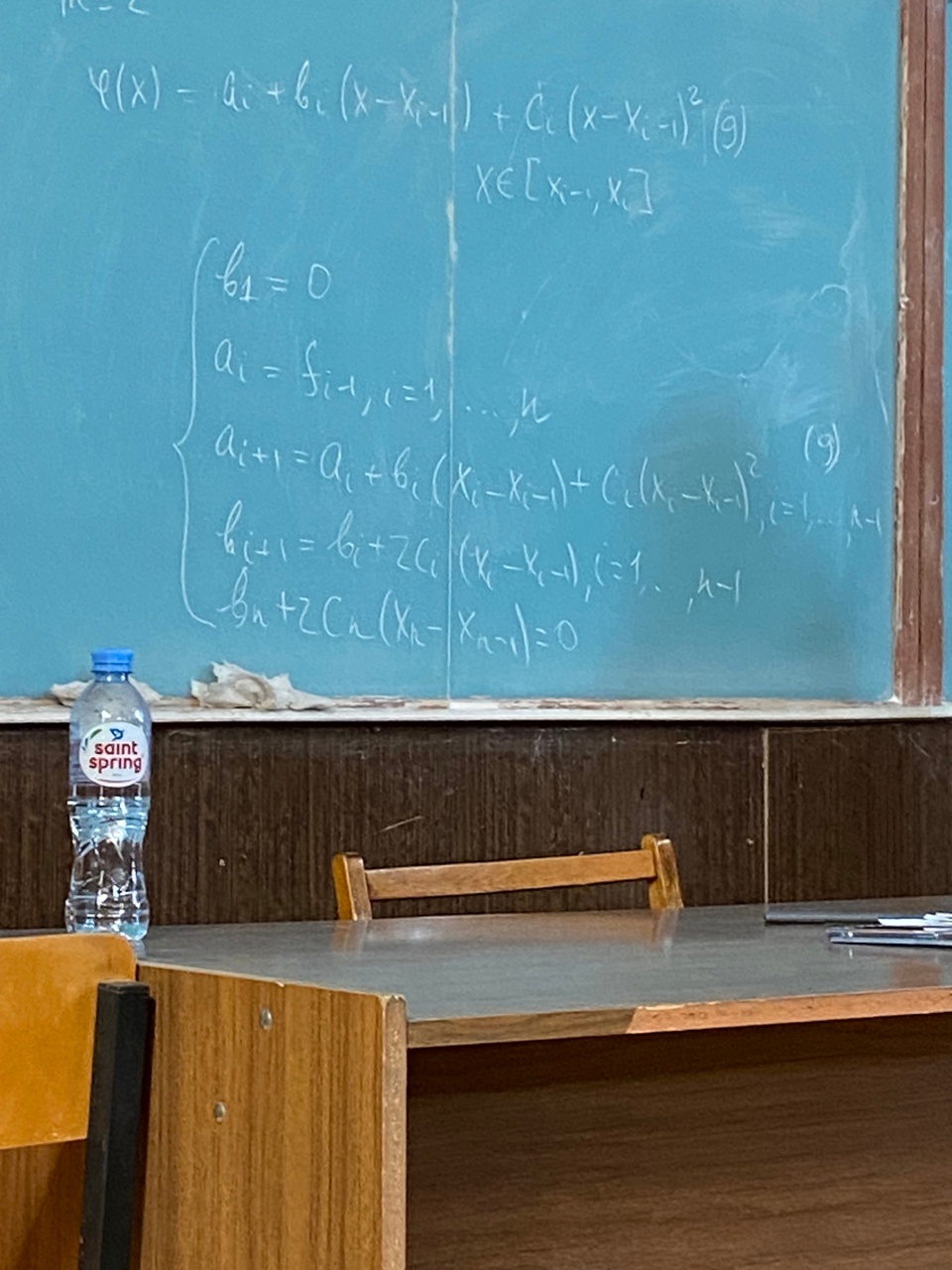
{Ф(x) = ai + bi(x-xi-1), x e [xi-1, xi] (7)

