

Лекция(6.05.05)

8.3.3 Применение метода Бубнова-Галеркина.

$K \subset H$, H - гильбертово пространство.

$$\begin{cases} hu = f \\ lu = 0 \end{cases} \quad (5) \text{ краевая задача} \quad h - \text{линейный дифференциальный оператор}$$

K – его область определения, H – его область значений.

$hu=f$ уравнение в H , т.е. $f \in H$ l - линейный оператор, определяющий граничные условия.

U – точное решение (5), V – приближенное решение (5).

Подразумевается $K_n \subset K \subseteq H$, K_n – конечномерное пространство.

φ_i , $i=1,n$ базис для K_n $l\varphi_i = 0$, $i = 1, n$ Вывод: $\forall \varphi \in K_n$ верно $l\varphi = 0$ доказательство очевидно (сами).

Ищем V как элемент из K_n , то есть $V = \sum_{i=0}^n \alpha_i \varphi_i$ (7) используя условие

$$\left(\begin{array}{c} \text{если } V \text{ решает } (5) \\ \text{невязка основного уравнения на элементе } V \\ \text{и базис из } K_n \end{array}, \varphi_j \right) = 0, \quad j = 1, n \quad (8) \mid \text{запись принципа Бубнова-Галеркина}$$

$\forall f, g \in H$ определено (f, g) Если $H = L_2[a, b]$ $(f, g) = \int_a^b f(x)g(x)dx$

$$\left(L \left(\sum_{i=1}^n \alpha_i \varphi_i \right) - f, \varphi_j \right) = 0, \quad j = 1, n \quad \text{Использовали линейность } L$$

$$\left(\sum_{i=1}^n \alpha_i \cdot L(\varphi_i), \varphi_j \right) - (f, \varphi_j), \quad j = 1, n$$

$$\boxed{\sum_{i=1}^n \alpha_i \left(L(\varphi_i), \varphi_j \right) = (f, \varphi_j), \quad j = 1, n} \quad (9)$$

Так L и f заранее заданы и так как K_n мы сами придумали, то (9) представляет собой систему линейных уравнений, относительно неизвестных α_i

(9) в матричном виде:

$$\begin{pmatrix} (L\varphi_1, \varphi_1) & \wedge & (L\varphi_n, \varphi_1) \\ \vdots & \wedge & \vdots \\ (L\varphi_1, \varphi_n) &] & (L\varphi_n, \varphi_n) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{pmatrix} \quad (10) \quad \text{Если решение данной системы}$$

$(\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$ найдено, то (7) дает приближенное решение задачи (5) по методу Бубнова-Галеркина.

Замечание 1: так как ограничения на (5) отсутствуют мы ничего не можем сказать про матрицу системы (10) и заранее не можем гарантировать существование единственного решения этой системы. Эту часть нужно выяснить учитывая специфику задачи (5).

Замечание 2: Если нужно решить (5) с помощью метода Бубнова-Галеркина более точно, то нужно взять другое K_n , то есть либо увеличить размерность пространства, либо полностью сменить базисные функции и взять такой базис, который лучше соответствует исходной задаче.

Основная цель метода Бубнова-Галеркина.

1. Подробное обоснование метода с помощью Функционального Анализа (Марчук), там есть теорема о сходимости для некоторого частного случая задачи (5).

2. Если без функана:

$Lu=f$ если u -решение, то $Lu-f=0$. Если некоторый элемент гильбертова пространства нуль, то он ортогонален всем элементам этого пространства.

$\left[\varphi_i \mid i = 1, 2, \dots \right]$ - базис в H . Тогда для u верно $(Lu - f, \varphi_i) = 0 \quad i = 1, 2, \dots$ (11)

Это система бесконечной размерности, поэтому в БГ берут конечный базис и требуют ортогональности невязки конечного набора базисных функций. $(Lv - f, \varphi_i) = 0 \quad i = 1, n$

8.3.4 (пример)

$$\begin{cases} \omega'' + x\omega' + \omega = 2x \\ \omega(0) = 1 \quad \omega(1) = 0 \end{cases} \quad (2) \quad \omega(x) = u(x) + \eta(x), \quad \text{тогда } \begin{cases} u'' + xu' + u = \dots = 4x - 1 \\ u(0) = 0 \quad u(1) = 0 \end{cases} \quad (1)$$

где $\eta(x) = 1 - x$

$$Lu = u'' + xu' + u \quad (12) \quad L(\alpha u + \beta v) = \alpha L(u) + \beta L(v) \quad l_0 u = u(0) \quad l_1 u = u(1) \quad f = 4x - 1$$

Таким образом (1) можно записать в виде (5):

$$(5^*) \begin{cases} Lu = f \\ l_0 = 0 \\ l_1 = 0 \end{cases} \quad H: L_2[0,1] \quad \text{К-пусть об этом думает тот, кто ставит задачу}$$

K_n : нами забыта

$$\varphi_1(x) = x(x-1) \quad \varphi_2(x) = x^2(x-1) \quad \varphi_3(x) = x^3(x-1) \quad \varphi_i(x) = x^i(x-1), \quad i=1,n \quad \varphi_i(0)=0 \quad \varphi_i(1)=0$$

\Rightarrow ищем решение в виде:

Берем $n=3$

$$v = \alpha_1 \varphi_1 + \alpha_2 \varphi_2 + \alpha_3 \varphi_3 \quad (7^*)$$

Теперь можно составить матрицу и правую часть системы (10)

$$\varphi_i(x) = x^{i+1} - x^i \quad i = 1, n$$

$$(f, \varphi_i) = \int_0^1 (4x-1)(x^{i+1} - x^i) dx \quad i = 1, n \quad (13)$$

$$(L\varphi_i, \varphi_j) = \int_0^1 (L\varphi_i(x))\varphi_j(x) dx = (\text{пол.ст.} j+1) \quad L\varphi_i = \varphi_i'' + x\varphi_i' + \varphi_i - (\text{пол.ст.} i+1) =$$

$$= \int_0^1 (\text{пол.ст.} i+j+2) dx \quad (\text{можно взять аналитически})$$

Таким образом, система (10) принимает вид:

$$\begin{pmatrix} \int_0^1 (L\varphi_1)\varphi_1 dx & \int_0^1 (L\varphi_2)\varphi_1 dx & \int_0^1 (L\varphi_3)\varphi_1 dx \\ \int_0^1 (L\varphi_1)\varphi_2 dx & \int_0^1 (L\varphi_2)\varphi_2 dx & \int_0^1 (L\varphi_3)\varphi_2 dx \\ \int_0^1 (L\varphi_1)\varphi_3 dx & \int_0^1 (L\varphi_2)\varphi_3 dx & \int_0^1 (L\varphi_3)\varphi_3 dx \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \int_0^1 (4x-1)\varphi_1 dx \\ \int_0^1 (4x-1)\varphi_2 dx \\ \int_0^1 (4x-1)\varphi_3 dx \end{pmatrix} \quad (10^*)$$

Д/з - доделать задачу до конца (до разностной схемы)

$$v(x) = \alpha_1 x(x-1) + \alpha_2 x^2(x-1) + \alpha_3 x^3(x-1)$$

Таким образом, чтобы найти решение задачи в виде (7**) нужно вручную посчитать 12 интегралов и решить линейную систему уравнений размерности (3*3)

Разобрали пример, в котором вычисление интегралов и решение системы линейных уравнений не является самодельным (???) (это просто способ приближенного решения ДУ)

8.3.5 Развитие метода Бубнова-Галеркина до метода конечных элементов (проекционно-сеточные методы).

Рассмотрим (2) сведем ее к (1)

$$\begin{cases} u'' + xu' + u = 4x - 1 \\ u(0) = 0 \quad u(1) = 0 \end{cases} \quad (1) \quad H : L_2[0,1] \\ K : \text{сейчас не наше дело}$$

$$x \in [0,1]$$

базис – сетка: $x_i = ih$, $i=0,n$, $h=1/n$

$$(14) \quad \Phi_i(x) = \begin{cases} 0, \text{ если } x \notin [x_{i-1}, x_{i+1}] \\ 1 - \text{такая, если } x \in [x_{i-1}, x_{i+1}] \end{cases} \quad \Phi_i(0) = 0 \quad i = 1, n-1 \\ \Phi_i(1) = 0$$

$$\text{Дальше ищем: } v = \sum_{i=1}^{n-1} \alpha_i * \Phi_i(x) \quad \text{В качестве } K_{n-1} = \left\{ \sum_{i=1}^{n-1} \alpha_i \Phi_i \right\}$$

В этом состоит метод конечных элементов для решения задачи (1), для произвольной краевой задачи (5) метод конечных элементов применяется с базисными функциями с компактным носителем.

Для построения таких базисных функций вводится сетка. Базисная функция определяется как ненулевая на небольшом участке сетки и нулевая на оставшемся.

Для моделирования шапочки используется линейная функция или полином небольшой функции (Марчук, Вержицкий).

8.3.6 Другие методы решения краевых задач.

$$K_n \subset K \subset H$$

$$\begin{cases} Lu = f & (0) \\ lu = 0 & \end{cases} \quad K_n : \Phi_i, \quad i = 1, n, \text{ где } (6) l\Phi_i = 0, \quad i = 1, n$$

$$v = \sum_{i=1}^n \alpha_i \Phi_i \quad (7)$$

I. Метод коллокации

Вводят сетку для значений аргумента функции $v(x)$, $u(x)$. На сетке фиксируем n точек α_i , $i=1,n$ в формуле (7) подбирают так, чтобы невязка уравнения (5) на функции v в указанных точках обращалась в нуль. $L(\sum \alpha_i \Phi_i(x)) - f = 0$ (15) для точек x_j , $j=1,n$ (эти точки должны быть внутри области и не попадать на границы задачи). Д/з выписать этот критерий для задачи (1).

