**Решение системы дифференциальных уравнений (ДУ) с использованием разностных схем**

Решение системы дифференциальных уравнений (ДУ) с использованием разностных схем — это численный метод, при котором дифференциальные операторы заменяются численными приближениями дифференциальных операторов.

Разностная схема — это способ приближенного решения дифференциальных уравнений с помощью конечной системы алгебраических уравнений, которые получаются заменой производных разностными отношениями на некоторой сетке.

Сетка — это множество точек, на которых определяются значения функции, решающей дифференциальное уравнение. Сетка может быть равномерной или неравномерной, одномерной или многомерной, в зависимости от типа уравнения и области, в которой оно решается.

**Явные и неявные разностные схемы**

Явный и неявный метод — это два вида разностных схем, которые отличаются тем, как определяются значения сеточной функции на новом слое по времени.

1. Явный метод использует значения сеточной функции только на предыдущем слое, то есть известные значения.
2. Неявный метод использует значения сеточной функции как на предыдущем, так и на новом слое, то есть неизвестные значения.

Это означает, что для решения неявной схемы нужно решать систему алгебраических уравнений, в то время как для решения явной схемы достаточно подставить известные значения в формулу.

Отличие явного и неявного метода влияет на их свойства, такие как устойчивость, сходимость и точность:

- Устойчивость означает, что решение разностной схемы не увеличивается по амплитуде при увеличении времени.

- Сходимость означает, что решение разностной схемы стремится к решению дифференциального уравнения при уменьшении шагов сетки.

- Точность означает, что решение разностной схемы близко к решению дифференциального уравнения.

**В общем случае, неявные схемы более устойчивы, чем явные, но менее точны и требуют больше вычислительных ресурсов.**

Явные схемы более просты в реализации, но имеют ограничения на выбор шагов сетки, чтобы обеспечить устойчивость.

Какой метод лучше, зависит от конкретной задачи и требований к решению.

Мы имеем дело с численным решением начально-краевой задачи для уравнения теплопроводности одномерных нестационарных процессов (также известного как уравнение диффузии) в виде:  
  
где u(x, t) - искомая функция, x и t - пространственная и временная координаты соответственно.

### **Начальные и граничные условия:**

1. Начальное условие задаётся как ( u(x, 0) = \sin(x) ), что соответствует начальному распределению температуры.
2. Граничные условия задаются как ( u(0, t) = 0 ) (на левой границе области) и ( u(\frac{\pi}{2}, t) = e^{-t} ) (на правой границе области).

Точное решение этой задачи задаётся функцией ( u(x, t) = e^{-t}\sin(x) ), используемой здесь для проверки точности численных схем.

### **Явная разностная схема:**

Была использована явная конечно-разностная схема, где производная по времени аппроксимируется через конечную разность вперёд, а вторая производная по пространственной координате - через центральную разность. Это может быть записано как:



где индекс ( i ) обозначает шаг по пространству, а ( n ) - шаг по времени. ( r ) - число Куранта, задаётся как ( r = \frac{\Delta t}{(\Delta x)^2} ). Для устойчивости метода требуется, чтобы ( r \leq \frac{1}{2} ).

### **Неявная разностная схема:**

Неявная схема устойчива для любого ( r ), и выражается как:



Это приводит к системе линейных алгебраических уравнений с трёхдиагональной матрицей, которая может быть решена с помощью метода прогонки или функции scipy.linalg.solve, как в предоставленном коде.

### **Выбор шагов и построение сетки:**

В выборке использовались равномерные пространственные шаги ( \Delta x ) и временные шаги ( \Delta t ), обозначенные как dx и dt соответственно.

### **Выбор времени T:**

Время T не было указано в условии задачи, и поэтому оно должно быть подобрано исследователем таким образом, чтобы достигнуть достаточно малой ошибки между численным и точным решением за приемлемое время вычислений.

### **Графическое представление решений**:

Используются библиотеки matplotlib и seaborn для визуализации результатов численного решения и сравнения его с точным решением.