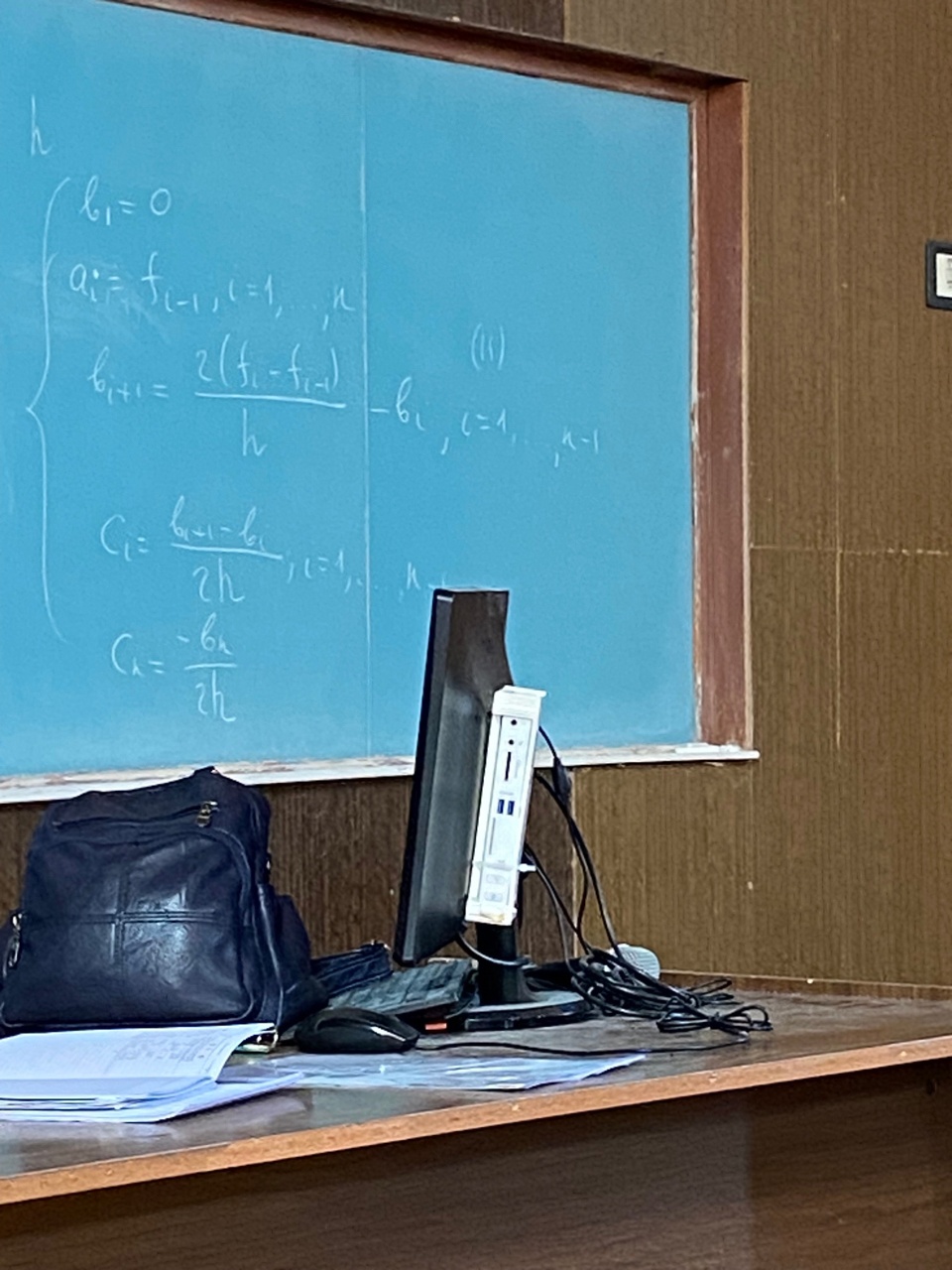
{Ф(xi-1) = ai

{ai = fi-1  
{bi = (fi – ai) / (xi - x0)

