

## Курс «Вычислительные алгоритмы теории автоматического управления».

### **Лекция 4. Методы решения краевых задач для ОДУ. Интегрирование систем с запаздыванием. Моделирование систем с распределенными параметрами. Методы интерполирования функций.**

Методы приближенного решения краевых задач для ОДУ. Уравнения непрерывных систем с запаздыванием. Численные методы интегрирования уравнений с запаздывающим аргументом. Моделирование систем с распределенными параметрами. Численное решение уравнений в частных производных методом сеток. Устойчивость и сходимость линейных разностных схем. Методы решения линейных интегральных уравнений. Методы интерполирования функций.

#### **Методы приближенного решения краевых задач для ОДУ.**

Рассмотрим двухточечную краевую задачу для нормальной системы обыкновенных дифференциальных уравнений.

$$\begin{cases} \dot{x} = F(t, x) \\ G[x(a)] = 0, \\ D[x(b)] = 0 \end{cases}$$

где  $t \in [a, b]; x \in \mathbb{R}^n; G: \mathbb{R}^n \rightarrow \mathbb{R}^k; D: \mathbb{R}^n \rightarrow \mathbb{R}^{n-k}$ .

Точное (аналитическое) решение краевых задач вызывает гораздо большие трудности, чем решение задач Коши. Поэтому обычно такие задачи решаются с помощью приближенных методов. По типу представления результатов приближенного решения методы разделяют на две группы: приближенно-аналитические, дающие приближенное решение краевой задачи на отрезке  $[a, b]$  в виде некоторой конкретной функции, и, собственно численные или сеточные методы, дающие каркас приближенного решения на заданной на интервале  $[a, b]$  сетке. Можно провести следующую классификацию методов решения краевых задач для ОДУ: методы сведения к задаче Коши (метод пристрелки или стрельбы, метод дифференциальной прогонки, метод редукции); метод конечных разностей; метод коллокации; проекционные методы (метод Галеркина); вариационные методы (наименьших квадратов, метод Рунге) и прочее /Вержбицкий с295/.

#### *Решение краевой задачи методом стрельбы.*

При численном решении краевой задачи обычно предполагается, что у нее существует единственное решение. Однако, на практике, проверка условий существования и единственности достаточно трудно проверять, особенно из-за неизбежных ошибок вычислений. Метод стрельбы (или пристрелки) состоит в сведении краевой задачи к задаче Коши, для которой существует, как уже было показано, много приближенных вычислительных методов. Рассмотрим задачу отыскания  $n$  составляющих вектора  $p = (p_1, p_2, \dots, p_n)^T$ , при которых решение задачи Коши следующего вида  $\dot{x} = F(t, x); x(a) = p$  будет совпадать с решением краевой задачи. Очевидно, что при этом должны

выполняться следующие соотношения  $\begin{cases} G[x(a, p)] = 0 \\ D[x(b, p)] = 0 \end{cases}$ . Искомые составляющие (переменные)

вектора  $p \in \mathbb{R}^n$  можно искать следующим образом. Сначала из системы уравнений  $k$ -го порядка

(в общем случае уравнения нелинейные) вида  $G[x(a, p)] = 0$  находим  $(n - k)$ -мерное параметрическое семейство решений, которое должно существовать в предположении, что краевая задача корректно поставлена. Такое семейство решений запишем в виде следующих соотношений:

$$\begin{cases} x_i(a) = p_i = \alpha_i(p_{k+1}, \dots, p_n); i = 1, 2, \dots, k \\ x_j(a) = p_j; j = k + 1, \dots, n \end{cases},$$

где  $p_j, j = k + 1, \dots, n$  - некоторые произвольные параметры. Решение  $x^{(a)}(t, p_{k+1}, \dots, p_n)$  задачи Коши:

$$\begin{cases} \dot{x}^{(a)} = F(t, x^{(a)}) \\ x_i^{(a)}(a) = \alpha_i(p_{k+1}, \dots, p_n), i = 1, \dots, k \\ x_j^{(a)}(a) = p_j, j = k + 1, \dots, n \end{cases}$$

будет также являться решением краевой задачи, если выполняются уравнения  $D[x^{(a)}(b, p_{k+1}, \dots, p_n)] = 0$ .

Данные уравнения для вычисления  $(n - k)$  неизвестных параметров  $p_i, i = k + 1, \dots, n$  часто называют уравнениями «сшивания».

Аналогично можно поступить, если известно, что  $k$  - параметрическое семейство решений системы уравнений  $(n - k)$  - го порядка относительно  $n$  переменных  $D[x(b, p_1, \dots, p_n)] = 0$  можно записать в виде:

$$\begin{cases} x_i(b) = \beta_i(p_1, \dots, p_k); i = 1, 2, \dots, n - k \\ x_{n-k+j}(b) = p_j; j = 1, 2, \dots, k \end{cases},$$

где  $p_j, j = 1, \dots, k$  - некоторые произвольные постоянные. Тогда решение  $x^{(b)}(t, p_1, \dots, p_k)$  задачи Коши:

$$\begin{cases} \dot{x}^{(b)} = F(t, x^{(b)}) \\ x_i^{(b)}(b) = \beta_i(p_1, \dots, p_k), i = 1, \dots, n - k \\ x_{n-k+j}^{(b)}(b) = p_j, j = 1, \dots, k \end{cases}$$

будет решением краевой задачи, если  $p_j, j = 1, \dots, k$  удовлетворяют уравнениям сшивания  $G[x^{(b)}(a, p_1, \dots, p_k)] = 0$ .

Выбор уравнений сшивания зависит от того, что больше  $k$  или  $(n - k)$ , так как численно легче решать систему меньшего порядка.

#### Линейная краевая задача.

Пусть требуется найти решение линейной краевой задачи вида:

$$\begin{cases} \dot{x} = A(t)x + f(t) \\ G[x(a)] = 0 \\ D[x(b)] = 0 \end{cases},$$

где  $t \in [a, b]; x, f \in \mathbb{R}^n; G: \mathbb{R}^n \rightarrow \mathbb{R}^k; D: \mathbb{R}^n \rightarrow \mathbb{R}^{n-k}; A(t) \in \mathbb{R}^{n \times n}$ .

Общее решение линейной краевой задачи (10) будем разыскивать в виде

$x(t) = x^{(0)}(t) + \sum_{i=1}^n p_i \bar{x}^{(i)}(t)$ . Здесь  $x^{(0)}(t) \in \mathbb{R}^n$  - решение неоднородной задачи Коши:

$$\begin{cases} \dot{x}^{(0)} = A(t)x^{(0)} + f(t) \\ x^{(0)}(a) = 0 \end{cases},$$

где  $t \in [a, b]$ , а  $\bar{x}^{(i)}(t) \in R^n, i = 1, 2, \dots, n$  решения однородных задач Коши вида

$$\begin{cases} \dot{x}^{(i)} = A(t)x^{(i)} \\ x^{(i)}(a) = (0, \dots, 0, \underset{i\text{-место}}{1}, 0, \dots, 0)^T, i = 1, \dots, n; t \in [a, b]. \end{cases}$$

Отсюда вытекает, что параметры  $p_i, i = 1, \dots, n$  должны быть решениями системы линейных алгебраических уравнений  $n$ -го порядка:

$$G[\sum_{i=1}^n p_i \bar{x}^{(i)}(a)] = 0 \text{ и } D[x^{(0)}(b) + \sum_{i=1}^n p_i \bar{x}^{(i)}(b)] = 0.$$

Эта система решений должна иметь единственное решение при условии, что краевая задача поставлена корректно и решения однородных задач Коши линейно независимы. Заметим, что для вычисления значений  $p_i, i = 1, \dots, n$  требуются только знание компонент векторов  $\bar{x}^{(i)}(a), \bar{x}^{(i)}(b)$ . После вычисления  $p_i, i = 1, \dots, n$  из системы алгебраических уравнений, искомое решение краевой

задачи можно найти из соотношения  $x(t) = x^{(0)}(t) + \sum_{i=1}^n p_i \bar{x}^{(i)}(t)$ , которое будет совпадать с

$$\text{решением задачи Коши для уравнения } \begin{cases} \dot{x} = A(t)x + f(t) \\ x(a) = (p_1, \dots, p_n)^T, t \in [a, b]. \end{cases}$$

Следует отметить, что все задачи Коши решаются численно, поэтому после реализации метода стрельбы получают приближенное решение. Если системы линейных алгебраических уравнений плохо обусловлена, то возможна большая потеря точности. Обычно это происходит, если у однородной линейной задачи имеются решения, сильно различающиеся скоростью изменения по параметру  $t$ .

### Пример.

Построим алгоритм метода пристрелки для решения линейной краевой задачи для уравнения  $(t+1)x^{(4)} + (4+t)\ddot{x} + 3x = 6t + 3, t \in [0, 1]$ . Краевые условия на левом конце  $t = 0$  имеют вид

$$\begin{cases} 2x(0) + \ddot{x}(0) = -2 \\ x(0) - 2\ddot{x}(0) = 1 \end{cases}, \text{ а на правом конце } t = 1, \begin{cases} x(1) + \ddot{x}(1) = 3 \\ \dot{x}(1) + \ddot{x}(1) = 2 \end{cases}. \text{ Приведем заданную задачу к}$$

нормальной форме, введя новые переменные:  $x_1(t) = x(t); x_2(t) = \dot{x}(t); x_3(t) = \ddot{x}(t); x_4(t) = \dddot{x}(t)$ . В векторной форме уравнения нормальной системы можно записать в следующем виде:

$$\dot{x} = A(t)x + f(t), \text{ где } x = \{x_1, x_2, x_3, x_4\}^T, A(t) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{3}{t+1} & 0 & -\frac{4+t}{t+1} & 0 \end{pmatrix}, f(t) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \frac{6t+3}{t+1} \end{pmatrix}.$$

Краевые условия соответственно примут вид  $\begin{cases} 2x_1(0) + x_4(0) = -2 \\ x_1(0) - 2x_4(0) = 1 \end{cases}, \begin{cases} x_1(1) + x_3(1) = 3 \\ x_2(1) + x_4(1) = 2 \end{cases}$ . Отсюда из первой

системы алгебраических уравнений найдем  $x_1(0) = -0.6, x_4(0) = -0.8$ . Запишем теперь алгоритм стрельбы.

$$1. \text{ Решается неоднородная задача Коши вида } \begin{cases} \dot{x}^{(0)} = A(t)x^{(0)} + f(t) \\ x^{(0)}(0) = (-0.6, 0, 0, -0.8)^T. \end{cases}$$

2. Решаются две задачи Коши для однородной системы дифференциальных уравнений вида

$$\begin{cases} \dot{x}^{(1)} = A(t)x^{(1)} \\ x^{(1)}(0) = (0, 1, 0, 0)^T \end{cases} \text{ и } \begin{cases} \dot{x}^{(2)} = A(t)x^{(2)} \\ x^{(2)}(0) = (0, 0, 1, 0)^T \end{cases}.$$

3. Построим векторную функцию  $\hat{x}(t) = x^{(0)}(t) + p_2 x^{(1)}(t) + p_3 x^{(2)}(t)$  (так как переменные  $p_1, p_4$  уже определены).

