Политех Петра Великого, 2 курс, осень 2022/23

Подготовка к экзамену по вычислительной математике

Лектор: Устинов Сергей Михайлович

Содержание

Вопрос 1. Конечные разности и их свойства. Таблица конечных разностей.

1.1 Конечные разности и их свойства.

Definition 1: Конечная разность

Пусть значения некоторой функции $f(x)$ известны лишь для дискретного множества значений независимой переменной $x \in \{x_0 \dots x_m\}$. Выражение

$$\Delta_h f(x_k) = f(x_k + h) - f(x_k) = f(x_0 + (k+1)h) - f(x_0 + kh) \tag{1}$$

называют *конечной разностью* (*разностным оператором*) первого порядка.

Поскольку величины x_0 и h постоянны для рассматриваемого множества, целесообразно, не умаляя общности, перейти к новой переменной $k = \frac{x_k - x_0}{h}$, которая принимает целые значения $0 \dots m-1$. Тогда функция $f(x)$ становится функцией целочисленной переменной $f(k)$, и можно будет опустить индекс $h = const$.

$$\Delta f(k) = \Delta f_k = f(k+1) - f(k) = f_{k+1} - f_k$$

Теперь обратимся к некоторым свойствам конечных разностей, отмечая тесную связь между ними и свойствами производных, что является основой большинства конечно-разностных выражений.

1. $\alpha = const \Rightarrow \Delta\alpha = 0$

   $$\alpha = const \Rightarrow \forall k \quad f(k+1) - f(k) = \alpha - \alpha = 0$$

2. $\Delta(\alpha f(k)) = \alpha \Delta f(k)$
3. $\Delta(f(k) \pm g(k)) = \Delta f(k) \pm \Delta g(k)$
4. $\Delta(f(k) \cdot g(k)) = \Delta f(k) \cdot g(k+1) + f(k)\Delta g(k)$

   $$\Delta(f(k) \cdot g(k)) = f(k+1)g(k+1) - f(k)g(k) =$$
   $$f(k+1)g(k+1) - f(k)g(k) + f(k)g(k+1) - f(k)g(k+1) =$$
   $$\Delta f(k) \cdot g(k+1) + f(k)\Delta g(k)$$

   Заметим, что аналогичными преобразованиями можно было получить и другой вид, в котором функции идут в другом порядке:

   $$\Delta(g(k) \cdot f(k)) = \Delta g(k) \cdot f(k+1) + g(k)\Delta f(k)$$

5. Конечная разность от полинома степени s равна полиному степени $s-1$.

   $$\Delta k^s = (k+1)^s - k^s = sk^{s-1} + \frac{s(s-1)}{2}k^{s-2} + \dots$$

6. Конечная разность высокого порядка.
   Подобно дифференциалам и производным высокого порядка, соответствующие конечные разности строятся на основе рекуррентных соотношений. Так конечная разность порядка $s+1$ строится следующим образом:

   $$\Delta^{s+1} f_k = \Delta(\Delta^s f_k) = \Delta^s f_{k+1} - \Delta^s f_k$$

   По индукции можно доказать следующее утверждение:

Theorem 1: О конечных разностях высокого порядка

$$\Delta^s f_k = \sum_{i=0}^{s} (-1)^i C_s^i f_{k+s-i} \tag{2}$$

1.2 Таблица конечных разностей.

Аналогично тому, как в непрерывном случае строилась таблица производных, рассмотрим конечные разности для наиболее популярных функций.

1. $\Delta a^k = a^{k+1} - a^k = a^k(a-1)$
   Заметим, что число 2 в условиях конечных разностей играет роль, схожую с экспонентой в непрерывном случае: $(e^x)' = e^x$.
2. $\Delta \sin(k) = \sin(k+1) - \sin(k) = 2\sin\left(\frac{1}{2}\right)\cos\left(k+\frac{1}{2}\right)$
3. $\Delta \cos(k) = \cos(k+1) - \cos(k) = -2\sin\left(\frac{1}{2}\right)\sin\left(k+\frac{1}{2}\right)$
4. $\Delta \log(k) = \log(k+1) - \log(k) = \log\left(1+\frac{1}{k}\right)$

Вопрос 2. Суммирование функций. Формула Абеля суммирования по частям.

2.1 Суммирование функций.

Обратимся к уравнению

$$\Delta F(k) = \varphi(k) \tag{3}$$

До сих пор мы занимались прямой задачей: по заданной функции $F(k)$ необходимо определить функцию $\varphi(k)$. Теперь обратимся к обратной задаче: по заданной функции $\varphi(k)$ необходимо восстановить функцию $F(k)$. Ситуация подобна нахождению функции по ее производной в непрерывных терминах. В этом случае появляется возможность ее решения при помощи интеграла

$$\int_a^b h(x)dx = \int_a^b f'(x)dx = f(b) - f(a)$$

Аналогично, решение обратной задачи (3) позволяет, в свою очередь, успешно решать задачу суммирования функции $\varphi(k)$. Запишем уравнение (3) последовательно для $k = m, m+1, \ldots, N-1$ и результаты просуммируем.

$$\begin{aligned}
&F(m+1) - F(m) = \varphi(m)\\
&F(m+2) - F(m+1) = \varphi(m+1)\\
&F(m+3) - F(m+2) = \varphi(m+2)\\
&\ldots\\
&F(N) - F(N-1) = \varphi(N-1)\\
&F(N) - F(m) = \sum_{k=m}^{N-1} \varphi(k)
\end{aligned}$$

Иными словами

$$\sum_{k=m}^{N-1} \varphi(k) = \sum_{k=m}^{N-1} \Delta F(k) = F(N) - F(m) \tag{4}$$

Выражение (4) является дискретным аналогом формулы Ньютона-Лейбница. В дополнение следует заметить, что она выводилась в предположении, что $N > m$.

Рассмотрим некоторые примеры суммирования функций. Результаты отдаленно напоминают таблицу интегралов для непрерывных функций, а оператор суммы сопоставляется определенному интегралу:

$$\sum_{k=0}^{N-1} \leftrightarrow \int_0^N$$

1. Найти $\sum_{k=0}^{N-1} a^k$

Функция $F(k)$, удовлетворяющая условию $\Delta F(k) = a^k$ легко находится из таблицы конечных разностей: $F(k) = \dfrac{a^k}{a-1}$. Тогда

$$\sum_{k=0}^{N-1} a^k = \frac{\left(a^N - 1\right)}{a-1} - \frac{a^0 - 1}{a-1} = \frac{1-a^N}{1-a}$$

2. Найти $\displaystyle\sum_{k=0}^{N-1} \cos\left(k + \frac{1}{2}\right)$

   Аналогичным образом найдем функцию из таблицы конечных разностей.

$$\sum_{k=0}^{N-1} \cos\left(k + \frac{1}{2}\right) = \sum_{k=0}^{N-1} \frac{\sin(k)}{2\sin\left(\frac{1}{2}\right)} = \frac{\sin(N)}{2\sin\left(\frac{1}{2}\right)}$$

3. Найти $\displaystyle\sum_{k=0}^{N-1} k^2$

   Воспользуемся свойством №5 конечных разностей: полином степени 2 является конечной разностью полинома степени 3. Рассмотрим

$$\begin{aligned}
&\Delta k^3 = (k+1)^3 - k^3 = 3k^2 + 3k + 1\\
&\Delta k^2 = (k+1)^2 - k^2 = 2k + 1\\
&\Delta k = (k+1) - k = 1
\end{aligned}$$

   Попробуем составить подходящую линейную комбинацию этих полиномов. Подбирая коэффициенты и используя свойства конечных разностей №1 и 2, получаем

$$\Delta\left(\frac{k^3}{3} - \frac{k^2}{2} + \frac{k}{6}\right) = \frac{1}{3}\left(3k^2 + 3k + 1\right) - \frac{1}{2}\left(2k + 1\right) + \frac{1}{6} = k^2 + k + \frac{1}{3} - k - \frac{1}{2} + \frac{1}{6} = k^2$$

   Получаем

$$\begin{aligned}
&F(k) = \frac{k^3}{3} - \frac{k^2}{2} + \frac{k}{6}\\
&\sum_{k=0}^{N-1} k^2 = \frac{N^3}{3} - \frac{N^2}{2} + \frac{N}{6}
\end{aligned}$$

2.2 Суммирование по частям.

Суммирование *по частям* вводится как прием, аналогичный интегрированию по частям в непрерывном случае. Для вывода формулы запишем уравнение интегрирования по частям в несколько другом виде.

Введем три функции: $u(t), v(t), U(t)$, где $U(t) = \displaystyle\int_0^t u(\tau)d\tau$. Рассмотрим производную выражения $U(t)v(t)$:

$$\frac{d}{dt}\left(U(t)v(t)\right) = \frac{dU(t)}{dt}v(t) + U(t)\frac{dv(t)}{dt} = u(t)v(t) + U(t)\frac{dv(t)}{dt}$$

Перенесем второе слагаемое из правой части равенства в левую и проинтегрируем

$$\int_a^b u(t)v(t)dt = \int_a^b \frac{d}{dt}\left(U(t)v(t)\right)dt - \int_a^b U(t)\frac{dv(t)}{dt}dt = U(t)v(t)\Big|_{t=a}^{t=b} - \int_a^b U(t)\frac{dv(t)}{dt}dt$$

Теперь обратимся к суммированию. Аналогично введем три функции: $u(k), v(k), U(k)$, где $U(k) = \sum_{i=0}^{k} u(i)$. $\Delta U(k) = U(k+1) - U(k) = u(k+1)$. Воспользуемся ранее полученной формулой конечной разности для произведения:

$$\Delta\left(U(k)v(k)\right) = v(k+1)\Delta U(k) + U(k)\Delta v(k) = v(k+1)u(k+1) + U(k)\Delta v(k)$$

Перенесем второе слагаемое из правой части равенства в левую и просуммируем обе его части:

Theorem 2: Формула Абеля суммирования по частям

$$\sum_{k=p}^{N} u(k+1)v(k+1) = U(k)v(k)\Big|_{k=p}^{k=N+1} - \sum_{k=p}^{N} U(k)\Delta v(k) \tag{5}$$

Пример: требуется найти $\sum_{k=0}^{N} ka^k$.

Возьмем функции $u(k) = a^{k-1}, v(k) = k-1$. Тогда

$$U(k) = \sum_{i=0}^{k} a^{k-1} = \frac{1}{a}\frac{a^{k+1}-1}{a-1}$$
$$\Delta v(k) = \Delta k = 1$$

$$\sum_{k=0}^{N} ka^k = \frac{1}{a}\frac{a^{N+2}-1}{a-1}\cdot(((N+1)-1)-(0-1)) - \sum_{k=0}^{N}\frac{1}{a}\frac{a^{k+1}-1}{a-1} =$$
$$= \frac{(N+1)(a^{N+2}-1)}{a(a-1)} - \frac{1}{a(a-1)}\left(a\sum_{k=0}^{N}a^k - (N+1)\right) =$$
$$= \frac{(N+1)(a^{N+2}-1)}{a(a-1)} - \frac{1}{a(a-1)}\left(a\frac{a^{N+1}-1}{a-1} - (N+1)\right) =$$
$$= \frac{(a-1)(N+1)(a^{N+2}-1) - a^{N+3} + a^2 + (a-1)(N+1)}{a(a-1)^2} =$$
$$= \frac{(a-1)(N+1)a^{N+1} - a^{N+2} + a}{(a-1)^2} = \frac{Na^{N+2} - (N+1)a^{N+1} + a}{(a-1)^2} =$$
$$= \frac{a\left(Na^{N+1} - (N+1)a^N + 1\right)}{(a-1)^2}$$

Вопрос 3. Разностное уравнение, его порядок. Линейные разностные уравнения первого порядка и порядка выше первого.

3.1 Разностное уравнение, его порядок.

Первоначально обратимся к дифференциальным уравнениям.

Definition 2: Дифференцильное уравнение

Соотношение

$$F\left(t, z(t), z'(t), \ldots, z^{(s-1)}(t)\right) = 0$$

где t – независимая переменная, функция F задана, функция $z(t)$ – искомая, называется *дифференциальным уравнением порядка* s.

При этом уравнение может быть разрешено относительно старшей производной

$$z^{(s)}(t) = f\left(t, z(t), z'(t), \ldots, z^{(s-1)}(t)\right) \tag{6}$$

Порядок уравнения s, определяемый номером старшей производной, является важной характеристикой уравнения (6). Так он определяет количество начальных условий, необходимых для однозначного решения. Если дифференциальное уравнение является линейным относительно функции $z(t)$ и ее производных, то величина s задает количество линейно независимых решений и т.д.
Рассмотрим разностный аналог дифференциального уравнения

$$F\left(k, f(k), \Delta f(k), \Delta^2 f(k), \ldots, \Delta^s f(k)\right) = 0 \tag{7}$$

где k – независимая целочисленная переменная, функция F задана, функция $f(k)$ – искомая. Казалось бы, логично считать порядок этого уравнения равным s, руководствуясь номером старшей конечной разности, как это было с производными в уравнении (6). Рассмотрим, однако, следующий пример:

$$2\Delta^3 f_k + 3\Delta^2 f_k - f_k = 0, \quad f_k \equiv f(k)$$

Выразим все конечные разности через значения функции в различных точках и получим

$$2(f_{k+3} - 3f_{k+2} + 3f_{k+1} - f_k) + 3(f_{k+2} - 2f_{k+1} + f_k) - f_k = 0,$$
$$2f_{k+3} - 3f_{k+2} = 0$$

Задаваясь только одним начальным условием f_0 вместо ожидаемых трех и последовательно полагая значение $k = -2, -1, 0, 1, \ldots$ шаг за шагом воспроизводим f_k для любого значения k. В этом и есть различие между дифференциальными и разностными уравнениями. Снижение ожидаемого порядка произошло за счет сокращения слагаемых. По этой причине в общем случае для определения порядка разностного уравнения будем выражать все конечные разности через значения функции. Тогда, после всех упрощений порядок разностного уравнения будет определяться разностью между наибольшим и наименьшим значениями аргумента функции $f(k)$. В дальнейшем будем записывать разностные уравнения в следующем виде.

Definition 3: Разностное уравнение

Уравнение вида

$$\Phi\left(k, f(k), f(k+1), \ldots, f(k+s)\right) = 0$$

где k – независимая переменная, функция Φ задана, функция $f(k)$ – искомая, называется *разностным уравнением порядка* $s = (k+s) - k$.