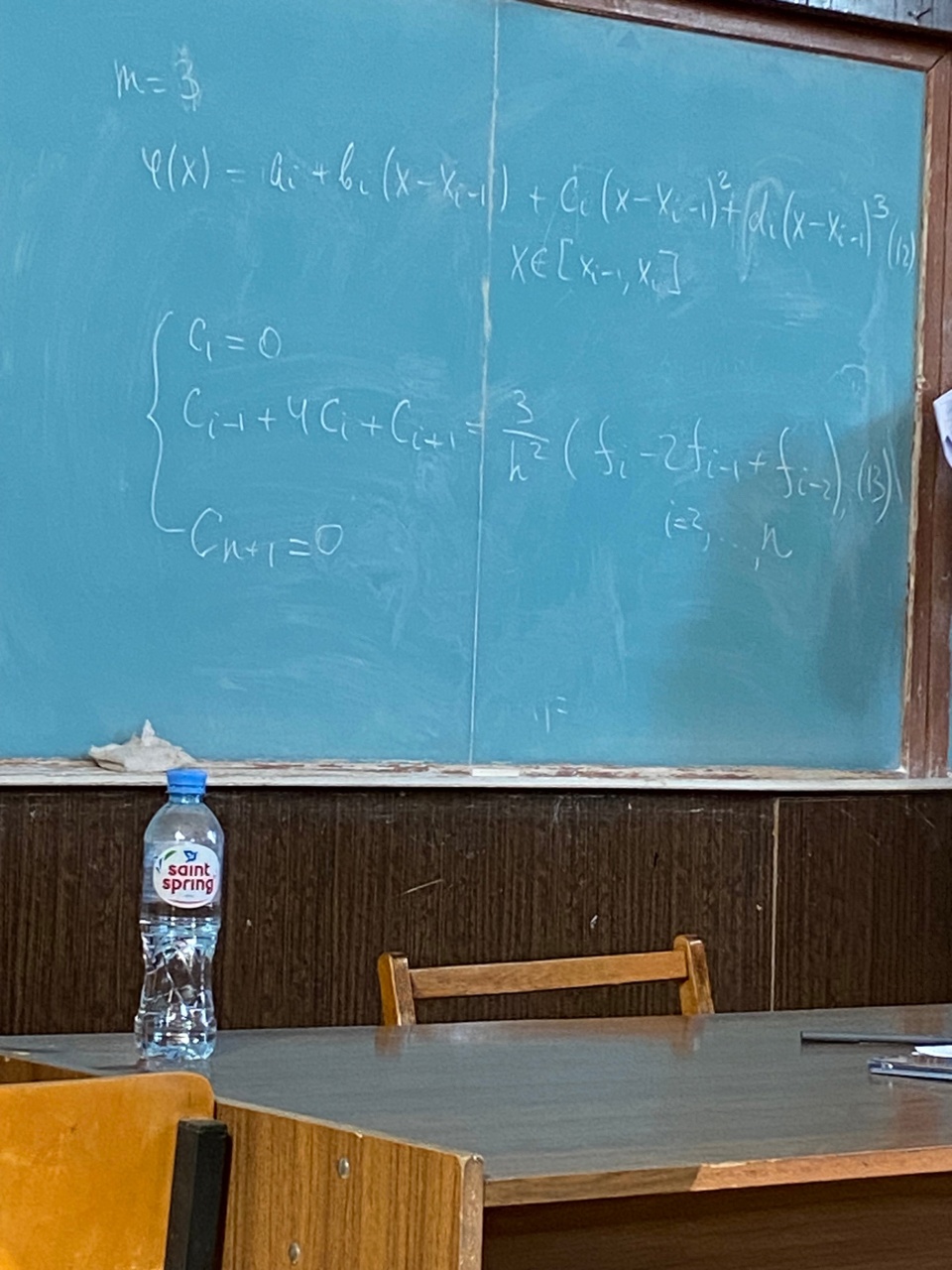
Сплайны второго порядка (m = 2)

