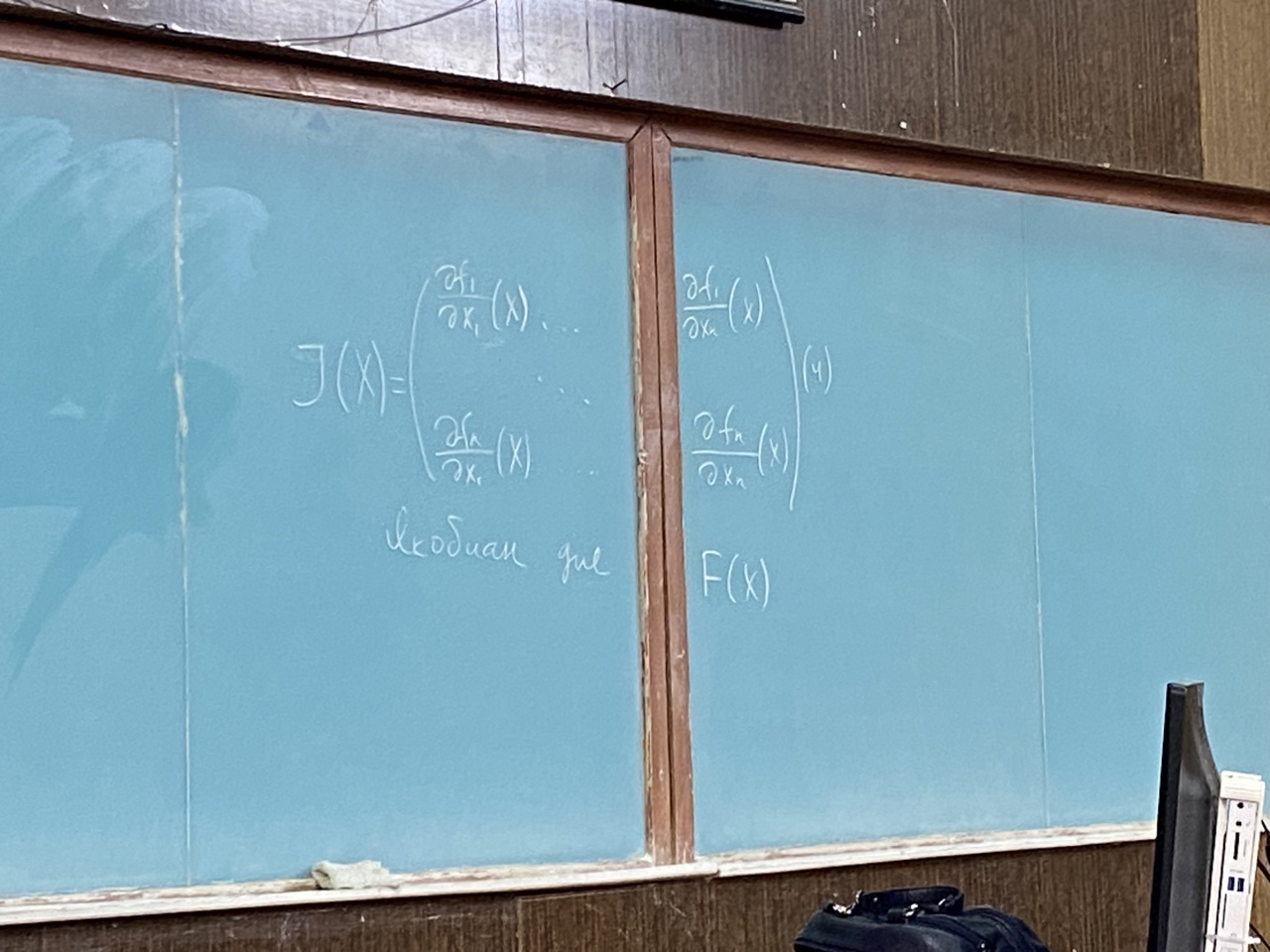
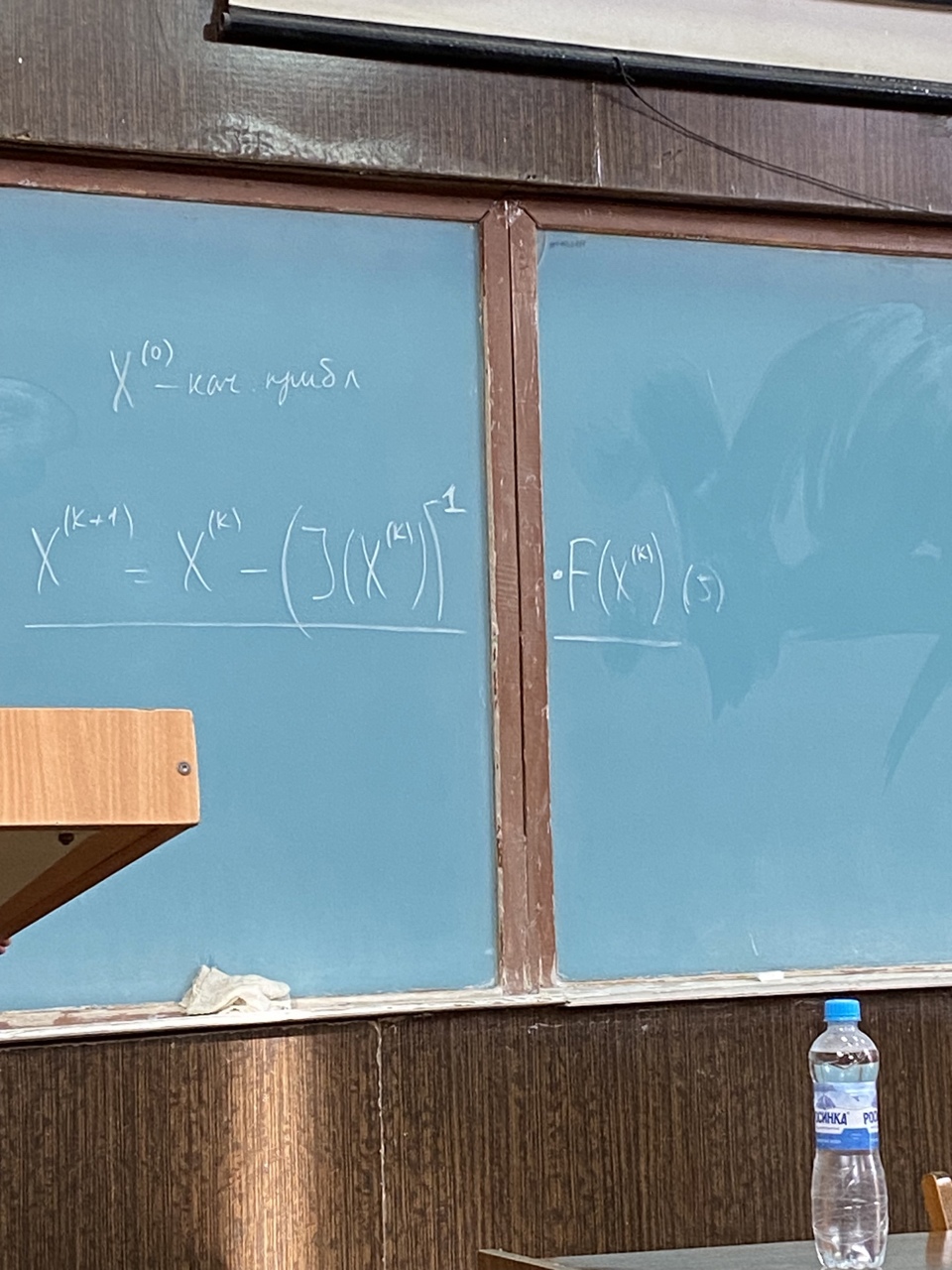
Метод Ньютона