или в виде, разрешенном относительно функции с наибольшим значением аргумента

$$f(k+s) = \Phi_1\left(k, f(k), \ldots, f(k+s-1)\right)$$

Для его решения достаточно последовательно полагать $k = 0, 1, 2, \ldots$

$$\begin{aligned} f(s) &= \Phi_1\left(0, f(0), \ldots, f(s-1)\right), \\ f(s+1) &= \Phi_1\left(1, f(1), \ldots, f(s)\right), \\ f(s+2) &= \Phi_1\left(2, f(2), \ldots, f(s+1)\right), \\ &\ldots \end{aligned}$$

Такое построение решения называют *пошаговым методом решения разностного уравнения*, который всегда дает решение, когда заданы s начальных условий.

3.2 Линейное разностное уравнение первого порядка

Обратимся к уравнению

$$y(k+1) = \alpha \cdot y(k) + \varphi(k), \quad y(0) = y_0 \tag{8}$$

где α – постоянный коэффициент, $\varphi_k = \varphi(k)$ – заданная функция k, $y_k = y(k)$ – искомая функция. Если $\varphi(k) = 0$, уравнение называется однородным, в противном случае – неоднородным. Начнем решать уравнение (8) пошаговым методом.

$$\begin{aligned} &y_1 = \alpha \cdot y_0 + \varphi_0, \\ &y_2 = \alpha \cdot y_1 + \varphi_1 = \alpha\left(\alpha \cdot y_0 + \varphi_0\right) + \varphi_1 = \alpha^2 \cdot y_0 + \alpha \cdot \varphi_0 + \varphi_1, \\ &y_3 = \alpha \cdot y_2 + \varphi_2 = \alpha\left(\alpha^2 \cdot y_0 + \alpha \cdot \varphi_0 + \varphi_1\right) + \varphi_2 = \alpha^3 \cdot y_0 + \alpha^2 \cdot \varphi_0 + \alpha \cdot \varphi_1 + \varphi_2, \\ &\ldots \end{aligned}$$

По индукции можно доказать, что

$$y_n = \alpha^n \cdot y_0 + \sum_{k=0}^{n-1} \alpha^k \cdot \varphi_{n-1-k}$$

Для важного частного случая, когда $\varphi(k)$ – постоянная функция ($\varphi_k = \beta = const$):

$$y_n = \alpha^n \cdot y_0 + \left(\sum_{k=0}^{n-1} \alpha^k\right) \cdot \beta = \alpha^n \cdot y_0 + \frac{1-\alpha^n}{1-\alpha} \cdot \beta$$

3.3 Линейное разностное уравнение порядка выше первого

Перейдем к уравнению порядка s:

$$y(k+s)+\alpha_1 y(k+s-1)+\alpha_2 y(k+s-2)+\cdots+\alpha_s y(k)=\varphi(k) \tag{9}$$

где α_i – постоянные коэффициенты, $\varphi(k)$ – заданная функция, $y(k)$ – искомая функция.
Рассмотрим некоторые свойства частных и общих решений систем линейных разностных уравнений.

Theorem 3

Пусть $y_1(k), y_2(k), \ldots, y_p(k)$ – частные решения линейного однородного уравнения

$$y(k+s)+\alpha_1 y(k+s-1)+\alpha_2 y(k+s-2)+\cdots+\alpha_s y(k)=0 \tag{10}$$

то любая их линейная комбинация

$$c_1 y_1(k)+c_2 y_2(k)+\cdots+c_p y_p(k)$$

где c_i – произвольные постоянные, также будет частным решением этого уравнения.

Theorem 4

Если s частных решений однородного уравнения $y_1(k), y_2(k), \ldots, y_s(k)$ – линейно независимы, то

$$y(k)=\sum_{i=1}^{s} c_i y_i(k) \tag{11}$$

является общим решением однородного уравнения.

Theorem 5

Общее решение линейного неоднородного уравнения (9) представляется в виде суммы частного его решения $y_{частн}(k)$ и общего решения линейного уравнения (11)

$$y(k)=y_{частн}(k)+\sum_{i=1}^{s} c_i y_i(k)$$

Решение неоднородного уравнения (9) начинается с решения однородного уравнения (10). Это решение будем искать в виде $u(k)=C\gamma^k$, где $C=const$. Здесь уместно вспомнить, что в дифференциальном уравнении порядка s с постоянными коэффициентами частные решения ищутся в форме $z(t)=C\cdot\exp(\lambda_k t)$. Подставим $u(k)=C\gamma^k$ в уравнение (10) и после сокращения получаем уравнение

$$\gamma^s+\alpha_1\gamma^{s-1}+\cdots+\alpha_s=0 \tag{12}$$

которое получило название *характеристического уравнения*. Оно, с учетом кратности, имеет s корней, каждому из которых соответствует частное решение.

1. Каждому простому вещественному корню γ_r соответствует частное решение $u_r(k) = c_r\gamma_r^k$, являющееся одним из слагаемых в общем решении.
2. Каждой простой паре комплексно-сопряженных корней $\gamma_{r,r+1} = (\alpha_r \pm i\beta_r)$ соответствуют комплексные частные решения, являющиеся линейно независимыми

$$u_r(k) = (\alpha_r + i\beta_r)^k, \quad u_{r+1}(k) = (\alpha_r - i\beta_r)^k$$

или вещественные частные решения

$$u_r(k) = \rho_r^k \cos(k\varphi_r), \quad u_{r+1}(k) = \rho_r^k \sin(k\varphi_r), \quad \rho_r = \sqrt{\alpha_r^2 + \beta_r^2}, \quad tg(\varphi_r) = \frac{\beta_r}{\alpha_r}$$

В общем решении им сопоставляются два слагаемых (вещественный вариант)

$$c_r\rho_r^k \cos(k\varphi_r) + c_{r+1}\rho_r^k \sin(k\varphi_r)$$

3. Если среди корней встречаются кратные, то корню γ_r кратности p соответствуют частные решения

$$u_r(k) = \gamma_r^k, \quad u_{r+1}(k) = k\gamma_r^k, \ldots, u_{r+p-1}(k) = k^{p-1}\gamma_r^k$$

Решения эти линейно зависимы, и в общем решении им сопоставляются слагаемые

$$c_r\gamma_r^k + c_{r+1}k\gamma_r^k + \cdots + c_{r+p-1}k^{p-1}\gamma_r^k = Q_{p-1}(k)\gamma_r^k$$

где $Q_{p-1}(k)$ – полином от k степени $p-1$.

Вопрос 4. Разделенные разности и их связь с конечными разностями.

Для равноотстоящих узлов таблицы конечные разности являются хорошей характеристикой изменения функции, аналогичной производной для непрерывного случая. При произвольном расположении узлов таблицы целесообразно ввести понятие *разделенной разности*.

Definition 4: Разделенная разность

Разделенные разности нулевого порядка совпадают со значениями функции, а разности первого порядка определяются равенством

$$f(x_{n-1}; x_n) = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \tag{13}$$

Аналогично строятся разделенные разности высших порядков. При этом разности k-го порядка определяются через разности $(k-1)$-го порядка по формуле

$$f(x_0; x_1; \ldots; x_k) = \frac{f(x_1; x_2; \ldots; x_k) - f(x_0; x_1; \ldots; x_{k-1})}{x_k - x_0} \tag{14}$$

Подобно конечным разностям, разделенные тоже можно выразить через значения функции в различных точках. По индукции можно доказать следующее равенство:

$$f(x_i; x_{i+1}; \ldots; x_{i+k}) = \sum_{j=i}^{i+k} \frac{f(x_j)}{\prod\limits_{j \neq i} (x_j - x_i)} \tag{15}$$

Отсюда следует важное свойство разделенных разностей: они являются симметричными функциями своих аргументов.

$$f(x_n; x_{n-1}) = f(x_{n-1}; x_n)$$

Если в исходной таблице узлы равноотстоящие, то для описания поведения функции можно использовать как конечные разности, так и разделенные. Установим связь между ними. Обобщенную формулу можно доказать по индукции.

$$f(x_i; x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{\Delta f_i}{h} \quad \ldots$$

Theorem 6: Связь конечных и разделенных разностей

$$f(x_i; x_{i+1}; \ldots; x_{i+k-1}; x_{i+k}) = \frac{\Delta^k f_i}{k!\, h^k} \tag{16}$$

Вопрос 5. Аппроксимация функций. Задача интерполирования.

5.1 Аппроксимация функций.

В данной теме аппроксимация означает *замену одной функциональной зависимости другой*. Поскольку на практике часто возникает потребность дифференцировать, интегрировать или использовать эту функцию в различных расчетах, целесообразно выбирать аппроксимирующую функцию, исходя из простоты ее вида. Возможность выбора обосновывается следующей теоремой.

Theorem 7: Теорема Вейерштрасса

Пусть $f(x)$ непрерывна на $[a,b]$. Тогда

$$\forall \varepsilon > 0 \quad \exists P_n(x) \quad n = n(\varepsilon): \quad \max |f(x) - P_n(x)| < \varepsilon$$

Однако, эта теорема лишь гарантирует существование, но не дает гарантии, что такой полином можно построить при помощи практического алгоритма.
Для того чтобы можно было сравнивать различные варианты аппроксимации, следует ввести критерий близости. Например, максимум модуля отклонения исходной функции $f(x)$ от аппроксимирующей $g(x)$ на заданном промежутке:

$$\delta = \max_{x \in [a,b]} |f(x) - g(x)| \tag{17}$$

или так называемый «среднеквадратичный критерий»

$$\rho^2 = \int_a^b (f(x) - g(x))^2 \, dx \tag{18}$$

В случае если $f(x)$ определена таблично заданным набором точек, может быть использован аналог критерия (18)

$$\rho^2 = \sum_{k=1}^{m} (f(x_k) - g(x_k))^2 \tag{19}$$

Лучшей оказывается аппроксимирующая функция, обладающая наименьшей величиной δ или ρ^2. *Заметим, что* только решаемая задача диктует выбираемый критерий близости, который, в свою очередь, позволяет выбрать лучшую аппроксимацию.

5.2 Постановка задачи интерполирования

Будем приближать исходную функцию, заданную таблично $X = \{x_0, x_1, \ldots, x_m\}, F = \{f(x) \,|\, x \in X\}$ обобщенным многочленом

$$Q_m(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \cdots + a_m\varphi_m(x) = \sum_{k=0}^{m} a_k\varphi_k(x) \tag{20}$$

где $\{\varphi_k\}$ – заданный набор линейно независимых функций, а коэффициенты a_i подлежат определению. В качестве критерия близости выбирается совпадение значений $f(x)$ и $Q_m(x)$ в узлах таблицы

$$Q_m(x_i) = f(x_i), \qquad i = 0,1,\ldots,m \tag{21}$$

Definition 5: Интерполяционный многочлен

Полином (20) называется *интерполяционным многочленом*, а x_k – *узлами интерполирования*.

Равенства (21) представляют собой СЛАУ относительно искомых коэффициентов обобщенного многочлена $a_0, a_1, \ldots, a_m$. Эта система имеет единственное решение, если ее определитель отличен от нуля:

$$\det \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \ldots & \varphi_m(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \ldots & \varphi_m(x_1) \\ \ldots & \ldots & \ldots & \ldots \\ \varphi_0(x_m) & \varphi_1(x_m) & \ldots & \varphi_m(x_m) \end{pmatrix} \neq 0$$

Наиболее популярной является полиномиальная аппроксимация:

$$\varphi_k(x) = x^k \qquad Q_m(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_m x^m$$

Definition 6: Определитель Вандермонда

Определитель СЛАУ для случая полиномиальной аппроксимации называется определителем Вандермонда и имеет следующий вид:

$$\begin{vmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & \ldots & x_n^n \end{vmatrix}$$

Определитель Вандермонда отличен от нуля, и задача имеет единственное решение, если узлы интерполирования $x_0, x_1, \ldots, x_m$ различны.

Вопрос 6. Интерполяционный полином Лагранжа. Остаточный член полинома Лагранжа.

Непосредственное численное решение представляет значительные трудности. С одной стороны, это связано с заметным объемом вычислений для нахождения a_k. С другой стороны, малое изменение данных таблицы $(x_k, f(x_k))$ часто приводит к сильному изменению решения (особенно для близко расположенных узлов интерполирования). В связи с этим, попробуем построить полином, не прибегая к решению системы.
С этой целью введем следующие функции:

$$\omega(x) = (x - x_0)(x - x_1)\dots(x - x_m)$$
$$\omega_k(x) = \frac{\omega(x)}{(x - x_k)} = (x - x_0)(x - x_1)\dots(x - x_{k-1})(x - x_{k+1})\dots(x - x_m)$$

В этих обозначениях запишем следующий полином

Definition 7: Интерполяционный полином Лагранжа

$$Q_m(x) = \sum_{k=0}^{m} \frac{\omega_k(x)}{\omega_k(x_k)} f(x_k) \tag{22}$$

По построению это многочлен степени m. Определим его значения в узлах интерполирования x_i. Так как для $x = x_i$ полином $\omega_k(x)$ равен нулю, если только $i \neq k$, то для $Q_m(x_i)$ получаем

$$Q_m(x_i) = \sum_{k=0}^{m} \frac{\omega_k(x_i)}{\omega_k(x_k)} f(x_k) = \frac{\omega_i(x_i)}{\omega_i(x_i)} f(x_i) = f(x_i), \qquad i = 0,1,\dots,m$$

То есть в узлах интерполяции значения полинома совпадают со значениями функции.
Теперь обратимся к погрешности интерполяционного полинома. Исходная функция $f(x)$ может быть представлена в виде

$$f(x) = Q_m(x) + R_m(x)$$

где $Q_m(x)$ – интерполяционный полином, а $R_m(x)$ носит название *остаточного члена интерполяционного полинома.*

Theorem 8: Об остаточном члене полинома Лагранжа

Пусть $f(x)$ на промежутке $[a, b]$ имеет непрерывные производные вплоть до $m+1$ порядка, то остаточный член $R_m(x)$ можно представить в виде:

$$R_m(x) = f(x) - Q_m(x) = \frac{f^{(m+1)}(\eta)}{(m+1)!}\omega(x), \qquad \eta \in [a, b] \tag{23}$$

При этом $\omega(x)$ определяется как и прежде.
Доказательство. Рассмотрим вспомогательную функцию

$$\varphi(z) = f(z) - Q_m(z) - K\omega(z) \tag{24}$$

где K – некоторая постоянная. Пусть x_k – узлы интерполирования, а x – точка, в которой оценивается погрешность ($x \neq x_k$). Легко заметить, что функция $\varphi(z)$ равна нулю во всех узлах интерполирования. Выберем константу K так, чтобы $\varphi(x) = 0$

$$K = \frac{f(x) - Q_m(x)}{\omega(x)} = \frac{R_m(x)}{\omega(x)}$$

Таким образом, $\varphi(z)$ имеет по меньшей мере $m+2$ нуля (все узлы интерполирования и точка x). Тогда по теореме Ролля первая производная $\varphi(z)$ имеет по меньшей мере $m+1$ нуль, вторая производная – не менее m нулей, а $(m+1)$-я производная $\varphi^{(m+1)}(z)$ имеет по меньшей мере один нуль. Обозначим такую точку за η. Тогда, последовательно дифференцируя (24), получаем

$$\varphi^{(m+1)}(\eta) = f^{(m+1)}(\eta) - 0 - K(m+1)! = 0$$

Подставляя в это равенство выражение для K, получаем формулу для $R_m(x)$, совпадающую с ожидаемой.