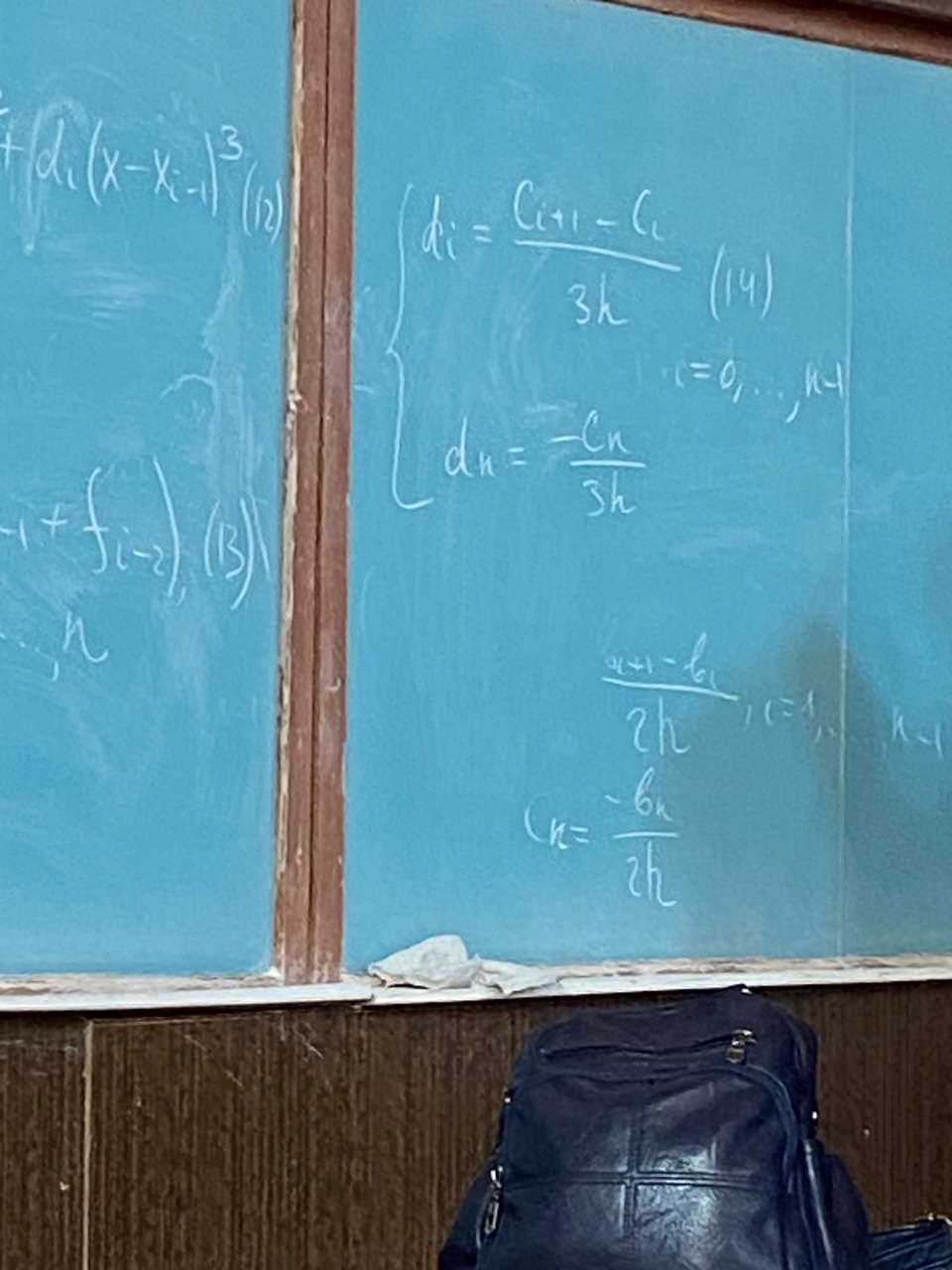
Ф(x) ai + bi (x – xi-1\_ + Ci(x-xi-1­)2 , x e [xi-1, xi] (9)