II Метод наименьших квадратов (МНК).

В функции (7) коэффициенты α_i выбирают так, чтобы норма невязки

$$\left\| L\left(\sum_{i=1}^n \alpha_i \Phi_i\right) - f \right\|_H \rightarrow \min \quad (16) \quad \text{Выписать этот критерий для задачи (1).}$$

III МНК на дискретной системе точек.

Вводят сетку для функций $v(x)$, $u(x)$. На сетке фиксируют m точек $(x_j, j=1,m)$ $m>n$

$$\sum_{j=1}^m \left(L\left(\sum_{i=1}^n \alpha_i \Phi_i(x_j)\right) - f(x_j) \right)^2 \rightarrow \min \quad (17) \quad \text{Также выписать критерий.}$$

Лекция (11.05.05)

Замечание 1: в методах коллокации, наименьших квадратов, МНК на дискретной сетке в качестве базисных функций можно использовать функции с компактным носителем.

Замечание 2: Когда рассматривали метод БГ, предполагали, что задача является линейной (с линейными однородными граничными условиями). Тот же метод можно применить для нелинейных диффуров, лишь бы граничные условия были линейными и однородными.

Если граничные условия не линейны, то не сможем искать условие в виде разложения по базису, если основное уравнение будет нелинейным, то это не страшно.

Если диф. не линейный, то принцип изменяется так, что просто системы (9) и (10) будут не линейными системами.

Интерес к методу БГ это просто часть современного метода конечных элементов.

9 Решение задачи Коши для жесткой системы ЛДУ (линейные диффуры).

Введение:

$$(1) \begin{cases} \frac{du}{dx} = Au \\ u(x_0) \end{cases} \quad \begin{matrix} A \in R^{m \times m} \\ u_0 \in R^m \quad u(x) \in R^m \\ x \in [x_0, x_0 + T] \end{matrix}$$

Зачем ее решать численно, если умеем решать аналитически? Оказывается на практике, что решить численно можно быстрее и более точно, чем аналитически (собственные числа могут быть комплексные, кратные и т.д.)

Такие задачи встречаются (химическая кинематика, электротехника, механика сплошной среды, ядерный реактор, теория управления).

9.1 Методы РК для решения задачи Коши системы диффуров порядка m .

Рассмотрим задачу более общую, чем задача (1).

$$\begin{cases} \frac{du}{dx} = f(x, u) \\ u(x_0) = u_0 \end{cases} \quad \begin{matrix} u(x) \in R^m \quad f(x, u) \in R^m \\ u_0 \in R^m \quad x \in [x_0, x_0 + T] \in R \end{matrix}$$

$$\begin{cases} v_0 = u_0 \\ x_{n+1} = x_n + h \\ v_{n+1} = v_n + h \cdot f(x_n, v_n) \end{cases} \quad \begin{matrix} v_n \in R^m, \quad v_{n+1} \in R^m \end{matrix}$$

РК 2 порядка (I) (3)

$$\begin{cases} v_0 = u_0 \\ x_{n+1} = x_n + h \\ v_{n+1} = v_n + hf\left(x_n + \frac{h}{2}, v_n + \frac{h}{2}f(x_n, v_n)\right) \end{cases}$$

$$\left. \begin{aligned} v_{n+1} &= v_n + h \sum_{i=1}^q p_i K_i(h) \\ \text{PK 2 порядка (II) (4) где } K_i(h) &= f\left(x_n + \alpha_i h, v_n + \sum_{j=1}^{i-1} \beta_{ij} K_j(h)\right) \\ \text{см 8.2.2} \end{aligned} \right\} (5)$$

Если речь идет о системе (1*), то $p_i, \alpha_i, \beta_{ij}$ – числа, а $K_i(h)$ – вектор из R^m

9.2 Методы РК для решения линейной системы (1).

Эйлер для (1):

$$\left. \begin{aligned} v_0 &= u_0 \\ x_{n+1} &= x_n + h \\ v_{n+1} &= v_n + hf(x_n, v_n) = v_n + hAv_n = (E + hA)v_n \end{aligned} \right\} (6)$$

$$\text{и } \left. \begin{aligned} v_0 &= u_0 \\ x_{n+1} &= x_n + h \\ v_{n+1} &= v_n + h \cdot A \left(v_n + \frac{h}{2} \cdot A v_n \right) = \left(E + hA + \frac{h^2 A^2}{2} \right) v_n \\ &\quad f\left(x_n + \frac{h}{2}, v_n + \frac{h}{2}f(x_n, v_n)\right) \end{aligned} \right\} (7)$$

Метод РК (второй) выписать самим.

9.3 Свойства решений задачи Коши для жесткой системы линейных уравнений.

$$(1***) \frac{du}{dx} = Au, u \in R^m, A \in R^{m \times m}$$

Определение: система (1***) называется жесткой, если $\max_{\operatorname{Re} \lambda_i < 0} |\operatorname{Re} \lambda_i| >> \min_{\lambda_i, i=1,m} |\operatorname{Re} \lambda_i|$ (9), где λ_i – собственные числа матрицы A.

Рассмотрим свойства жестких систем на примере системы второго порядка.

$$(1****) \begin{cases} \frac{du}{dx} = Au, & u \in R^2 \\ u(x_0) = u_0 \end{cases}$$

Пусть (λ_i, C_i) – собственное число и собственный вектор матрицы A ($i=1,2$) и пусть $\lambda_1 \ll \lambda_2 < 0$.

Тогда $u(x) \equiv 0$ является равновесием, оно устойчиво и узел.

Решение (1****) $u_0 = a_1 C_1 + a_2 C_2$, $a_{1,2}$ – коэффициенты разложения по собственным векторам.
 $u(x) = a_1 C_1 e^{\lambda_1(x-x_0)} + a_2 C_2 e^{\lambda_2(x-x_0)}$ (10)

Нетрудно видеть, что первое слагаемое формулы (10) стремиться к нулю быстрее, чем второе. Таким образом, решение состоит из быстро затухающей и медленно затухающей компоненты. Участок интегрирования, при котором $a_1 C_1 e^{\lambda_1(x-x_0)}$ дает заметный вклад в функцию (10)

называется пограничным слоем. После прохождения пограничного слоя вклад первого слагаемого становится маленьким и решение определяется вторым слагаемым: $u(x) \approx a_2 C_2 e^{\lambda_2(x-x_0)}$ (11).

Вывод: основное свойство жестких систем – наличие быстро затухающих компонент и медленная динамика решения после прохождения пограничного слоя. При этом, в том примере, что мы рассматривали: $\forall u_0 \in R^m, u_0 \neq 0$ верно $\lim_{x \rightarrow +\infty} u(x) = 0$ (12)

Сохраняется ли (12) у численных решений?

9.4 Вспомогательная задача.

$$(A) \begin{cases} \frac{du}{dx} = \lambda u & u \in R, u_0 \in R, x \in R, \\ u(x_0) = u_0 & \lambda < 0 - \text{число} \end{cases}$$

$$u(x) = u_0 e^{\lambda(x-x_0)} \quad \text{Tак как } \lambda < 0 \Rightarrow \lim_{x \rightarrow +\infty} u(x) = 0$$

Применим Эйлера.

$$v_0 = u_0, \quad x_{n+1} = x_n + h, \quad v_{n+1} = v_n + hf(x_n, v_n) = v_n + h\lambda v_n = (1 + h\lambda) v_n$$

$$\text{Если интегрировать с постоянным шагом, то: } v_{n+1} = (1 + h\lambda)^{n+1} v_0 \quad (13)$$

$$\text{Выясним, при каких условиях } \lim_{n \rightarrow +\infty} v_n = 0 \quad (14)$$

Утверждение: для выполнения (14) $\Leftrightarrow |1+h\lambda| < 1$: для того, чтобы при $\forall u_0$ было верно (14) необходимо и достаточно $|1+h\lambda| < 1$ (15).

Доказательство очевидно.

Для того, чтобы при \forall начальном условии задачи Коши в системе А численное решение, формируемое методом Эйлера стремилось к нулю необходимо и достаточно, чтобы шаг численного интегрирования удовлетворял условию (15).

$$\text{Таким образом, } |1+h\lambda| < 1 \Rightarrow -1 < 1 + h\lambda < 1 \Rightarrow -2 < h\lambda < 0 \Rightarrow h < 2/|\lambda|$$

$0 < h < \frac{2}{|\lambda|}$

(16)!!! - условие на шаг интегрирования.

Если в (A) решение быстро стремиться к нулю, то шаг интегрирования нужно брать очень маленьким.

К сожалению, ограничения на шаг в формуле (16) не является свойством используемого метода Эйлера. Такое ограничение на шаг проявляется при \forall рассмотренном нами методе РК.