4. Выбираем такие значения переменных  $p_2, p_3$ , чтобы функция  $\hat{x}(t)$  удовлетворяла краевым условиям

$$\begin{cases} \hat{x}_1(1) + \hat{x}_3(1) = 3 \\ \hat{x}_2(1) + \hat{x}_4(1) = 2 \end{cases}. \text{ То есть } \begin{cases} [x_1^{(0)}(1) + p_2 x_1^{(1)}(1) + p_3 x_1^{(2)}(1)] + [x_3^{(0)}(1) + p_2 x_3^{(1)}(1) + p_3 x_3^{(2)}(1)] = 3 \\ [x_2^{(0)}(1) + p_2 x_2^{(1)}(1) + p_3 x_2^{(2)}(1)] + [x_4^{(0)}(1) + p_2 x_4^{(1)}(1) + p_3 x_4^{(2)}(1)] = 2 \end{cases}.$$

5. После вычисления  $p_2, p_3$  находится решение  $\hat{x}(t)$ , которое является решением задачи Коши

$$\begin{cases} \dot{\hat{x}} = A(t)\hat{x} + f(t) \\ \hat{x}(0) = (-0.6, p_2, p_3, -0.8)^T \end{cases}. \text{ Первая компонента } \hat{x}_1(t) \text{ вектора } \hat{x}(t) \text{ является решением исходной}$$

краевой задачи.

*Подходы к решению уравнений сшивания для нелинейных краевых задач.*

Для краевых задач второго порядка, иногда третьего порядка, как правило, получается одно уравнение сшивания. Однако, начиная с краевых задач четвертого порядка, обычно получается нелинейная система уравнений сшивания. Рассмотрим, например, уравнение

$G[x(a, \bar{p})] = \Phi(\bar{p}) = 0$ , где  $\Phi, G, \bar{p} \in \mathbb{R}^k$ . Если известно начальное приближение  $\bar{p}^{(0)}$ , то можно использовать для решения нелинейной краевой задачи метод Ньютона. Основную трудность при расчете итераций в этом случае составляет вычисление соответствующей матрицы Якоби.

Решение краевых задач для ОДУ в системе Matlab осуществляется с помощью функции *bvp4c*. Типичная форма вызова данной функции имеет вид:

$$sol = bvp4c(odefun, bcfun, solinit),$$

где:

аргумент *odefun* - представляет собой указатель на функцию, содержащую схему расчета правых частей системы ОДУ, то есть  $dydx = odefun(x, y)$ ;

аргумент *bcfun* - представляет собой указатель на функцию, задающую граничные условия, то есть  $res = bcfun(xa, yb)$ , где  $xa, yb$  - вектора - столбцы заданных значений решения в концевых точках отрезка  $[a, b]$ ;

аргумент *solinit* содержит ссылку на структуру, которая инициализирует начальное решение; данная структура использует функцию *bvpinit* ( $solinit = bvpinit(x, yinit, params)$ ) и содержит следующие поля:

*solinit.x* - содержит сетку граничных значений, где  $a = solinit.x(1)$  и  $b = solinit.x(end)$ ;

*solinit.y* - начальное приближение решения, такое, что  $solinit.y(:, i)$  соответствует узлам сетки  $solinit.x(:, i)$ ;

*params* - дополнительный вектор параметров, необходимых для задания граничных условий.

### Уравнения непрерывных систем с запаздыванием.

Непрерывные системы с запаздыванием обычно описывают с помощью дифференциального уравнения  $n$ -го порядка с  $l$  отклонениями аргумента. Отклонения, в общем случае, носят переменный характер. Такое уравнение можно записать в следующем виде:

$$\dot{x}^{(m_0)} = f[t, x(t), \dots, x^{(m_{0-1})}(t), x(t - \tau_1(t)), \dots, x^{(m_1)}(t - \tau_1(t)), \dots, x^{(m_l)}(t - \tau_l(t))],$$

где  $0 < \tau_1(t) < \dots < \tau_l(t)$ ,  $\max_{1 \leq i \leq l} [m_i] = n$ . Здесь, под обозначением  $x^{(k)}(t - \tau_i(t))$  понимается  $k$ -я

производная от функции  $x(z)$ , взятая в точке  $z = t - \tau_i(t)$ . Пусть задана начальная точка  $t_0$ .

Каждое отклонение  $\tau_i > 0$  определяет начальное множество  $E_{t_0}^i$ , состоящее из точки  $t_0$  и тех значений  $t - \tau_i(t)$ , для которых справедливы соотношения:  $[t - \tau_i(t)] < t_0$ ,  $t \geq t_0$ . Введем

множество  $E_{t_0} = \sum_{i=1}^l E_{t_0}^i$ . Зададим на множестве  $E_{t_0}$  непрерывные функции

$\phi_k(t), k = 0, 1, 2, \dots, n-1$ . Обычно, наиболее естественно, рассматривать случай, когда:

$\phi_k(t) \equiv \phi_0^{(k)}(t), k = 0, 1, 2, \dots, n-1$ . Основная начальная задача для исходного

дифференциального уравнения заключается в определении  $(n-1)$  раз непрерывного

дифференцируемого решения  $x(t)$  при  $t \geq t_0$  и условиях, задаваемых следующими соотношениями:

$$x^{(k)}(t_0 + 0) = x_0^{(k)} = \phi_k(t_0), k = 0, 1, \dots, (n-1) \text{ и } x^{(k)}(t - \tau_i(t)) \equiv \phi_k(t - \tau_i(t)),$$

если  $[t - \tau_i(t)] < t_0, i = 1, 2, \dots, l$ .

Уравнение  $n$ -го порядка с отклоняющимся аргументом можно заменить, так же, как и для уравнений без отклонений аргумента, соответствующей системой уравнений первого порядка. Однако, такая замена задач, не будет полностью эквивалентна при введении ограничения (наиболее естественного для различных приложений), что начальные функции  $\phi_k(t)$  должны быть производными одной и той же функции  $\phi_k(t) \equiv \phi_0^{(k)}(t), k = 0, 1, 2, \dots, n-1$ . В этом случае приходится дополнительно решать задачу назначения начальных условий.

*Свойства решений дифференциальных уравнений с запаздывающим аргументом.*

*Метод интегрирования по шагам /Эльсгольц, Норкин с17/.*

Рассмотрим основную начальную задачу для простейшего дифференциального уравнения первого порядка с одним запаздывающим аргументом:  $\dot{x}(t) = f(t, x(t), x(t - \tau)); x \in \mathbb{R}^n$ , где  $\tau > 0, x(t) = \phi_0(t)$ , при  $t_0 - \tau \leq t \leq t_0$ . Наиболее естественным методом решения этой задачи является, так называемый, метод шагов, заключающийся в том, что решение  $x(t)$  рассматриваемой задачи определяется из дифференциальных уравнений без запаздывания:

$\dot{x}(t) = f(t, x(t), \phi_0(t - \tau))$  при  $t_0 \leq t \leq t_0 + \tau$ ;  $x(t_0) = \phi_0(t_0)$ . Если предположить, что решение

$x(t) = \phi_1(t)$  этой начальной задачи существует на всем отрезке  $[t_0, t_0 + \tau]$ , аналогично получим:

$\dot{x}(t) = f(t, x(t), \phi_1(t - \tau))$  при  $t_0 + \tau \leq t \leq t_0 + 2\tau$ ,  $x(t_0 + \tau) = \phi_1(t_0 + \tau)$ , ...

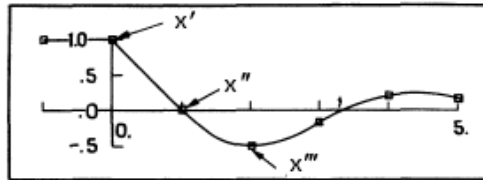
$\dot{x}(t) = f(t, x(t), \phi_n(t - \tau))$  при  $t_0 + n\tau \leq t \leq t_0 + (n+1)\tau$ ,  $x(t_0 + n\tau) = \phi_n(t_0 + n\tau)$ , ... .., где  $\phi_i(t)$

решение рассматриваемой начальной задачи на отрезке  $t_0 + (i-1)\tau \leq t \leq t_0 + i\tau$ . Этот метод дает возможность определить решение  $x(t)$  на некотором конечном отрезке и одновременно доказывает существование решения в окрестности точки  $(t_0, \phi(t_0))$ , если функции  $\phi, f$  непрерывны в рассматриваемой области переменных. Единственность решения будет достигаться, если функция  $f$  удовлетворяет одному из условий, обеспечивающих единственность решения уравнения:  $\dot{x}(t) = f(t, x(t), \phi_0(t-\tau))$  без отклонений аргумента, например условию Липшица по второму аргументу.

*Пример.* Рассмотрим уравнение  $\dot{x}(t) = -x(t-1)$ , где  $\phi(t) = 1$  при  $-1 \leq t \leq 0$  ( $\tau = 1$ ). Применяя описанный

выше метод шагов, получим:  $x(t) = 1-t$  при  $-1 \leq t \leq 0$ ;  $x(t) = 1-t + \frac{(t-1)^2}{2!}$  при  $1 \leq t \leq 2$ ;

$$x(t) = 1-t + \frac{(t-1)^2}{2!} - \frac{(t-2)^3}{3!} \text{ при } 2 \leq t \leq 3; \dots\dots$$



Заметим, что решение имеет разрывные производные, несмотря на то, что правая часть дифференциального уравнения и начальная функция принадлежат  $C^\infty$ . Это происходит из-за того, что начальная функция не удовлетворяет дифференциальному уравнению. Однако с каждым шагом  $\tau$  эти разрывы все более сглаживаются.

Применение метода шагов становится затруднительным, если запаздывание  $\tau$  мало, по сравнению с отрезком, на котором требуется определить решение. Рассмотрим теперь более общее уравнение:  $\dot{x}(t) = f(t, x(t-\tau_1), \dots, x(t-\tau_l))$ .

**Теорема (существования и единственности решения начальной задачи) /Эльсгольц, Норкин с31/.** Если в уравнении  $\dot{x}(t) = f(t, x(t-\tau_1), \dots, x(t-\tau_l))$  все  $\tau_i > 0, i = 1, 2, \dots, l$ , а функция  $f$  непрерывна в окрестности точки  $(t_0, \phi(t_0), \phi(t_0-\tau_1), \dots, \phi(t_0-\tau_l))$  и удовлетворяет условию Липшица по всем аргументам, начиная со второго, а начальная функция  $\phi$  непрерывна на начальном множестве  $E_{t_0}$ , то существует единственное решение  $x(t)$  основной начальной задачи для рассматриваемого уравнения при  $t \in [t_0, t_0 + h]$ , где  $h$  достаточно мало.