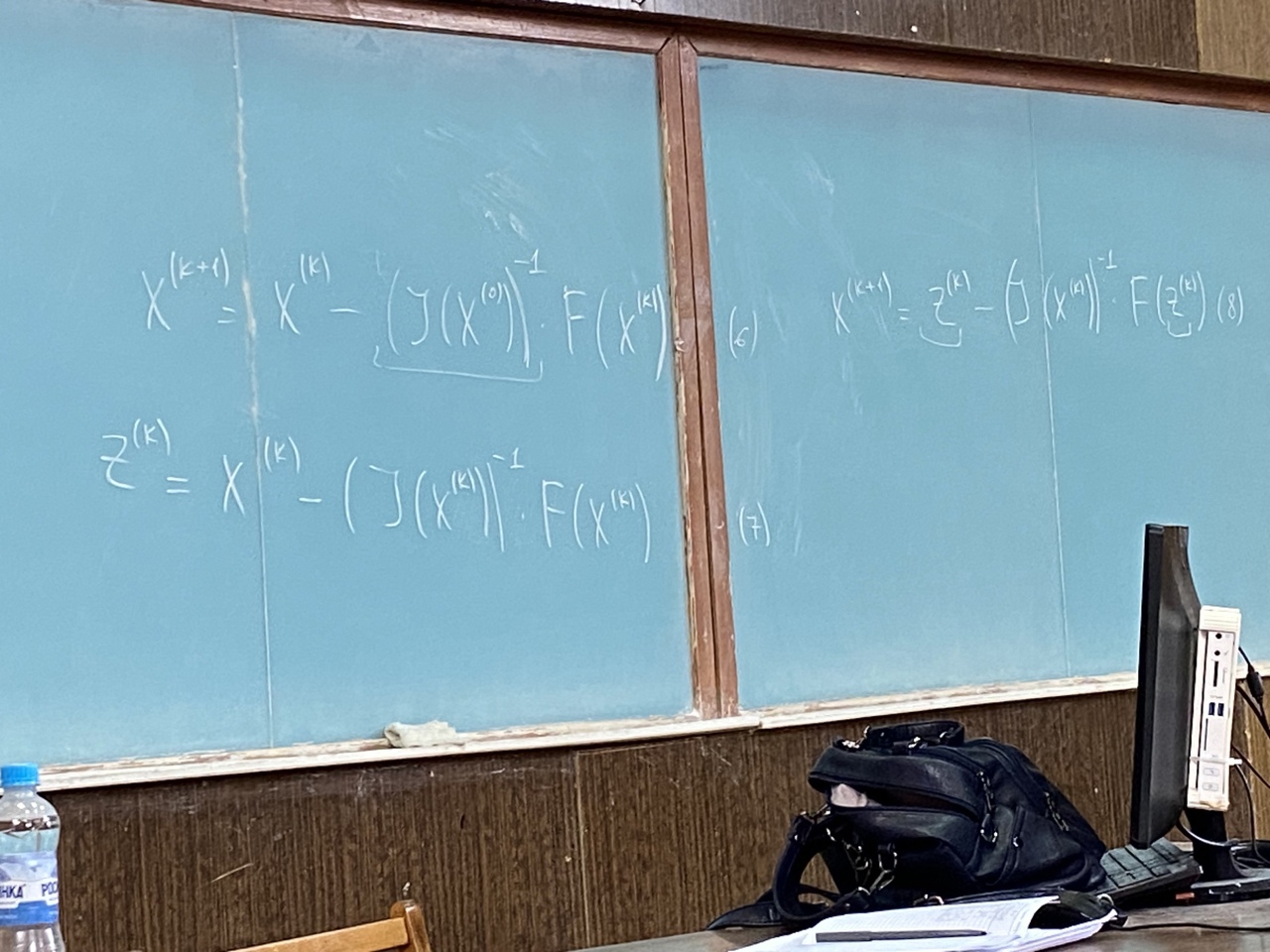


Метод Ньютона отлично сходится.

Модификации метода Ньютона.   
1. Модифицированный метод Ньютона.  
2. Двуступенчатый

1. Главное отличие – якобиан (обратную матрицу) считаем в x0 один раз

2. большая формула, но этот метод сходится быстрее метода Ньютона (^3 скорость)+ матрицу Якоби считаем через раз



Метод простой итерации

Пусть необходимо решить систему нелинейных уравнений (3).