Эта теорема позволяет сделать очевидный, но важный вывод. Пусть $f(x)$ – это полином степени m. Тогда $f^{(m+1)}(\eta) = 0$. Следовательно, полином степени m *однозначно* воспроизводится интерполяционным полиномом по $m+1$ точке. Ясно также, что остаточный член во всех узлах интерполирования равен нулю.

В заключение стоит отметить, что, хотя о расположении точки η ничего не известно, очевидна зависимость величины η как от узлов интерполирования, так и от точки x, где оценивается погрешность, т.е. $\eta = \eta(x)$.

Остаточный член позволяет оценивать отклонение $L_m(x)$ от $f(x)$ для дифференцируемых функций тогда, когда удается оценить $f^{(m+1)}(x)$.

Полагая $M_{m+1} = \max |f^{(m+1)}(x)|$, получим $R_m(x) \leq \dfrac{M_{m+1}}{(m+1)!}|\omega(x)|$.

Вопрос 7. Выбор узлов интерполирования. Интерполяционный полином Ньютона для равно и неравноотстоящих узлов.

7.1 Выбор узлов интерполирования

Для уменьшения погрешности интерполирования обратимся к теореме об остаточном члене полинома Лагранжа при заданной степени полинома m. Поскольку величиной $f^{(m+1)}(\eta)$ трудно управлять, и возможна лишь оценка пределов ее изменения, задача уменьшения погрешности сводится к управлению величиной $|\omega(x)|$ за счет выбора узлов интерполирования. Рассмотрим два типичных на практике случая.
Случай 1. Задана степень полинома m и имеется таблица достаточно большой длины. Точка x^*, в которой вычисляется значение полинома, заранее известна. Требуется выбрать $m+1$ узел так, чтобы величина $|\omega(x^*)|$ была бы минимальна.
Результат очевиден. Нужно выбирать узлы интерполирования из таблицы, *ближайшие* к x^*. Использование любого другого узла вместо ближайшего неизбежно увеличивает значение

$$|\omega(x^*)| = |\,(x^* - x_0)\,(x^* - x_1)\ldots(x^* - x_m)\,|$$

Случай 2. Заданы степень полинома m и промежуток интерполирования $[a, b]$. Точка x^*, в которой вычисляется значение полинома, заранее не известна. Требуется выбрать узлы интерполирования так, чтобы в самом неблагоприятном случае расположения x^* погрешность была бы минимальна (т.н. *минимаксный критерий*)

$$\max_{[a,b]} |\omega(x^*)| \to \min$$

Интуитивно напрашивающееся предложение о равномерном задании узлов на промежутке оказывается ошибочным. Значения $|\omega(x)|$ в узлах интерполирования равны нулю, график напоминает «колокольчики», максимум которых достигается между узлами интерполирования. При выборе равноотстоящих узлов погрешность для x^*, близких к центру промежутка интерполирования оказывается небольшой, однако ближе к концам она сильно возрастает. Узлы интерполирования нужно симметрично сместить ближе к концам промежутка. Тогда высота центрального «колокольчика» увеличится, в то время как высота крайних уменьшится. Оптимальный выбор узлов интерполирования отвечает нулям так называемых ортогональных полиномов Чебышева, когда все «колокольчики» будут одинаковыми по высоте.

7.2 Интерполяционный полином Ньютона для равно и неравноотстоящих узлов.

Оценка погрешности на основе формулы

$$R_m(x) = \frac{f^{(m+1)}(\eta)}{(m+1)!}\omega(x)$$

выполняется крайне редко из-за известных трудностей, связанных с оценкой производной $f^{(m+1)}(\eta)$ особенно для таблично заданной функции. Поэтому на практике о

величине погрешности принято судить, сравнивая в заданной точке x^* значения интерполяционных полиномов соседних степеней $Q_m(x^*)$ и $Q_{m+1}(x^*)$. При недостаточной точности последовательно повышают степень полинома. Но для такой процедуры использование полинома Лагранжа оказывается неэффективным. При переходе к полиному следующей степени всю работу приходится выполнять заново. Целесообразно записать полином в таком виде, чтобы расчеты сводились к появлению лишь еще одного слагаемого в дополнение к ранее вычисленному $Q_m(x)$. С этой целью запишем первую разделенную разность

$$f(x;x_0) = \frac{f(x)-f(x_0)}{x-x_0}$$

и выразим из нее $f(x)$

$$f(x) = f(x_0) + (x-x_0)f(x;x_0) \tag{25}$$

Заметим, что первое слагаемое в первой части это интерполяционный полином нулевой степени, а второе слагаемое – погрешность полинома. Теперь запишем вторую разделенную разность

$$f(x;x_0;x_1) = \frac{f(x;x_0)-f(x_0;x_1)}{x-x_1}$$

выразим из нее первую разность через вторую и поставим в формулу (25)

$$f(x) = f(x_0) + (x-x_0)f(x_1;x_0) + (x-x_0)(x-x_1)f(x;x_0;x_1) \tag{26}$$

Продолжая этот процесс, получаем

$$\begin{aligned} f(x) =& f(x_0) + (x-x_0)f(x_1;x_0) + (x-x_0)(x-x_1)f(x_0;x_1;x_2) + \cdots + \\ &+ (x-x_0)(x-x_1)\dots(x-x_{m-1})f(x_0;x_1;x_2;\dots x_m) + \\ &+ (x-x_0)(x-x_1)\dots(x-x_m)f(x;x_0;x_1;x_2;\dots x_m) = \\ &= Q_m(x) + \omega(x)f(x;x_0;x_1;x_2;\dots x_m) \end{aligned}$$

Сумма первых k слагаемых порождает интерполяционный полином степени $k-1$, а последнее слагаемое является погрешностью интерполяционного полинома степени m. При этом структура полинома такова, что полином степени m получается как полином степени $m-1$ с добавлением еще одного слагаемого.

Definition 8: Интерполяционный полином Ньютона

Это интерполяционный полином в форме

$$Q_m(x) = Q_{m-1}(x) + (x-x_0)(x-x_1)\dots(x-x_{m-1})f(x_0;x_1;x_2;\dots x_m) \tag{27}$$

На практике вычисление разделенных разностей производится в рамках следующей таблицы, где появление новой разделенной разности более высокого порядка связано с построением еще одной диагонали

x_0	$f(x_0)$	$f(x_0;x_1)$	$f(x_0;x_1;x_2)$	$f(x_0;x_1;x_2;x_3)$	$f(x_0;x_1;x_2;x_3;x_4)$
x_1	$f(x_1)$	$f(x_1;x_2)$	$f(x_1;x_2;x_3)$	$f(x_1;x_2;x_3;x_4)$	
x_2	$f(x_2)$	$f(x_2;x_3)$	$f(x_2;x_3;x_4)$		
x_3	$f(x_3)$	$f(x_3;x_4)$			
x_4	$f(x_4)$				

Пускай узлы таблицы задания функции являются равноотстоящими. В этом случае разделенные разности можно заменить на конечные, тем самым избежав деления на разность значений аргумента. Используя

$$\frac{x-x_k}{h}=\frac{x-x_0-kh}{h}=\frac{x-x_0}{h}-k=t-k$$

и формулу связи разделенных и конечных разностей

$$f(x_i;x_{i+1};\ldots;x_{i+k-1};x_{i+k})=\frac{\Delta^k f_i}{k!\,h^k}$$

получим версию полинома Ньютона для равноотстоящих узлов

$$Q_m(x_0+ht)=f(x_0)+\frac{t}{1!}\Delta f(x_0)+\frac{t(t-1)}{2!}\Delta^2 f(x_0)+\cdots+\frac{t(t-1)\ldots(t-m+1)}{m!}\Delta^m f(x_0) \tag{28}$$

Новая независимая переменная t принимает в узлах таблицы целые значения, а вычисление конечных разностей реализуется подобно разделенным разностям по следующей таблице

x_0	$f(x_0)$	Δf_0	$\Delta^2 f_0$	$\Delta^3 f_0$	$\Delta^4 f_0$
x_1	$f(x_1)$	Δf_1	$\Delta^2 f_1$	$\Delta^3 f_1$	
x_2	$f(x_2)$	Δf_2	$\Delta^2 f_2$		
x_3	$f(x_3)$	Δf_3			
x_4	$f(x_4)$				

Вопрос 8. Сплайн-интерполяция. Подпрограммы SPLINE и SEVAL. Интерполирование по Эрмиту. Обратная задача интерполирования.

8.1 Интерполирование сплайнами.

На практике интерполяционные полиномы высоких степеней строят крайне редко. Это связано с тем, что их коэффициенты крайне чувствительны к погрешностям исходных данных. Сравнительно малое изменение узлов интерполирования x_k или значений функции $f(x_k)$ приводит к сильному изменению вида самого полинома. Одним из возможных решений является разбиение большой исходной таблицы на участки, для каждого из которых строится интерполяционный полином относительно невысокой степени. Однако, в основном требуется, чтобы аппроксимирующая функция была гладкой, а функция, составленная из различных полиномов, в узлах сопряжения не имеет производной. Выходом из положения является использование сплайн-интерполяции. Вообще, сплайн – это некий инструмент, используемый при построении чертежей. Дадим математической модели более формальное определение.

Обратимся к таблично заданной функции: $X = \{x_1, \ldots, x_N\}, F = \{f(x) \mid x \in X\}$. Число узлов равно N, а их нумерация начинается с единицы. На каждом промежутке $[x_k, x_{k+1}]$ будем строить интерполяционный полином третьей степени

$$S_k(x_{k+1}) = a_k + b_k (x - x_k) + c_k (x - x_k)^2 + d_k (x - x_k)^3 \tag{29}$$

Количество полиномов, как и промежутков, равно $N - 1$, и каждый полином имеет 4 параметра. Таким образом, всего в наличии $4N - 4$ параметра. Потребуем, чтобы во всех внутренних точках были равны значения соседних полиномов, их первых и вторых производных.

$$\begin{cases} S_k(x_{k+1}) = S_{k+1}(x_{k+1}) \\ S'_k(x_{k+1}) = S'_{k+1}(x_{k+1}) \\ S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}) \end{cases} \qquad k = 1, \ldots, N-2$$

То есть выполнялось суммарно $3(N - 2) = 3N - 6$ уравнений. Еще N уравнений отражают требования интерполирования

$$S_k(x_k) = f_k; \qquad k = 1, \ldots, N-1; \qquad S_{N-1}(x_N) = f_N$$

Общее число задаваемых уравнений достигает $4N - 6$. При наличии $4N - 4$ параметров появляется возможность выполнить еще два условия. Их задание необязательно – все требования интерполяции и сопряжения соседних полиномов уже выполнены, но это целесообразно сделать для однозначного решения задачи. Различные кубические сплайны отличаются друг от друга заданием этих двух требований, которые, как правило, записываются для двух крайних точек x_1 и x_N. К этим двум дополнительным условиям целесообразно выдвинуть следующие два требования. С одной стороны, их лучше задавать так, чтобы полная система уравнений решалась по возможности более просто. С другой стороны, они должны максимально соответствовать характеру

поведения функции в начале и в конце промежутка интерполирования. Рассмотрим на примерах.

Пример 1. $S_1''(x_1) = 0; \quad S_{N-1}''(x_N) = 0$. Этот сплайн получил название *естественного кубического сплайна.* Такие условия и название оправдываются только при использовании в механике. В общем случае равенство нулю второй производной на краях промежутка не является обязательным свойством экспериментальных данных, отражаемых таблицей.
Пример 2. По первым четырем точкам таблицы строится интерполяционный полином третьей степени $Q_3(x)$, и его третья производная приравнивается третьей производной $S_1(x)$. Аналогично, по последним четырем точкам строится интерполяционный полином $\tilde{Q}_3(x)$, и его третья производная приравнивается третьей производной последнего полинома $S_{N-1}(x)$.

$$Q_3'''(x_1) = S_1'''(x_1); \qquad \tilde{Q_3}'''(x_N) = S_{N-1}'''(x_N)$$

Такие условия не только отвечают характеру поведения функции в начале и в конце промежутка интерполирования, но и достаточно просты (третья производная от полинома третьей степени постоянна). Именно они и учитываются в рассматриваемых программах `SPLINE` и `SEVAL`.

8.2 Подпрограммы SPLINE и SEVAL.

Первая из них – `SPLINE(N, X, F, B, C, D)`, оформленная как процедура, решает систему уравнений относительно b_k, c_k, d_k.
N – число точек;
X, **F** – векторы, элементами которых являются x_k и f_k;
B, **C**, **D** – векторы с коэффициентами b_k, c_k, d_k полиномов (29) – результаты работы `SPLINE`.

Вторая программа `SEVAL(N, U, X, F, B, C, D)`, оформленная как функция, использует результаты работы `SPLINE` и вычисляет значение сплайна в заданной точке **U**.

8.3 Интерполирование по Эрмиту.

До этого интерполяция происходила только по значениям функции. Существует ряд задач, задаваемых более широкими условиями. В частности, если в таблице помимо значений функции присутствуют ее производные, и от интерполяционного полинома требуется совпадение с данными этой таблицы, то такая задача называется *интерполированием по Эрмиту.* Рассмотрим пример.
Для следующих входных данных требуется построить интерполяционный полином, удовлетворяющий всем условиям таблицы.

x	x_0	x_1	x_2
$f(x)$	$f(x_0)$	$f(x_1)$	$f(x_2)$
$f'(x)$	$f'(x_0)$	$f'(x_1)$	—
$f''(x)$	—	$f''(x_1)$	—

Выпишем таблицу в виде системы уравнений:

$$\begin{cases} H(x_k) = f(x_k), & k = 0, 1, 2 \\ H'(x_k) = f'(x_k), & k = 0, 1 \\ H''(x_k) = f''(x_k), & k = 1 \end{cases} \tag{30}$$

Система (30) содержит 6 уравнений. Для ее однозначного решения полином $H(x)$ должен иметь 6 коэффициентов, т.е. быть полиномом пятой степени. Общее правило очевидно: степень интерполяционного полинома Эрмита на единицу меньше общего числа условий таблицы. Вспоминая случай с полиномом Лагранжа, для построения которого решение системы оказалось необязательным, возникает вопрос – нельзя ли и полином Эрмита воспроизвести сразу в готовом виде? Ответ оказывается положительным, однако общая формула довольна громоздка. Форма записи будет проще, если исходная система симметрична, т.е. число и вид условий во всех узлах одинаковые.