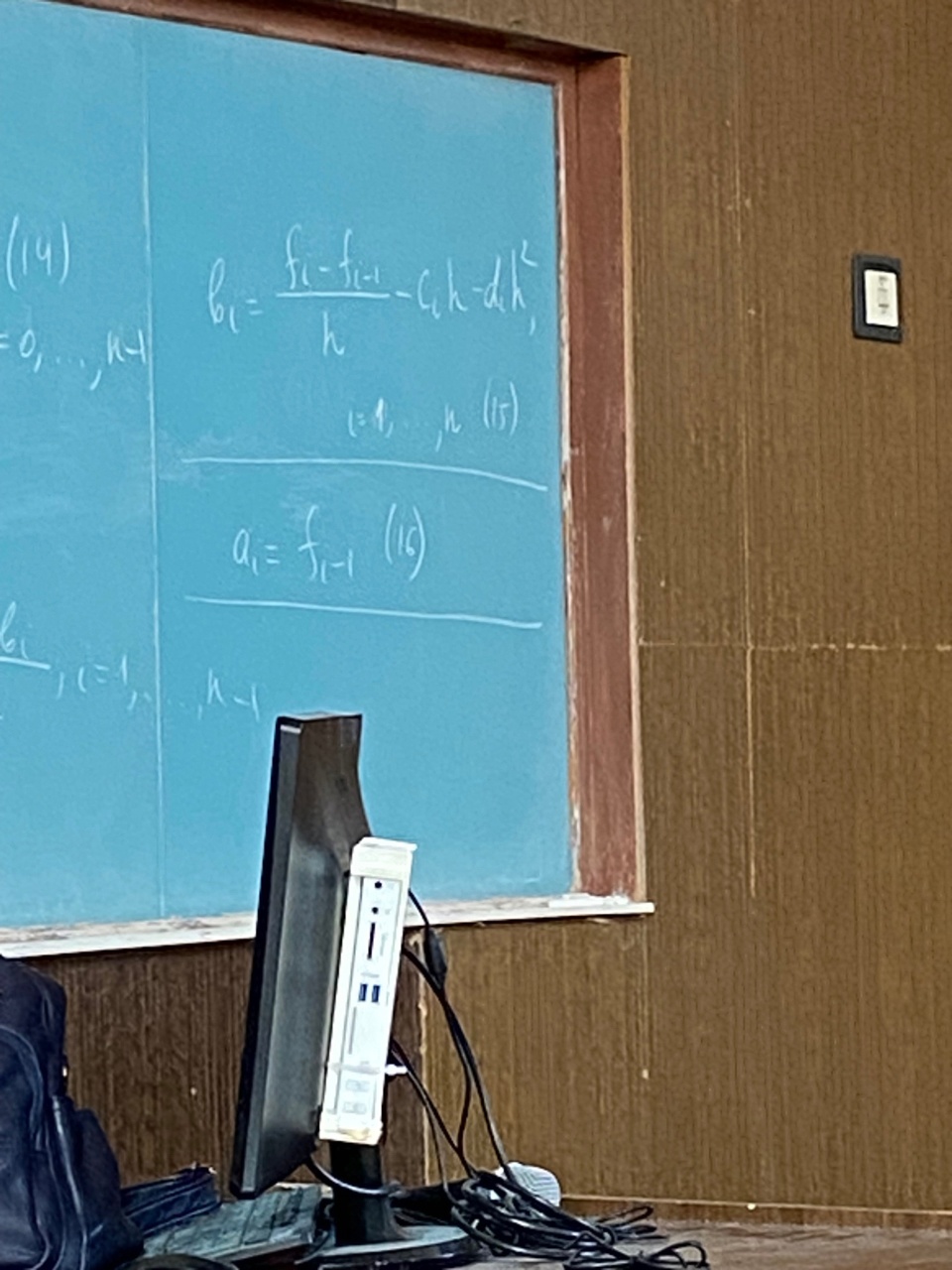




Сплайн 3го порядка (добавляется новое слагаемое di)