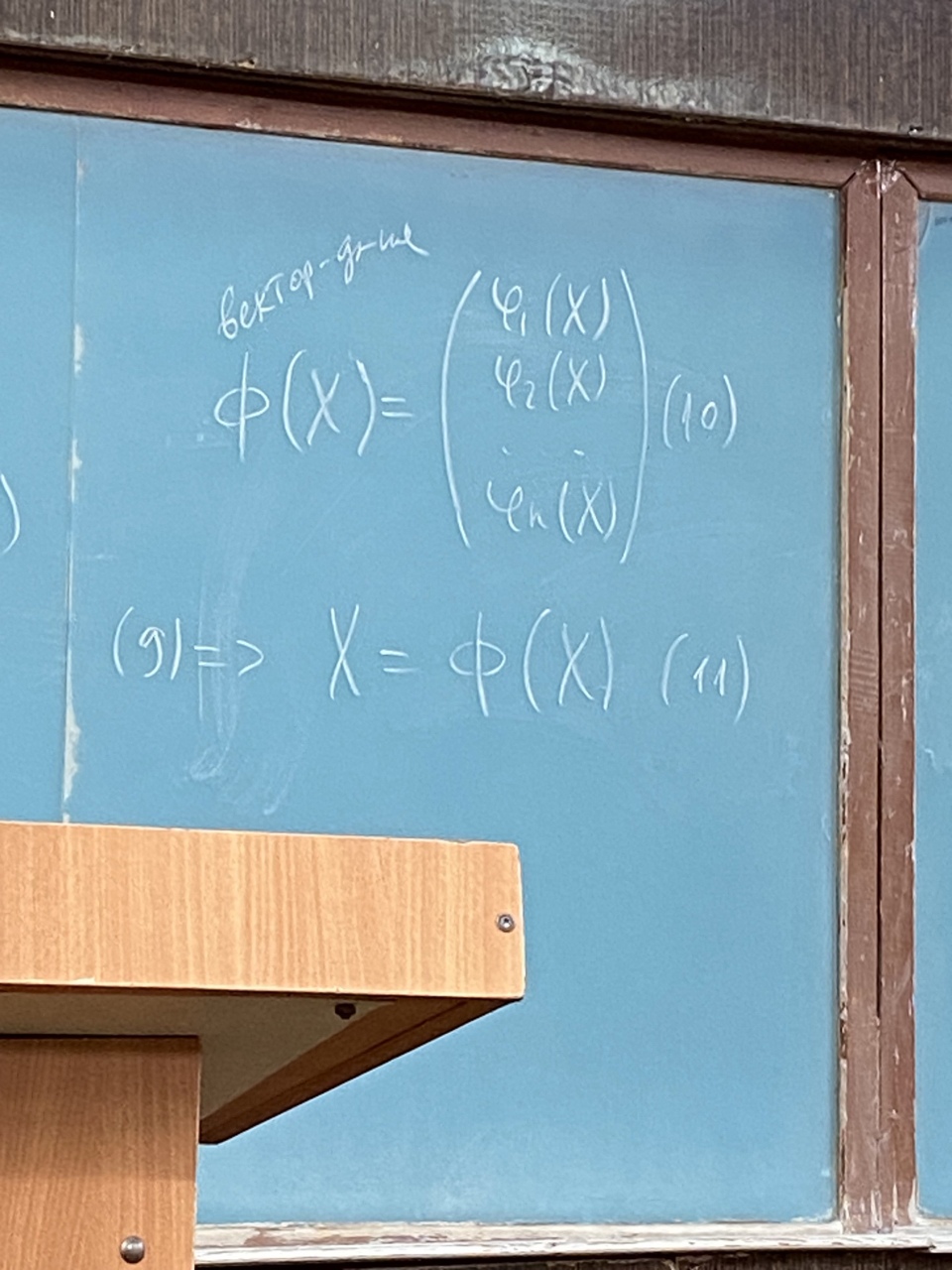
F(x) = 0 (3)

Преобразуем её к следующему виду:

x1 = ф1(x1, ... , xn)  
x2 = ф2(x1, ... xn)

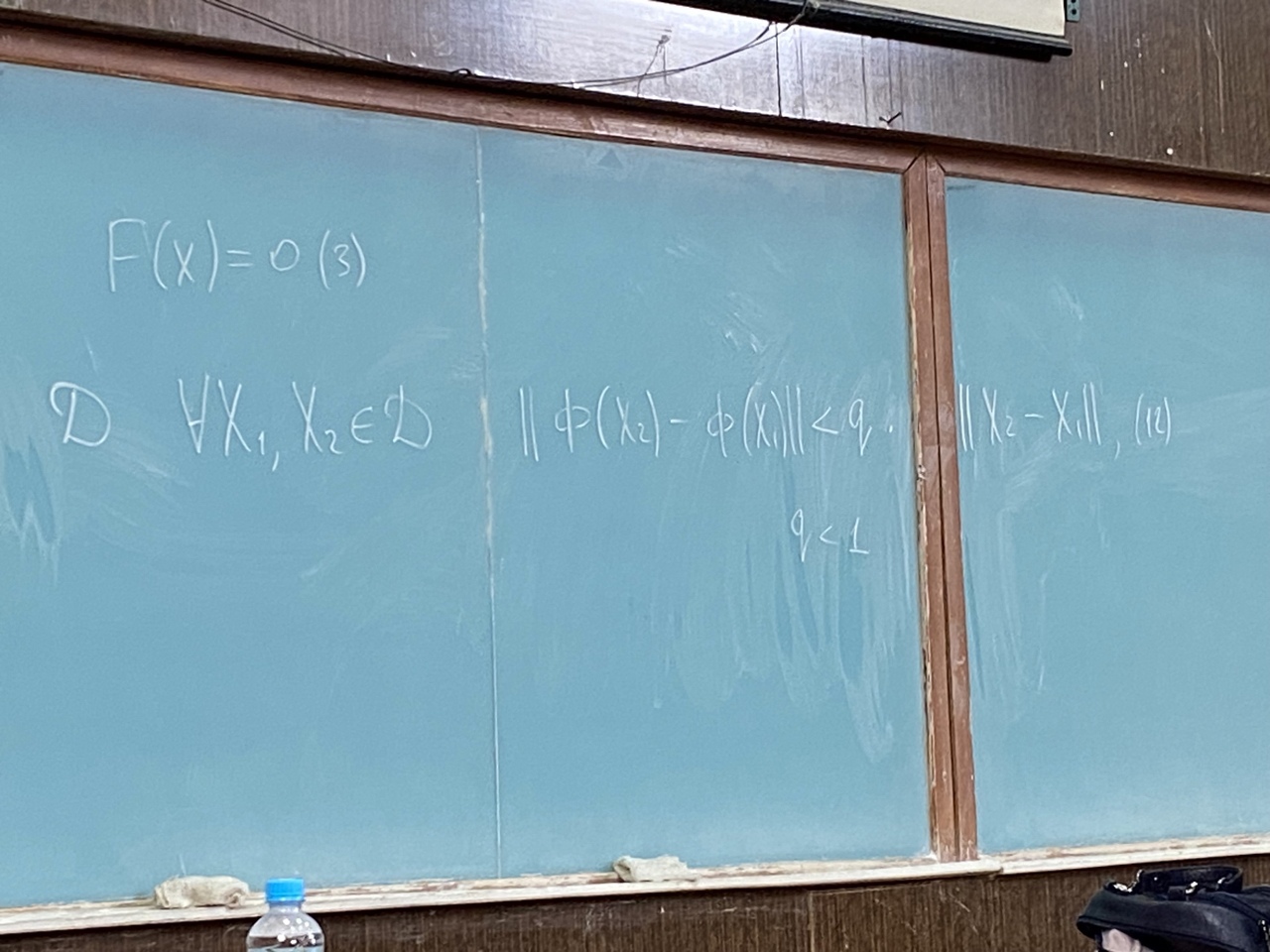
...

xn = фn(x1, ..., xn)



Ф(x) – сжимающий оператор на каком-то множестве D, если для любых x1, x2 выполняется следующее свойство: норма Ф(x2) – Ф(x1) < q \* ||x2 – x1||, где q – число меньше единицы.