Отметим, что полином Эрмита второй степени

$$H_2(x) = f(x_0) + \frac{x - x_0}{1!} f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0)$$

с одной стороны является частичной суммой ряда Тейлора, а с другой удовлетворяет условиям

$$H_2(x_0) = f(x_0); \qquad H_2'(x_0) = f'(x_0); \qquad H_2''(x_0) = f''(x_0)$$

что позволяет назвать его еще и интерполяционным полиномом Эрмита с одним узлом интерполирования.

8.4 Обратная интерполяция.

До этого была рассмотрена так называемая *прямая* задача интерполирования, в рамках которой по заданному значению x^* требовалось оценить значение функции $f(x^*)$. В обратной же задаче для такой же таблицы требуется восстановить такое значение аргумента, при котором функция принимает заданное значение. На практике чаще всего используется один из следующих способов.

Способ 1. Меняются местами строки таблицы, в качестве узлов интерполирования выступают значения функции, по которым строится интерполяционный полином для обратной функции. Подставляя в него данное f^* находим искомый x^*. Такой подход возможен, если обратная функция на заданном участке интерполирования существует, то есть исходная функция строго монотонна, что бывает совсем не всегда.

Способ 2. По исходной таблице строится обычный интерполяционный полином $Q_m(x)$ с узлами x_k, а затем решается уравнение $Q_m(x) = f^*$. Для полиномов до 4 степени ответ может быть получен даже аналитически, а в других случаях это уравнение решается численно. В случае немонотонной функции, краевом для предыдущего способа, в этот раз будет найдено несколько корней, из которых необходимо будет выбрать отвечающий поставленной задаче.

Вопрос 9. Квадратурные формулы левых, правых и средних прямоугольников, трапеций, Симпсона. Малые и составные формулы, их остаточные члены.

9.1 Простейшие квадратурные формулы

Definition 9: Квадратурные формулы

Это формулы для вычисления значения определенного интеграла. Их получение – одно из многочисленных возможных приложений интерполяционных полиномов.

Пусть требуется вычислить некий интеграл

$$I = \int_a^b f(x)dx$$

точное значение которого определить весьма сложно или невозможно. Тогда исходная функция может быть аппроксимирована интерполяционным полиномом $f(x) = Q_m(x) + R_m(x)$ и интеграл от интерполяционного полинома порождает некоторую квадратурную формулу

$$I = \int_a^b f(x)dx \approx \int_a^b Q_m(x)dx \tag{31}$$

а интеграл от остаточного члена определяет ее погрешность

$$\varepsilon = \int_a^b R_m(x)dx \tag{32}$$

Будем последовательно подставлять в (31) полиномы различных степеней, начиная с нулевой ($Q_0(x) = f(x_0)$ $\quad I \approx (b-a)f(x_0)$. Наиболее популярными являются следующие три варианта выбора узла x_0:

1. Формула левых прямоугольников

$$x_0 = a, \qquad I \approx (b-a)f(a) \tag{33}$$

2. Формула правых прямоугольников

$$x_0 = b, \qquad I \approx (b-a)f(b) \tag{34}$$

3. Формула средних прямоугольников

$$x_0 = \frac{a+b}{2}, \qquad I \approx (b-a)f(\frac{a+b}{2}) \tag{35}$$

Причины таких названий легко понять из геометрических иллюстраций, откуда видно, что площадь под заданной функцией аппроксимируется площадью соответствующего прямоугольника.
Интегрирование полинома первой степени с узлами интерполирования $x_0 = a$ и $x_1 = b$

$$Q_1(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$$

порождает квадратурную *формулу трапеций*

$$I \approx \frac{b-a}{2} \left(f(a) + f(b) \right) \tag{36}$$

а интегрирование полинома второй степени с узлами $x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b$ приводит к квадратурной *формуле Симпсона*

$$I \approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2} \right) + f(b) \right) \tag{37}$$

9.2 Погрешность малых квадратурных формул

Оценку погрешности будем выполнять на основе двух теорем о средних:

Theorem 9

1. Пусть $f(x)$ и $g(x)$ непрерывны на $[a, b]$, а $g(x)$ еще и знакопостоянна. Тогда найдется такая точка c, что

$$\int_a^b g(x) f(x) dx = f(c) \int_a^b g(x) dx$$

2. Пусть $f(x)$ непрерывна на $[a, b]$, и заданы N точек $x_k \in [a, b]$. Найдется точка $\eta \in [a, b]$ такая, что

$$\frac{1}{N} \sum_{k=1}^{N} f(x_k) = f(\eta)$$

Остаточный член интерполяционного полинома нулевой степени имеет вид

$$R_0(x) = \frac{x - x_0}{1!} f'(\eta)$$

Полезно заметить, что точка η зависит от x, т.е. $\eta = \eta(x)$, как это уже отмечалось при выводе остаточного члена интерполяционного полинома.
Последовательно подставляя значения x_0, вычислим интеграл (32):

$$\varepsilon_{лев.\ пр.} = \int_a^b (x-a) f'(\eta) dx = f'(\eta^*) \int_a^b (x-a) dx = \frac{(b-a)^2}{2} f'(\eta^*) \tag{38}$$

$$\varepsilon_{прав.\ пр.} = \int_a^b (x-b) f'(\eta) dx = f'(\eta^*) \int_a^b (x-b) dx = -\frac{(b-a)^2}{2} f'(\eta^*) \tag{39}$$

Для формулы средних прямоугольников все немного не так, ведь $\left(x - \frac{a+b}{2}\right)$ меняет знак. Для оценки погрешности в этом случае воспользуемся разложением $f(x)$ в ряд в точке $\frac{a+b}{2} = t$

$$f(x) = f(t) + \frac{x-t}{1!}f'(t) + \frac{(x-t)^2}{2!}f''(t)$$

интегрируя его на $[a, b]$. Интеграл от первого слагаемого дает формулу средних прямоугольников, от второго равен нулю, а третий определяет погрешность

$$\varepsilon_{ср.\ пр.} = \frac{1}{2}\int_a^b \left(x - \frac{a+b}{2}\right)^2 f''(\eta)dx = \frac{f''(\eta^*)}{2}\int_a^b \left(x - \frac{a+b}{2}\right)^2 dx = \frac{(b-a)^3}{24}f''(\eta^*) \tag{40}$$

Для оценки погрешности формулы трапеций проинтегрируем остаточный член полинома первой степени

$$\varepsilon_{трап} = \int_a^b \frac{(x-a)(x-b)}{2!}f''(\eta)dx = f''(\eta^*)\int_a^b \frac{(x-a)(x-b)}{2}dx = -\frac{(b-a)^3}{12}f''(\eta^*) \tag{41}$$

Для интеграла от остаточного члена полинома второй степени условия теоремы о среднем не выполняются, и погрешность формулы Симпсона определяется по другому. По этим же трем узлам строится полинома Эрмита уже третьей степени с двумя условиями в центральной точке, погрешность которого и интегрируется.

$$\varepsilon_{Симпс} = -\frac{(b-a)^5}{2880}f^{(4)}(\eta) \tag{42}$$

9.3 Составные квадратурные формулы

Definition 10

В большинстве случаев подынтегральная функция не описывается удовлетворительно полиномами первой и второй степени. Поэтому, для достижения необходимой точности исходный промежуток разбивается на такие малые промежутки, где указанная аппроксимация удачна, на каждом из этих промежутков применяется выбранная квадратурная формула, а результаты складываются. Такие формулы получили название *составных* к.ф.

Разобьем исходный промежуток $[a, b]$ на N равных промежутков $[x_k, x_{k+1}]$

$$h = \frac{b-a}{N}, \qquad x_k = x_0 + kh, \qquad x_0 = a, \qquad x_N = b$$

На каждом промежутке применим квадратурную формулу и сложим:

1. Левые прямоугольники

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k)f(x_k) = \frac{b-a}{N} f(x_k)$$
$$I_{лев.\ пр.} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k)$$

2. Правые прямоугольники

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k)f(x_{k+1}) = \frac{b-a}{N} f(x_{k+1})$$
$$I_{прав.\ пр.} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_{k+1}) = \frac{b-a}{N} \sum_{k=1}^{N} f(x_k)$$

3. Средние прямоугольники

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx (x_{k+1} - x_k)f(x_k + h/2) = \frac{b-a}{N} f(x_k + h/2)$$
$$I_{сред.\ пр.} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{N} \sum_{k=0}^{N-1} f(x_k + h/2) = \frac{b-a}{N} \sum_{k=0}^{N-1} f\left(x_k + \frac{b-a}{2N}\right)$$

4. Трапеции

$$I_k = \int_{x_k}^{x_{k+1}} f(x)dx \approx \frac{x_{k+1} - x_k}{2} \left(f(x_{k+1}) + f(x_k)\right) = \frac{b-a}{2N} \left(f(x_{k+1}) + f(x_k)\right)$$
$$I_{трап} = \sum_{k=0}^{N-1} I_k \approx \frac{b-a}{2N} \sum_{k=0}^{N-1} \left(f(x_{k+1} + f(x_k))\right) = \frac{b-a}{2N} \left(f(a) + 2\sum_{k=1}^{N-1} f(x_k) + f(b)\right)$$

Для получения составной формулы Симпсона будем выбирать N всегда четным, а исходный промежуток разобьем на $N/2$ равных промежутков $[x_k, x_{k+2}]$ длиной $2(b-a)/N$.На каждом из них интеграл равен

$$I_k = \int_{x_k}^{x_{k+2}} f(x)dx \approx \frac{x_{k+2} - x_k}{6} \left(f(x_{k+2}) + 4f(x_{k+1}) + f(x_k)\right) =$$
$$= \frac{b-a}{3N} \left(f(x_{k+2}) + 4f(x_{k+1}) + f(x_k)\right)$$

Суммируя по всем промежуткам, получаем *составную формулу Симпсона*

$$I_{Симпс} = \frac{b-a}{3N} \left[f(a) + 4(f_1 + f_3 + \cdots + f_{N-1}) + 2(f_2 + f_4 + \cdots + f_{N-2}) + f(b)\right] \quad (43)$$

9.4 Погрешности составных квадратурных формул

Для оценки погрешностей составных формул вычисляем погрешность каждого участка и складываем.

1. Левые прямоугольники

$$\varepsilon_k = \frac{(x_{k+1} - x_k)^2}{2} f'(\eta_k) = \frac{(b-a)^2}{2N^2} f'(\eta_k)$$
$$\varepsilon_{\text{лев. пр.}} = \sum_{k=1}^{N} \varepsilon_k = \frac{(b-a)^2}{2N} \left[\frac{1}{N} \sum_{k=1}^{N} f'(\eta_k) \right] = \frac{(b-a)^3}{24N^2} f''(\eta)$$

2. Правые прямоугольники

$$\varepsilon_{\text{прав. пр.}} = -\frac{(b-a)^2}{2N} f'(\eta)$$

3. Средние прямоугольники

$$\varepsilon_k = \frac{(x_{k+1} - x_k)^3}{24} f''(\eta_k) = \frac{(b-a)^3}{24N^3} f''(\eta_k)$$
$$\varepsilon_{\text{сред. пр.}} = \sum_{k=1}^{N} \varepsilon_k = \frac{(b-a)^3}{24N2} f''(\eta)$$

4. Трапеции

$$\varepsilon_k = -\frac{(x_{k+1} - x_k)^3}{12} f''(\eta_k) = -\frac{(b-a)^3}{12N^3} f''(\eta_k)$$
$$\varepsilon_{\text{трап}} = \sum_{k=1}^{N} \varepsilon_k = -\frac{(b-a)^3}{12N^2} f''(\eta)$$

5. Формула Симпсона
   Тут следует учитывать, что длина каждого участка в два раза больше, чем ранее, а число таких участков в два раза меньше – $N/2$.

$$\varepsilon_k = -\frac{(x_{k+2} - x_k)^5}{2880} f^{(4)}(\eta_k) = -\frac{(b-a)^5}{90N^5} f^{(4)}(\eta_k)$$
$$\varepsilon_{\text{Симпс}} = \sum_{k=1}^{N/2} \varepsilon_k = -\frac{(b-a)^5}{180N^4} \left[\frac{1}{N/2} \sum_{k=1}^{N/2} f^{(4)}(\eta_k) \right] = -\frac{(b-a)^5}{180N^4} f^{(4)}(\eta)$$

Заметим, что все эти формулы описываются общей зависимостью

$$\varepsilon = \alpha \frac{(b-a)^{p+1}}{N^p} f^{(p)}(\eta) \tag{44}$$

Вопрос 10. Общий подход к построению квадратурных формул. Квадратурные формулы Ньютона-Котеса, Чебышева, Гаусса.

10.1 Общий подход к построению квадратурных формул.

Заметим, что все полученные выше простейшие квадратурные формулы имеют следующий вид

$$\int_a^b f(x)dx \approx \sum_{k=1}^{S} A_k f(x_k) \tag{45}$$

Узлы x_k и *веса* A_k квадратурной формулы получались на основе интегрирования соответствующих интерполяционных полиномов. Поставим задачу несколько иначе. Требуется выбрать узлы и веса так, чтобы формула (45) была бы *точной* для полиномов заданной степени. Логика таких требований очевидна. Если подынтегральная функция хорошо аппроксимируется этим полиномом, то и формула (45) обеспечит требуемую погрешность решения задачи. В противном случае, промежуток $[a, b]$ всегда можно разбить на достаточно малые промежутки и применить составные квадратурные формулы.
Потребуем, чтобы формула (45) была бы точна для полинома нулевой степени $f(x) = \alpha = const$. Вынося константу α из-под знаков интеграла и суммы и сокращая на нее, имеем

$$\sum_{k=1}^{S} A_k = b - a$$

Второе уравнение получим, требуя точности для полинома первой степени и поставляя с этой целью $f(x) = x$.

$$\sum_{k=1}^{S} A_k x_k = \frac{b^2 - a^2}{2}$$

Это же требование для $f(x) = x^2$ выглядит следующим образом

$$\sum_{k=1}^{S} A_k x_k^2 = \frac{b^3 - a^3}{3}$$

а в общем случае для $f(x) = x^N$ условие с номером $N + 1$ имеет вид

$$\sum_{k=1}^{S} A_k x_k^N = \frac{b^{N+1} - a^{N+1}}{N + 1}$$

Объединяя все уравнения, получим систему из $N + 1$ уравнения относительно $2S$ неизвестных x_k и A_k. Эта система является общей для многих семейств квадратурных формул, отличающихся друг от друга дополнительными условиями, накладываемыми на x_k и A_k.