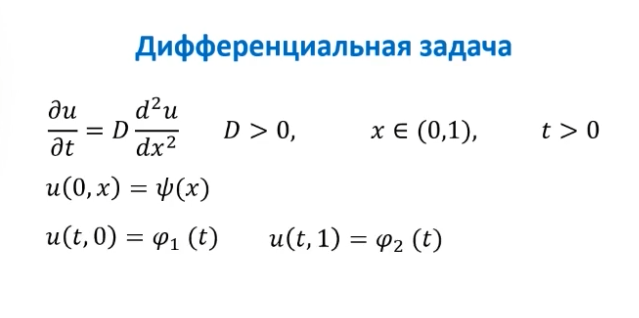
### **Вывод результатов**:

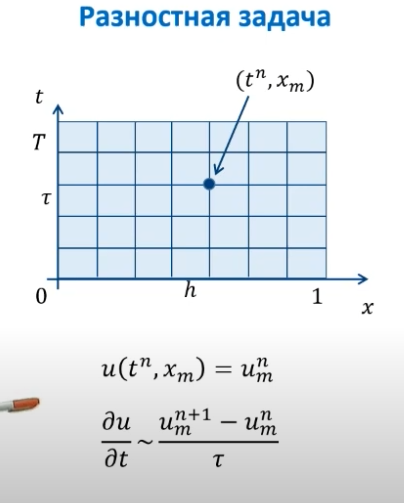
После расчёта численного решения (для явной и неявной схем отдельно), скрипт вычисляет абсолютные ошибки численных схем по сравнению с точным решением, а затем выводит их в консоль вместе со значениями численного решения в последний момент времени T и создает графики решений.

### **Теория, лежащая в основе реализации**:

1. **Разностные схемы** - численные методы для аппроксимации дифференциальных уравнений, заменяющие производные конечными разностями.
2. **Устойчивость схем** - важное свойство численного метода, гарантирующее, что ошибка не будет возрастать при увеличении числа шагов.
3. **Метод прогонки (TDMA)** - алгоритм решения трёхдиагональных систем линейных уравнений, характерный для неявной схемы.
4. **Начально-краевые задачи** - свойство физической или инженерной задачи, состоящей из уравнения, начальных условий и граничных условий.

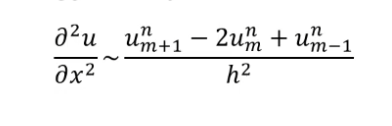
Стандартная постановка задачи:





Производную мы аппроксимируем самым простым образом,

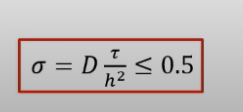
Отношение разности функций к превращению аргумента, то есть превращение функции к превращению аргумента.



Для второй производной по пространству мы записываем её как разность первых производных, и получаем выражение через значение функции, аппроксимацию на трёх точках.