9.5 Проблема решения жесткой системы линейных дифференциальных уравнений методами РК (на примере системы 2^{го}-порядка из #9.3).

$$(1^{***}) \begin{cases} \frac{du}{dx} = Au, & u \in R^2 \quad A \in R^{2 \times 2} \\ u(x_0) = u_0 \quad u_0 \in R^2 \\ x \in [x_0, x_0 + T] \end{cases} \quad \text{в предположении, что } \lambda_1 < < \lambda_2 < 0 \text{ и } u \equiv 0 - \text{устойчивый узел.}$$

Исследуем свойство численного решения, полученного методом Эйлера.

$$v_0 = u_0, \quad x_{n+1} = x_n + h \quad v_{n+1} = (E + hA)v_n \quad \text{см. (6)}$$

$$\text{Если } h = \text{const} \quad v_{n+1} = (E + hA)^{n+1} v_0$$

$$1 \text{ способ рассуждения: для того, чтобы } \lim_{n \rightarrow +\infty} v_n = 0 \Leftrightarrow p(E + hA) < 1 \quad (17)$$

Для того, чтобы численное решение сохраняло свойство точного, необходимо и достаточно, чтобы матрица (???) метода Эйлера
(???)

(17) позаимствовали из раздела о линейной алгебре.

Пусть λ_i, C_i – собственная пара А, тогда $(E + hA)C_i = C_i + h\lambda_i C_i = (1 + h\lambda_i)C_i$

Таким образом, $(1 + h\lambda_i)C_i$ – собственная пара $E + hA$.

Для того чтобы (17) было верно нужно $|1 + \lambda_1 h| < 1, |1 + \lambda_2 h| < 1$

$$0 < h < \frac{2}{|\lambda_1|} \quad 0 < h < \frac{2}{\lambda_2} \quad \text{Откуда } 0 < h < \frac{2}{|\lambda_1|}, \text{ так как } \lambda_1 \ll \lambda_2 < 0$$

Вывод: для того, чтобы численное решение исходной системы линейных уравнений с равновесием типа узел вело себя как аналитическое необходимо брать лишь маленький шаг интегрирования. Этот шаг определяется отрицательным собственным числом с наибольшим модулем действительной части.

Рассуждение 2:

$$\begin{aligned} \text{Эйлер: } & \begin{cases} v_0 = u_0 \\ u_{n+1} = x_n + h \\ v_{n+1} = (E + hA)^{n+1} v_0 \end{cases} \end{aligned}$$

Мысли:

$v_0 = a_1 C_1 + a_2 C_2$ Разложение начального условия по собственным векторам.

Тогда $v_{n+1} = a_1 (1 + \lambda_1 h)^{n+1} C_1 + a_2 (1 + \lambda_2 h)^{n+1} C_2$ - разложение очередной точки, полученной методом Эйлера по собственным векторам матрицы А. $|1 + h\lambda| < 1$ (15)

Если (15) будет нарушено для какого-либо λ_i , то решение, полученное методом Эйлера будут неограниченно возрастать – это основная проблема решения жестких систем. После прохождения пограничного слоя медленно меняющееся решение приходиться вычислять с невероятно маленьким шагом и любое его увеличение имеет катастрофические последствия (исследуемые траектории начинают бешено возрастать).

Лекция(12.04.05). #9.6 Постановка задачи Коши для системы диф. уравнений. Неявные методы РК.

Обычно записываются в виде:

$$(1^*) \left\{ \begin{array}{l} \frac{du}{dx} = f(x, u) \quad u(x) \in R^m \\ u(x_0) = u_0 \quad u_0 \in R^m, x \in R \\ \\ v_0 = u_0 \quad x_{n+1} = x_n + h \\ v_{n+1} = v_n + h \sum_{i=1}^q p_i K_i(h), \text{ где} \\ K_i(h) = f \left(x_n + \alpha_i h, v_n + h \sum_{j=1}^q \beta_{ij} K_j(h) \right), i = 1, q \end{array} \right\} \quad (5^*) \quad \text{Здесь } \alpha_i, p_i, \beta_{ij} \text{ – числа, } K_i(h) \in R^m$$

Методы РК (5) являются явными, так как $K_i(h)$ зависит от $K_1(h), \dots, K_{i-1}(h)$ и не зависит от K_i, K_{i+1}, \dots, K_q

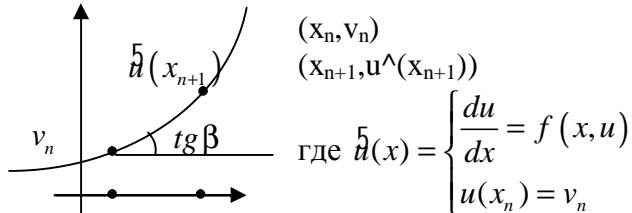
(5*) – неявные методы РК не дают возможности выразить K_i через предыдущие коэффициенты K . В этих формулах каждый из K_i может зависеть от всех остальных K_j

$$\begin{cases} v_0 = u_0 \\ x_{n+1} = x_n + h \\ v_{n+1} = v_n + hF(x_n, v_n, h, v_{n+1}) \end{cases} \quad (5**)$$

(5**) – нельзя решить относительно v_{n+1} явно.

На основе (5**) вводятся понятия порядка метода и локальной погрешности метода.

Определение: p - порядок, если $F(x_n, v_n, h, v_{n+1}) - tg\beta = O(h^p)$



Определение: $x_{n+1} e_{n+1} = \tilde{h}(x_{n+1}) - v_{n+1}$ – локальная погрешность

Утверждение: если метод порядка $p \Rightarrow$ порядок локальной погрешности – $p+1$

Пример неявных методов.

Неявный метод Эйлера:

$$\begin{cases} v_0 = u_0 & x_{n+1} = x_n + h \\ v_{n+1} = v_n + hK_1(h) & \text{Очевидно, } v_{n+1} = v_n \\ K_1(h) = f(x_n + h, v_n + hK_1(h)) \end{cases}$$

Метод 2 порядка

$$\text{PK: } \begin{cases} v_0 = u_0 & x_{n+1} = x_n + h \\ v_{n+1} = v_n + hK_2(h) & \\ K_1(h) = hf(x_n, v_n) & K_2(h) = f\left(x_n + \frac{h}{2}, v_n + \frac{h}{2}K_2(h)\right) \end{cases} \quad \begin{cases} v_0 = u_0 & x_{n+1} = x_n + h \\ v_{n+1} = v_n + \frac{h}{2}(K_1(h) + K_2(h)) & \\ K_1(h) = f(x_n, v_n) & \end{cases}$$

применение неявных методов РК к решению линейных систем ДУ. Эйлер 1-го порядка – неявный.

Неявный РК 2 порядка (I)

$$v_0 = u_0 \quad x_{n+1} = x_n + h \quad v_{n+1} = v_n + hK_2(h) \quad K_2(h) = A\left(v_n + \frac{h}{2}K_2(h)\right)$$

$$\left(E - \frac{h}{2}A\right) \cdot K_2(h) = Av_n \quad K_2(h) = \left(E - \frac{h}{2}A\right)^{-1} \cdot Av_n \quad (w)$$

$$v_{n+1} = v_n + \left(E - \frac{h}{2}A\right)^{-1} Ahv_n \quad (7)$$

$$\left. \begin{cases} \frac{du}{dx} = 5u \\ u(0) = 1 \end{cases} \quad v_{n+1} = v_n + \frac{5h}{\left(1 - \frac{5h}{2}\right)} v_n \end{cases} \right| \quad (7**)$$

При решении задачи Коши для линейной системы ДУ(1) на каждом шаге приходиться решать линейную систему уравнений для отыскания K_2 . В такой ситуации полезно найти обратную матрицу и организовать вычисление по формуле (w).

Специалист по решению линейных систем должен знать, когда нужно решать линейную систему, а когда обращать матрицу. Решение задачи Коши неявными методами – пример ситуации, когда нужно обращать матрицу.

#9.9 Применение неявных методов РК для решения задачи Коши для жестких систем линейных дифференциальных уравнений (на примере системы 2 порядка).

$$x \in [x_0, x_0 + T] \quad (1^{**}) \begin{cases} \frac{du}{dx} = Au \\ u(x) \in R^2 \quad u_0 \in R^2 \quad x \in R \quad A \in R^{2 \times 2} \\ u(x_0) = u_0 \end{cases}$$

для которой $u \equiv 0$ – устойчивый узел.

(λ_i, C_i) – собственная пара $A \quad \lambda_1 << \lambda_2 < 0$

Тогда $\forall u(x)$ – точное решение (1^{**}) ведет себя $\lim_{x \rightarrow +\infty} u(x) = 0$

Сохраняется ли такое свойство для численных траекторий, полученных неявным методом.
Используем неявный метод Эйлера.

$$v_0 = u_0 \quad x_{n+1} = x_n + h \quad v_{n+1} = (E - hA)^{-1} v_n \quad (18)$$

Пусть начальное условие из задачи Коши раскладывается по собственным векторам следующим образом: $v_0 = a_1 C_1 + a_2 C_2$ (19)

Если C_i – собственный вектор A , то $AC_i = \lambda_i C_i$ C_i – собственный вектор $E - hA$

$$(E - hA)C_i = (1 - h\lambda_i)C_i$$

$$(E - hA)^{-1} \cdot C_i = \frac{1}{1 - h\lambda_i} C_i \quad \text{то } 1 - h\lambda_i > 1 \text{ и вектор } C_i \text{ является собственным для } (E - hA)^{-1}$$

если $\neq 0$

$$v_1 = (E - hA)^{-1} v_0$$

$$v_1 = a_1 \frac{1}{(1 - h\lambda_1)} C_1 + a_2 \frac{1}{1 - h\lambda_2} C_2$$

$$v_{n+1} = a_1 \left(\frac{1}{1 - h\lambda_1} \right)^{n+1} C_1 + a_2 \left(\frac{1}{1 - h\lambda_2} \right)^{n+1} C_2 \quad (20)$$

при $n \rightarrow \infty$

$$\forall v_0 \in R^2, \lim_{n \rightarrow \infty} v_n = 0 \quad (21)$$

Таким образом применение неявного метода Эйлера к жесткой системе (задачи Коши (1^{***})) сохраняет свойство (21) независимо от шага интегрирования.