*Математические модели линейных систем с запаздыванием. Характеристический квазиполином.*

Рассмотрим линейное однородное уравнение вида:  $\sum_{p=0}^n \sum_{j=0}^l a_{pj} x^{(p)}(t-\tau_j) = 0$ . Найдем

частные решения этого уравнения в виде  $x(t) = e^{kt}$ , где  $k$  – некоторая постоянная. Тогда для

определения значения  $k$  получим следующее характеристическое уравнение:

$$\sum_{p=0}^n \sum_{j=0}^l a_{pj} k^p \exp[-k\tau_j] = 0. \text{ Левая часть уравнения } \sum_{p=0}^n \sum_{j=0}^l a_{pj} k^p \exp[-k\tau_j] = 0 \text{ называется}$$

характеристическим квазиполиномом. Очевидно, что уравнение  $\Phi(k) = 0$  имеет бесконечное множество корней и каждому корню  $k_i$  соответствует решение  $e^{k_i t}$ . Кратным корням  $k_i$  уравнения, имеющим кратность  $\alpha_i$ , соответствуют не только решение  $e^{k_i t}$ , но и решения  $t e^{k_i t}, \dots, t^{\alpha_i-1} e^{k_i t}$  и,

следовательно, если ряд  $\sum_{i=1}^{\infty} P_i(t) e^{k_i t}$ , где  $P_i(t)$  - многочлены с произвольными постоянными коэффициентами степени  $\alpha_i - 1$ , сходится и допускает  $n$ -кратное дифференцирование, то его сумма является решением линейного однородного дифференциального уравнения

$$\sum_{p=0}^n \sum_{j=0}^l a_{pj} x^{(p)}(t - \tau_j) = 0.$$

### **Численные методы интегрирования уравнений с запаздывающим аргументом.**

#### **Метод Эйлера.**

Пусть задана система нелинейных уравнений вида  $\dot{y} = F(t, y(t), y(t - \tau))$ , где  $y(t), y(t - \tau) \in \mathbb{R}^n, \tau \in \mathbb{R}, F = (f_1, \dots, f_n)^T, y(t) = \phi(t), t \in [t_0 - \tau, t_0]$ . Рассмотрим возможность интегрирования данной системы с постоянным аргументом запаздывания с помощью модифицированного метода Эйлера, который задается следующей схемой:

$$\begin{cases} y^{(\frac{k+1}{2})} = y^{(k)} + \frac{h}{2} f(t^{(k)}, y^{(k)}) \\ y^{(k+1)} = y^{(k)} + h f(t^{(\frac{k+1}{2})}, y^{(\frac{k+1}{2})}) \\ t^{(k+1)} = t^{(k)} + h \\ t^{(\frac{k+1}{2})} = t^{(k)} + \frac{h}{2} \end{cases}$$

Можно показать, что данная модификация метода Эйлера имеет второй порядок точности ( $p=2$ ). Таким образом, для получения решения в точке  $t^{(k)}$  надо получить предварительное решение в точке  $t^{(\frac{k+1}{2})} = t^{(k)} + \frac{h}{2}$ . Соответственно от этих точек надо брать запаздывание  $\tau$ , то есть надо

найти значение решения в точках  $t^{(k)} - \tau$  и  $t^{(k)} + \frac{h}{2} - \tau$ .

Таким образом, чтобы определить значение  $y(t^{(k)} - \tau)$  нужно выполнить следующие действия. Если значение  $t^{(k)} - \tau$  лежит левее начальной точки  $t^{(0)}$ , то  $y(t^{(k)} - \tau)$  определяется из начальных условий. Если  $t^{(k)} - \tau$  совпадает с одним из узлов правее точки  $t^{(0)}$ , тогда  $y(t^{(k)} - \tau)$  принимает значение функции в этом узле. Если величина  $t^{(k)} - \tau$  не совпадает ни с одним узловым значением  $t^{(k)}, k = 0, 1, 2, \dots$ , она лежит внутри некоторого отрезка  $[t_i, t_{i+1}], i < k$  и можно, по

значениям  $y(t^{(i-1)}), y(t^{(i)}), y(t^{(i+1)})$ , построить некоторый интерполяционный многочлен  $P_3$  (например, многочлен Лежандра или кубический сплайн 3-го порядка) для определения приближенного значения  $y(t^{(k)} - \tau) \approx P_3(t^{(k)} - \tau)$ . Таким образом, схема расчета значения решения в новой точке для системы  $\dot{y} = F(t, y(t), y(t - \tau))$  будет выглядеть следующим образом:

$$\begin{cases} y^{(\frac{k+1}{2})} = y^{(k)} + \frac{h}{2} f(t^{(k)}, y(t^{(k)} - \tau)) \\ y^{(k+1)} = y^{(k)} + hf(t^{(\frac{k+1}{2})}, y(t^{(\frac{k+1}{2})} - \tau)) \\ t^{(k+1)} = t^{(k)} + h \\ t^{(\frac{k+1}{2})} = t^{(k)} + \frac{h}{2} \end{cases}$$

*Применение методов Рунге-Кутты с постоянной длиной шага для постоянного запаздывания.*

При применении методов Рунге-Кутты, сразу возникает вопрос, как задавать, или вычислять значения в точках  $(t^{(k)} + c_j h - \tau)$ . Но если запаздывание постоянное и длина шага должна быть выбрана в соответствии со следующим соотношением  $\tau = mh$ , где  $m$  - некоторое целое число, то естественно использовать уже вычисленные  $m$  шагов назад значения решения. Это можно интерпретировать, как последовательное решение уравнения  $\dot{y} = F(t, y(t), \varphi(t - \tau))$  для интервала

$[t^{(0)}, t^{(0)} + \tau]$ , а затем уравнений  $\begin{cases} \dot{y} = F(t, y(t), z^{(1)}(t)) \\ \dot{z}^{(1)} = F(t - \tau, z^{(1)}(t), \varphi(t - 2\tau)) \end{cases}$  для интервала  $[t^{(0)} + \tau, t^{(0)} + 2\tau]$ .

Затем уравнений  $\begin{cases} \dot{y} = F(t, y(t), z^{(1)}(t)) \\ \dot{z}^{(1)} = F(t - \tau, z^{(1)}(t), z^{(2)}(t)) \\ \dot{z}^{(2)} = F(t - 2\tau, z^{(2)}(t), \varphi(t - 3\tau)) \end{cases}$  для интервала  $[t^{(0)} + 2\tau, t^{(0)} + 3\tau]$  и т.д. То

есть, это точный численный аналог приведенного выше «метода шагов». Таким образом, процесс решения уравнения  $\dot{y} = F(t, y(t), y(t - \tau))$  с постоянной величиной запаздывания можно свести к решению последовательности систем обыкновенных дифференциальных уравнений, решаемых методом Рунге-Кутты с постоянным шагом  $p$ -го порядка.

*Методы с переменной длиной шага.*

Однако, в общем случае, приведенные выше методы решения задач с запаздывающим аргументом не позволяют произвольно менять длину шага, и применение их, например, к уравнениям с переменной величиной запаздывания вызывает значительные трудности и может приводить к большим ошибкам численного интегрирования. Для того, чтобы избежать этого, необходимо использование глобальной аппроксимации решения. Наиболее подходящими методами для реализации такого являются многошаговые методы типа Адамса или методы Рунге-Кутты с переменным шагом интегрирования или, так называемые, непрерывные многошаговые методы.

В пакете MatLab имеется численная процедура dde23, которая автоматически определяет нарушения непрерывности в предыстории и определяет те точки нарушения непрерывности, порядок которых достаточно мал для того, чтобы повлиять на выполнение вычислений. Так как в процессе интегрирования системы с запаздывающим аргументом поведение системы сглаживается, то процедура проверяет возможность увеличения шага интегрирования со временем.



Для получения общего решения используется кубическая полиномиальная аппроксимация, которая затем корректируется в процессе перевычисления формул.

### **Моделирования систем с распределенными параметрами.**

Основные теоретически проблемы управления объектами с распределенными параметрами связаны с разработкой законов управления объектами, описываемыми дифференциальными уравнениями с распределенными параметрами, а именно: решение задач управляемости, наблюдаемости, финитного управления, синтеза, оптимизации и многих других. К сожалению, только в ряде случаев удастся свести поставленные задачи к решению в рамках конечномерных пространств фазового пространства. Поэтому для нахождения решения вышеуказанных проблем приходится привлекать совершенно иной математический аппарат. Соответственно, приходится применять и совершенно другие методы моделирования систем, такие как моделирование систем с трансцендентными и иррациональными передаточными функциями, разнообразные сеточные модели с активными и пассивными элементами.

*Основные типы линейных уравнений второго порядка.*

*Параболический тип.* Уравнение параболического типа описывает процессы теплопроводности и диффузии и определяется условием:  $B^2 - 4A \cdot C = 0$

*Гиперболический тип.* Уравнения гиперболического типа описывают колебательные системы или волновые движения, и определяются условием:  $B^2 - 4A \cdot C > 0$ .

*Эллиптический тип.* Уравнения эллиптического типа описывают установившиеся процессы и определяются условием:  $B^2 - 4A \cdot C < 0$ .

*Пример.*

$u_t(t) = u_{xx}$ ,  $B^2 - 4A \cdot C = 0$ , - параболическое уравнение;

$u_{tt}(t) = u_{xx}$ ,  $B^2 - 4A \cdot C = 4 > 0$ , гиперболическое уравнение;

$u_{xx} + u_{yy} = 0$ ,  $B^2 - 4A \cdot C = -4 < 0$  - эллиптическое уравнение;

$yu_{xx} + u_{yy} = 0$ ,  $B^2 - 4A \cdot C = -4y$  - при:  $y > 0$  - эллиптическое;  $y = 0$  - параболическое;  $y < 0$  - гиперболическое.

*Постановка граничной (нестационарной) задачи.*

Если объект управления является нестационарным, то есть описывается нестационарным уравнением вида  $L(t, D)u + f(x, t) = 0$ , где  $L(t, D) = \frac{\partial}{\partial t} + \frac{\partial^2}{\partial x_1^2} + \dots + \frac{\partial^2}{\partial x_n^2}$ , то необходимо задавать краевые условия – совокупность граничных и начальных условий.

**Начальные условия** — значения фазовых переменных и/или их функций внутри исследуемой области, заданные в нулевой момент времени  $u|_{t=0} = \phi(x)$ , где  $\phi(x)$  - заданная функция.

Различают граничные условия первого, второго и третьего рода.

*Граничные условия первого рода* (условия Дирихле) включают в себя значения фазовых переменных, заданные на границе области  $u|_{\Gamma} = \phi_{\Gamma}(x)$ .

*Граничные условия второго рода* (условия Неймана) включают в себя поток фазовой переменной через границу области  $k \frac{du}{dn} \big|_{\Gamma} = q_{\Gamma}(x)$ , где  $n$  - вектор нормали к границе области,  $q_{\Gamma}(x)$  - заданная скорость потока, через границу.