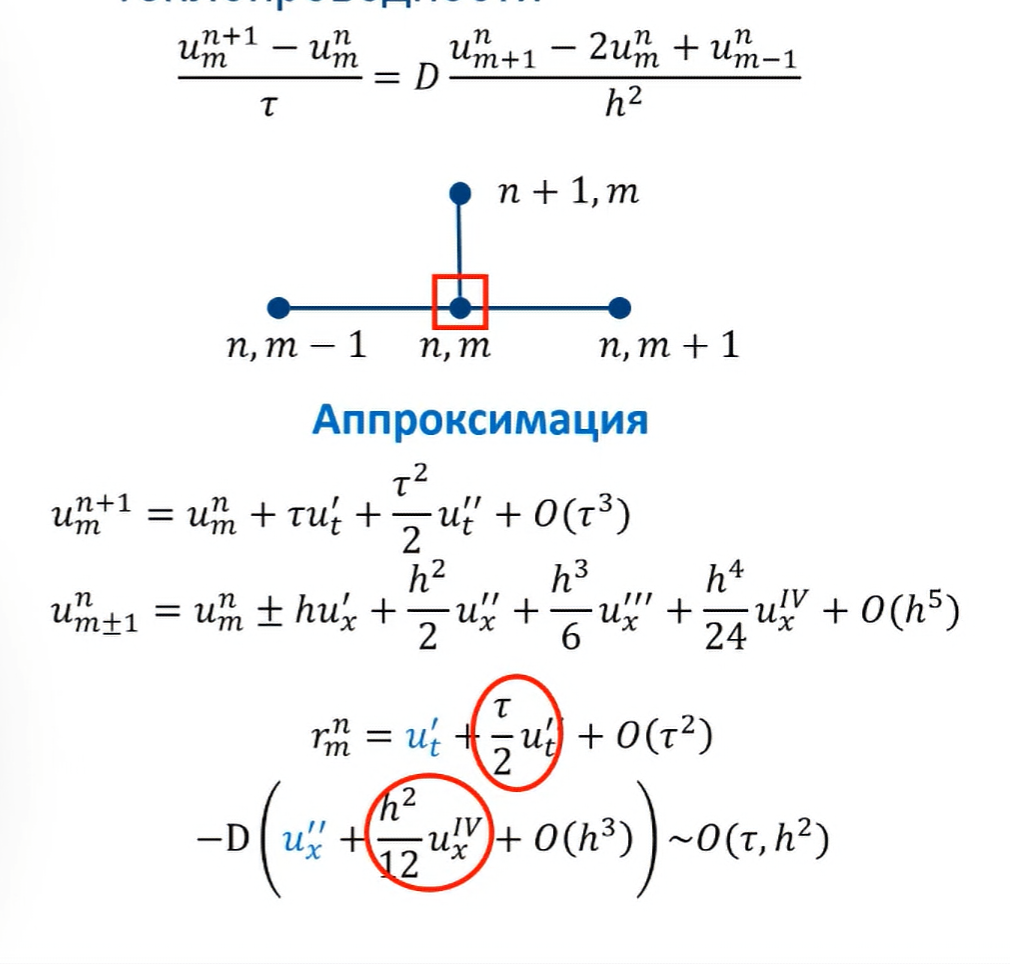
Задача принимает вид:

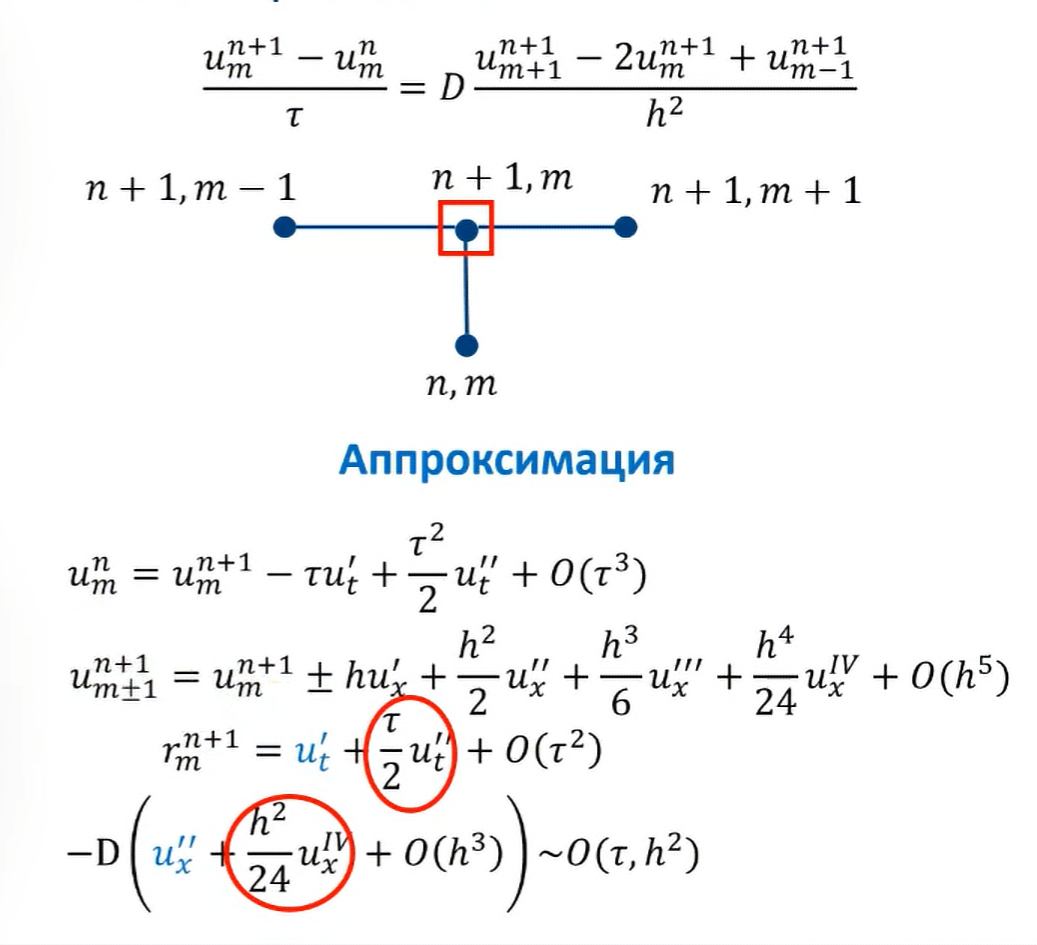


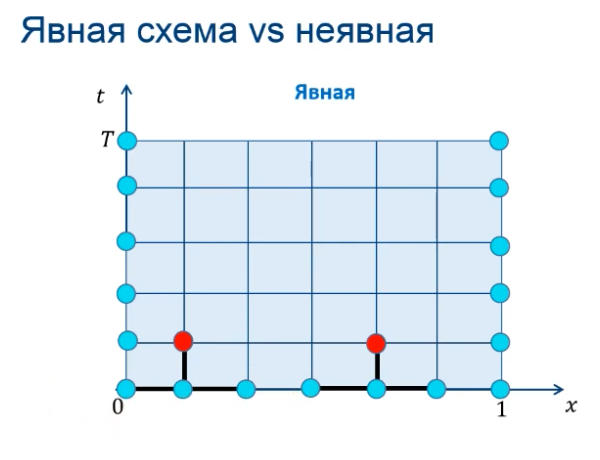
Для проверки на устойчивость явной схемы используем число Куранта:  


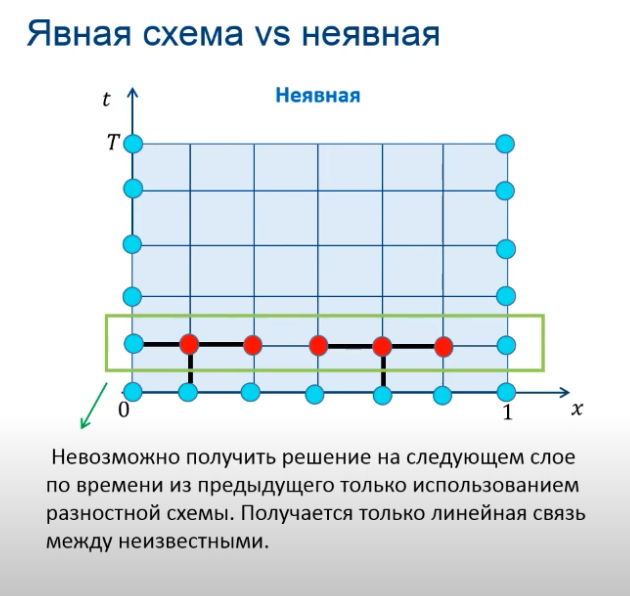
**ОСОБЕННОСТЬ НЕ ЯВНОЙ СХЕМЫ, ОНА БЕЗУСЛОВНО УСТОЙЧИВА**

**Явная разностная схема:**



**Неявная разностная схема:**  







  
**Прямой ход алгоритма прогонки:**  


Когда выразили Xm начинаем обратный ход прогонки. Через n мы находим n-1, через него n-2, и так далее.