То же самое в математическом определении:

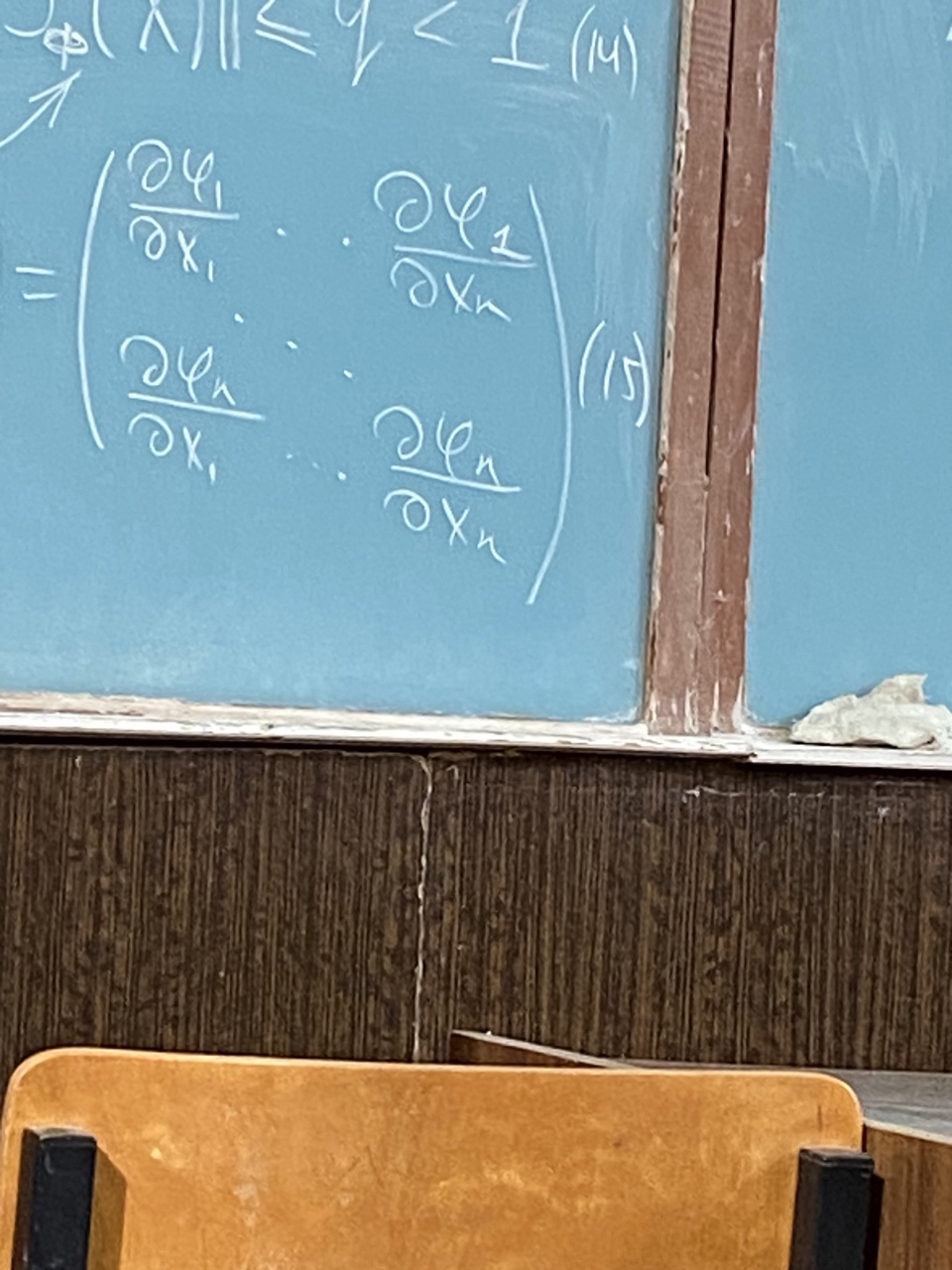


Условие (12) – условие Липшица.

Xk+1 = Ф(Xk) (13), если xk e D

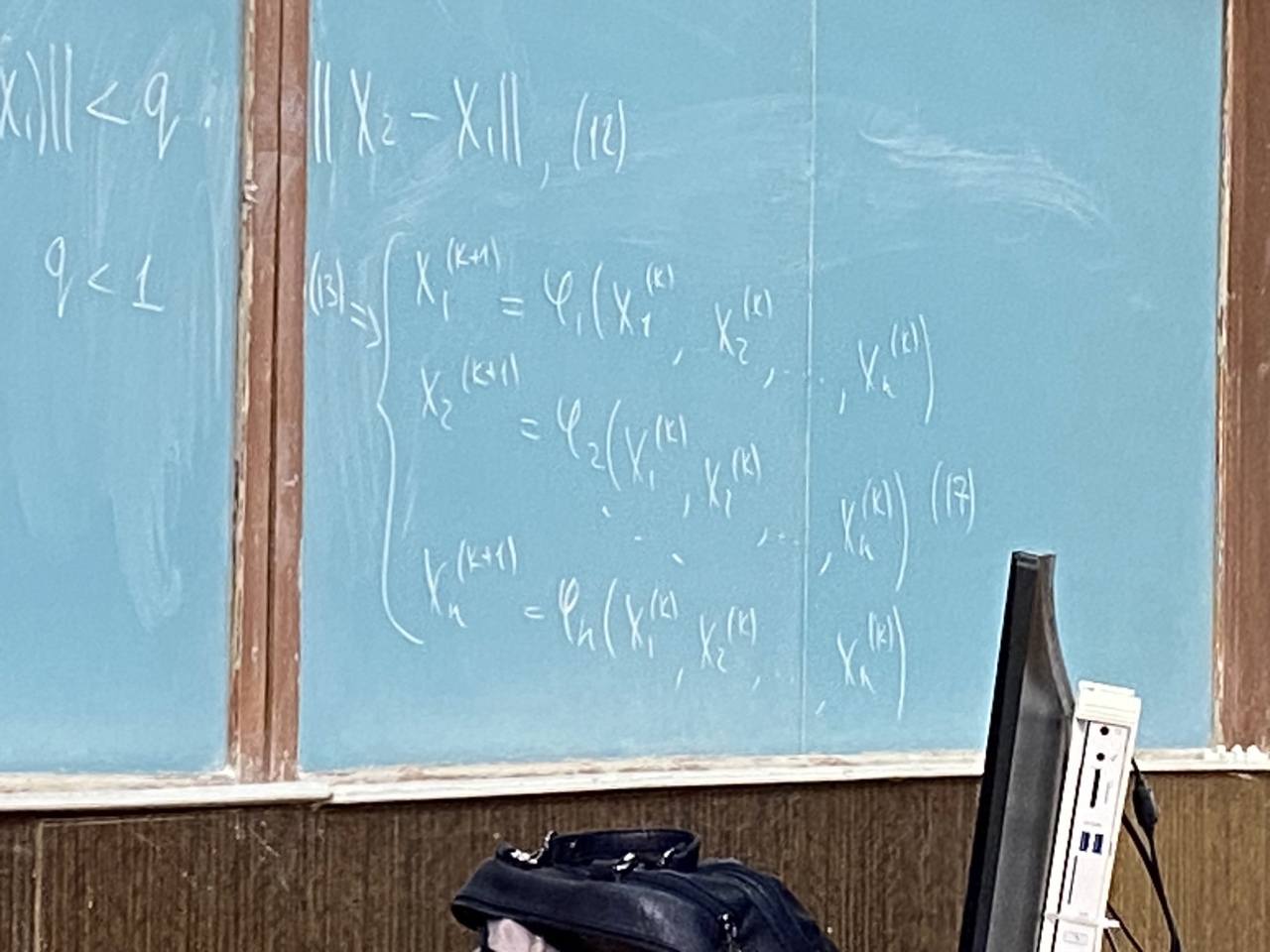
**Теорема о сходимости итерационного процесса.**