*Граничные условия третьего рода* — уравнение, связывающее фазовую переменную и ее производные  $P(\frac{du}{dn}, u(x)) = 0$ , так называемые уравнения баланса. Классический пример уравнения баланса — уравнение теплового баланса, когда тепловой поток с границы объекта зависит от температуры.

### **Численное решение уравнений в частных производных методом сеток.**

*Численное решение задачи Дирихле для эллиптических уравнений.*

Рассмотрим задачу Дирихле для уравнения Лапласа  $u_{xx} + u_{yy} = 0$  в прямоугольной области  $0 \leq x \leq 1$ ,  $0 \leq y \leq 1$ . Пусть граничные условия задаются в следующем виде  $u(x, 0) = g_1(y)$ ,  $u(1, y) = g_2(y)$ ,  $u(x, 1) = g_3(y)$ ,  $u(0, y) = g_4(y)$ .

Предварительно вспомним понятие конечно – разностной аппроксимации для частных производных. Если исходить из разложения Тейлора функции двух переменных:

$$u(x+h, y) = u(x, y) + u_x(x, y)h + u_{xx}(x, y)\frac{h^2}{2!} + \dots,$$

$$u(x-h, y) = u(x, y) - u_x(x, y)h + u_{xx}(x, y)\frac{h^2}{2!} - \dots,$$

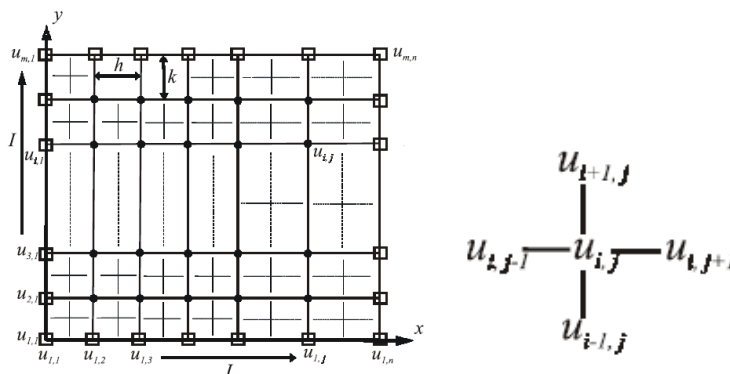
то можно записать следующие аппроксимации частных производных:

$$u_x(x, y) \approx \frac{u(x+h, y) - u(x, y)}{h}, \quad u_{xx}(x, y) \approx \frac{1}{h^2} [u(x+h, y) - 2u(x, y) + u(x-h, y)],$$

$$u_y(x, y) \approx \frac{u(x, y+k) - u(x, y)}{k}, \quad u_{yy}(x, y) \approx \frac{1}{k^2} [u(x, y+k) - 2u(x, y) + u(x, y-k)].$$

Однако, обычно, при расчетах пользуются центральными аппроксимациями.

Построим в плоскости  $(x, y)$  равномерную сетку.



Введем следующие обозначения

$$u_{i,j} = u(x_j, y_i), \quad u_{i+1,j} = u(x_j, y_i + k), \quad u_{i-1,j} = u(x_j, y_i - k), \quad u_{i,j-1} = u(x_j - h, y_i), \quad u_{i,j+1} = u(x_j + h, y_i);$$

$$u_x(x_j, y_i) = \frac{1}{2h}(u_{i,j+1} - u_{i,j-1}), \quad u_y(x_j, y_i) = \frac{1}{2k}(u_{i+1,j} - u_{i-1,j}),$$

$$u_{xx}(x_j, y_i) = \frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}), \quad u_{yy}(x_j, y_i) = \frac{1}{k^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}).$$

Тогда конечно-разностная аппроксимация уравнения Лапласа примет следующий вид:

$$\frac{1}{h^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \frac{1}{k^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = 0.$$

Если принять условие  $k = h$ , то уравнение Лапласа приводится к виду:

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = 0. \text{ Разрешив данное уравнение относительно величины } u_{i,j},$$

получим следующее выражение  $u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$ . Отметим, что все выражения

вычисляются для внутренних узлов сетки, так как значения функции на границах заданы. Таким образом, если задать начальные значения решения уравнения Лапласа в узлах сетки и затем последовательно уточнять их в соответствии с заданным шаблоном

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}), \text{ то итерационный процесс будет сходиться (релаксировать) к}$$

точному решению. Для численного решения задачи Дирихле методом релаксации можно использовать следующий алгоритм.

1. Присвоить величинам  $u_{i,j}$  во внутренних узлах сетки некоторые численные значения (например, равные среднему значению всех граничных условий).
2. Пересчитать значения во всех внутренних точках сетки в соответствии с шаблоном (маской)

$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1})$ , заменяя старое значение средним значением, вычисленным по четырем соседним точкам.

Недостатком данной схемы является низкая скорость сходимости итерационного процесса. Для улучшения скорости сходимости обычно используют специальные методы, основанные на особенностях представления конечно-разностной аппроксимации уравнения Лапласа в матричной форме.

#### Явные разностные схемы.

Явные разностные схемы используются для решения уравнений, в которые входят производные по времени. В качестве примера, можно привести смешанную краевую задачу для уравнения теплопроводности  $u_t = a^2 \Delta u + f(x, t)$ ,  $0 \leq x \leq X; 0 \leq t \leq T$  с граничными (краевыми) условиями:

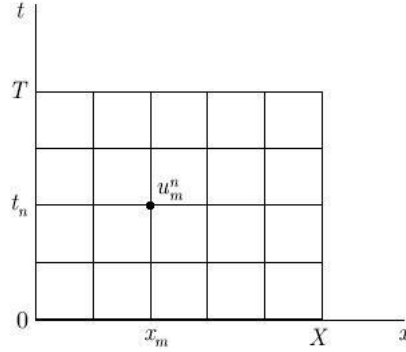
$$\begin{cases} A_1 \frac{\partial u(t, x)}{\partial x} \Big|_{(t, 0)} + B_1 u(t, 0) = \phi_1(t) \\ -A_2 \frac{\partial u(t, x)}{\partial x} \Big|_{(t, X)} + B_2 u(t, X) = \phi_2(t) \end{cases},$$

и начальными условиями  $u(0, x) = u_0(x)$ .

Для решения таких задач, область определения искомой функции покроем расчетной сеткой с узлами в точках:

$$\{t_n, x_m\}, n=0, \dots, N; m=0, \dots, M; t_n = n\tau; x_m = mh; \tau = T / N; h = X / M ,$$

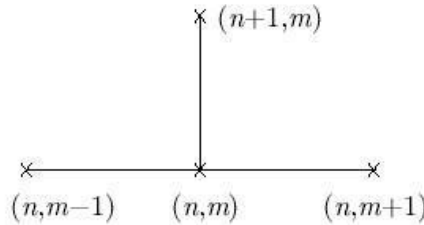
где  $\tau, h$  - шаги сетки по времени  $t \in [0, T]$  и по пространству  $x \in [0, X]$ . Приближенным решением задачи назовем сеточную функцию  $\{u_m^n\}$ . Верхний индекс в такой форме записи традиционно указывает на номер слоя по времени, нижний (нижние) – на номер узла сетки по пространственной координате.



Тогда, например, явную разностную схему для уравнения теплопроводности  $u_t = a^2 \Delta u + f(x, t)$  во внутренних узлах сетки, можно записать в следующем виде:

$$\frac{u_m^{n+1} - u_m^n}{\tau} = a \frac{u_{m-1}^n - 2u_m^n + u_{m+1}^n}{h^2} + f_m^n, \quad f_m^n = f(t_n, x_m) \in w_f^\tau$$

Шаблон расчетной схемы, представляющий собой конфигурацию расчетных узлов в области интегрирования, в этом случае будет иметь вид.



Проведем теперь аппроксимацию начальных данных и краевых условий:

$$\begin{aligned} u_m^0 &= u_0(x_m), m=0:M, u_0 \in w_\tau; \\ -A_1 \frac{u_1^n - u_0^n}{h} + B_1 u_0^n &= \phi_1(t_n), n=1:N; \phi_1 \in w_\phi^\tau; \\ A_2 \frac{u_m^n - u_{M-1}^n}{h} + B_2 u_m^n &= \phi_2(t_n), n=1:N; \phi_2 \in w_\phi^\tau. \end{aligned}$$

Предположим, для определенности, что  $A_k, B_k \geq 0, k=1, 2$ . Расчет ведется по рекуррентной формуле на каждом временном слое от  $n=1$  до  $n=N$  и от  $m=1$  до  $m=M-1$  во внутренних узлах. Слой  $n=0, (t=t_0)$  соответствует начальным данным; лучи  $m=0, (x=t_0)$  и  $m=M, (x=x_M)$  - левому и правому краевым условиям.

Запишем явную схему в виде:  $u_m^{n+1} = u_m^n + \frac{\tau}{h^2}(u_{m-1}^n - 2u_m^n + u_{m+1}^n) + \tau f_m^n$ . Используя данное выражение, последовательно, на каждом слое можно вычислить сеточную функцию во внутренних узлах области интегрирования. Для завершения расчета слоя  $t = t_{n+1}$  необходимо вычислить  $u_0^{n+1}$  и  $u_M^{n+1}$ , для чего разрешим левое и правое краевые условия относительно этих величин:

$$u_0^{n+1} = \frac{A_1 u_1^{n+1} + h\phi_1^{n+1}}{A_1 + hB_1}, \quad u_M^{n+1} = \frac{A_2 u_{M-1}^{n+1} + h\phi_{M-1}^{n+1}}{A_2 + hB_2},$$

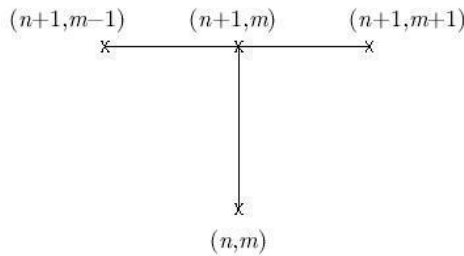
где  $u_1^{n+1}$  и  $u_{M-1}^{n+1}$  уже вычислены ранее.