10.2 Квадратурные формулы Ньютона-Котеса

Узлы квадратурной формулы здесь выбираются равноотстоящими

$$h = \frac{b-a}{S-1}, \qquad x_k = a + (k-1)h, \qquad x_1 = a \quad x_S = b$$

В данном случае система является линейной относительно S неизвестных A_k и легко решаема. Ее определитель является определителем Вандермонда, что обеспечивает единственность решения. Для составных квадратурных формул при удвоении N (числа внутренних промежутков) в половине возникающих узлов x_k значения функции $f(x_k)$ уже вычислялись ранее и могли быть сохранены, что позволяет сократить объем вычислений вдвое.
Как результат, имеем систему из S уравнений с S неизвестными A_k, и эти квадратурные формулы оказываются гарантированно точными для полиномов степени $N = S - 1$.
Заметим, что, решая систему последовательно для $S = 1, 2, 3$, можно прийти к уже знакомым формулам прямоугольников, трапеций и Симпсона.

10.3 Квадратурные формулы Чебышева

Когда значения $f(x_k)$ определены с заметной погрешностью, удобно установить равные веса и решать задачу только относительно x_k.

$$\int_a^b f(x)dx \approx A\sum_{k=1}^{S} f(x_k)$$

Получаем систему из $S + 1$ уравнения относительно такого же числа неизвестных x_k (S) и A $(+1)$, что позволяет в случае успешного решения получить формулы, гарантированно точные для полиномов степени $N = S$. Однако, учитывая нелинейность системы, вопросы существования и единственности выходят на первый план. Оказывается, что система решается единственным образом для $S \in [1; 7] \cup \{9\}$. С.Н.Берштейном было показано, что для других значений S формулы Чебышева не существуют.

10.4 Квадратурные формулы Гаусса

В этих формулах на узлы и веса не накладываются никакие дополнительные условия, и все свободные $2S$ параметров используются при решении системы из $2S$ уравнений. В отличии от формул Чебышева, формулы Гаусса существуют для любого числа узлов. Они гарантированно точны для полиномов степени $N = 2S - 1$ и называются *формулами наивысшей алгебраической степени точности*.

На практике, для получения A_k и x_k нет никакой необходимости обращаться к системе. Результаты ее решения для промежутков $[-1, 1]$ и $[0, 1]$ приведены в различных справочниках, а пользователю достаточно лишь воспользоваться заменой переменных в (45). Так для промежутка $[-1, 1]$ эта замена выглядит следующим образом: $x = \frac{a+b}{2} + \frac{b-a}{2}t$. При изменении значений t от -1 до 1, переменная x пробегает

значения от a до b. Учитывая $dx = \frac{b-a}{2}dt$, формула (45) имеет вид

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right)dt \approx \frac{b-a}{2}\sum_{k=1}^{S} A_k f\left(\frac{a+b}{2} + \frac{b-a}{2}t_k\right) \quad (46)$$

где веса A_k и узлы t_k берутся из справочника.

Вопрос 11. Адаптивные квадратурные формулы. Подпрограмма QUANC8.

Рассмотренный ранее простой алгоритм вычисления интеграла на основе составных квадратурных формул со сравнением результата для N и $2N$, отличается достаточной надежностью, но его быстродействие может быть заметно повышено. Например, если функция в основном ведет себя стабильно, а на каком-то малом участке резко меняется, то имеет смысл требовать подсчет с малым шагом только на этом участке, а не на всем рассматриваемом промежутке.

Целесообразным представляется построение алгоритма, который был бы способен *адаптироваться* к виду функции и выбирать достаточно малый шаг там, где функция меняется быстро и характеризуется большими производными, и относительно большой шаг там, где функция меняется медленно. На этом пути возможны два варианта: минимизировать погрешность при заданном объеме вычислений или минимизировать объем вычислений при заданных требованиях к погрешности. В рассматриваемой программе реализован второй подход.

В основу положена квадратурная формула Ньютона-Котеса с девятью узлами, т.е. восемью промежутками между ними, что и оправдывает название программы. Ее составная формула имеет погрешность вида (32) для $p = 10$.

Рассмотрим промежуток длиной h_k внутри $[a, b]$ и введем для него следующие обозначения:

I_k – точное значение интеграла на этом промежутке

P_k – значение интеграла, вычисленное по квадратурной формуле с девятью узлами

Q_k – значение интеграла, вычисленное по той же формуле, примененной к двум половинам этого промежутка (по сути используется составная формула с вдвое большим значением N)

Учитывая вид (32) для $p = 10$, для погрешностей P_k и Q_k последовательно имеем

$$I_k - P_k \approx 2^p \left(I_k - Q_k\right); \qquad I_k \approx \frac{2^p Q_k - P_k}{2^p - 1}; \qquad I_k - Q_k \approx \frac{Q_k - P_k}{2^p - 1} = \frac{Q_k - P_k}{1023}$$

Обозначая требуемую абсолютную погрешность вычисления интеграла на всем промежутке $[a, b]$ за ε_A, считаем промежуток h_k «принятым», а интеграл на нем вычисленным, если выполняется неравенство

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\varepsilon_A \tag{47}$$

Множитель $\dfrac{h_k}{b-a}$ является весовым коэффициентом и отражает вклад погрешности на промежутке h_k в общую погрешность для всего промежутка. Возможно также использование и относительной погрешности ε_R

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\varepsilon_R \left|\tilde{I}_k\right| \tag{48}$$

где $\tilde{I}_k$ – оценка вычисления интеграла по всему промежутку. Следует, однако, помнить, что использование критерия относительной погрешности усложняется, если

значение $\tilde{I}_k$ оказывается нулевым или близким к нулю. В программе `QUANC8` пользователю предоставляется возможность использовать один из трех вариантов контроля погрешности на основе формул (47) и (48)

$$\left|\frac{Q_k - P_k}{1023}\right| \le \frac{h_k}{b-a}\max\left(\varepsilon_A; \varepsilon_R\left|\tilde{I}_k\right|\right) \tag{49}$$

1. $\varepsilon_R = 0, \quad \varepsilon_A \neq 0$ – контроль абсолютной погрешности
2. $\varepsilon_R \neq 0, \quad \varepsilon_A = 0$ – контроль относительной погрешности
3. $\varepsilon_R \neq 0, \quad \varepsilon_A \neq 0$ – контроль «смешанной» погрешности

В последнем случае делается попытка избежать упомянутых неприятных ситуаций со значениями $\tilde{I}_k$, близкими к нулю.
Адаптация программы к виду функции и реализация переменного шага интегрирования реализуются в соответствии со следующим алгоритмом. Вычисляются P_k и Q_k применительно ко всему промежутку. Если погрешность еще достаточно велика, промежуток делится пополам, значения подынтегральной функции $f(x)$, вычисленные на правой половине промежутка, запоминаются, и все повторяется для левой половины промежутка. Такое обращение каждый раз к левой половине текущего промежутка продолжается до тех пор, пока крайний слева промежуток не будет принят. После этого обрабатывается ближайший к нему правый промежуток. Запоминание значений $f(x)$ повышает быстродействие алгоритма.
В программе реализовано два ограничения сверху на объем вычислений. Во-первых, деление промежутка пополам продолжается не более 30 раз. По достижении этой величины соответствующий интеграл на нем считается вычисленным, а промежуток «принятым», независимо от условия (49). Число таких промежутков, принятых с нарушением условия (49), содержится в целой части выходного значения переменной `FLAG`. Длина каждого такого промежутка крайне мала ($\approx 10^{-9}$), и подобная ситуация, как правило, связана с разрывами подынтегральной функции или ее «зашумлением» вычислительной погрешностью. Во-вторых, вводится ограничение сверху на количество вычислений подынтегральной функции $f(x)$. Если этот предел достигнут, то информация о точке x^*, где возникла трудность, отражена в дробной части выходного значения переменной `FLAG`.
Например, для `FLAG` $= 91.21$, число промежутков равно 91, а

$$\frac{b - x^*}{b-a} = 0.21; \qquad x^* = b - 0.21(b-a)$$

Программа имеет следующие параметры
FUN – имя подпрограммы-функции, вычисляющей значение подынтегральной функции $f(x)$;
A, B – нижний и верхний пределы интегрирования;
ABSERR, RELERR – границы абсолютной ε_A и относительной ε_R погрешностей.
Остальные параметры – выходные со следующим смыслом:
RESULT – значение интеграла, определенное программой;
ERREST – оценка погрешности, выполненная программой и удовлетворяющая (49);
NOFUN – количество вычислений подынтегральной функции $f(x)$, использованных для получения результата;
FLAG – индикатор надежности результата. Нулевое значение этой переменной отвечает относительной надежности результата, а ненулевое, как уже отмечалось, свидетельствует об отклонениях от нормального хода выполнения программы.

Вопрос 12. Задача численного дифференцирования. Влияние вычислительной погрешности.

12.1 Задача численного дифференцирования.

Предлагаемая задача ставится следующим образом. Для таблично заданной функции $f(x)$ требуется оценить значения производной функции в узлах таблицы.
Идея, лежащая в основе численного дифференцирования, крайне проста и уже использовалась при получении квадратурных формул. Исходная функция аппроксимируется интерполяционным полиномом

$$f(x) = Q_m(x) + R_m(x)$$

и производная от полинома дает формулу численного дифференцирования

$$\frac{df(x_k)}{dx} \approx \frac{dQ_m(x_k)}{dx}$$

а производная от остаточного члена позволяет оценить погрешность этой операции

$$\varepsilon = \frac{dR_m(x_k)}{dx}$$

Ограничимся случаем, когда узлы таблицы будут равноотстоящими с шагом $h = x_{k+1} - x_k$. Начнем с полинома первой степени, построенного по двум узлам x_k и x_{k+1}

$$Q_1(x) = \frac{x - x_{k+1}}{x_k - x_{k+1}} f(x_k) + \frac{x - x_k}{x_{k+1} - x_k} f(x_{k+1})$$

Дифференцируя его и полагая последовательно $x = x_k$ и $x = x_{k+1}$, получаем

$$\frac{df(x_k)}{dx} \approx \frac{f_{k+1} - f_k}{h} \tag{50}$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+1} - f_k}{h} \tag{51}$$

Хотя правые части обоих выражений равны, формулы получились принципиально различными. Выполним аналогичные операции для полинома второй степени

$$\begin{aligned} Q_2(x) =& \frac{(x - x_{k+1})(x - x_{k+2})}{(x_k - x_{k+1})(x_k - x_{k+2})} f(x_k) + \frac{(x - x_k)(x - x_{k+2})}{(x_{k+1} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+1}) + \\ &+ \frac{(x - x_k)(x - x_{k+1})}{(x_{k+2} - x_k)(x_{k+1} - x_{k+2})} f(x_{k+2}) \end{aligned}$$

Последовательно полагая $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$, получаем

$$\frac{df(x_k)}{dx} \approx \frac{-3f_k + 4f_{k+1} - f_{k+2}}{2h} \tag{52}$$

$$\frac{df(x_{k+1})}{dx} \approx \frac{f_{k+2} - f_k}{2h} \tag{53}$$

$$\frac{df(x_{k+2})}{dx} \approx \frac{3f_{k+2} - 4f_{k+1} + f_k}{2h} \tag{54}$$

Для оценки погрешности всех формул необходимо продифференцировать остаточный член $R_m(x)$. Для полинома первой степени он имеет вид

$$R_1(x) = \frac{(x-x_k)(x-x_{k+1})}{2!} f''(\eta)$$

$$\frac{dR_1(x)}{dx} = \frac{x-x_{k+1}}{2!} f''(\eta) + \frac{x-x_k}{2!} f''(\eta) + \frac{(x-x_k)(x-x_{k+1})}{2!} f'''(\eta)\eta'(x)$$

Спасает ситуацию то, что оценивать погрешность нужно в узлах интерполирования. Так как при $x = x_k$ и $x = x_{k+1}$ два слагаемых из трех в этой формуле обращаются в ноль

$$\varepsilon_1(x_k) = \frac{dR_1(x_k)}{dx} = -\frac{h}{2} f''(\eta) \tag{55}$$

$$\varepsilon_1(x_{k+1}) = \frac{dR_1(x_{k+1})}{dx} = \frac{h}{2} f''(\eta) \tag{56}$$

Эти два выражения задают погрешность численного дифференцирования для формул (50) и (51) соответственно. Аналогично продифференцируем погрешность $R_2(x)$

$$R_2(x) = \frac{(x-x_k)(x-x_{k+1})(x-x_{k+2})}{3!} f'''(\eta)$$

Последовательно подставляя в результат $x = x_k$, $x = x_{k+1}$ и $x = x_{k+2}$, получаем

$$\varepsilon_2(x_k) = \frac{dR_2(x_k)}{dx} = \frac{h^2}{3} f'''(\eta) \tag{57}$$

$$\varepsilon_2(x_{k+1}) = \frac{dR_2(x_{k+1})}{dx} = -\frac{h^2}{6} f'''(\eta) \tag{58}$$

$$\varepsilon_2(x_{k+2}) = \frac{dR_2(x_{k+2})}{dx} = \frac{h^2}{3} f'''(\eta) \tag{59}$$

Заметим, что на меньшую погрешность можно рассчитывать, используя формулу (53), которая и является наиболее популярной на практике. Формулы (52) и (54) используются для дифференцирования в начале и в конце таблицы соответственно.

Если интерполяционный полином второй степени продифференцировать дважды, то получается простейшая формула для второй производной

$$\frac{d^2 f(x_{k+1})}{dx^2} \approx \frac{f_{k+2} - 2f_{k+1} + f_k}{h^2} \tag{60}$$

Практически важным является вопрос о выборе шага h для формул численного дифференцирования. Ограничение сверху накладывается величиной погрешности ε_2, а снизу – точностью задания табличных данных для $f(x)$.

12.2 Влияние погрешности задания функции на точность.

В качестве примера вновь обратимся к простейшей формуле для первой производной (50). Пусть в ней значение f_{k+1} определено с погрешностью Δ_{k+1}, а значение f_k – Δ_k. Тогда общая погрешность $\varepsilon(h)$ складывается из двух погрешностей

$$\varepsilon(h) = \varepsilon_1(h) + \varepsilon_2(h)$$

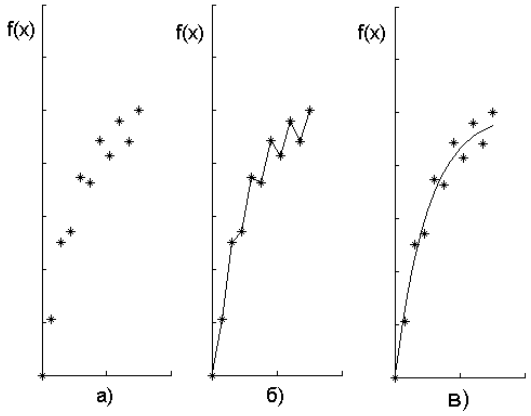
первая из которых задается формулой (56) и примерно линейно убывает с уменьшением шага h, а вторая зависит от Δ_k и Δ_{k+1}. Оценивая полную погрешность сверху, получаем

$$|\varepsilon(h)| \leq |\varepsilon_1(h)| + |\varepsilon_2(h)| = \frac{h}{2}\,|f''(\eta)| + \frac{|\Delta_{k+1} - \Delta_k|}{h} \tag{61}$$

Оптимальное значение шага h_{opt} отвечает ситуации, когда оба слагаемых равны друг другу. На практике в точном определении h_{opt} нет необходимости, важно лишь знать о характере зависимости (61).