Пусть в некоторой области D выполняется условие Якобиана ||Y(X)|| <= q < 1 (14), где

Y(X) = 

Погрешность сходимости процесса:   
||xk+1-x\*|| < q/(1-q) ||Xk+1-Xk|| < E (16)

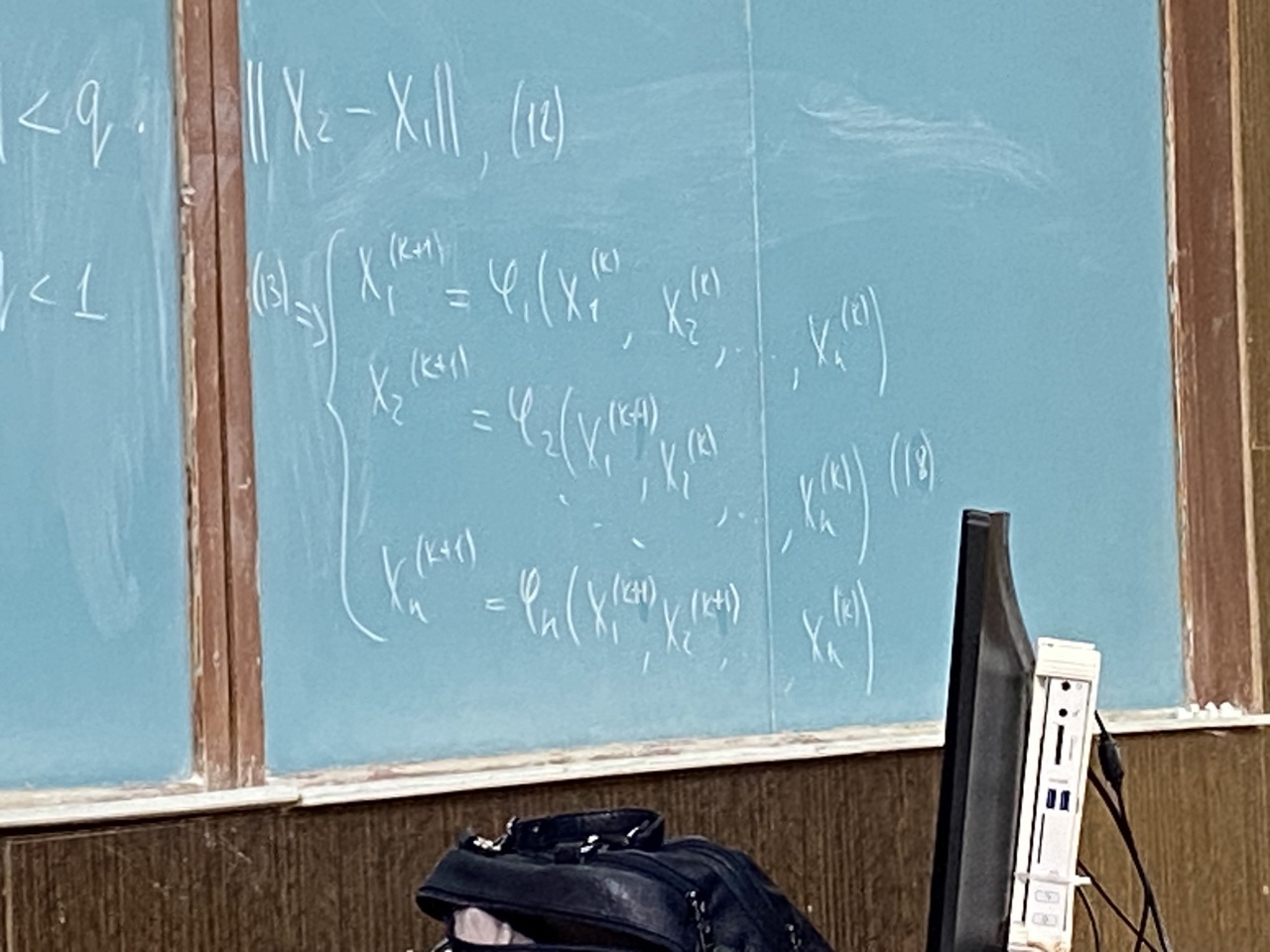
Запишем формулу (13) в координатной форме:



Основной недостаток: не факт, что при следующей итерации все сойдется.

Метод Зейделя

Главное отличие: на каждой итерации x2 считаем по x1.



Метод градиентного спуска для поиска минимума функции и для решения системы уравнения.

f1 (x) = 0

...

...

fn(x) = 0

(19)

Пусть задана система нелинейных уравненй 19

M(X) = f21(X) + ... + f2n(X) (20)

X\*: M(X\*) – min M(X)

Минимум этой функции равен

X\* удовлетворяет (19), потому что M(X\*) = 0 => (19) – верна для точки X\*

Задача нахождения решения системы (19) сводится к нахождению минимума функции (20). Найдем градиент функции M:

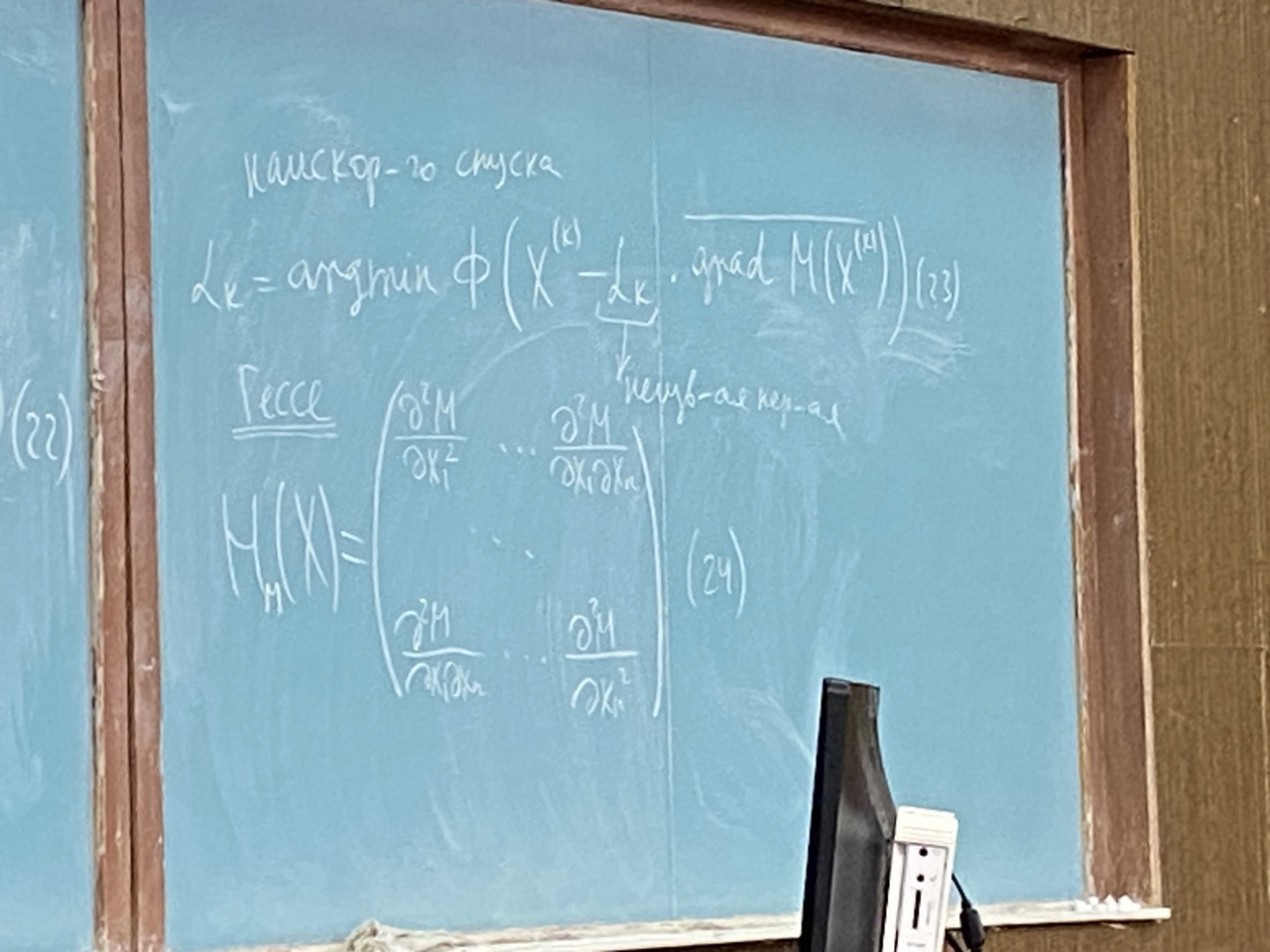
gradM = (dM/dx1, ..., dM/dxm) => метод градиентного спуска имеет следующий вид:

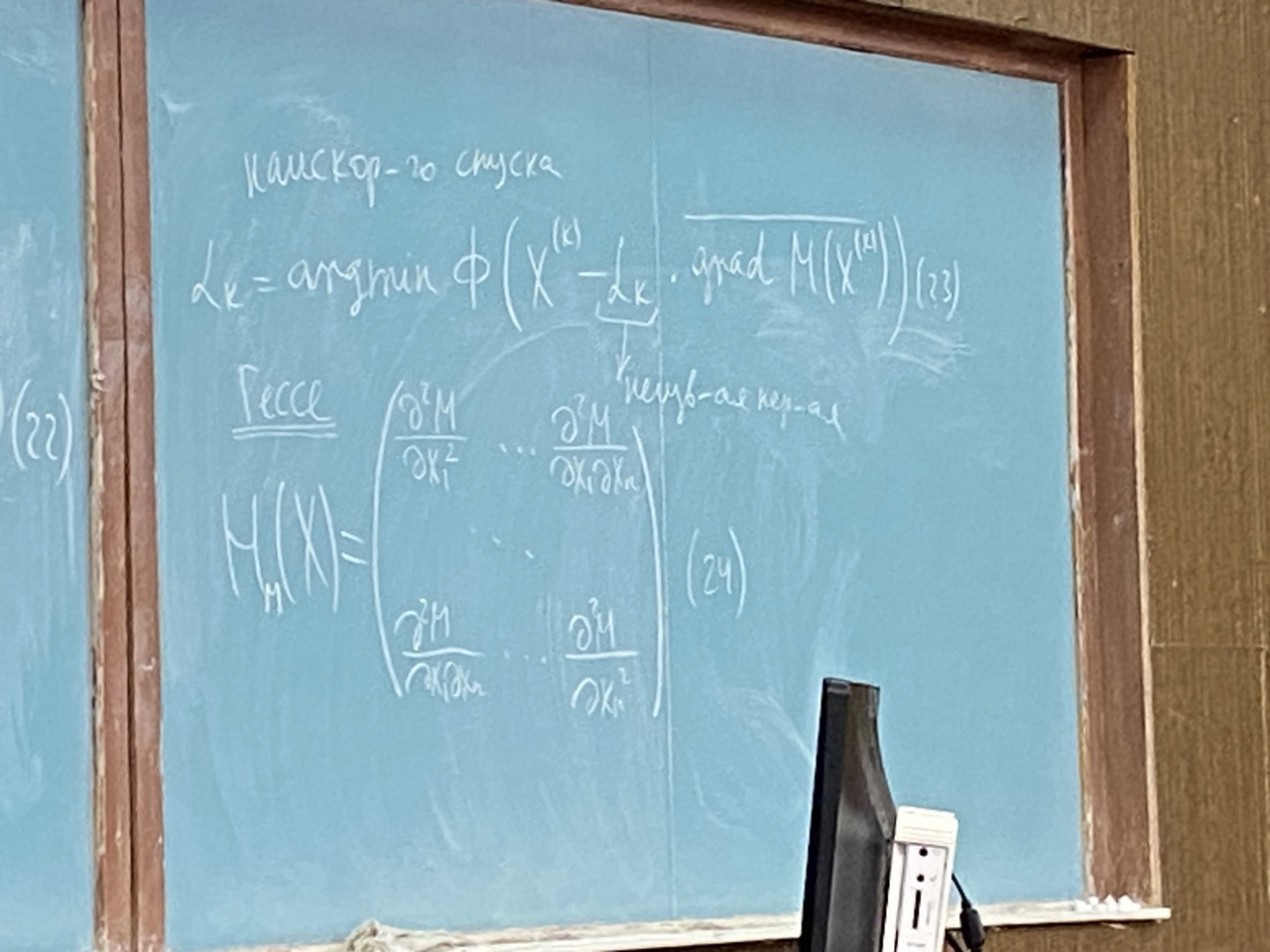
X(k)= X(k)– ak \* gradM(X|k|) (23)

проблема в нахождении ak, и e (0, 1) (задаем произвольно)

Есть метод наискорейшего градиентного спуска (там ak мы находим на каждом шаге)

Гессе

Ньютон выявил, что a достаточно заменить на обратную матрицу Гессе



**Аппроксимация функции**

Задача восстановления аналитической функции по отдельным значениям называется аппоркимацией.