#### Неявные разностные схемы.

При реализации явной схемы вычислений, данные на следующем слое по времени находятся непосредственно из данных на предыдущем слое без решения алгебраических систем уравнений. Если же на верхнем временном слое для определения значений сеточной функции необходимо решать систему алгебраических уравнений, то такая схема называется неявной. Простейшая неявная разностная схема имеет следующий вид (при значении  $a = 1$ ):

$$\frac{u_m^{n+1} - u_m^n}{\tau} = \frac{u_{m-1}^n - 2u_m^n + u_{m+1}^n}{h^2} + f_m^n.$$

Шаблон такой схемы можно изобразить следующим образом:



Неявная схема с краевыми и начальными условиями будет определяться следующими уравнениями:

$$\begin{aligned} (A_1 + hB_1)u_0^{n+1} - A_1 u_1^{n+1} &= h\phi_1^{n+1}, m = 0 \\ \frac{\tau}{h^2} u_{m-1}^{n+1} - (1 - 2\frac{\tau}{h^2})u_m^{n+1} + \frac{\tau}{h^2} u_{m+1}^{n+1} &= u_m^n + \tau f_m^n, m = 1, 2, \dots, M-1 \\ (A_2 + hB_2)u_M^{n+1} - A_2 u_{M-1}^{n+1} &= h\phi_{M-1}^{n+1}, m = M. \end{aligned}$$

Данные уравнения можно записать в матричной форме:

$$\begin{pmatrix} (A_1 + hB_1) & -A_1 & 0 & \dots & 0 \\ \frac{\tau}{h^2} & -(1 - 2\frac{\tau}{h^2}) & \frac{\tau}{h^2} & \dots & 0 \\ 0 & \frac{\tau}{h^2} & -(1 - 2\frac{\tau}{h^2}) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & -A_2 & (A_2 + hB_2) \end{pmatrix} \begin{pmatrix} u_0^{(n+1)} \\ u_1^{(n+1)} \\ u_2^{(n+1)} \\ \dots \\ u_M^{(n+1)} \end{pmatrix} = \begin{pmatrix} h\phi_1^{n+1} \\ u_1^n + \tau f_1^n \\ u_2^n + \tau f_2^n \\ \dots \\ h\phi_2^{n+1} \end{pmatrix}$$

Решение такой системы находится последовательно снизу вверх. Для нахождения решения на следующем шаге во времени следует вновь решить аналогичную систему. Алгоритм решения такой системы линейных алгебраических уравнений называется прогонкой.

### **Устойчивость и сходимость линейных разностных схем.**

Пусть  $Lu = F$  и  $L_\tau u_\tau = F_\tau$  - операторные обозначения исходной дифференциальной и аппроксимирующей ее разностной задачи. Здесь  $L, L_\tau$  дифференциальный и разностный операторы, соответственно;  $u \in \Omega, u_\tau \in \Omega_\tau$  - решения дифференциального и разностного уравнений;  $F \in w, F_\tau \in w_\tau$  - правая часть исходного уравнения и ее проекция на расчетную сетку. Индекс  $\tau$  в этой операторной записи указывает на всю совокупность сеточных параметров (то есть, по сути, для дискретной задачи указывается некоторое параметрическое семейство задач).

**Определение.** Говорят, что решение  $u_\tau$  сходится к  $\bar{U}_\tau$  при  $\tau \rightarrow 0$ , если  $\|u_\tau - \bar{U}_\tau\| \rightarrow 0$ . Здесь  $\bar{U}_\tau$  - проекция точного решения соответствующей дифференциальной системы  $Lu = F$  на разностную сетку  $\tau$ . Причем, если имеет место оценка  $\|u_\tau - \bar{U}_\tau\| \leq c\tau^p$ , где  $c$  - некоторая постоянная, не зависящая от параметра  $\tau$ , то сходимость имеет порядок  $p$ .

**Определение.** Говорят, что разностная задача является устойчивой: если из соотношений  $L_\tau u_\tau - F_\tau = \xi_\tau$  и  $L_\tau v_\tau - F_\tau = \eta_\tau$  следует, в смысле выбранной нормы,  $\|u_\tau - v_\tau\| \leq c_2(\|\xi_\tau\| + \|\eta_\tau\|)$ ; причем эта оценка равномерная, то есть постоянная  $c_2$  не зависит параметра  $\tau$ . Или, если, при любой правой части  $F_\tau$ , задача имеет единственное решение  $u_\tau$ , причем  $\|u_\tau\| \leq c_3 \|F_\tau\|$ , и данная оценка равномерная по сеточным параметрам.

**Теорема (Лакса – Рябенского).** Решение линейной разностной задачи сходится к решению дифференциальной, если разностная задача устойчива и аппроксимирует дифференциальную задачу на ее решении. При этом порядок аппроксимации совпадает с порядком сходимости.

*Программные пакеты для решения уравнений в частных производных и моделирования систем с распределенными параметрами.*

*Пакет PDE Toolbox (Partial Differential Equations) MatLab.*

Пакет реализован в виде интегрированной среды основных инструментальных средств и функций численных методов решения уравнений с частными производными для основных задач математической физики и двумерной расчетной области. В пакете реализованы проекционные процедуры с конечными элементами на базе метода Галеркина, решение скалярных краевых задач

эллиптического, параболического и гиперболического типов. Осуществляется визуальный принцип задания как стационарных, так и нестационарных краевых условий. В пакете предусмотрены развитые графические средства отображения двух и трех - трехмерных решений.

#### *Пакет FlexPDE.*

Программа FlexPDE предназначена для построения сценарных моделей решения дифференциальных уравнения в частных производных методом конечных элементов. По сценарию, написанному пользователем, FlexPDE производит операции, необходимые для того, чтобы преобразовать описание систему в УЧП в модель для расчета методом конечных элементов. Таким образом, пользователь имеет возможность готовить для решения начальную краевую задачу в своих обозначениях. Число решаемых уравнений в системе ограничивается только мощностью компьютера. Решаемые уравнения могут быть как линейными, так и нелинейными.

#### *Пакет ANSYS.*

Пакет ANSYS – это совокупность различных программных модулей, предназначенных для проведения конечно-элементного анализа в различных областях инженерной деятельности. Данный пакет имеет тесные связи с программными пакетами геометрического проектирования, например, таким как AutoCad. В состав пакета входят следующие модули:

Multiphysics – программа, обеспечивающая расчетные возможности для модулей различных инженерных дисциплин;

Mechanical – программа конструкционного (прочностного) и термического анализа;

Structural – программа прочностного анализа, включая геометрические и физически нелинейные задачи;

Emag – электромагнитные расчеты;

CFX - модуль анализа гидро- и газодинамических процессов, многофазных потоков;

LS-DYNA – программа высоконелинейных расчетов, предназначенная для численного моделирования взаимодействий многих тел;

DesignSpace – расчеты статической прочности, устойчивости по Эйлеру, теплопроводность с поддержкой сборок

Rigid Dynamics – программа моделирования механических систем, соединенных шарнирными соединениями и т.д.

#### **Методы решения линейных интегральных уравнений.**

Рассмотрим задачу оценки входа стационарного распределенного блока по наблюдениям за его выходом. Как известно, если параметры распределенного объекта не зависят от времени, то соответствующий этому объекту распределенный блок называется стационарным распределенным блоком. В этом случае импульсная распределенная переходная функция стационарного блока зависит только от трех независимых аргументов и имеет вид:  $G(x, t, \xi, \tau) = G(x, \xi, t - \tau)$ . Динамика такого блока инвариантна относительно сдвигов во времени. Поэтому за начало отсчета времени можно принять нулевой момент времени, то есть  $t \geq 0$ . Интегральный оператор, в этом случае, описывается соотношением:

$$Q(x, t) = \int_0^t \int_D G(x, \xi, t - \tau) \cdot u(\xi, \tau) \cdot d\xi \cdot d\tau,$$

где  $Q(x, t)$  - выходной сигнал распределенного блока,  $u(\xi, \tau)$  - входной сигнал блока в пространственной точке  $\xi$  в момент времени  $\tau$ ,  $G(x, \xi, t)$  - функция Грина. Таким образом, для решения обратной задачи нахождения входа  $u(x, t)$  по наблюдаемому выходу  $Q(x, t)$  необходимо решить соответствующее интегральное уравнение.

В общем случае, интегральным уравнением называется уравнение, в котором неизвестная функция стоит под знаком интеграла. Например, уравнение относительно функции  $u(x)$  вида:

$$\int_a^b G(x, \xi, u(\xi)) d\xi = F(x, u(x)),$$

является стационарным интегральным уравнением и, в общем случае, нелинейным.

*Пример.*

Рассмотрим неоднородное уравнение с частными производными с начальными и граничными условиями вида:

$$u'_t = a^2 \cdot u''_{xx} + f(x, t, u), \quad u|_t = 0, \quad u|_\Gamma = 0, \quad x \in D.$$

Отсюда можно записать следующее решение:

$$u(x, t) = \int_0^t \int_D G(x, \xi, t - \tau) \cdot f(\xi, \tau, u(\xi, \tau)) \cdot d\xi \cdot d\tau,$$

где  $G(x, \xi, t)$  - искомая функция Грина данной задачи.

Ограничимся, в дальнейшем рассмотрении, только линейными задачами. Типичным представителем такого класса является интегральное уравнение Фредгольма второго рода:

$$u(x) - \lambda \cdot \int_a^b K(x, \xi) \cdot u(\xi) \cdot d\xi = f(x), \quad x, \xi \in [a, b].$$

Ядро  $K$  этого уравнения задано в квадрате  $[a, b] \times [a, b]$ . Если ядро  $K$  отлично от нуля только в половине указанного квадрата – треугольнике  $0 \leq \xi \leq x$ , то данное уравнение называется уравнением Вольтерра второго рода.

*Метод последовательных приближений.*

Запишем уравнение Фредгольма второго рода в виде:

$$u(x) = \lambda \cdot Au(x) + f(x),$$

где  $Au(x) = \int_a^b K(x, \xi) \cdot u(\xi) \cdot d\xi.$

Тогда, при условии ограниченности ядра  $K$  и малости  $\lambda$ , решение интегрального уравнения Фредгольма второго рода может быть найдено методом последовательных приближений. То есть последовательным вычислением с помощью следующих соотношений:

$$y^{(k+1)}(x) = \lambda \cdot Ay^{(k)}(x) + f(x).$$

Запишем следующие соотношения:

$$y^{(k+1)}(x) = \lambda \cdot \int_a^b K(x, \xi) \cdot y^{(k)}(\xi) \cdot d\xi + f(x), \quad z^{(k+1)}(x) = u(x) - y^{(k+1)}(x).$$

Тогда для последовательности  $\{z^{(k)}(x)\}$  можно записать уравнение:

$$z^{(k+1)}(x) = \lambda \cdot \int_a^b K(x, \xi) \cdot z^{(k)}(\xi) \cdot d\xi.$$



Отсюда найдем следующую оценку:

$$\|z^{(k)}(x)\|_C \leq |\lambda| \cdot \|K(x, \xi)\|_C \cdot (b-a).$$

Следовательно, если  $q = |\lambda| \cdot \|K(x, \xi)\|_C \cdot (b-a) \in (0,1)$ , то оператор  $A$  является сжимающим.