Д/з – исследовать неявные методы РК для той же самой жесткой системы уравнений (1^{ый} метод)
Для решения задачи Коши системы ДУ разработали много методов. Одним из классов являются неявные методы РК. Они позволяют вычислить медленно сходящуюся траекторию жесткой системы с достаточно большим шагом интегрирования.

Метод, который строится на основе полиномов Лежандра.

Слагаемых: q штук; порядок метода: $2q$.

Мы рассмотрели понятие жесткой системы д/у применительно к линейной системы с постоянной матрицей.

$$\begin{cases} \frac{du}{dx} = A(x) \cdot u + f(x) \\ u(x_0) = u_0 \\ u \in R^m \quad x \in [x_0, x_0 + T] \end{cases} \quad (*) - \text{линейная система с переменной матрицей.}$$

Система жесткая на участке $x \in [x', x'']$, если $\forall x \in [x', x''] \max_{\lambda_i(x) < 0} |\operatorname{Re} \lambda_i(x)| >> \min_{\lambda_i(x), i=i..m} |\operatorname{Re} \lambda_i(x)|$

Здесь $\lambda_i(x)$ – собственные числа $A(x)$

$$(*) \begin{cases} \frac{du}{dx} = f(x, u) \\ u(x_0) = u_0 \end{cases} \quad x \in [x', x'']$$

10 Численное интегрирование.

Квадратурные формулы интерполяционного типа (КВФ).

Дано $x_i, f_i \quad i=0, n$ (1) Можно построить интерполяционный полином.

$P_n(x) = a_0 + a_1 x + \dots + a_n x^n$, который $P_n(x_i) = f_i$ (2) для $\forall x_i, f_i$ такой полином существует и

единственен, в котором $x_i \neq x_j$ и $P_n(x) = \sum_{i=0}^n L_{ni}(x) \cdot f(x_i)$ (3)

$$L_{ni}(x) = -\frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (4)$$

(3), (4) можно записать через $\omega(x) = (x - x_0)(x - x_1) \dots (x - x_n)$ ($n+1$ ишук)

$$P_n(x) = \sum_{i=0}^n \frac{\omega(x)}{\omega'(x_i)(x - x_i)} \cdot f(x_i) \quad (5)$$

погрешность интерполяции в т. x $r_n(x) = f(x) - P_n(x)$. Очевидно, что $r_n(x_i) = f_i - P_n(x_i) = 0 \quad i=0, n$ (6)

$$\text{Теорема: } r_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega(x) \quad (7)$$

Пусть ищем $\int_a^b f(x) dx = I$ Заменим $f(x)$ на $P_n(x)$

$$\oint = \int_a^b P_n(x) dx = \int_a^b \left(\sum_{i=0}^n \frac{\omega(x)}{\omega'(x_i)(x - x_i)} \cdot f(x_i) \right) dx = \sum_{i=0}^n f_i \cdot \int_a^b \frac{\omega(x) dx}{\omega'(x_i)(x - x_i)} = \sum_{i=0}^n f_i C_i$$

число, которое зависит от расположения $x_0 \dots x_n$

$$\text{Функция } \oint = \sum_{i=0}^n a_i f_i, \text{ где } C_i = \int_a^b \frac{\omega(x) dx}{\omega'(x_i)(x - x_i)} \quad i = 0, n$$

КВФ – нужно для приближенного вычисления интеграла для $I = \int_a^b f(x) dx$

$$\psi \stackrel{\text{def}}{=} I - \oint = \int_a^b (f(x) - P_n(x)) dx = \int_a^b r_n(x) dx \quad (9)$$

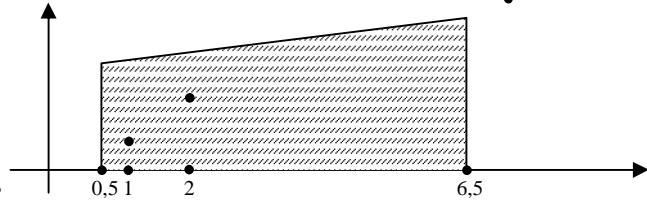
Лекция (21 мая).

Квадратурные формулы как универсальный способ приближенного отыскания интегралов.

Формулы из #10.1 позволяют приблизенно считать интегралы в самых нестандартных ситуациях.
Пример:

x_i	1	2	7
f_i	5	9	15

$$\int_{0,5}^{6,5} f(x) dx = ?$$



В этой ситуации мы можем построить полином 2 степени проходящий через 3 точки.

$$P_2(x) = \frac{(x-2)(x-7)}{(1-2)(1-7)} \cdot 5 + \frac{(x-1)(x-7)}{(2-1)(2-7)} \cdot 9 + \frac{(x-1)(x-2)}{(7-1)(7-2)} \cdot 15 \quad (*)$$

Построили интерполяционный полином второй степени, проходящий через точки в таблице.

$$I = \int_{0,5}^{6,5} f(x) dx \quad \hat{P} = \int_{0,5}^{6,5} P_2(x) dx \quad \text{Считаем, что } I = \hat{P}$$

Полином интерполирует функцию на $[1;6,5]$, экстраполирует на $[0,5;1]$

Для оценки погрешности численного интегрирования можно использовать следующие рассуждения: $\psi = I - \hat{P}$

Знаем: $f(x) = P_2(x) + r_2(x)$, где $r_2(x) = \frac{f'''(\xi)}{3!}(x-1)(x-2)(x-7)$ причем

$$\xi \in [\min(x, 1), \max(x, 7)]$$

$$\text{Поэтому } \psi = I - \hat{P} = \int_{0,5}^{6,5} r_2(x) dx = \int_{0,5}^{6,5} \frac{f'''(\xi)}{3!}(x-1)(x-2)(x-7) dx \quad (***)$$

На основе (***) можно построить более конкретную оценку погрешности.

Пример: $\int_0^1 f(x) \ln x dx$ $f(x)$ – похожа на полином $x_0 = 0, x_n = 1, x_i = ih, h = \frac{1}{n}$

$f(x) \rightarrow$ (строим для нее полином) $\rightarrow P_n(x)$

$$P_n(x_i) = f_i \quad i = 0, n \quad f(x) = P_n(x) + r_n(x)$$

$$\text{Тогда } I = \int_0^1 P_n(x) \ln x dx + \int_0^1 r_n(x) \ln x dx$$

$$\hat{P} = \int_0^1 P_n(x) \ln x dx = \int_0^1 \left(\sum_{i=0}^n \frac{\omega(x) f(x_i)}{\omega'(x_i)(x-x_i)} \right) \ln x dx = \sum_{i=0}^n \left(\int_0^1 \frac{\omega(x) \ln x dx}{\omega'(x_i)(x-x_i)} \right) f(x_i)$$

число, которое не зависит от $f(x)$

Таким образом	$\hat{P} = \sum_{i=0}^n f(x_i) C_i, \quad C_i - \text{числа}$
---------------	---

$$(\ast\ast\ast)$$

Вычисление приближенного интеграла сводиться к подсчету линейной комбинации.

(***) – квадратурная формула интерполяционного типа вычисления интеграла с весовой функцией $\ln x$. Чтобы оценить погрешность численного интегрирования нужно оценить ψ .

Вывод: чем отличается пример 1 от 2 – мы не хотим заменять подынтегральную функцию на полином, так как знаем, что вся подынтегральная функция не похожа на полином, мы заменяем ту часть подынтегральной функции, которая похожа на полином.

Квадратурная формула интерполяционного типа для вычисления интеграла с весовой функцией $p(x)$.

$$I = \int_a^b f(x) p(x) dx, \text{ где } p(x) \leq 0 \text{ или } p(x) \geq 0 \text{ при } x \in [a, b]$$

Заменим $f(x)$ на $P_n(x)$, что $P_n(x_i) = f_i$ и построим формулу типа:

$$\hat{P} = \sum_{i=0}^n f_i C_i \quad \text{где } C_i = \int_a^b \frac{u(x) \beta(x) dx}{w'(x_i)(x - x_i)} \quad \text{И приведем все это так, как в заголовке.}$$

Пример: $I = \int_{-1}^1 \frac{f(x) dx}{\sqrt{1-x^2}}$

10.3 Квадратурная формула Ньютона – Котоса.