Вопрос 13. Среднеквадратичная аппроксимация (дискретный случай). Понятие веса.

Критерий интерполирования, предполагающий совпадение исходной и аппроксимирующей функции в узлах таблицы, не является единственным. Обратимся к экспериментальным данным, представленным на рисунке (а).



Если выполнить по ним интерполяцию, то получится кривая на рис. (б). Маловероятно, чтобы ее вид отвечал исходной зависимости, положенной в основу таблицы. Вероятнее всего, что этой зависимости отвечает кривая, похожая на рис. (в), а отклонение экспериментальных данных от нее продиктовано сравнительно большой погрешностью измерений. В таком случае целесообразно использовать среднеквадратичный критерий

$$\rho^2 = \sum_{k=1}^{N} \left(f(x_k) - g(x_k)\right)^2$$

или, если аппроксимируемая функция задана непрерывно,

$$\rho^2 = \int_a^b \left(f(x_k) - g(x_k)\right)^2 dx$$

Близость функций по среднеквадратичному критерию еще не гарантирует малой величины их максимальной разности

$$\delta = \max_{[a,b]} |f(x) - g(x)|$$

Малое значение интеграла или суммы свидетельствует лишь о том, что почти на всем отрезке $[a, b]$ значения $f(x)$ и $g(x)$ мало отличаются друг от друга, хотя в отдельных точках или на небольших отрезках разность их значений может быть значительной. Проблема выбора аппроксимирующей функции решается так же, как и при интерполяции: $Q(x)$ выбирается в виде обобщенного многочлена:

$$Q_m(x) = \sum_{k=0}^{m} a_k \varphi_k(x) \tag{62}$$

где $\{\varphi_k\}$ – заданный набор линейно независимых функций, а коэффициенты a_k подлежат определению.

13.1 Дискретный случай. Весовые коэффициенты.

Функция $f(x)$ задается на дискретном множестве точек таблицей, а ее аппроксимация $Q_m(x)$ выбирается в виде обобщенного многочлена (62). Коэффициенты a_k выбираются из условия минимума величины ρ^2

$$\rho^2 = \sum_{i=1}^{N} \left(Q_m(x_i) - f(x_i)\right)^2 \to \min \tag{63}$$

Рассмотрим три варианта соотношения N и m.

1. $N = m + 1$
   Число коэффициентов a_k равно числу точек таблицы, решение задачи единственное, и им является интерполяционный полином, проходящий через все точки. Минимальное значение $\rho^2 = 0$.
2. $N < m + 1$
   Минимальное значение ρ^2 также равно нулю, но задача имеет бесконечное множество решений.
3. $N > m + 1$
   Это типичный случай среднеквадратичной аппроксимации. Более того, на практике часто $N \gg m+1$. Минимальное значение ρ^2 оказывается уже, как правило, ненулевым, а задача имеет единственное решение. Рассмотрим этот вариант подробнее.

Записываем необходимое условие экстремума

$$\frac{\partial \rho^2}{\partial a_k} = 0, \qquad k = 0,1,\ldots,m$$

и выполняем операцию дифференцирования:

$$\frac{\partial \rho^2}{\partial a_k} = 2 \sum_{i=1}^{N} \left(Q_m(x_i) - f(x_i)\right) \varphi_k(x_i) = 0$$

Подставляя в получившуюся формулу выражение для $Q_m(x)$, получаем систему линейных алгебраических уравнений относительно a_k:

$$a_0 \sum_{i=1}^{N} \varphi_0(x_i)\cdot\varphi_k(x_i) + a_1 \sum_{i=1}^{N} \varphi_1(x_i)\cdot\varphi_k(x_i) + \cdots + a_m \sum_{i=1}^{N} \varphi_m(x_i)\cdot\varphi_k(x_i) = \sum_{i=1}^{N} f(x_i)\cdot\varphi_k(x_i) \tag{64}$$

Если определитель системы (64) не равен нулю, задача имеет единственное решение. Самой популярной является аппроксимация полиномами, когда $\varphi_k(x) = x^k$, а система (64) принимает вид

$$\left(\sum_{i=1}^{N} 1\right) a_0 + \left(\sum_{i=1}^{N} x_i\right) a_i + \cdots + \left(\sum_{i=1}^{N} x^m\right) = \left(\sum_{i=1}^{N} f(x_i)\right)$$
$$\left(\sum_{i=1}^{N} x_i\right) a_0 + \left(\sum_{i=1}^{N} x_i^2\right) a_i + \cdots + \left(\sum_{i=1}^{N} x^{m+1}\right) = \left(\sum_{i=1}^{N} f(x_i)x_i\right)$$

$$\dots \tag{65}$$
$$\left(\sum_{i=1}^{N} x_i^m\right) a_0 + \left(\sum_{i=1}^{N} x_i^{m+1}\right) a_i + \dots + \left(\sum_{i=1}^{N} x^{2m}\right) = \left(\sum_{i=1}^{N} f(x_i) x_i^m\right)$$

При выполнении среднеквадратичной аппроксимации (другое название – «метод наименьших квадратов») возможна ситуация, когда исходные данные имеют различную точность. Если к каким-либо экспериментальным значениям доверие выше, т.е. они являются более надежными по сравнению с другими, это может быть учтено введением в критерий (63) положительных весовых коэффициентов p_i.

$$\rho^2 = \sum_{i=1}^{N} p_i \left(Q_m(x_i) - f(x_i)\right)^2 \to \min \tag{66}$$

Для тех точек, степень доверия к которым выше, и к которым аппроксимирующую кривую желательно провести ближе, чем к другим точкам, весовые коэффициенты следует задавать больше. Величину ρ^2 можно трактовать, как своеобразную «функцию штрафа». За отклонение $Q_m(x)$ от $f(x)$ в точке x_i к значению ρ^2 добавляется слагаемое («штраф») $\left(Q(x_i) - f(x_i)\right)^2$ тем большее, чем больше это отклонение. Если какая-то точка является более приоритетной и к ней аппроксимирующую кривую желательно провести ближе, с помощью весового коэффициента за отклонение в этой точке «штраф» должен быть увеличен.

На практике положительные весовые коэффициенты p_i часто задают так, чтобы их сумма была равна, например, единице или 100. Последнее часто удобно, но не обязательно. Если все коэффициенты умножить на одно и то же число, то, хотя, ρ^2 и изменится, решение задачи останется прежним. Важным являются отношения p_i друг к другу.

Вопрос 14. Среднеквадратичная аппроксимация (непрерывный случай). Понятие ортогональности.

Теперь обратимся к варианту непрерывного задания $f(x)$ на $[a,b]$

$$\rho^2 = \int_a^b \left(Q(x) - f(x)\right)^2 dx \to \min \tag{67}$$

СЛАУ относительно a_k сохранит прежний вид, только вместо сумм будут интегралы

$$a_0 \int_a^b \varphi_0(x)\cdot\varphi_k(x)dx + a_1 \int_a^b \varphi_1(x)\cdot\varphi_k(x)dx + \cdots + a_m \int_a^b \varphi_m(x)\cdot\varphi_k(x)dx = \int_a^b f(x)\cdot\varphi_k(x)dx \tag{68}$$

Аналогично дискретному случаю критерий ρ^2 может быть обобщен введением положительной весовой функции $p(x)$

$$\rho^2 = \int_a^b p(x)\left(Q(x) - f(x)\right)^2 dx \to \min \tag{69}$$

Все формулы сохранят прежний вид, а под знаком интеграла появится $p(x)$

$$a_0 \int_a^b p(x)\cdot\varphi_0(x)\cdot\varphi_k(x)dx + \cdots + a_m \int_a^b p(x)\cdot\varphi_m(x)\cdot\varphi_k(x)dx = \int_a^b p(x)\cdot f(x)\cdot\varphi_k(x)dx \tag{70}$$

Решение системы (70) значительно упрощается, если вместо произвольных линейно независимых функций $\{\varphi_k(x)\}$ воспользоваться *ортогональными* функциями $\{g_k(x)\}$.

Definition 11: Ортогональные функции

Последовательность функций $\{g_k(x)\}$ является ортогональной на промежутке $[a,b]$ с весом $p(x)$, если выполняются следующие условия

$$\int_a^b g_k(x)g_i(x)dx = \begin{cases} 0, \text{ если } i \neq k \\ A > 0, \text{ если } i = k \end{cases}$$

Если в дополнение $A = 1$, то такие функции называются *ортонормированными*.

Для ортогональных функций все интегралы, кроме одного, в левой части (70) равны нулю, матрица этой системы оказывается диагональной, и каждое уравнение дает готовое выражение для коэффициента

$$a_k = \frac{\int_a^b p(x)f(x)\varphi_k(x)dx}{\int_a^b p(x)\varphi_k^2(x)dx} \tag{71}$$

Если исходный базис не является ортогональным, его можно сделать таковым, используя процедуру Грама-Шмидта.

Вопрос 15. Ортогонализация по Шмидту. Примеры ортогональных полиномов.

15.1 Процедура ортогонализации Грама-Шмидта.

Для начала обозначим задачу. Пусть задан набор линейно независимых функций $\{\varphi_k(x)\}$. Требуется построить набор ортогональных функций $\{g_k(x)\}$, которые будут являться линейной комбинацией функций $\{\varphi_k(x)\}$. Аппроксимация, таким образом, будет выполняться в том же классе функций.
Введем следующее обозначение: $\tilde{g_k}(x)$ – функции ортогональные, но еще не нормированные. Очередная функция $\tilde{g_k}(x)$ строится так, чтобы она была ортогональна всем $\tilde{g_i}(x)$, построенным до нее.

Шаг 1.
$\tilde{g_0}(x) = \varphi_0(x)$. Нормируем ее.

$$\int_a^b p(x)\tilde{g_0}^2 dx = \alpha_0^2, \quad g_0(x) = \frac{\tilde{g_0}}{\alpha_0}$$

Шаг m.
Функция $\tilde{g_m}(x)$ строится с привлечением новой $\varphi_m(x)$ и добавлением линейной комбинации функций $g_k(x)$, построенных на предыдущих шагах.

$$\tilde{g_m}(x) = \varphi_m(x) - \sum_{k=0}^{m-1} C_{m,k} \cdot g_k(x)$$

Навесим на каждое слагаемое интеграл:

$$\int_a^b p(x) \cdot \tilde{g_m}(x) \cdot g_i(x)dx = \int_a^b p(x) \cdot \varphi_m(x) \cdot g_i(x)dx - \sum_{k=0}^{m-1} C_{m,k} \cdot \int_a^b p(x) \cdot g_k(x) \cdot g_i(x)dx = 0$$

Из условия ортогональности $\tilde{g_m}(x)$ и $g_i(x)$ интеграл в левой части уравнения равен нулю. Все интегралы под знаком суммы, кроме одного, тоже равны нулю и для $C_{m,i}$ имеем

$$C_{m,i} = \frac{\int_a^b p(x) \cdot \varphi_m(x) \cdot g_i(x)dx}{\int_a^b p(x) \cdot g_i^2(x)dx}, \quad \int_a^b p(x)\tilde{g_m}^2(x)dx = \alpha_m^2, \quad g_m(x) = \frac{\tilde{g_m}}{\alpha_m}$$

15.2 Примеры ортогональных полиномов.

Неотъемлемыми атрибутами понятия ортогональности являются промежуток интегрирования и весовая функция. Проблема различных промежутков, возникающих на практике, решается легко. Полиномы для стандартных промежутков ($[-1, 1]$, $[0, 1]$) приводятся в справочниках и учебниках, а к произвольному промежутку переходят обычной заменой переменных. Примером является следующая замена

$$x = \frac{a+b}{2} + \frac{b-a}{2}t, \qquad x \in [a, b], \quad t \in [-1, 1]$$

В приводимых примерах остановимся на промежутке $[-1, 1]$. Тогда главной отличительной особенностью различных полиномов будет весовая функция $p(x)$.

1. Ортогональные полиномы Лежандра

   Для этих полиномов весовая функция имеет популярный вид: $p(x) \equiv 1$ и сами они могут быть вычислены по формуле

   $$L_n(x) = \frac{(-1)^n}{n!\,2^n}\frac{d^n}{dx^n}\left[\left(1-x^2\right)^n\right], \qquad x \in [-1,1] \tag{72}$$

   Заметим, что $L_0(x) = 1, L_1(x) = x$, а следующие полиномы можно найти при помощи рекуррентной формулы:

   $$(n+1)L_{n+1}(x) - (2n+1)xL_n(x) + nL_{n-1}(x) = 0$$

   Применив ее, получаем

   $$L_2(x) = \frac{3x^2-1}{2}, \quad L_3(x) = \frac{5x^3-3x}{2}, \quad L_4(x) = \frac{35x^4-30x^2+3}{8}\dots$$

   В такой форме полиномы Лежандра ортогональны, но не нормированы. Квадрат нормы:

   $$\int_{-1}^{1} L_n^2(x)dx = \frac{2}{2n+1}$$

2. Ортогональные полиномы Чебышева

   Для этих полиномов весовая функция выглядит следующим образом:

   $$p(x) = \frac{1}{\sqrt{1-x^2}}.$$

   При $x \in [-1, 1]$ они могут быть вычислены по формуле

   $$T_0(x) = 1, \quad T_1(x) = x, \quad T_n(x) = \cos\left(n \cdot \arccos(x)\right) \tag{73}$$

   Как и для полиномов Лежандра, здесь имеет место следующее рекуррентное соотношение:

   $$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

   Применив ее, получаем

   $$T_2(x) = 2x^2-1, \quad T_3(x) = 4x^3-3x, \quad T_4(x) = 8x^4-8x^2+1\dots$$

   Полиномы Чебышева могут быть представлены и в другом виде:

   $$T_n(x) = \frac{(-2)^n n!}{(2n)!}\sqrt{1-x^2}\frac{d^n}{dx^n}\left[\left(1-x^2\right)^{n-1/2}\right] \tag{74}$$

   Квадраты нормы:

   $$(T_0, T_0) = \pi, \qquad (T_n, T_n) = \int_{-1}^{1}\frac{T_n^2(x)}{\sqrt{1-x^2}}dx = \frac{\pi}{2}$$

Вопрос 16. Обратная матрица, собственные числа и векторы. Задачи на матрицы. Норма матрицы, сходимость матричного степенного ряда, функции от матрицы.