Можно показать, что метод последовательных приближений эквивалентен представлению решения в виде ряда по степеням  $\lambda$ :

$$u(x) = \sum_{k=0}^{\infty} \lambda^k \cdot u_k(x).$$

Если подставить такое решение в исходное уравнение и приравнять коэффициенты при равных степенях  $\lambda$ , то получим следующие соотношения:

$$u_0(x) = f(x); \quad u_{k+1}(x) = \int_a^b K(x, \xi) \cdot u_k(\xi) \cdot d\xi, \quad k = 0, 1, \dots$$

$$y^{(k)}(x) = \sum_{i=0}^k \lambda^i \cdot u_i(x)$$

*Пример.*

Рассмотрим уравнение:  $u(x) - \lambda \cdot \int_0^{\frac{\pi}{2}} (\sin(x) \cdot \cos(\xi)) \cdot u(\xi) \cdot d\xi = 1$ . В соответствии с описанным

алгоритмом найдем:

$$u_0(x) = 1; \quad u_1(x) = \sin(x) \cdot \int_0^{\frac{\pi}{2}} \cos(\xi) \cdot d\xi = \sin(x);$$

$$u_2(x) = \sin(x) \cdot \int_0^{\frac{\pi}{2}} \cos(\xi) \cdot \sin(\xi) \cdot d\xi = \frac{1}{2} \cdot \sin(x);$$

$$u_3(x) = \frac{1}{4} \cdot \sin(x);$$

$$\dots$$

Отсюда можно записать:

$$u(x) = 1 + \left(\lambda + \frac{1}{2} \cdot \lambda^2 + \frac{1}{4} \cdot \lambda^3 + \dots\right) \cdot \sin(x) = 1 + \frac{2\lambda}{2-\lambda} \cdot \sin(x)$$

Ряд для функции  $u(x)$  будет сходиться при  $|\lambda| < 2$ .

*Разностный метод численного решения.*

Другой способ численного решения интегрального уравнения заключается во введении сетки  $\Omega_h$  на отрезке  $[a, b]$ , замене интеграла квадратурной формулой и решении полученной системы алгебраических уравнений. Этот простейший численный метод позволяет получать решение одномерных задач с хорошей точностью.

Рассмотрим линейную квадратурную формулу с узлами в точках  $x_i$  и весами  $c_i$ :

$$\int_a^b \Phi(\xi) \cdot d\xi \approx \sum_{i=1}^N c_i \cdot \Phi(x_i).$$

Введем в квадрате  $[a \leq x \leq b, a \leq \xi \leq b]$  сетку  $\Omega_{h \times h}$ . Тогда интегральное уравнение вида:

$$\int_a^b K(x, \xi, u(\xi)) \cdot d\xi = F(x, u(x)), \quad x \in [a, b],$$

можно заменить следующим приближением в виде системы алгебраических уравнений:

$$\sum_{i=1}^N c_i \cdot K(x_j, \xi_i, y(\xi_i)) = F(x_j, y(x_j)), \quad j = 1, 2, \dots, N.$$

Здесь  $y(x_j) = y_j$  - искомое сеточное решение. Полученная система алгебраических уравнений является, в общем случае, нелинейной.

Для линейного уравнения Фредгольма второго рода такая приближенная система примет вид:

$$y_j - \lambda \cdot \sum_{i=1}^N c_i \cdot K(x_j, \xi_i) \cdot y_i = f_j, \quad j = 1, 2, \dots, N$$

Данная система алгебраических уравнений является линейной и имеет заполненную матрицу. На практике сходимость решения обычно проверяют путем последовательных расчетов на сгущающихся сетках.

*Метод Галеркина или метод моментов.*

С помощью данного метода приближенное решение  $y(x)$  интегрального уравнения

$$\text{Фредгольма } u(x) - \lambda \cdot \int_a^b K(x, \xi) \cdot u(\xi) \cdot d\xi = f(x), \quad x, \xi \in [a, b]$$

ищется в виде разложения по некоторой полной системе функций  $\{\varphi_i(x)\}$ :

$$y(x) = f(x) + \lambda \cdot \sum_{i=1}^N c_i \cdot \varphi_i(x).$$

Подставим приближенное решение в интегральное уравнение и потребуем, чтобы невязка была ортогональна всем функциям  $\varphi_i(x)$ , входящим в состав приближенного решения. Тогда получим следующую систему линейных алгебраических уравнений **/Калиткин/** :

$$\sum_{j=1}^N a_{ij} \cdot c_j = b_i, \quad i = 1, 2, \dots, N, \text{ где}$$

$$a_{ij} = \int_a^b \varphi_i(\xi) \cdot \varphi_j(\xi) \cdot d\xi - \lambda \cdot \int_a^b \int_a^b K(x, \xi) \cdot \varphi_i(x) \cdot \varphi_j(\xi) \cdot dx \cdot d\xi,$$

$$b_i = \int_a^b \int_a^b K(x, \xi) \cdot \varphi_i(x) \cdot f(\xi) \cdot dx \cdot d\xi.$$

Если система функций  $\{\varphi_i(x)\}$  ортогональна, то есть  $\int_a^b \varphi_i(\xi) \cdot \varphi_j(\xi) \cdot d\xi = 0, i \neq j$ , то

метод эквивалентен приближенной замене ядра на специальное вырожденное ядро вида:

$$K(x, \xi) \approx \sum_{i=1}^N \varphi_i(x) \cdot \phi_i(\xi),$$

$$\text{где } \phi_i(\xi) = \int_a^b K(x, \xi) \cdot \varphi_i(x) \cdot dx.$$

### **Методы интерполирования функций.**

При получении численного решения задачи в виде некоторой функции  $y = f(x)$ , в большинстве случаев, она представляется в виде отдельных значений  $f_i = f(x_i)$  в точках  $x_i$ , распределенных на некоторой сетке  $\Omega_h$ . Данная сетка представляет собой дискретное множество значений аргумента  $x$  из некоторого множества  $\Omega$ ;  $x \in \Omega$ .

Довольно часто пользователю необходимо выполнить процедуру восстановления некоторого приближения  $\hat{f}(x)$  исходной функции  $y = f(x)$  по множеству значений  $\{f_i\}$ . Такую процедуру называют интерполяцией. При этом необходимо знать, какова норма ошибки  $\|f - \hat{f}\|$ .

Если же необходимо оценить значения восстанавливаемой функции на множестве  $\Omega_e \supset \Omega$ , которое превышает исходное множество заданных значений аргументов  $\Omega$ , то такую процедуру называют экстраполяцией.

*Кусочно – линейная интерполяция.*

Потребуем, чтобы функция  $\hat{f}(x)$  была линейна на каждом отрезке  $[x_{i-1}, x_i]; i = 1, 2, \dots, n$ . Здесь предполагается, что  $\hat{f}, f, x \in \mathfrak{R}$ . Тогда можно записать следующее равенство:

$$\hat{f}(x) = \frac{1}{x_i - x_{i-1}} \cdot ((x - x_{i-1}) \cdot f_i - (x - x_i) \cdot f_{i-1}); \quad x \in [x_{i-1}, x_i].$$

Введем в обращение следующие функции:

$$\varphi_0(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_0}; & x \in [x_0, x_1] \\ 0; & x \geq x_1 \end{cases}, \quad \varphi_n(x) = \begin{cases} \frac{x - x_{n-1}}{x_n - x_{n-1}}; & x \in [x_{n-1}, x_n] \\ 0; & x \leq x_{n-1} \end{cases},$$

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}; & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}; & x \in [x_i, x_{i+1}] \\ 0; & x \notin [x_{i-1}, x_{i+1}] \end{cases}; i = 1, 2, \dots, n-1$$

Носители функций  $\varphi_i(x)$ , то есть множества значений аргумента, на которых эти функции отличны от нуля, называются конечными элементами, а сами функции  $\varphi_i(x)$  называются базисными функциями  $i$ -го конечного элемента.

Таким образом, для оценки приближения  $\hat{f}(x)$  можно записать следующую формулу:

$$\hat{f}(x) = \sum_{i=0}^n \varphi_i(x) \cdot f_i.$$

**Теорема.** Пусть  $f(x)$  - непрерывная функция, заданная на отрезке  $[a, b]$  и удовлетворяющая условию Липшица вида:

$$|f(x') - f(x'')| \leq q \cdot |x' - x''|; \quad x', x'' \in [a, b].$$

Тогда будет справедлива следующая оценка:

$$|f(x) - \hat{f}(x)| \leq \frac{1}{2} \cdot q \cdot h; \quad x \in [a, b],$$

где  $h = \max_{i=1,2,\dots,n} |x_i - x_{i-1}|$ .

**Доказательство.**

Пусть  $x \in [x_{i-1}, x_i]$ ,  $h_i = x_i - x_{i-1}$ ,  $x = x_{i-1} + \alpha \cdot h_i$ ,  $\alpha = (x - x_{i-1}) / h_i$ ,  $\alpha \in (0, 1)$ . Тогда имеет место следующее равенство:

$$\hat{f}(x) = \alpha \cdot f_i + (1 - \alpha) \cdot f_{i-1}.$$

Отсюда можно записать следующие соотношения:

$$\begin{aligned} |f(x) - \hat{f}(x)| &= |\alpha \cdot f + (1 - \alpha) \cdot f - \alpha \cdot f_i - (1 - \alpha) \cdot f_{i-1}| \leq \\ &\leq \alpha \cdot |f - f_i| + (1 - \alpha) \cdot |f - f_{i-1}| \leq \alpha \cdot q \cdot |x - x_i| + (1 - \alpha) \cdot q \cdot |x - x_{i-1}| = \\ &= \alpha \cdot q \cdot (1 - \alpha) \cdot h_i + (1 - \alpha) \cdot q \cdot \alpha \cdot h_i = 2q \cdot \alpha \cdot (1 - \alpha) \cdot h_i \leq \frac{1}{2} \cdot q \cdot h_i \end{aligned}$$

Отсюда получим:

$$\|f(x) - \hat{f}(x)\|_C \leq \frac{1}{2} \cdot q \cdot h; \quad x \in [a, b]$$

Следует отметить, что эта оценка является неулучшаемой в данном классе функций.

**Пример.**

Найдем наилучшую (в среднеквадратичном смысле) оценку функции  $f(x)$ , заданную тремя значениями  $f_0, f_1, f_2$  в точках  $x_0, x_1, x_2$ . Примем, что  $x_2 - x_1 = x_1 - x_0 = h$ . В качестве формы базисных функций примем следующую аппроксимирующую функцию  $\varphi(x) = a + b \cdot (x - x_1)$ . Тогда неизвестные коэффициенты  $a, b$  найдем из условия:

$$J = \sum_{i=0}^2 (\varphi(x_i) - f_i)^2 \rightarrow \min.$$

Решение задачи имеет вид:

$$a = \frac{1}{3} \cdot (f_0 + f_1 + f_2), \quad b = \frac{1}{2h} \cdot (f_2 - f_0).$$

При этом минимальное среднеквадратичное отклонение равно:

$$J = \frac{1}{6} \cdot h^4 \cdot (f''_{xx})^2,$$

где  $f''_{xx} = \frac{1}{h^2} \cdot (f_0 - 2f_1 + f_2)$  - разностная производная 2 – го порядка.

В общем случае, задачу приближения функции  $f(x)$ , заданной дискретными отсчетами  $\{f_i\}, i=1,2,\dots,n$  на некотором отрезке  $[a,b]$ , можно представить в виде нахождения неизвестных параметров  $c_i$  при заданных базисных функциях  $\{\varphi_i(x)\}, i=1,2,\dots,n$ . При этом оценка  $\hat{f}(x)$  задается суммой вида:

$$\hat{f}(x) = \sum_{i=0}^n c_i \cdot \varphi_i(x).$$

Очевидно, что в пространстве  $L_2$ , при задании базовых функций в виде тригонометрических функций, минимальной ошибке будет соответствовать частичная сумма ряда Фурье.