Эти формулы являются частным случаем квадратурных формул общего вида (#10.1), для которых весовая функция $\rho(x) \equiv 1$, узлы интерполяции расположены на равномерной сетке и граничные узлы интерполяции совпадают с пределами интегрирования.

$$I = \int_a^b f(x) dx \quad h = \frac{b-a}{n}, \quad x_0 = a, \quad x_n = b, \quad x_i = a + ih, \quad i = \overline{0, n}$$

КВФ НК строится следующим образом: для $f(x)$ строим $P_n(x)$, который:

$$P_n(x_i) = f_i, \quad i = \overline{0, n} \quad I \stackrel{\text{def}}{=} \int_a^b f(x) dx \approx \hat{P} = \int_a^b P_n(x) dx = \sum_{i=0}^n c_i f_i, \quad \text{где } c_i = \int_a^b \frac{\omega(x) dx}{\omega'(x_i)(x - x_i)}, \quad i = \overline{0, n}$$

погрешность интерполяции:

$$\psi = I - \hat{P} = \int_a^b r_n(x) dx \stackrel{\text{no Th}}{=} (\text{o формуле представления погрешности интерполяции}) = \\ = \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \omega(x) dx = \begin{cases} \frac{f^{(n+1)}(\xi)}{(n+1)!} \int_a^b \omega(x) dx, & \text{если } n - \text{нечетно} \\ \frac{f^{(n+2)}(\xi)}{(n+2)!} \int_a^b \omega(x) dx, & \text{если } n - \text{четно} \end{cases} \quad (10)$$

таким образом для квадратурных формул НК есть формула представления погрешности (10). Нужно научиться строить и использовать квадратурные формулы НК. При построении формул НК удобно использовать отрезок $[0, nh]$, то есть $[a, b] = [a, a + (b-a)/n^*h]$. В качестве a используют 0, b заменяют на h . \Rightarrow для построения интерполяционного полинома используют сетку:

$$x_0 = 0, \quad x_n = nh, \quad x_i = ih, \quad i = \overline{0, n}$$

$$I = \int_0^{nh} f(x) dx = \sum_{i=0}^n c_i f_i + \psi = Ah \sum_{i=0}^n B_i f_i + \psi \quad \text{смотрите справочную распечатку.}$$

Определение:

Порядком квадратурной формулы НК называют число n -степень полинома, интерполирующего подынтегральную функцию.

Важной характеристикой квадратурной формулы НК является порядок погрешности (k) и точность (l).

	n	k	l
Трапец	1	3	1
Симпс	2	5	3
НК	3	5	3

Определение:

Формула имеет точность l , если для всех подынтегральных функций являющихся полиномами степени l , формула абсолютно точна, и среди полиномов степени $l+1$, найдется такой, для которого формула дает ненулевую погрешность.

10.4 Формула Симпсона.

пример: $I = \int_{-1}^1 f(x)dx$ $x_0 = -1, x_1 = 0, x_2 = 1, h = 1$ $f = \{f_0, f_1, f_2\}$

$$P_2(x) = \frac{(x-0)(x-1)}{(-1-0)(-1-1)} f_0 + \frac{(x+1)(x-1)}{(+1)(-1)} f_1 + \frac{(x+1)(x-0)}{(2)(1)} f_2 \quad \hat{I} = \sum_{i=0}^2 c_i f_i, \text{ где } c_i = \int_{-1}^1 (\dots) dx$$

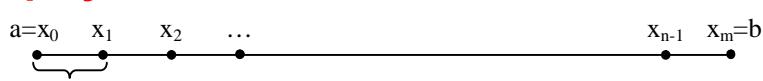
Построим оценку для погрешности формулы Симпсона: ($n=2$ – четно)

$$\Psi = \frac{f^{(n+2)}}{(n+2)!} \int_a^b \omega(x) dx \quad \Psi = \frac{f^{IV}(\xi)}{4!} \int_{-1}^1 x(x-1)(x+1)x dx = \frac{f^{IV}(\xi)}{4!} \int_{-1}^1 (x^2 - 1)x^2 dx =$$

$$= \frac{f^{IV}(\xi)}{4!} \left(\frac{x^5}{5} - \frac{x^3}{3} \right) \Big|_{-1}^1 = \frac{f^{IV}(\xi)}{24} \left(\frac{1}{5} - \frac{1}{3} \right) 2 = -\frac{f^{IV}(\xi)}{90}$$

10.5 Составные формулы Ньютона-Котоса. Лекция 44.

CHK (n , m)
НК число уч-в



$$I = \int_a^b f(x)dx \quad \hat{I} = \frac{b-a}{m} \text{-длина участков применения формулы Ньютона-Котоса.}$$

$$x_i = a + i \left(\frac{b-a}{m} \right), i = \overline{0, m}$$

$$h = \frac{\hat{I}}{n} \text{-расстояние между узлами на которых}$$

применяется формула НК порядка n (шаг НК(n))

$$x_i + \frac{j}{h} = x_i + jh = x_i + \frac{j}{n} \hat{I} = a + \left(i + \frac{j}{n} \right) \hat{I} \quad I \approx \hat{I} \leftarrow \text{CHK}(n, m) \quad I = \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)dx$$

$$\hat{I} = \sum_{i=0}^{m-1} \sum_{j=0}^n C_j f_i + \frac{j}{n} = (\text{Если использовать формулы из распечаток, то получим}) = \sum_{i=0}^{m-1} \sum_{j=0}^n AhB_j f_i + \frac{j}{n}$$

$$A, B_j \text{ - из распечатки } h = \frac{b-a}{n \cdot m}$$

$$\Psi \equiv I - \hat{I} = \sum_{i=0}^{m-1} \underbrace{\left(\int_{x_i}^{x_{i+1}} f(x)dx - \sum_{j=0}^n AhB_j f_i + \frac{j}{n} \right)}_{\text{это погрешность НК на } [x_i, x_{i+1}]} \quad$$

Утверждение:

Погрешность составной функции НК (n, m) является суммой погрешностей на участках $[x_i, x_{i+1}]$ при помощи НК(n)

#10.6 Анализ составной формулы Симпсона.

Формула Симпсона из распечатки: $\int_0^{2h} f(x)dx = \frac{h}{3} (f(0) + 4f(0+h) + f(2h)) - \frac{h^5}{90} f^{IV}(\xi)$

$$\text{Для } [x_i, x_{i+1}] \Rightarrow 2h = x_{i+1} - x_i \Rightarrow h = \frac{x_{i+1} - x_i}{2} \quad \xi \in [0, 2h]$$

$$\boxed{\int_{x_i}^{x_{i+1}} f(x)dx = \int_0^{2h} f(x_i + S)dS = (*) = \frac{h}{3} (f(x_i) + 4f(x_i + h) + f(x_i + 2h)) - \frac{h^5}{90} f^{IV}(\xi), \xi \in [x_i, x_{i+1}]}$$

CHK ($n=2, m$):

$$(**) \int_a^b f(x)dx = \sum_{i=0}^{m-1} \int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{i=0}^{m-1} \frac{h}{3} \left(f_i + f_{\frac{i+1}{2}} + f_{i+1} \right), \text{ где } f_i = f(x_i), f_{\frac{i+1}{2}} = f\left(x_{\frac{i+1}{2}}\right), f_{i+1} = f(x_{i+1})$$

Построим погрешность формулы (**):

$$\psi \equiv I - P = \sum_{i=0}^{m-1} \left(-\frac{h^5}{90} f^{IV}(\xi_i) \right), \xi_i \in [x_i, x_{i+1}] \quad h = \frac{b-a}{m \cdot n} \Big|_{n=2} = \frac{b-a}{2m}$$

Теорема: пусть $|f^{IV}(x)| \leq M$ при $x \in [a, b]$ тогда формулы Симпсона с точностью $\varepsilon > 0$, необходимо

$$\text{использовать } m \text{ такое, что } m \geq \sqrt[4]{\frac{(b-a)^5 M}{2880\varepsilon}}$$

Доказательство:

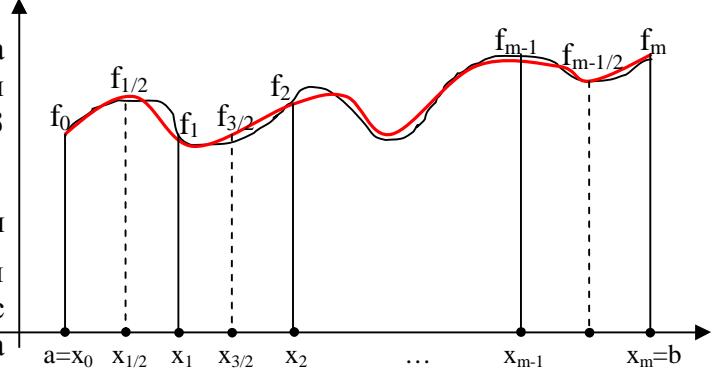
$$\max_{x \in [a, b]} |f^{IV}(x)| = M \Rightarrow |\psi| \leq \frac{h^5}{90} M \cdot m \text{ (очевидно)} = \left(\frac{b-a}{2m} \right)^5 \frac{1}{90} M \cdot m = \frac{(b-a)^5 M}{32 \cdot 90} \frac{1}{m^4} = \frac{(b-a)^5 M}{2880 m^4} \xrightarrow{m \rightarrow \infty} 0$$

путь

$$\Rightarrow \text{чтобы } |\psi| < \varepsilon, \text{ достаточно взять } \frac{(b-a)^5 M}{2880 m^4} < \varepsilon. \text{ Если брать } m^4 \geq \frac{(b-a)^5 M}{2880 \varepsilon} \Rightarrow |\psi| \leq \varepsilon.$$

Выводы:

- 1) Рисунок для составной формулы Симпсона:
Смысл СНК(2,m), в том, что интервал на каждом из участков $[x_i, x_{i+1}]$ заменяется участками наравне, проходящими через 3 точки: $f_i, f_{i+1/2}, f_{i+1}$
- 2) Если $|f^{IV}(x)| \leq M$ при $x \in [a, b]$, то интеграл
И можно посчитать с помощью формулы Симпсона СНК(2,m) можно посчитать с любой точностью, за счет выбора достаточно большого количества участков.
- 3) Если 4-я производная велика, то придется брать много участков, для обеспечения нужной точности.
- 4) Если количество участков увеличить вдвое, то погрешность уменьшиться в $2^4 = 16$ раз.
формула Симпсона – это очень хорошая формула:
 - при работе с методом РК (4-го порядка) для решения задачи Коши для дифференциального уравнения, при уменьшении шага в 2, погрешность падает в 16.
 - разностная схема лабы №2 – при уменьшении шага в 2 раза – погрешность падает в 4 раза.