16.1 Обратная матрица.

Definition 12: Матрица

Матрицей называется совокупность $n \times m$ скаляров a_{ij}, образующих прямоугольную таблицу из n строк и m столбцов.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

В литературе и, в частности, в этом конспекте матрицы обозначаются **жирным** шрифтом.

Definition 13: Обратная матрица

Для квадратной матрицы с ненулевым определителем ($\det(\mathbf{A}) \neq 0$) вводится понятие *обратной матрицы* $\mathbf{X} = \mathbf{A}^{-1}$, которая существует, является единственной и удовлетворяет условиям $\mathbf{A} \cdot \mathbf{X} = \mathbf{E}, \quad \mathbf{X} \cdot \mathbf{A} = \mathbf{E}$

16.2 Собственные числа и векторы.

Поставим задачу: для заданной матрицы $\mathbf{A}$ найти такие векторы $\mathbf{u}$, которые сохраняют свое направление после линейного преобразования.

$$\mathbf{y} = \mathbf{A}\mathbf{u} = \lambda\mathbf{u} \Leftrightarrow (\mathbf{A} - \lambda\mathbf{E})\,\mathbf{u} = 0 \tag{75}$$

Definition 14

Числа λ и векторы $\mathbf{u}$, удовлетворяющие уравнению (75) получили название *собственные значения* и *собственные векторы*.

Если определитель $\det(\mathbf{A} - \lambda\mathbf{E})$ не равен нулю, то система имеет единственное решение $\mathbf{u} = 0$. Для того, чтобы существовало решение, отличное от нулевого потребуем

равенство нулю определителя. Запишем это в покомпонентном виде:

$$\det\left(\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix} - \begin{pmatrix} \lambda & 0 & \dots & 0 \\ 0 & \lambda & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda \end{pmatrix}\right) =$$

$$= \det\begin{pmatrix} a_{11}-\lambda & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22}-\lambda & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm}-\lambda \end{pmatrix} = 0$$

Вычисление этого определителя дает:

$$\det(\mathbf{A} - \lambda\mathbf{E}) = (-1)^n\lambda^n + b_1\lambda^{n-1} + \dots + b_{n-1}\lambda + b_n = 0$$

Полином, стоящий в левой части уравнения, называется *характеристическим*, как и само уравнение. Оно имеет ровно n корней (с учетом кратности). Это и есть собственные значения матрицы $\lambda_1, \lambda_2, \dots, \lambda_n$. Они составляют спектр матрицы $\mathbf{A}$, а величина $\rho(\mathbf{A}) = \max|\lambda_i|$ называется *спектральным радиусом*. Для каждого λ_i можно найти решение $\mathbf{u}_i$, т.е. собственный вектор.
Поскольку собственные векторы находятся из решения однородной системы, то и неизвестными они оказываются с точностью до постоянного (ненулевого) множителя, т.е. собственные векторы однозначно определены по направлению, но их длины (нормы) остаются произвольными. Часто бывает удобно приводить их к единичной длине. Без доказательства оставим следующую теорему:

Theorem 10

Если все собственные значения матрицы различны, то отвечающие им собственные векторы – линейно-независимы.

16.3 Задачи на матрицы.

Не умаляя общности, будем считать исходные матрицы квадратными. Те же задачи можно доказать и для матриц в общем виде, достаточно лишь поменять соответствующие индексы.

Theorem 11

$$(\mathbf{A}\cdot\mathbf{B})^T = \mathbf{B}^T\cdot\mathbf{A}^T$$

Доказательство.

Рассмотрим отдельно левую и правую части равенства. Правая:

$$\mathbf{B}^T\cdot\mathbf{A}^T = \begin{pmatrix} b_{11} & \dots & b_{n1} \\ \dots & \dots & \dots \\ b_{1n} & \dots & b_{nn} \end{pmatrix} \cdot \begin{pmatrix} a_{11} & \dots & a_{n1} \\ \dots & \dots & \dots \\ a_{1n} & \dots & a_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} b_{11}\cdot a_{11} + \dots + b_{n1}\cdot a_{1n} & \dots & b_{11}\cdot a_{n1} + \dots + b_{n1}\cdot a_{nn} \\ \dots & \dots & \dots \\ b_{1n}\cdot a_{11} + \dots + b_{nn}\cdot a_{1n} & \dots & b_{1n}\cdot a_{n1} + \dots + b_{nn}\cdot a_{nn} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}\cdot b_{11} + \dots + a_{1n}\cdot b_{n1} & \dots & a_{n1}\cdot b_{11} + \dots + a_{nn}\cdot b_{n1} \\ \dots & \dots & \dots \\ a_{11}\cdot b_{1n} + \dots + a_{1n}\cdot b_{nn} & \dots & a_{n1}\cdot b_{1n} + \dots + a_{nn}\cdot b_{nn} \end{pmatrix}$$

Левая:

$$(\mathbf{A}\cdot\mathbf{b})^T = \left(\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & \dots & b_{1n} \\ \dots & \dots & \dots \\ b_{n1} & \dots & b_{nn} \end{pmatrix} \right)^T =$$

$$= \begin{pmatrix} a_{11}\cdot b_{11} + \dots + a_{1n}\cdot b_{n1} & \dots & a_{11}\cdot b_{1n} + \dots + a_{1n}\cdot b_{nn} \\ \dots & \dots & \dots \\ a_{n1}\cdot b_{11} + \dots + a_{nn}\cdot b_{n1} & \dots & a_{n1}\cdot b_{1n} + \dots + a_{nn}\cdot b_{nn} \end{pmatrix}^T =$$

$$= \begin{pmatrix} a_{11}\cdot b_{11} + \dots + a_{1n}\cdot b_{n1} & \dots & a_{n1}\cdot b_{11} + \dots + a_{nn}\cdot b_{n1} \\ \dots & \dots & \dots \\ a_{11}\cdot b_{1n} + \dots + a_{1n}\cdot b_{nn} & \dots & a_{n1}\cdot b_{1n} + \dots + a_{nn}\cdot b_{nn} \end{pmatrix}$$

В результате левая и правая части совпадают. Теорема доказана.

Theorem 12

$$(\mathbf{A}\cdot\mathbf{B})^{-1} = \mathbf{B}^{-1}\cdot\mathbf{A}^{-1}$$

Доказательство.

Умножим обе части равенства на $(\mathbf{A}\cdot\mathbf{B})$. Слева, по определению, получим единичную матрицу. Для выражения справа получаем

$$(\mathbf{A}\cdot\mathbf{B})\cdot\mathbf{B}^{-1}\cdot\mathbf{A}^{-1} = \mathbf{A}\cdot\left(\mathbf{B}\cdot\mathbf{B}^{-1}\right)\cdot\mathbf{A}^{-1} = \mathbf{A}\cdot\mathbf{E}\cdot\mathbf{A}^{-1} = \mathbf{A}\cdot\mathbf{A}^{-1} = \mathbf{E}$$

В результате левая и правая части совпадают. Теорема доказана.

Theorem 13

$$\left(\mathbf{A}^T\right)^{-1} = \left(\mathbf{A}^{-1}\right)^T$$

Доказательство.

Умножим обе части равенства на $\mathbf{A}^T$. Слева, по определению, получим единичную матрицу. Для выражения справа, используя первую задачу, получаем

$$\left(\mathbf{A}^{-1}\right)^T \cdot \mathbf{A}^T = \left(\mathbf{A} \cdot \mathbf{A}^{-1}\right)^T = \mathbf{E}^T = \mathbf{E}$$

В результате левая и правая части совпадают. Теорема доказана.

Theorem 14

$$\det\left(\mathbf{A}^{-1}\right) = \frac{1}{\det\left(\mathbf{A}\right)}$$

Доказательство.

Поскольку *определитель произведения равен произведению определителей*, получаем

$$\det\left(\mathbf{A}^{-1}\right) \cdot \det\left(\mathbf{A}\right) = \det\left(\mathbf{A}^{-1} \cdot \mathbf{A}\right) = \det\left(\mathbf{E}\right) = 1$$

При делении обеих частей равенства на $\det\left(\mathbf{A}\right)$ получим доказываемое утверждение. Теорема доказана.

Theorem 15

При умножении матрицы $\mathbf{A}$ на диагональную матрицу $\mathbf{D}$ *слева* $\mathbf{B} = \mathbf{D} \cdot \mathbf{A}$ все *строки* матрицы $\mathbf{A}$ умножаются на соответствующие диагональные элементы. При умножении матрицы $\mathbf{A}$ на диагональную матрицу $\mathbf{D}$ *справа* $\mathbf{B} = \mathbf{A} \cdot \mathbf{D}$ все *столбцы* матрицы $\mathbf{A}$ умножаются на соответствующие диагональные элементы.
Доказательство.

Очевидно и следует из определения умножения матриц.

Theorem 16

Собственные значения диагональной матрицы равны ее диагональным элементам.
Доказательство.

Пускай дана диагональная матрица $\mathbf{D}$. Рассмотрим определитель

$$\det\begin{pmatrix} d_1 - \lambda_1 & 0 & \dots & 0 \\ 0 & d_2 - \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n - \lambda_n \end{pmatrix} = 0$$

Другими словами

$$(d_1 - \lambda_1) \cdot (d_2 - \lambda_2) \dots (d_n - \lambda_n) = 0$$

Поочередно получаем $\lambda_i = d_i$. Теорема доказана.

Theorem 17

Собственные значения треугольной матрицы равны ее диагональным элементам.

Доказательство.

Разложив определитель треугольной матрицы по первому столбцу, убедимся, что он равен произведению ее диагональных элементов. Применим предыдущую задачу к диагональной матрице, элементами которой являются элементы главной диагонали данной. Теорема доказана.

Theorem 18

При умножении треугольных матриц одного вида получается треугольная матрица того же вида.

Доказательство.

Не умаляя общности, положим $\mathbf{A}$ и $\mathbf{B}$ нижними треугольными матрицами третьего порядка. В общем случае доказательство строится аналогично, меняется только объем записей.

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11} \cdot b_{11} + 0 \cdot b_{21} + 0 \cdot b_{31} & a_{11} \cdot 0 + 0 \cdot b_{22} + 0 \cdot b_{32} & a_{11} \cdot 0 + 0 \cdot +0 \cdot b_{33} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} + 0 \cdot b_{31} & a_{21} \cdot 0 + a_{22} \cdot b_{22} + 0 \cdot b_{32} & a_{21} \cdot 0 + a_{22} \cdot 0 + 0 \cdot b_{33} \\ a_{31} \cdot b_{11} + a_{32} \cdot b_{21} + a_{33} \cdot b_{31} & 0 + a_{32} \cdot b_{22} + a_{33} \cdot b_{32} & 0 + 0 + a_{33} \cdot b_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} c_{11} & 0 & 0 \\ c_{21} & c_{22} & 0 \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Таким образом $\mathbf{C}$ – тоже нижняя треугольная матрица. Теорема доказана.

Theorem 19

Для треугольной матрицы ее обратная матрица имеет тот же вид.

Доказательство.

Заметим, что единичная матрица – тоже треугольная.

$$\mathbf{E} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} = \mathbf{A} \cdot \mathbf{A}^{-1}$$

Поскольку $\mathbf{A}$ – треугольная, то по предыдущей теореме $\mathbf{A}^{-1}$ тоже должна быть треугольной, причем того же типа. Теорема доказана.

Theorem 20

Для произвольной квадратной матрицы сумма ее собственных значений равна сумме элементов главной диагонали, а произведение ее собственных значений равно ее определителю.

$$\sum_{k=1}^{N} \lambda_k = \sum_{k=1}^{N} a_{kk}$$
$$\prod_{k=1}^{N} \lambda_k = \det(\mathbf{A})$$

16.4 Нормы матриц.

Definition 15: Норма матрицы

Нормой матрицы $\|\mathbf{A}\|$ называется число, удовлетворяющее следующим аксиомам:

1. $\|\mathbf{A}\| \geq 0$, при этом $\|\mathbf{A}\| = 0 \Leftrightarrow \mathbf{A} = \mathbf{O}$
2. $\|\alpha\mathbf{A}\| = |\alpha| \, \|\mathbf{A}\|$
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{A} \cdot \mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$

Норма также называется *канонической*, если к тому же выполняются следующие аксиомы:

5. $a_{ik} \leq \|\mathbf{A}\| \qquad \forall i, k$
6. Если $\forall i, k \ |a_{ik}| \leq |b_{ik}|$, то $\|\mathbf{A}\| \leq \|\mathbf{B}\|$

Примерами матричной нормы являются:

$$\|\mathbf{A}\|_1 = \max_i \sum_{j=1}^{n} |a_{ij}| \, ; \qquad \|\mathbf{A}\|_2 = \max_j \sum_{i=1}^{n} |a_{ij}| \, ; \qquad \|\mathbf{A}\|_3 = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

Последняя называется *Евклидовой нормой*. Все эти нормы являются каноническими, т.е. удовлетворяют всем шести аксиомам.

16.5 Матричный ряд и матричные функции.

Наиболее простыми матричными функциями являются полиномы. Для их построения необходимо образовать степени матрицы, которые определены только для квадратных матриц, поэтому далее речь пойдет о них.

По определению $\mathbf{A}^k = \prod_{i=1}^{k} \mathbf{A}$, $\mathbf{A}^0 = \mathbf{E}$. Тогда матричный полином:

$$\mathbf{P}_n(\mathbf{A}) = c_0\mathbf{E} + c_1\mathbf{A} + c_2\mathbf{A}^2 + \cdots + c_n\mathbf{A}^n$$

Аргументом является квадратная матрица $m \times m$, и значением будет матрица той же размерности.
Теперь устремим n к бесконечности, т.е. формально перейдем к бесконечной сумме

$$\mathbf{P}(\mathbf{A}) = \sum_{\gamma=0}^{\infty} c_\gamma \mathbf{A}^\gamma \tag{76}$$

Такая сумма называется степенным матричным рядом относительно матрицы $\mathbf{A}$. Матричному ряду естественно сопоставить скалярный ряд

$$p(x) = \sum_{\gamma=0}^{\infty} c_\gamma x^\gamma$$

Матричный ряд будем называться *сходящимся*, если сходятся все m^2 скалярных рядов для элементов матрицы $\mathbf{P}(\mathbf{A})$. Введенное понятие нормы позволяет установить достаточное условие сходимости матричного ряда. Введем матрицу $\mathbf{U}^{(\gamma)} = c_\gamma \mathbf{A}^\gamma$. Обозначим ее элементы за $u_{kj}^{(\gamma)}$, а элементы матрицы $\mathbf{P}(\mathbf{A})$ за p_{kj}. Тогда с учетом выполнения шести аксиом для канонической нормы имеем цепочку неравенств

$$|p_{kj}| = \left|\sum_{\gamma=0}^{\infty} u_{kj}^{(\gamma)}\right| \le \sum_{\gamma=0}^{\infty} \left|u_{kj}^{(\gamma)}\right| \le \sum_{\gamma=0}^{\infty} \|c_\gamma \mathbf{A}^\gamma\| = \sum_{\gamma=0}^{\infty} |c_\gamma| \, \|\mathbf{A}^\gamma\| \le \sum_{\gamma=0}^{\infty} |c_\gamma| \, \|\mathbf{A}\|^\gamma$$

В результате *достаточным* условием сходимости матричного ряда является выполнение условия

$$\|\mathbf{A}\| < R \tag{77}$$

являющегося, в свою очередь, условием абсолютной сходимости скалярного степенного ряда, стоящего последним в цепочке неравенств. Здесь R – радиус сходимости скалярного степенного ряда.