*Полиномиальная интерполяция.*

Теоретической основой полиномиальной интерполяции является следующая теорема Вейерштрасса.

**Теорема /Галанин/.** Для любой непрерывной функции  $f(x)$ , заданной на отрезке  $[a,b]$ , существует полином  $P_n(x)$ , приближающий функцию  $f(x)$  с любой наперед заданной точностью, то есть:

$$\forall \varepsilon > 0, \exists P_n(x): \|f(x) - P_n(x)\|_C < \varepsilon$$

Рассмотрим следующий полином степени  $n$  вида:

$$P_n(x) = a_0 + a_1 \cdot x + \dots + a_n \cdot x^n,$$

проходящий через точки  $(x_i, f_i), i=0,1,\dots,n$ . Тогда можно записать следующие уравнения для определения коэффициентов этого полинома:

$$a_0 + \sum_{k=1}^n a_k \cdot x_i^k = f_i; i=0,1,\dots,n.$$

Можно показать, что указанный интерполяционный полином существует и является единственным. Заметим, что полиномиальная интерполяция представляет собой частный случай

приближения функции суммами вида  $\sum_{i=0}^n c_i \cdot \varphi_i(x)$ , где  $\varphi_i(x) = x^i$ . Поэтому, в общем случае,

обычно интерполяцию рассматривают с помощью системы функций Чебышева, которая состоит из  $n+1$  функций вида  $\varphi_i(x)$ , любая линейная комбинация которых не может иметь  $n+1$  корней на отрезке интерполирования. Известно, что система функций  $1, x, x^2, \dots, x^n$  образует систему Чебышева на любом отрезке.

*Интерполяционный полином в форме Лагранжа.*

Рассмотрим интерполяционный полином в форме Лагранжа, который имеет вид:

$$L_n(x) = \sum_{k=0}^n f_k \cdot \prod_{\substack{i=0 \\ i \neq k}}^n \frac{(x - x_i)}{(x_k - x_i)}.$$

Введем следующее обозначение:  $\omega(x) = \prod_{i=0}^n (x - x_i)$ . Тогда можно определить базисные полиномы в следующем виде:

$$\varphi_k(x) = \prod_{\substack{i=0 \\ i \neq k}}^n \frac{(x - x_i)}{(x_k - x_i)} = \frac{\omega(x)}{(x - x_k) \cdot \omega'_x(x_k)}.$$

Тогда полином Лагранжа может быть записан в виде:

$$L_n(x) = \sum_{k=0}^n f_k \cdot \varphi_k(x).$$

Остаточный член  $r_n(x) = f(x) - L_n(x)$  интерполяционного многочлена определяется следующим соотношением:

$$r_n(x) = \frac{1}{(n+1)!} \cdot \omega(x) \cdot f_x^{(n+1)}(\xi); \quad \xi \in [a, b]$$

Если  $\|f_x^{(n+1)}(x)\|_C = M_{n+1}$ , а  $x_i - x_{i-1} = h; i = 1, 2, \dots, n$  то тогда можно записать следующую оценку:

$$|f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)} \cdot h^{n+1}.$$

*Интерполяционный полином в форме Ньютона.*

Введем в рассмотрение понятие разделенных разностей.

Будем говорить, что разделенная разность нулевого порядка совпадает со значениями функции  $f(x_i)$  в точках  $x_i, i = 0, 1, \dots, n$ .

Разделенной разностью первого порядка назовем выражение:

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_j - x_i}.$$

Разделенной разностью второго порядка:

$$f(x_i, x_j, x_l) = \frac{f(x_j, x_l) - f(x_i, x_j)}{x_l - x_i}$$

Разность  $k$ -го порядка имеет вид:

$$f(x_1, x_2, \dots, x_{k+1}) = \frac{f(x_2, x_3, \dots, x_{k+1}) - f(x_1, x_2, \dots, x_k)}{x_{k+1} - x_1}.$$

Справедливо равенство:

$$f(x_1, x_2, \dots, x_k) = \sum_{j=1}^k \frac{f(x_j)}{\prod_{\substack{i=1 \\ i \neq j}}^k (x_j - x_i)}$$

Отсюда в частности следует, что значение разделенной разности  $f(x_1, x_2, \dots, x_k)$  не зависит от порядка следования аргументов.

**Теорема.** Интерполяционный полином  $P_n(x)$  может быть записан в форме Ньютона в виде:

$$P_n(x) = f(x_0) - (x - x_0) \cdot f(x_0, x_1) + (x - x_0) \cdot (x - x_1) \cdot f(x_0, x_1, x_2) + \dots \\ + (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{n-1}) \cdot f(x_0, x_1, \dots, x_n)$$

**Интерполяционный полином Эрмита.**

Пусть задано  $m + 1$  точка отрезка  $[a, b]$ :  $a = x_0 < x_1 < \dots < x_m = b$ , в каждой из которых известно  $n_i$  величин  $f^{(0)}(x_i), f^{(1)}(x_i), \dots, f^{(n_i-1)}(x_i)$ . Значение  $n_i$  называется кратностью узла. Построим интерполяционный полином, проходящий через точки  $(x_i, f_i)$  так, что в каждой точке  $x_i$  он и его производные до порядка  $n_i - 1$  принимали заданные значения. Такой полином  $H_n(x)$

имеет степень  $n = \sum_{i=1}^m n_i - 1$  и называется интерполяционным полиномом Эрмита.

Полином Эрмита должен удовлетворять системе из  $n + 1$  уравнения вида:

$$H_n^{(k)}(x_i) = f^{(k)}(x_i); k = 0, 1, \dots, n_i - 1; i = 0, 1, \dots, m.$$

Отсюда видно, что коэффициенты полинома линейно выражаются через значения  $f^{(k)}(x_i)$ .

Поэтому полином  $H_n(x)$  можно искать с помощью следующего соотношения:

$$H_n(x) = \sum_{i=0}^m \sum_{k=0}^{n_i-1} c_{ki}(x) \cdot f^{(k)}(x_i).$$

Простейшим примером полинома Эрмита является полином Тейлора, построенный в одной точке по значению функции и значениям ее первых производных до  $n$  порядка включительно, и, который имеет вид:

$$T_n(x) = \sum_{k=0}^n \frac{1}{k!} \cdot f^{(k)}(x_0) \cdot (x - x_0)^k.$$

Очевидно, что точность приближения с помощью такого полинома может быть оценена с помощью соотношения:

$$f(x) - T_n(x) = \frac{1}{(n+1)!} \cdot f^{(n+1)}(\xi) \cdot (x - x_0)^{n+1},$$

где  $\xi \in (x_0, x)$  (или  $\xi \in (x, x_0)$ ).

**Устойчивость интерполяционного полинома.**

Пусть для функции  $f(x)$  построен интерполяционный полином  $L_n(x)$  на некоторой сетке.

Рассмотрим ситуацию, когда функция  $f(x)$  не имеет  $n + 1$  ограниченной производной.

Было показано, что полиномы, интерполирующие функцию  $f(x) = |x|$  на равномерной сетке на отрезке  $[-1, 1]$ , таковы, что:

$$\|f(x) - L_n(x)\|_C \xrightarrow{n \rightarrow \infty} \infty$$

При этом полином  $L_n(x)$  не сходится к функции  $f(x)$  ни в одной точке отрезка  $[-1, 1]$ , за исключением точек  $-1, 0, 1$ .

В общем случае верна теорема Фабера, которая утверждает, что для любой последовательности сеток  $\{\Omega_n\}, \Omega_n \in [a, b]$ , существует непрерывная на отрезке  $[a, b]$  функция  $f(x)$  такая, что  $\{L_n(x)\} \not\rightarrow f(x)$ .

Есть и обратная теорема Марцинкевича, которая утверждает, что для любой непрерывной на отрезке  $[a, b]$  функции  $f(x)$  существует последовательность сеток  $\{\Omega_n\}$ , для которой  $\{L_n(x)\} \rightarrow f(x)$ .

Однако, такие сетки необходимо строить для каждой функции отдельно /Калиткин/.

*Пример (Рунге).*

Рассмотрим следующую функцию  $f(x) = \frac{1}{1 + 25x^2}$ . Глобальная интерполяция на равномерной сетке дает расходимость на участках  $|x| \in (0.73, 1]$  при бесконечном увеличении числа точек разбиения. Это связано с ростом нормы производной данной функции при увеличении порядка производной.

*Методы сплайн – интерполяции.*

Чтобы избавиться от недостатков глобальной полиномиальной интерполяции часто применяется кусочно – полиномиальная интерполяция.

Пусть задана сетка на отрезке  $[a, b]$  вида:  $a = x_0 < x_1 < \dots < x_n = b$ . Построим полином третьей степени  $S_3(x)$ , на каждом отрезке  $[x_{i-1}, x_i]$  длиной  $h_i = x_i - x_{i-1}$ , следующего вида:

$$S_3(x) = a_i + b_i \cdot (x - x_{i-1}) + c_i \cdot (x - x_{i-1})^2 + d_i \cdot (x - x_{i-1})^3.$$

Потребуем, чтобы полином  $S_3(x)$  принимал заданные значения на концах отрезка  $[x_{i-1}, x_i]$ :

$$\begin{aligned} S_3(x_{i-1}) &= y_{i-1} = a_i \\ S_3(x_i) &= y_i = a_i + b_i \cdot h_i + c_i \cdot h_i^2 + d_i \cdot h_i^3 \end{aligned}$$

Очевидно, что число неизвестных параметров (коэффициентов полинома  $S_3(x)$ ) превышает в два раза число уравнений. Поэтому введем дополнительные условия – потребуем непрерывности первой и второй производных во внутренних точках сетки:

$$\begin{aligned} S'_3(x) &= b_i + 2c_i \cdot (x - x_{i-1}) + 3d_i \cdot (x - x_{i-1})^2 \\ S''_3(x) &= 2c_i + 6d_i \cdot (x - x_{i-1}) \end{aligned}$$

Условие непрерывности позволяет получить следующие уравнения:

$$\begin{aligned} b_i + 2c_i \cdot h_i + 3d_i \cdot h_i^2 &= b_{i+1}, \quad i = 1, 2, \dots, n-1 \\ 2c_i + 6d_i \cdot h_i &= 2c_{i+1} \end{aligned}$$

Таким образом, получим  $4n - 2$  уравнений для  $4n$  неизвестных. Введем еще два уравнения, полагая  $S''_3(x)$  равной нулю на концах отрезка  $[a, b]$ :

$$\begin{aligned} 2c_1 &= 0 \\ 2c_n + 6d_n \cdot h_n &= c_{n+1} = 0 \end{aligned}$$

Исключим из полученных уравнений параметры  $a_i, b_i, d_i$  с помощью следующих соотношений:



$$d_n = -\frac{c_n}{3h_n}, \quad d_i = \frac{c_{i+1} - c_i}{3h_i},$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - c_i \cdot h_i - \frac{(c_{i+1} - c_i) \cdot h_i}{3},$$

$$b_n = \frac{y_n - y_{n-1}}{h_n} - c_n \cdot h_n + \frac{c_n \cdot h_n}{3}$$

Тогда получим следующие уравнения для нахождения коэффициентов  $c_i$ :

$$c_{i-1} \cdot h_{i-1} + (2h_{i-1} + 2h_i) \cdot c_i + c_{i+1} \cdot h_i = 3\left(\frac{1}{h_i}(y_i - y_{i-1}) - \frac{1}{h_{i-1}}(y_{i-1} - y_{i-2})\right),$$

где  $c_1 = c_{n+1} = 0$ .