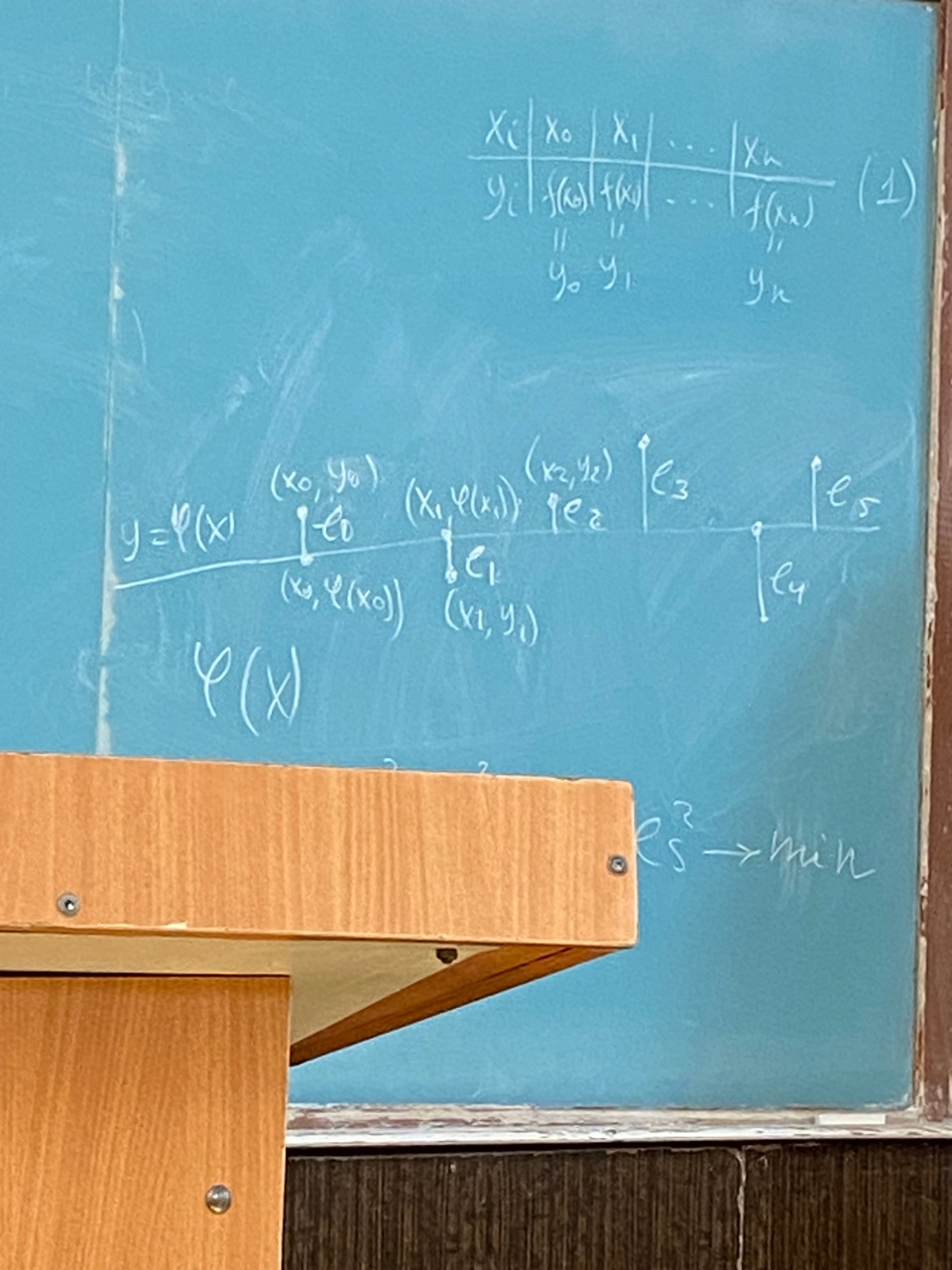


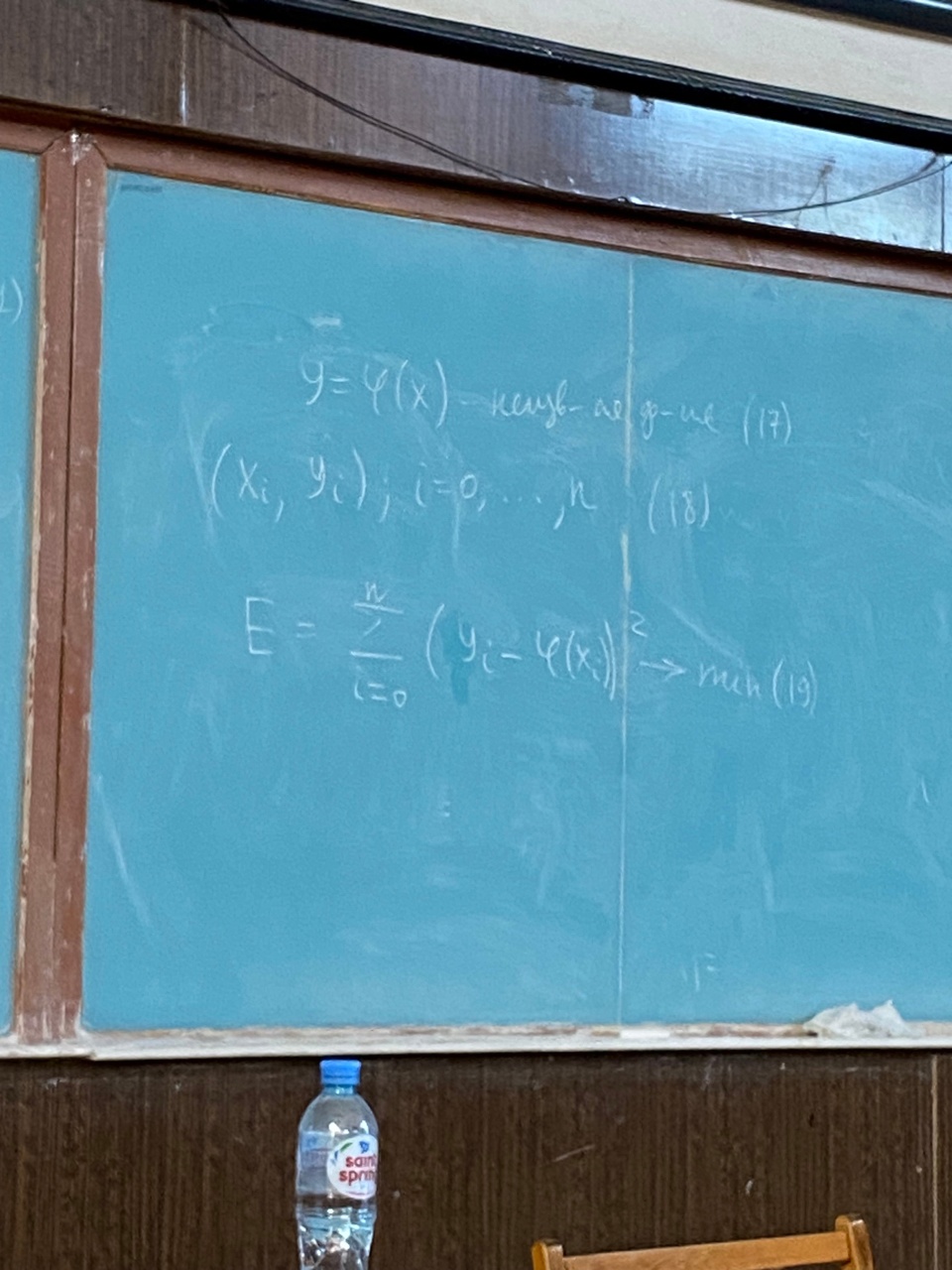


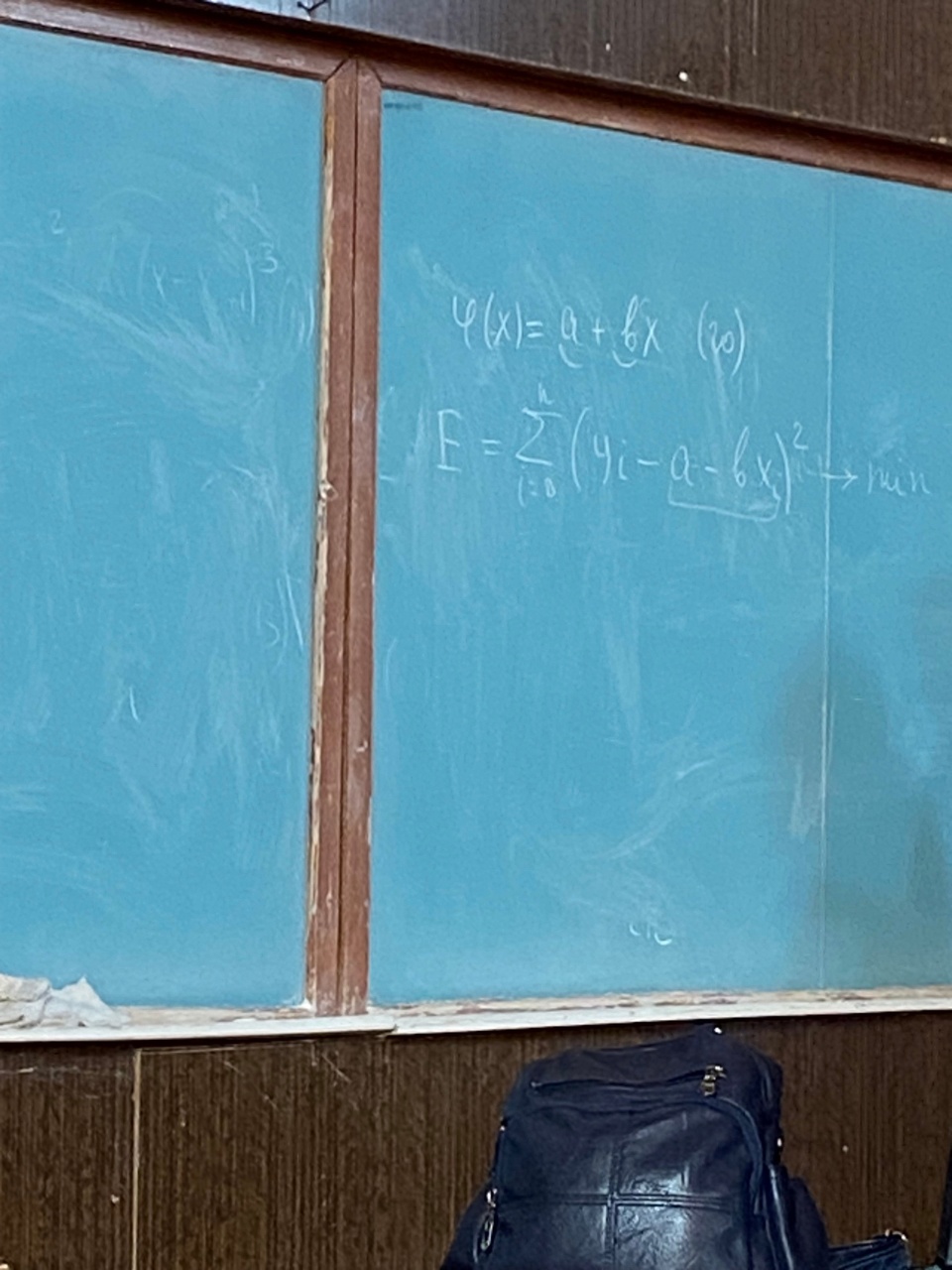
Сглаживание. Метод наименьших квадратов

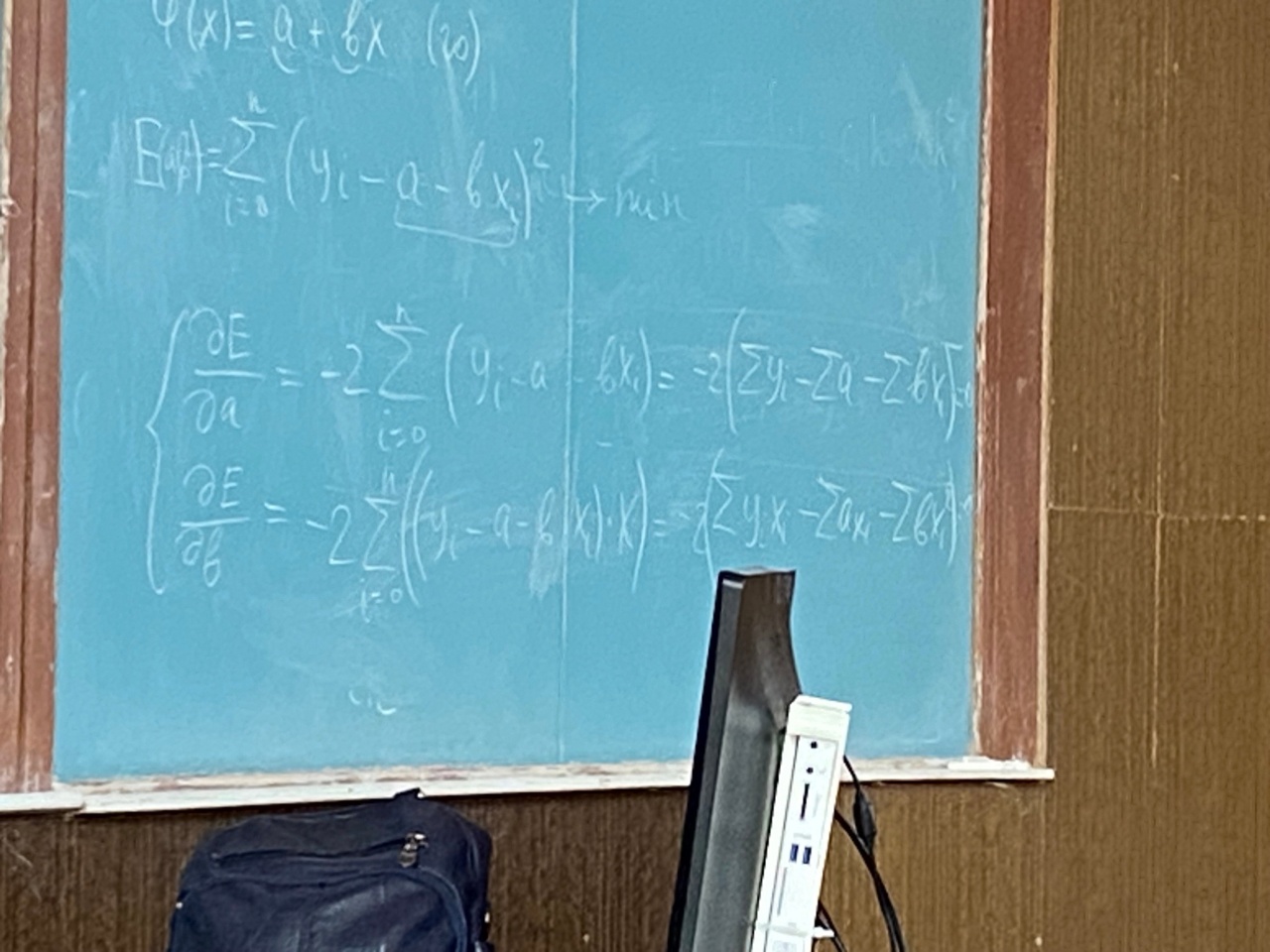
Пусть задана функция в виде таблицы 1

Будем производить аппроксимацию этой функции сглаживанием.









Ф(x) = e^(a+bx)

ln(Ф(x)) = a + bx = ln(y)

В (21) вместо y -> ln(y)

Ф(x) = a\*by

ln(Ф(x)) = ln(a) + (ln(b \* x)) = ln(y)

Дифференцирование функций с помощью интерполяционных многочленов

С помощью многочлена Лагранжа 1го порядка.

L1(x) = (x-x1)/(x0-x1) \* f(x0) + (x-x0)/(x1-x0) \* f(x1) (11)

Возьмем равномерную сетку с шагом h, то L1 будет выглядеть следующим образом:

L1(x) = y0 (x- x1) / -h + y1 (x-x0) / h

k = (x – x0) / h

x – x0 = kh

x – x1 = h(k-1)

L1(k) = y0(k+1) - y1k b (23)

Найдем dx от этого многочлена:

(L1(k­))1x = (y1 – y0) \* k’x = (y1 – y0) / h -> f’(x1) = (y1 – y0) / h

L2(k) = (½)y0(k-1)(k-2) = y1k(k-2) + (1/2)y2k(k-1) (24)

f’(x) ~ L’2(x) = 1/h(1/2\*y0 ( 2k-3 ) = y1(2k-2) + (1/2) y2 (2k-1) (25)