#10.7 Заключительные замечания.

- 1) Для того, чтобы вычислить интегралы с максимально возможной точностью, разрабатываются 2 подхода:
 - a. на участке $[a, b]$ применяется формула наивысшего порядка точности, то есть по заданному количеству узлов, строиться квадратурная формула, являющаяся абсолютно точной для полиномов максимально возможной степени. Формулы НК(n) являются абсолютно точными для полиномов степени n (или выше).

- b. Есть теоремы о том, что используя $n+1$ узел, можно построить квадратурную формулу, являющуюся абсолютно точной для полиномов степени $2n+1$. Такие формулы называются квадратурами Гаусса. Они строятся по схеме (10.1) и используют специальную схему построения узлов интерполяции, то есть узлы x_0, \dots, x_n выбираются корнями полинома $P_n(x)$, который взят из системы полиномов, ортогональных на участке $[a,b]$. (примеры таких формул в справочнике) Этот подход развивался в докомпьютерную эпоху, когда люди хотели сэкономить на количестве вычислений.
- 2) Использование составных формул на участках $[x_i, x_{i+1}]$, которые имеют разную длину. Если $f(x)$ на участке $[x_i, x_{i+1}]$ имеет большую 4 производную, разбиение должно быть мелким и наоборот. Такое неравномерное разбиение участка $[a,b]$ может строиться по ходу работы метода. (Мэтьюз, Финк(эль) «4M в MatLab») (адаптивная квадратура)

#11 Численное решение задачи Коши для нестандартного уравнения теплопроводности.

Основная задача #: выяснить какие сложности таит в себе задача исследования задач Математической Физики.

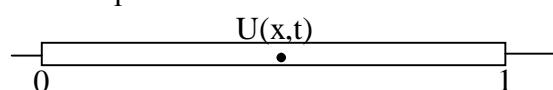
Исследуем разностную схему, понятие вычислительной устойчивости, исследуем погрешность аппроксимации и попробуем доказать сходимость.

$$(1) \begin{cases} \text{нач.условия} \\ \text{краевые} \\ \text{условия} \end{cases} \left\{ \begin{array}{l} U_t = U_{xx} + f(x,t), x \in [0,1], t \in [0,T] \\ U(x,0) = \phi(x) \\ \alpha_1 U(0,t) + \alpha_2 U'_x(0,t) = \mu_1(t), |\alpha_1| + |\alpha_2| \neq 0 \\ \beta_1 U(1,t) + \beta_2 U'_x(1,t) = \mu_2(t), |\beta_1| + |\beta_2| \neq 0 \end{array} \right. \quad U(x,t) = ?$$

(1) – задача Коши для уравнения в частных производных, по смыслу задачи $t \in [0, \infty]$ и ограничение $t \in [0, T]$ вводится чтобы удобнее было строить численное решение.

Смысл задачи (1):

$U(x,t)$ – температура в точке стержня с координатой x в момент времени t .



$\phi(x)$ – распределение температуры в начальный момент времени.

функции μ_1, μ_2 и коэффициенты α_i, β_i , $i=1,2$ определяют тепловой режим на концах стержня в любой момент времени.

$f(x,t)$ – функция плотности источников или стоков тепла на стержне.

Уравнение и система уравнений типа (1) используется в задачах химической кинематики, моделях биологических популяций и при моделировании движения фронта при боевых действиях.

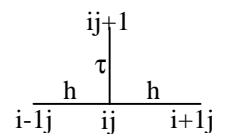
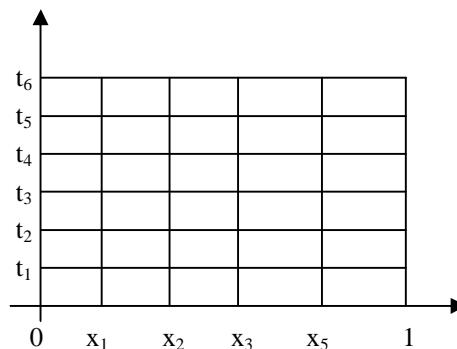
Сетка и разностная схема:

$$h = \frac{1}{n}, x_i = ih, i = \overline{0, n} \quad x_0 = 0 \quad x_n = 1 \quad \tau = \frac{T}{m} \quad t_j = j\tau, j = \overline{0, m}$$

$U_{ij} = U(x_i, t_j)$ - точное решение задачи (1) в узлах сетки

V_{ij} - точное решение разностной схемы в узлах сетки.

$$(2) \begin{cases} \frac{V_{ij+1} - V_{ij}}{\tau} - \frac{V_{i+1j} - 2V_{ij} + V_{i-1j}}{h^2} = f(x_i, t_j) \\ V_{i0} = \varphi(x_i) \\ \alpha_1 V_{0j} + \alpha_2 \frac{V_{1j} - V_{0j}}{h} = \mu_1(t_j) \\ \beta_1 V_{nj} + \beta_2 \frac{V_{nj} - V_{n-1j}}{h} = \mu_2(t_j) \end{cases}$$



11.1 Постановка задачи и запись разностной схемы. Лекция №45

Сетка: $h = \frac{1}{n}$, $x_i = ih$, $i = \overline{0, n}$ $x_0 = 0$ $x_n = 1$ $\tau = \frac{T}{m}$ $t_j = j\tau$, $j = \overline{0, m}$

$$(2-1) \frac{V_{ij+1} - V_{ij}}{\tau} - \frac{V_{i+1j} - 2V_{ij} + V_{i-1j}}{h^2} = f(x_i, t_j), i = \overline{1, n-1}, j = \overline{0, m}$$

$$(2-2) V_{i0} = \varphi(x_i)$$

$$(2-3) \alpha_1 V_{0j} + \alpha_2 \frac{V_{1j} - V_{0j}}{h} = \mu_1(t_j) \quad j = \overline{1, m}$$

$$(2-4) \beta_1 V_{nj} + \beta_2 \frac{V_{nj} - V_{n-1j}}{h} = \mu_2(t_j)$$

Значения сеточной функции U_{ij} и V_{ij} соответствующие одному компоненту времени $t=t_j$ называются одним слоем.

Рассмотрим задачу (1) и разностную схему (2).

V_{31} - приближенное значение температуры стержня при $x=x_3$ в момент времени t_1 .

\otimes - эти узлы соответствуют границе стержня, значения сеточной функции V в этих узлах определяют температуру на границах стержня.

$\otimes \times \otimes$ - узлы участвующие в аппроксимации граничных условий.

$$\frac{h}{i-1j} \frac{h}{ij} \frac{h}{i+1j}$$

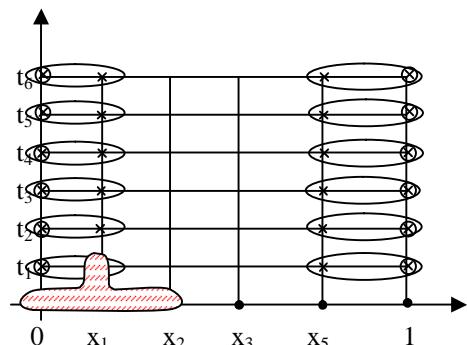
Это шаблон, аппроксимирующий основное уравнение. Его можно прикладывать к узлам: $i = \overline{i, n-1}$ $j = \overline{0, m-1}$

Способ решения разностной схемы.

находим $V_{i0} = \varphi_i$, $i = \overline{0, n}$, затем (2*-2) схема решается послойно.

для слоя $j=(0, m-1)$ вычисляем:

$$V_{ij+1} = \left(\frac{V_{i+1j} - 2V_{ij} + V_{i-1j}}{h^2} + f_{ij} \right) \tau + V_{ij}, i = \overline{1, n-1} \text{ (2*-1) затем (2-3) и (2-4):}$$



$$V_{0j} = \frac{\left(\mu_1(t_j) - \frac{\alpha_2}{n} V_{1j} \right)}{\alpha_1 - \frac{\alpha_2}{h}} \quad (2*-3) \quad V_{nj} = \frac{\left(\mu_2(t_j) + \frac{\beta_2}{h} V_{n-1j} \right)}{\beta_1 + \frac{\beta_2}{n}} \quad (2*-4)$$

Переходим к следующему условию.