Существует два основных подхода – это интерполяция и сглаживание.

Рассмотрим интерполяцию:

Интерполяцию – это приближенное или точное нахождение значений какой-либо величины по известным отдельным значениям этой величины или значением других величин, связанных с данной.

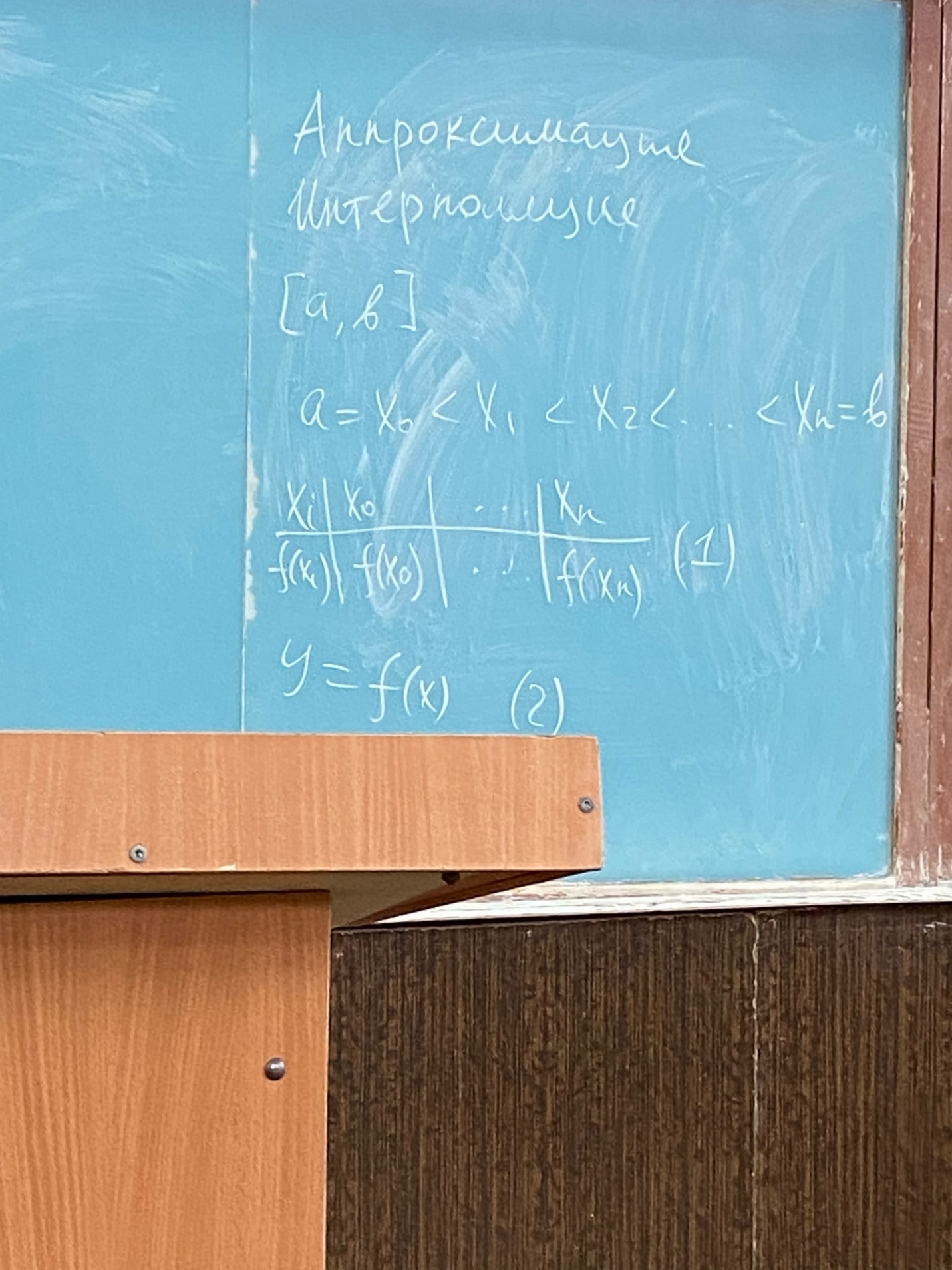
Пусть на отрезке [a, b] задано некоторое разбиение x0, x1 , ... , xn

a = x0 < x1 < x2 < ... < xn

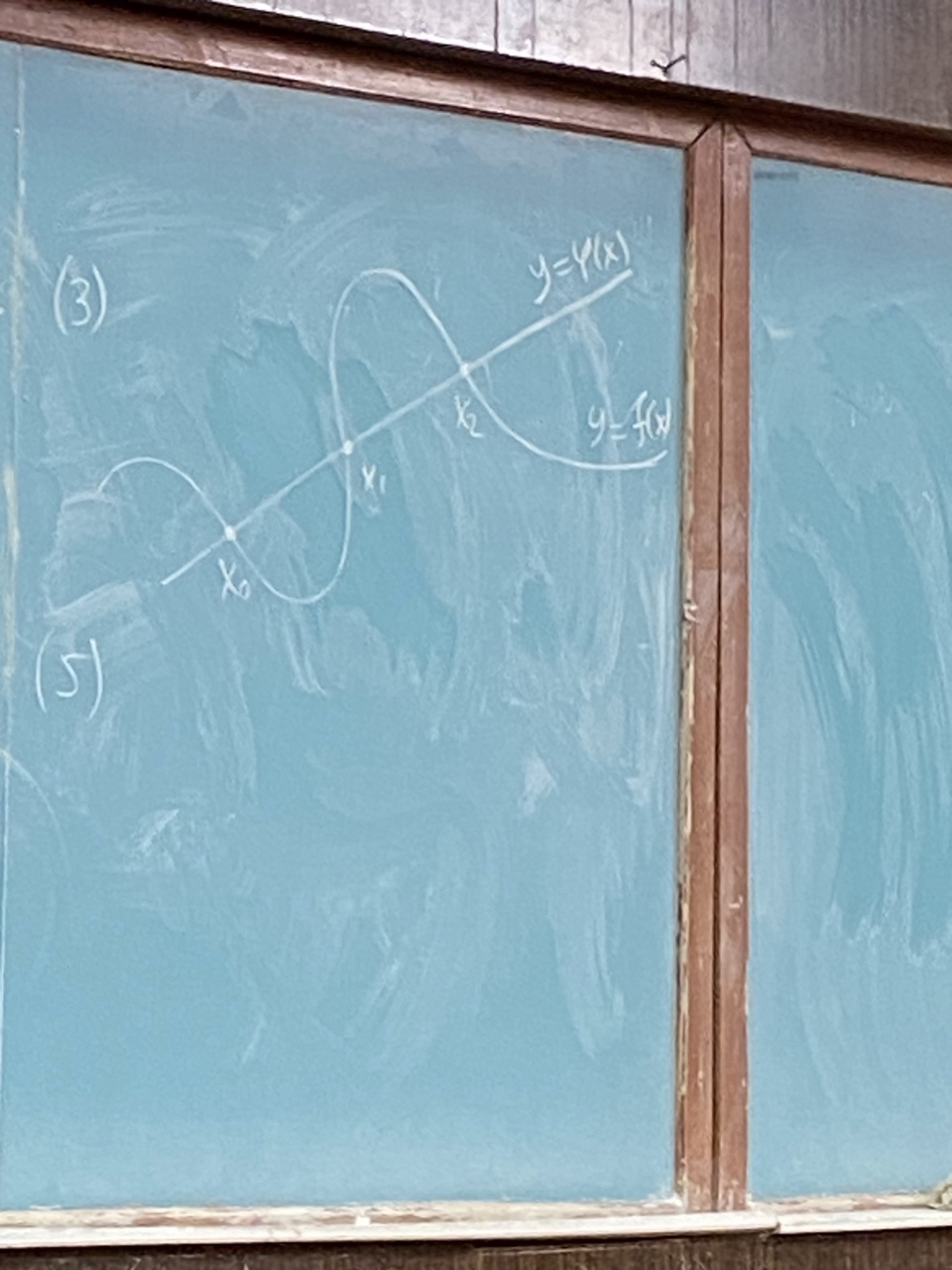
И известно значение функции f в точках этого разиения.