Таким образом, получена трехдиагональная систем линейных уравнений, которая может быть легко решена. После нахождения параметров  $c_i$  можно далее вычислить и другие параметры полинома интерполяции  $S_3(x)$ , который называется интерполяционным кубическим сплайном.

**Теорема.** Пусть  $f(x) \in C^4(a, b)$ ,  $M_4 = \|f^{(4)}(x)\|_C$  и  $S_3(x)$  - кубический сплайн. Тогда выполняются следующие соотношения:

$$\begin{aligned} \|f(x) - S_3(x)\|_C &\leq C_1 \cdot M_4 \cdot h^4, \\ \|f'(x) - S_3'(x)\|_C &\leq C_2 \cdot M_4 \cdot h^3, \\ \|f''(x) - S_3''(x)\|_C &\leq C_3 \cdot M_4 \cdot h^2 \end{aligned}$$

Отсюда следует, что для указанного класса функций сплайн и его первые две производные обладают свойствами сходимости. То есть сплайн  $S_3(x)$  можно дважды дифференцировать. Такой сплайн является глобальным, так как его коэффициенты определяются данными на всей сетке сразу.

В общем случае, сплайном степени  $m$  называется локально заданная функция  $S_m(x)$ , заданная на отрезке  $[a, b]$  с указанными точками разбиения, которая непрерывна вместе со своими производными вплоть до некоторого порядка  $p$ , и на каждом отрезке разбиения  $[x_{i-1}, x_i]$  совпадает с алгебраическим полиномом  $P_{m,i}(x)$  степени  $m$ . Разность  $m - p$  называется дефектом сплайна. Как видно, кубический сплайн обладает дефектом, равным единице.

Если кубический сплайн имеет дефект равный двум, то вторая производная, в общем случае, не будет непрерывной. Поэтому необходимо использовать другие дополнительные условия. В частности, это могут быть значения производной в точках разбиения:  $y'_i = s_i = S'_3(x_i)$ . Тогда выражение для такого сплайна может быть записано на отрезке  $[x_{i-1}, x_i]$  в явном виде:

$$S_3(x) = \frac{(x - x_i)^2 \cdot (2(x - x_{i-1}) + h_i)}{h_i^3} \cdot y_{i-1} + \frac{(x - x_{i-1})^2 \cdot (2(x_i - x) + h_i)}{h_i^3} \cdot y_i +$$

$$+ \frac{(x - x_i)^2 \cdot (x - x_{i-1})}{h_i^2} \cdot s_{i-1} + \frac{(x - x_{i-1})^2 \cdot (x - x_i)}{h_i^2} \cdot s_i$$

Очевидно, что такой сплайн является локальным и совпадает с интерполяционным полиномом Эрмита на каждом отрезке  $[x_{i-1}, x_i]$ .

*B – сплайны.*

Данные локальные функции представляют собой конструкции, связанные с финитными функциями, то есть гладкими функциями, которые определяются на всей действительной оси, но отличны от нуля лишь на некотором конечном интервале (носителе) /Завьялов/.

Расширим сетку  $\Omega_h \in [a, b]$ ,  $a = x_0 < x_1 < \dots < x_N = b$ , добавив дополнительно точки:

$$x_{-n} < \dots < x_{-1} < a; \quad b < x_{N+1} < \dots < x_{N+n}.$$

Далее определим базовую функцию:

$$\varphi_n(x, x_i) = (-1)^{n+1} \cdot (n+1) \cdot (x - t)_+^n,$$

где  $x_+^n = \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases}$ . Построим для этой базовой функции разделенные разности  $(n+1)$ -го порядка

по значениям аргумента  $t = x_i, \dots, x_{i+n+1}$ . В результате получим функции переменной  $x$ :

$$B_n^i(x) = \varphi_n(x; x_i, \dots, x_{i+n+1}), i = -n, \dots, N-1$$

Так как для разделенной разности  $(n+1)$ -го порядка от функции  $f(x)$  по точкам  $x_i, \dots, x_{i+n+1}$  справедливо равенство:

$$f(x_i, \dots, x_{i+n+1}) = \sum_{k=i}^{i+n+1} \frac{f(x_k)}{\omega'_{n+1,i}(x_k)},$$

где  $\omega_{n+1,i}(x) = \prod_{j=i}^{i+n+1} (x - x_j)$ , то получим:

$$B_n^i(x) = (-1)^{n+1} \cdot (n+1) \cdot \sum_{k=i}^{i+n+1} \frac{(x - x_k)_+^n}{\omega'_{n+1,i}(x_k)}, i = -n, \dots, N-1.$$

Используя тождество:

$$(x - t)_+^n = (x - t)^n + (-1)^{n+1} \cdot (t - x)_+^n,$$

можно получить следующую формулу:

$$B_n^i(x) = (n+1) \cdot \sum_{k=i}^{i+n+1} \frac{(x_k - x)_+^n}{\omega'_{n+1,i}(x_k)}, i = -n, \dots, N-1.$$

Очевидно, что функции  $B_n^i(x), i = -n, \dots, N-1$  являются сплайном степени  $n$  с дефектом один на сетке узлов  $x_i, \dots, x_{i+n+1}$  и обладают следующими свойствами:

$$B_n^i(x) = \begin{cases} > 0, & x \in (x_i, x_{i+n+1}) \\ \equiv 0, & x \notin (x_i, x_{i+n+1}) \end{cases}, \quad \int_{-\infty}^{\infty} B_n^i(x) dx = 1.$$

Можно показать, что функции  $B_n^i(x), i = -n, \dots, N-1$  линейно независимы и образуют базис в пространстве  $S_{n,1}(a, b)$  всех сплайнов порядка  $n$  с дефектом один на отрезке  $[a, b]$ . Таким образом, всякий сплайн  $S(x) \in S_{n,1}(a, b)$  может быть единственным образом записан в виде:

$$S(x) = \sum_{i=-n}^{N-1} b_i \cdot B_n^i(x),$$

где  $b_i$  - некоторые постоянные коэффициенты. Эту запись сплайна называют его представлением через  $B$  – сплайны.

*Реализация функций интерполяции в среде Matlab.*

В среде Matlab определены базовые функции аппроксимации данных полиномами по методу наименьших квадратов — полиномиальной регрессии. Это выполняет следующая функция:

$$\text{polyfit}(x, y, n).$$

Функция возвращает вектор коэффициентов полинома  $p(x)$  степени  $n$ , который с наименьшей среднеквадратичной погрешностью аппроксимирует функцию  $y(x)$ .

Для моделирования данных с помощью полученного интерполяционного полинома применяется функция:

$$\text{polyval}(p, s),$$

с помощью которой вычисляется значение полинома  $p(x) = p_1 \cdot x^n + p_2 \cdot x^{n-1} + \dots + p_n \cdot x + p_{n+1}$  в точке  $x = s$ .

Пакет Spline Toolbox среды Matlab позволяет конструировать сплайны, интерполировать и аппроксимировать одномерные и многомерные данные. Построенные сплайны записываются в специальном формате, который дает возможность визуализировать сплайны, производить арифметические и другие операции с ними. Spline Toolbox поддерживает работу со следующими основными формами записи сплайнов:

- кусочно-полиномиальная форма (pp – форма);
- $B$ - форма;
- рациональные сплайны, числитель и знаменатель которых могут быть представлены как в кусочно-полиномиальной форме, так и  $B$ -сплайнами;
- сплайн - поверхности, или сплайны типа тонкой пластинки (st-форма);
- сплайны, зависящие от нескольких переменных, представленные тензорными произведениями.

В пакете используется основная базовая функция, обращение к которой имеет вид:

$$yy = \text{int erpl}(ksi, y, xx, method),$$

где  $ksi$  - массив узлов,  $y$  - массив значений табличной функции,  $xx$  - значения абсцисс, для которых требуется вычислить значения интерполянта и записать в массив  $yy$ , а входной аргумент  $method$  определяет способ интерполяции: 'nearest' (по ближайшим соседям), 'linear' (кусочно-линейная), 'spline' (кубические сплайны), 'phchip' или 'cubic' (полиномы Эрмита на конечных носителях). Указание 'phchip' или 'cubic' приводит к вызову соответствующих функций *phchip* или *spline*, которыми можно пользоваться отдельно без вызова функции *int erpl*. То есть следующие обращения приводят к одинаковым результатам:

$$yy = \text{spline}(ksi, y, xx) \Leftrightarrow yy = \text{int erpl}(ksi, y, xx, 'spline')$$

$$yy = \text{pchip}(ksi, y, xx) \Leftrightarrow yy = \text{int erpl}(ksi, y, xx, 'pchip')$$

Другой способ обращения к этим функциям используется для построения кубических сплайнов и полиномов Эрмита на конечных носителях:

$$pp = \text{spline}(ksi, y) \Leftrightarrow pp = \text{int erpl}(ksi, y, 'spline', 'pp')$$

$$pp = pchip(ksi, y) \Leftrightarrow pp = \text{interpl}(ksi, y, 'pchip', 'pp')$$

Вызов данных функций позволяет получать структуры *pp* с информацией о сплайне, которая позволяет работать с другими вспомогательными функциями Spline ToolBox.

Например, вызов функции  $yy = \text{spline}(x, y, xx)$  позволяет интерполировать значения функции *y* в точках *xx* внутри области определения функции, на основе кубических сплайнов. А функция  $y = \text{ppval}(pp, xx)$  позволяет вычислять значения кусочно – гладкого полинома, заданного в виде *pp* - формы для значений аргумента *xx*.

Пакет Curve Fitting Toolbox среды Matlab предназначен для решения различных прикладных задач подгонки, аппроксимации и интерполяции данных с широкими визуальными возможностями представления полученных результатов. Данный пакет включает в себя интерактивные средства предварительной обработки данных, сравнения стандартных моделей и разработки моделей пользователя, подгонки с помощью стандартных и робастных методов и анализа качества аппроксимации. Пакет Curve Fitting Toolbox содержит полиномиальную модель подгонки, экспоненциальную, Гауссову, Фурье и многие другие. Пакет содержит средства идентификации аномальных данных, а также функции сглаживания Савитского–Голея и скользящего среднего.

Среда Matlab позволяет импортировать данные в пакет Curve Fitting Toolbox из различных файловых форматов.