11.2 Модельные задачи.

$$(1^*) \begin{cases} U_t = U_{xx} + f(x, t) \\ U(x, 0) = \varphi(x) \\ U(0, t) = \mu_1(t) \quad U(1, t) = \mu_2(t) \\ x \in [0, 1], t \in [0, T] \end{cases} \text{ Сетка: } h = \frac{1}{n}, x_i = ih, i = \overline{0, n} \quad x_0 = 0 \quad x_n = 1 \quad \tau = \frac{T}{m} \quad t_j = j\tau, j = \overline{0, m}$$

$$(3) \begin{cases} (V_t)_{ij} - \left(V_{xx} \right)_{ij} = f_{ij}, i = \overline{1, n-1}, j = \overline{0, m-1} \\ V_{i0} = \varphi_i, i = \overline{0, n-1} \\ V_{0j} = \mu_{1j}, V_{nj} = \mu_{2j}, j = \overline{1, m} \end{cases}$$

В следующих пунктах будем изучать разностную схему (3) для задачи (1*)

11.3 Теоретические конструкции, необходимые для определения понятия вычислительной устойчивости разностной схемы.

1. Пространство на слое: R_{n+1}^{n-1}

$V = (0, V_1, V_2, \dots, V_{n-1}, 0) \in R_{n+1}^{n-1}$ пространство состоит из векторов с $n+1$ компонентами, но 1 и последняя – нулевые. Такой вектор соответствует значениям сеточной функции, удовлетворяющей граничным условиям на некотором фиксированном слое.

2. 1-я разностная краевая задача на собственные функции и собственные значения:

$$\begin{cases} x \in [0, 1], x_i = ih, h = \frac{1}{n}, i = \overline{0, n} \\ -\frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} = \lambda V_i, i = \overline{1, n-1} \quad (4) \text{ или кратко} \\ V_0 = 0, V_n = 0 \end{cases}$$

$$\left\{ \begin{array}{l} -\left(V_{xx} \right)_i = \lambda V_i \\ V_0 = V_n = 0 \end{array} \right. \quad (4^*)$$

Очевидно, что система (4) – это линейная система уравнений для вектора из пространства на слое. Есть теоремы:

Теорема 1:

$$V^{(k)} = \sin(\pi kx) \in R_{n+1}^{n-1} \Rightarrow V_i^{(k)} = \sin(\pi kx_i), i = \overline{0, n}$$

$$U^{(k)} = \frac{4}{h^2} \sin^2 \left(\frac{\pi k}{2n} \right), k = \overline{1, n-1}$$

\Rightarrow задача (4) имеет $n-1$ собственное число, каждому из которых соответствует собственный вектор.

Теорема 2:

Решение задачи (4) в пространстве на слое образует ортогональный базис.

$$\left(V^{(l)}, V^{(k)} \right) \stackrel{\text{def}}{=} \sum_{i=0}^n V_i^{(l)} V_i^{(k)} {}^T h = 0, \quad k, l = \overline{1, n-1}, \quad k \neq l$$

Следствие:

$$V \in R_{n+1}^{n-1} \exists! \text{ представление вида } V = \sum_{k=1}^{n-1} a_k V^{(k)}, \text{ m.e. } V_i = \sum_{k=1}^{n-1} a_k \sin(\pi k x_i)$$

Задача (4) является разностным аналогом дифференциальной задачи Штурма – Лиувилля на собственные числа и собственные функции.

$$\begin{cases} -U''(x) = \lambda U \\ U(0) = U(1) = 0 \\ x \in [0, 1] \end{cases}$$

Теорема:

$$\text{Решение (5): } \lambda^{(k)} = (\pi k)^2 \quad U^{(k)}(x) = \sin(\pi k x), \quad k = 1, 2, 3, \dots$$

У (4) задачи конечное число собственных чисел, а у (5) – бесконечное.

3. Выражение V_{xx} как оператор в пространстве на слое R_{n+1}^{n-1} .

Пусть $V \in R_{n+1}^{n-1}$. Определим ω следующим образом:

$$(6) \begin{cases} \omega_0 = 0, \quad \omega_n = 0 \\ \omega_i \stackrel{\text{def}}{=} -\left(V_{xx}\right)_i = -\frac{V_{i+1} - 2V_i + V_{i-1}}{h^2}, \quad \forall i = \overline{1, n-1} \end{cases}$$

Таким образом, вектору $V \in R_{n+1}^{n-1}$ ставим в соответствие вектор $\omega \in R_{n+1}^{n-1}$ и определяются формулами (6).

Утверждение:

$$\text{Пусть } V \in R_{n+1}^{n-1} \text{ и } V = \sum_{k=1}^{n-1} a_k V^{(k)} \Rightarrow \omega = \sum_{k=1}^{n-1} \left(a_k \lambda^{(k)} \right) V^{(k)} \quad (8)$$

$$\omega_i = \frac{V_{i+1} - 2V_i + V_{i-1}}{h^2} = -\frac{1}{n^2} \left(\sum_{k=1}^{n-1} a_k V_{i-1}^{(k)} - 2 \sum_{k=1}^{n-1} a_k V_i^{(k)} + \sum_{k=1}^{n-1} a_k V_{i+1}^{(k)} \right)$$

Доказательство:

$$-\sum_{k=1}^{n-1} a_k \left(\frac{V_{i+1}^{(k)} - 2V_i^{(k)} + V_{i-1}^{(k)}}{h^2} \right) = \sum_{k=1}^{n-1} a_k \lambda^{(k)} V_i^{(k)} \quad (7) \quad \forall i = \overline{1, n-1}$$

Для компонент $\omega_0 = \omega_n = 0 \Rightarrow$ утверждение доказано.

4. Первая вспомогательная задача. Задача об остывании стержня, начальное распределение температур которого соответствует к –й гармонике. (с 0-й температурой на концах).

$$\begin{cases} U_t - U_{xx} = 0 \\ U(0, t) = U(1, t) = 0 \\ U(x, 0) = \sin(\pi k x), \quad x \in [0, 1], \quad t \in [0, T] \end{cases}$$

точным решением является функция $U(x, t) = e^{-(\pi k)^2 t} \sin(\pi k x)$

Свойства решения:

- С течением времени температура стержня выравнивается к нулю:

$$\forall x \in [0, 1] \lim_{t \rightarrow \infty} U(x, t) = 0 \quad \lim_{t \rightarrow \infty} \left(\max_{x \in [0, 1]} |U(x, t)| \right) = 0$$

стремление к нулю является монотонным.

План: мы построим разностную схему для задачи (1**) и изучим поведение решения разностной схемы. Если оно ведет себя так, как аналитическое (U), то разностную схему будем называть вычислительно устойчивой. Если поведение решения разностной схемы существенно отличается от поведения дифференциальной задачи, разностную схему назовем аналитически неустойчивой.

11.4 Вычислительная устойчивость разностной схемы.

Построим разностную схему для решения первой вспомогательной задачи (1**):

$$\begin{cases} \frac{V_{ij+1} - V_{ij}}{\tau} - \frac{V_{i+1j} - 2V_{ij} + V_{i-1j}}{h^2} = 0, i = \overline{1, n-1}, j = \overline{0, m} \\ V_{0j} = V_{nj} = 0, j = \overline{1, m} \\ V_{i0} = \sin(\pi kx_i), i = \overline{0, n} \end{cases}$$

Решение: $\begin{cases} V_{ij+1} = V_{ij} + \tau (V_{xx})_{ij}, i = \overline{1, n-1} \\ V_{0j+1} = 0, V_{nj+1} = 0 \end{cases}$ (10) - показывает как вычислять решение на слое $j+1$

Обозначения: $\overline{V}_0 = \{V_{i0}\}, i = \overline{0, n}$ $\overline{V}_0 \in R^{n-1}_{n+1}$; $\overline{V}_j = \{V_{ij}\}, i = \overline{0, n}$ $\overline{V}_j \in R^{n-1}_{n+1}$

Результат применения оператора $(-V_{xx})$ к вектору \overline{V}_j обозначим $\overline{\omega}_j$

$$(11) \begin{cases} \overline{V}_{j+1} = \overline{V}_j - \tau \overline{\omega}_j \\ \overline{V}_0 = V^{(k)} = \sin(\pi kx) \end{cases}$$

$$\overline{\omega}_0 = -\lambda^{(k)} \overline{V}_0 \Rightarrow \overline{V}_1 = \overline{V}_0 + \lambda^{(k)} \overline{V}_0 = (1 + \lambda^{(k)}) \overline{V}_0$$

$$\overline{V}_2 = \overline{V}_1 + \lambda^{(k)} \tau \overline{V}_1 \Rightarrow \overline{V}_{j+1} = (1 + \lambda^{(k)} \tau) \overline{V}_j \quad (12)$$

Для того, чтобы решение разностной схемы монотонно убывало при $j \rightarrow \infty$, необходимо и достаточно, чтобы $\forall k = \overline{1, n-1} \quad |1 + \lambda^{(k)} \tau| < 1$

Определение:

Разностная схема (9) для решения задачи (1**) называется вычислительно устойчивой, на сетки размерности (n, m) , если выполняется условие: $\forall k = \overline{1, n-1} \quad |1 + \lambda^{(k)} \tau| < 1$ (13).

Лекция №46.

Определение вычислительной устойчивости основано на следующем утверждении.

Утверждение:

Решение разностной схемы (9) для задачи (1**) монотонно стремиться к 0 при $j \rightarrow +\infty$ смысле $\|\overline{V}_{j+1}\| \leq \|\overline{V}_j\| \quad \lim_{j \rightarrow \infty} \|\overline{V}_j\| = 0$ тогда и только тогда, когда $|1 + \lambda^{(k)} \tau| < 1$, где $k \in \{1, n-1\}$

При доказательстве этого утверждения используется свойство:

Утверждение:

Если $V = \sum_{k=1}^{n-1} a_k V^{(k)}$, то $\omega = \sum_{k=1}^{n-1} (\lambda^{(k)} a_k) V^{(k)}$

Если $V = V^{(k)}$, то $\omega = \lambda^{(k)} V^{(k)}$

Докажем утверждение, которое покажет, какое ограничение на шаг сетей наложит (13)

Утверждение:

Чтобы разностная схема, для решения (1*) была вычислительно устойчива на сетке (n,m)

достаточно чтобы $\boxed{\tau < \frac{h^2}{2}}$

Определение:

Разностная схема для задачи (1*) на сетке (n,m) называется вычислительно устойчивой нужно $\forall k$ (13).