Таблица 1 назвается сеткой. Точки x0, x1 ... называются узлами сетки или узлами интерполяции.

Если расстояние между точками xk, xk-1 = h (3) , то сетка называется равномерной.



Функция y = ф(x) (4) называется интерполяционной на отрезке [a, b] для f(x), если на каждой точке ф(xk) = f(xk) (5)



Интерполяционным многочленом или полиномом степени n Ln(x) = a0 + a1x+ ... +anxn (6)

такой, что Ln(xk) = f(xk) для всех k

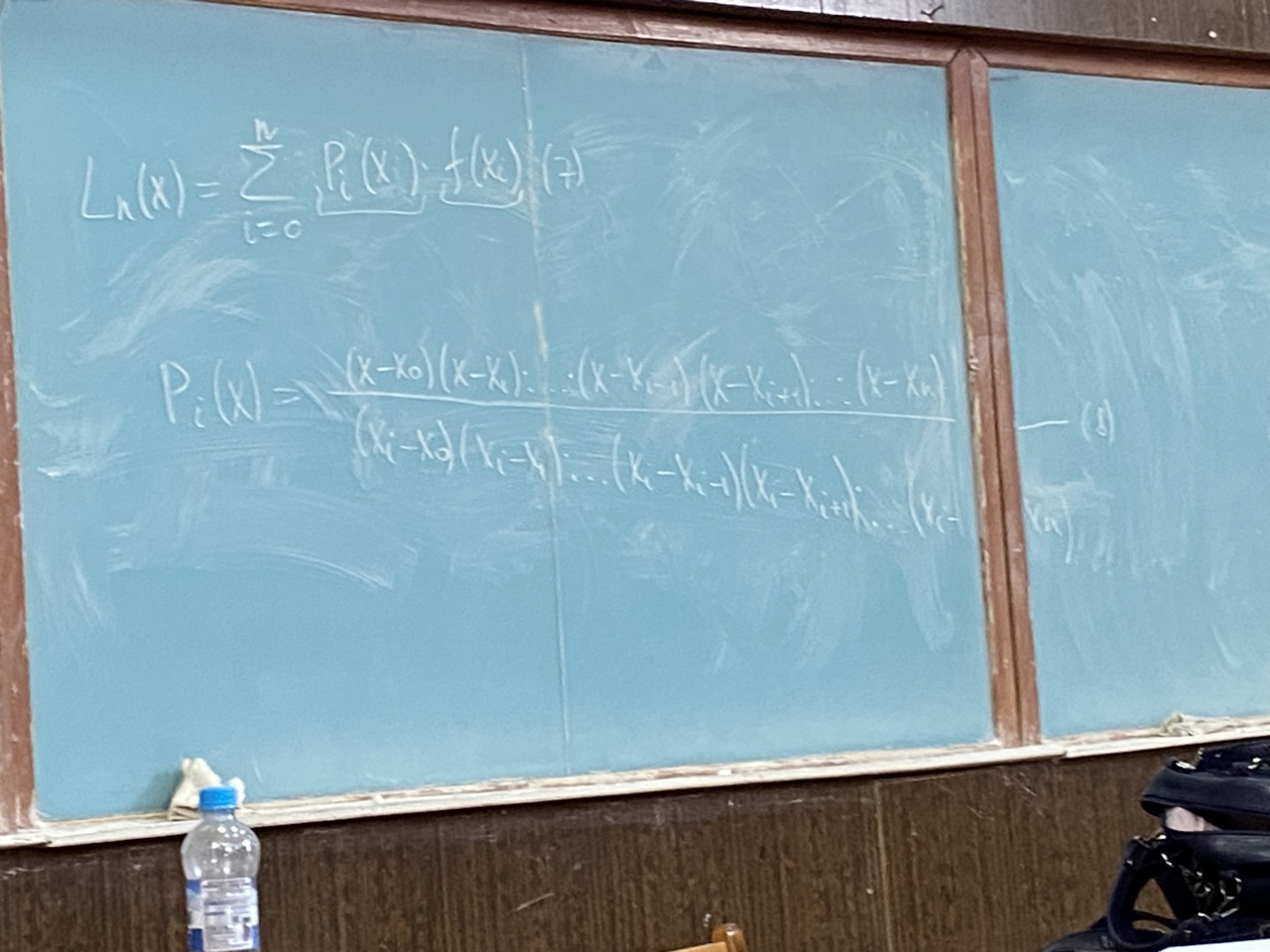
Интерполяционные многочлены нужны для того, чтобы упрощать функцию.

Например, узнать приближенное значение функции в какой-либо точке.

Если мы находим значение в [a, b] , то такая задача называется интерполирование или интерполированием в узком смысле.   
Если мы находим значение вне [a, b] , то такая задача называется экстраполирование или интерполяция в широком смысле.

Интерполяционный многочлен Лагранжа имеет вид: Ln(x) = SUM(Pi(x)\*f(x1)) (7)

P и f здесь это базисные многочлены, которые имеют следующий вид:



P1(x1) = 1 (9) =>   
Lm(xi) = f(xi) (10)

Рассмотрим многочлен Лагранжа первого порядка:

xi x0 x1

yi y0 y1

P1(x) = (x-x0)/(x1-x0) => P0(x) = (x-x1)/(x0-x1)

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

Взяли функцию и построили прямую.

L2(x) = ((x-x0)\*(x-x1))/((x0-x1)(x0-x2))\*y0 + (x-x0)(x-x1)/((x1-x0)(x1-x0)) \* y1.