Доказательство: $-2 < -\lambda^{(k)} < 0 \quad 0 < \lambda^{(k)} \tau < 2$

$\tau > 0$ – по смыслу, так как шаг по времени $\lambda^{(k)} = \frac{4}{h^2} \sin^2 \frac{\pi k}{2n} > 0, k = 1, n-1$

Из определения вычислительной устойчивости, требуем, чтобы $\forall k = \overline{1, n-1}$

$$\frac{4}{h^2} \sin^2 \left(\frac{\pi k}{2n} \right) \cdot \tau < 2 \Rightarrow \tau < \frac{h^2}{2 \sin^2 \left(\frac{\pi k}{2n} \right)} \quad (14^*) \Rightarrow \text{с гарантией } \tau < \frac{h^2}{2} \quad (14)$$

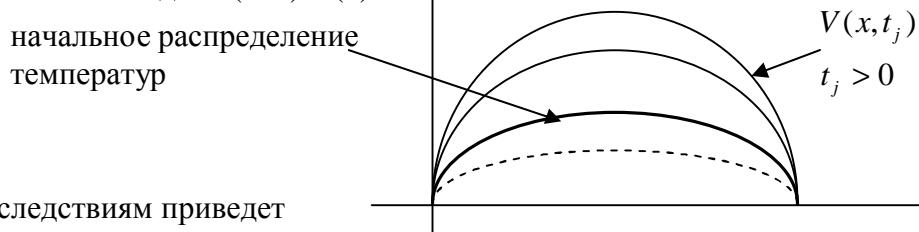
- Чтобы схема была вычислительно устойчивой шаг по времени должен быть много меньше чем шаг по пространству.

Пример: $h=10^{-2} \Rightarrow \tau < 10^{-4}/2 = 0,5 \cdot 10^{-4}$

- Переход от оценки (14*) к (14) не является слишком грубым, если сетка по x является достаточно густой и n достаточно велико, то $\sin^2 \frac{\pi(n-1)}{2n} \approx 1$.

В #11 мы изучаем свойство разностной схемы (3) для задачи (1*) и решаем разностную схему (9) и доказываем утверждение при каких условиях решение задачи (9) ведет себя в соответствие с решением (1**)

Определение вычислительной устойчивости и определение вычислительной устойчивости формулируются на основе свойств задачи (1**) и (9).

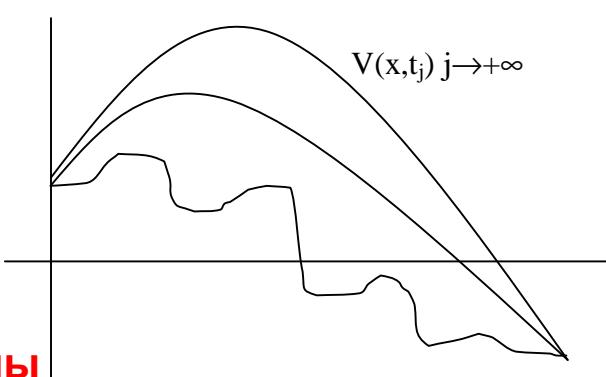


Протокол:

1. Обсудим к каким последствиям приведет нарушение условия (14) при решении задачи (9).
2. Обсудим, к каким последствиям приведет нарушение задачи (14*)

Если начальное распределение температур задачи (1*) представлено в машине с некоторой погрешностью, которая имеет составляющую гармонику $\sin(\pi kx)$, для которой условие

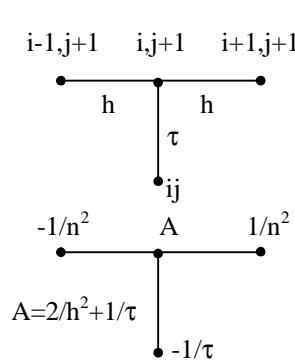
$$|1 - \lambda^{(k)} \tau| \geq 1 \Rightarrow \psi(x, t_j) \rightarrow \infty \quad j \rightarrow \infty$$



12 Неявные разностные схемы для решения уравнения теплопроводности.

Рассмотрим модельную задачу (1*) из #11:

$$(1^*) \begin{cases} U_t = U_{xx} + f(x, t) \\ U(x, 0) = \varphi(x) \\ U(0, t) = \mu_1(t), U(1, t) = \mu_2(t) \\ x \in [0, 1], t \in [0, T] \end{cases} \text{Сетка: } h = \frac{1}{n}, x_i = ih, i = \overline{0, n}, x_0 = 0, x_n = 1, \tau = \frac{T}{m}, t_j = j\tau, j = \overline{0, m}$$



$$(15) \begin{cases} \frac{V_{ij+1} - V_{ij}}{\tau} - \frac{V_{i+1,j+1} - 2V_{ij+1} + V_{i-1,j+1}}{h^2} = f(x_i, t_j) \\ f_{ij+1} \stackrel{\text{def}}{=} \Phi_{ij}, i = \overline{1, n-1}, j = \overline{0, m-1} \\ V_{i0} = \varphi_i, i = \overline{0, n} \\ V_{0j} = \mu_{1j}, V_{nj} = \mu_{2j}, j = \overline{1, m} \end{cases}$$

Это чисто неявная схема: схему (15) решают послойно, 0-й слой – из начальных условий. При нахождении каждого последующего слоя, нужно решать систему с трехдиагональной матрицей.

Для отыскания $j+1$ слоя, используется j -й слой:

$$V_{0j+1} = \mu_{1j+1}, V_{nj+1} = \mu_{2j+1}$$

$$(16) \boxed{-\frac{1}{h^2}V_{i-1,j+1} + AV_{ij+1} - \frac{1}{h^2}V_{i+1,j+1} = f_{ij+1} + \frac{V_{ij}}{\tau}}$$

Для модельной задачи (1^{*}) и неявной схемы (15) можно построить вспомогательную задачу (1***) и неявную разностную схему, аналогичную (15) \Rightarrow можно ввести определение вычислительной устойчивости (15).

Оказывается (15) вычислительно устойчива при любом выборе шага h и τ . (Эту теорему доказать самим)

Неявная схема с весом $\frac{1}{2}$.

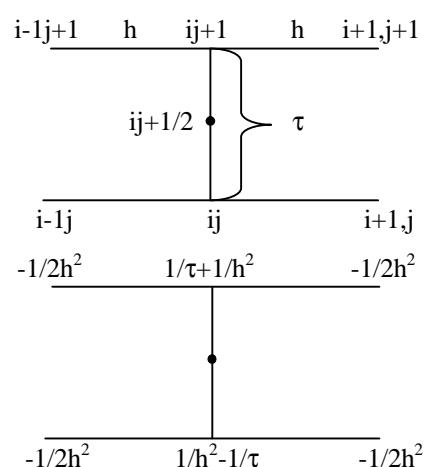
Запишем вычислительно устойчивую схему для любых h и τ и будем более быструю сходимость чем, чисто неявная и явная схема.

$$(17) \begin{cases} \frac{V_{ij+1} - V_{ij}}{\tau} - \frac{1}{2} \frac{V_{i+1,j} - 2V_{ij} + V_{i-1,j}}{h^2} - \frac{1}{2} \frac{V_{i+1,j+1} - 2V_{ij+1} + V_{i-1,j+1}}{h^2} = \\ = f(x_i, t_{j+1/2}), i = \overline{1, n-1}, j = \overline{0, m-1} \\ V_{i0} = \varphi_i, i = \overline{0, n} \\ V_{0j} = \mu_{1j}, V_{nj} = \mu_{2j}, j = \overline{0, m} \quad f_{ij+1/2} \stackrel{\text{def}}{=} \Phi_{ij} \end{cases}$$

схема (17) является неявной и решается послойно.

начальный слой: $V_{i0} = \varphi_i, i = \overline{0, n}$ слой $j+1$ находится на основе j -го, при этом решается система с 3-х диагональной матрицей.

$$V_{0j+1} = \mu_{1j+1}, V_{nj+1} = \mu_{2j+1}$$



- решается на каждом слое.

Определение вычислительной устойчивости и ее анализ проводится аналогично #11.

Итог по #11-12:

Пусть дана задача Коши для линейного параболического уравнения (например (1) или (1^{*})). В этом случае предпочтительнее использовать неявные разностные схемы улучшенного порядка

аппроксимации (например схему (17)) Она вычислительно устойчива при $\forall h$ и τ и имеет 2-й порядок сходимости по τ и по h .

Наряду с линейными задачами имеется нелинейный мир, например: $U_t = U_{xx} + f(x, t) + U^2 \dots$

Неявная схема с рассмотренными выше 2 шаблонами приводит к тому, что на каждом слое приходиться решать систему большей размерности: $n+1$, поэтому для нелинейных задач часто используется явную разностную схему с обычным шаблоном, так как каждый новый слой выражается через предыдущий по формулам и нет необходимости в системах.