Definition 16: Матричная функция

Если матричный ряд сходится, то матрицу $\mathbf{P}(\mathbf{A})$ называют *матричной функцией*.

Примеры матричных функций:

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!} \qquad \cos(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{(-1)^k \mathbf{A}^{2k}}{(2k)!}$$

$$\sin(\mathbf{A}) = \sum_{k=0}^{\infty} \frac{(-1)^k \mathbf{A}^{2k+1}}{(2k+1)!} \qquad (\mathbf{E} - \mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k$$

Вопрос 17. 7 теорем о матричных функциях.

Предварительно введем следующее определение

Definition 17: Подобная матрица

Пусть задана матрица $\mathbf{A}$ и некоторая неособенная матрица $\mathbf{S}$ (т.е. $\det(\mathbf{S}) \neq 0$ и существует S^{-1}). Всякая матрица $\mathbf{B} = \mathbf{SAS}^{-1}$ называется *подобной* матрице $\mathbf{A}$. Очевидно, что и $\mathbf{A} = \mathbf{S}^{-1}\mathbf{BS}$ подобна $\mathbf{B}$.

Theorem 21

Подобные матрицы $\mathbf{A}$ и $\mathbf{B} = \mathbf{SAS}^{-1}$ имеют одинаковые собственные значения. При этом, если собственному значению λ матрицы $\mathbf{A}$ отвечает собственный вектор $\mathbf{u}$, то у матрицы $\mathbf{B}$ этому же собственному числу λ соответствует собственный вектор $\mathbf{Su}$.

Доказательство.

Так как $\mathbf{SS}^{-1} = \mathbf{E}$, то $\det\left(\mathbf{SS}^{-1}\right) = \det(\mathbf{S})\det(\mathbf{S}^{-1}) = \det(\mathbf{E}) = 1$. Для характеристических полиномов $\mathbf{A}$ и $\mathbf{B}$ имеем

$$\begin{aligned}\det\left(\mathbf{B} - \lambda\mathbf{E}\right) = \det\left(\mathbf{SAS}^{-1} - \lambda\mathbf{SS}^{-1}\right) = \det\left(\mathbf{S}\left(\mathbf{A} - \lambda\mathbf{E}\right)\mathbf{S}^{-1}\right) = \\ = \det\left(\mathbf{S}\right) \cdot \det\left(\mathbf{A} - \lambda\mathbf{E}\right) \cdot \det\left(\mathbf{S}^{-1}\right) = \det\left(\mathbf{A} - \lambda\mathbf{E}\right)\end{aligned}$$

Характеристические полиномы для обеих матриц совпали, следовательно, совпали и их корни, т.е. собственные значения. Для доказательства второй части теоремы в равенстве $\mathbf{Au} = \lambda\mathbf{u}$ заменим матрицу $\mathbf{A}$ на подобную ей $\mathbf{SAS}^{-1}$:

$$\mathbf{S}^{-1}\mathbf{BSu} = \lambda\mathbf{u}$$

Теперь, умножив обе части равенства на $\mathbf{S}$ слева, получим требуемый результат:

$$\mathbf{B}\left(\mathbf{Su}\right) = \lambda\left(\mathbf{Su}\right)$$

Theorem 22

Если матрицы $\mathbf{A}$ и $\mathbf{B}$ подобны, то их матричные функции также подобны. Иными словами, если $\mathbf{B} = \mathbf{SAS}^{-1}$, то $\mathbf{f}(\mathbf{B}) = \mathbf{Sf}(\mathbf{A})\,\mathbf{S}^{-1}$
Доказательство.

Первоначально определим $\mathbf{B}^k$

$$\mathbf{B}^k = \left(\mathbf{SAS}^{-1}\right)^k = \mathbf{SAS}^{-1}\mathbf{SAS}^{-1}\ldots\mathbf{SAS}^{-1} = \mathbf{SAA}\ldots\mathbf{S}^{-1} = \mathbf{SA}^k\mathbf{S}^{-1}$$

$$\mathbf{f}(\mathbf{B}) = \sum_{k=0}^{\infty} c_k\mathbf{B}^k = \mathbf{S}\left(\sum_{k=0}^{\infty} c_k\mathbf{A}^k\right)\mathbf{S}^{-1} = \mathbf{Sf}(\mathbf{A})\,\mathbf{S}^{-1}$$

Theorem 23

Матрица $\mathbf{A}$ с различными собственными значениями $\lambda_1, \lambda_2, \ldots, \lambda_m$ (нет кратных) подобна некоторой диагональной матрице $\mathbf{\Lambda}$, на главной диагонали которой стоят собственные значения матрицы $\mathbf{A}$, то есть $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$
Доказательство.

Пусть $\mathbf{u}_k$ – собственные векторы матрицы $\mathbf{A}$. Обозначим за $\mathbf{U}$ матрицу, столбцами которой являются все $\mathbf{u}_k$. Тогда

$$\mathbf{AU} = \mathbf{A}\begin{pmatrix} | & | & & & | \\ | & | & & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \ldots & \mathbf{u}_m \\ | & | & & & | \\ | & | & & & | \end{pmatrix} = \begin{pmatrix} | & | & & & | \\ | & | & & & | \\ \lambda_1\mathbf{u}_1 & \lambda_2\mathbf{u}_2 & \ldots & \ldots & \lambda_m\mathbf{u}_m \\ | & | & & & | \\ | & | & & & | \end{pmatrix} =$$

$$= \begin{pmatrix} | & | & & & | \\ | & | & & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \ldots & \mathbf{u}_m \\ | & | & & & | \\ | & | & & & | \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 & 0 & \ldots & 0 \\ 0 & \lambda_2 & 0 & \ldots & 0 \\ 0 & 0 & \lambda_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \lambda_m \end{pmatrix} = \mathbf{U\Lambda}$$

Умножая полученное равенство поочередно справа и слева на $\mathbf{U}^{-1}$, получаем требуемое

$$\mathbf{\Lambda} = \mathbf{U}^{-1}\mathbf{AU}, \qquad \mathbf{A} = \mathbf{U\Lambda U}^{-1}$$

Заметим, что помимо доказательства теоремы, была определена матрица подобия $\mathbf{U}$, состоящая из линейно независимых столбцов $\mathbf{u}_k$ и, следовательно, неособенная.
В общем случае, при наличии кратных собственных значений у матрицы $\mathbf{A}$, вместо матрицы $\mathbf{\Lambda}$, возникает клеточно-диагональная матрица, где каждая клетка представляет собой так называемый *канонический ящик Жордана.*
Исключительно для простоты изложения дальнейшие теоремы будут доказываться только для матриц с различными собственными значениями, однако результаты справедливы и для более общего случая.

Theorem 24

Если собственные значения матрицы $\mathbf{A}$ обозначить через $\lambda_1, \lambda_2, \ldots, \lambda_m$, то собственными значениями матрицы $\mathbf{f}(\mathbf{A})$ будут числа $f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_m)$

Доказательство.

$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$. По теореме о подобии матричных функций $\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\,\mathbf{U}$. Представим $\mathbf{f}(\mathbf{\Lambda})$ в покомпонентном виде

$$\mathbf{f}(\mathbf{\Lambda}) = \sum_{k=0}^{\infty} c_k \mathbf{\Lambda}^k = \begin{pmatrix} \sum\limits_{k=0}^{\infty} c_k \lambda_1^k & 0 & \ldots & \ldots & 0 \\ 0 & \sum\limits_{k=0}^{\infty} c_k \lambda_2^k & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & \sum\limits_{k=0}^{\infty} c_k \lambda_m^k \end{pmatrix} =$$

$$= \begin{pmatrix} f(\lambda_1) & 0 & \ldots & \ldots & 0 \\ 0 & f(\lambda_2) & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & f(\lambda_m) \end{pmatrix}$$

И тогда

$$\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1} f(\mathbf{\Lambda})\,\mathbf{U} = \mathbf{U}^{-1} \begin{pmatrix} f(\lambda_1) & 0 & \ldots & \ldots & 0 \\ 0 & f(\lambda_2) & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & f(\lambda_m) \end{pmatrix} \mathbf{U}$$

что и означает, что $f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_m)$ являются собственными значениями матрицы $\mathbf{f}(\mathbf{A})$, так как преобразование подобия не меняет собственных значений.

Следствие 1. Из вышеприведенных формул непосредственно следует, что матричный ряд $\mathbf{f}(\mathbf{A})$ существует тогда и только тогда, когда существуют *все* скалярные степенные ряды, стоящие на диагонали матрицы $\mathbf{f}(\mathbf{\Lambda})$, а у тех, в свою очередь, необходимым и достаточным условием существования является выполнение условий $\forall \lambda_k \ |\lambda_k| < R$. Таким образом, это условие является необходимым и достаточным условием сходимости матричного степенного ряда. На практике вопрос о сходимости матричного ряда решается в такой последовательности: сначала находится радиус сходимости R соответствующего скалярного ряда, а затем проверяется выполнение условия выше для всех собственных значений.

Следствие 2. Поскольку условие $\|\mathbf{A}\| < R$ является лишь достаточным условием сходимости матричного степенного ряда, а условие выше необходимым и достаточным, то из совместного рассмотрения обоих условий можно заключить, что все собствен-

ные значения матрицы не превышают ее любую каноническую норму.

$$|\lambda_k| \leq \|\mathbf{A}\|$$

Theorem 25

Две любые функции матрицы $\mathbf{A}$ коммутируют между собой:

$$\mathbf{f}(\mathbf{A}) \cdot \mathbf{g}(\mathbf{A}) = \mathbf{g}(\mathbf{A}) \cdot \mathbf{f}(\mathbf{A})$$

Доказательство.

$$\mathbf{f}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\,\mathbf{U} \text{ и } \mathbf{g}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{g}(\mathbf{\Lambda})\,\mathbf{U}$$

В силу того, что диагональные матрицы всегда коммутируют:

$$\mathbf{f}(\mathbf{A})\,\mathbf{g}(\mathbf{A}) = \mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\,\mathbf{U}\mathbf{U}^{-1}\mathbf{g}(\mathbf{\Lambda})\,\mathbf{U} = \mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\,\mathbf{g}(\mathbf{\Lambda})\,\mathbf{U} =$$
$$= \mathbf{U}^{-1}\mathbf{g}(\mathbf{\Lambda})\,\mathbf{f}(\mathbf{\Lambda})\,\mathbf{U} = \mathbf{U}^{-1}\mathbf{g}(\mathbf{\Lambda})\,\mathbf{U}\mathbf{U}^{-1}\mathbf{f}(\mathbf{\Lambda})\,\mathbf{U} = \mathbf{g}(\mathbf{A})\,\mathbf{f}(\mathbf{A})$$

Theorem 26: Формула Кели-Гамильтона

Всякая матрица удовлетворяет своему характеристическому уравнению. Пусть $Q(\lambda) = (-1)^m\lambda^m + b_1\lambda^{m-1} + \cdots + b_m = 0$ – характеристическое уравнение любой матрицы $\mathbf{A}$. Тогда

$$Q(\mathbf{A}) = (-1)^m\mathbf{A}^m + b_1\mathbf{A}^{m-1} + \cdots + b_m\mathbf{E} \equiv \mathbf{O}$$

Доказательство.

Матрица с различными собственными значениями подобна диагональной матрице: $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$. По теореме о подобии матричных функций $\mathbf{A}^k = \mathbf{U}\mathbf{\Lambda}^k\mathbf{U}^{-1}$. Подставим в

$$Q(\mathbf{A}) = (-1)^m\mathbf{A}^m + b_1\mathbf{A}^{m-1} + \cdots + b_m\mathbf{E}$$

вместо $\mathbf{A}$ ее выражение через $\mathbf{\Lambda}$:

$$Q(\mathbf{A}) = (-1)^m\mathbf{U}\mathbf{\Lambda}^m\mathbf{U}^{-1} + b_1\mathbf{U}\mathbf{\Lambda}^{m-1}\mathbf{U}^{-1} + \cdots + b_m\mathbf{E} =$$
$$= \mathbf{U}\left((-1)^m\mathbf{\Lambda}^m + b_1\mathbf{\Lambda}^{m-1} + \cdots + b_m\mathbf{E}\right)\mathbf{U}^{-1}$$

В скобках стоит диагональная матрица с характеристическими полиномами на главной диагонали, в которые подставлены собственные значения, и значит тождественно равными нулю. Тогда матрица в скобках – нулевая, и теорема доказана.

Theorem 27: Формула Лагранжа-Сильвестра

Любая функция матрицы $\mathbf{A}$, имеющей различные собственные значения, может быть представлена в виде:

$$\mathbf{f}(\mathbf{A}) = \sum_{k=1}^{m} \frac{(\mathbf{A}-\lambda_1\mathbf{E})\dots(\mathbf{A}-\lambda_{k-1}\mathbf{E})(\mathbf{A}-\lambda_{k+1}\mathbf{E})\dots(\mathbf{A}-\lambda_m\mathbf{E})}{(\lambda_k-\lambda_1)\dots(\lambda_k-\lambda_{k-1})(\lambda_k-\lambda_{k+1})\dots(\lambda_k-\lambda_m)} f(\lambda_k) = \quad (78)$$

$$= \sum_{k=1}^{m} \mathbf{T}_k f(\lambda_k)$$

Доказательство.

Подставим функцию $f(x)$ в виде интерполяционного полинома Лагранжа $L_{m-1}(x)$, взяв в качестве узлов собственные значения матрицы $\mathbf{A}$ $\lambda_1, \lambda_2, \dots, \lambda_m$:

$$f(x) = L_{m-1}(x) + R_{m-1}(x)$$

Подставим в эту формулу $\mathbf{A}$ вместо x:

$$\mathbf{f}(\mathbf{A}) = \mathbf{L}_{m-1}(\mathbf{A}) + \mathbf{R}_{m-1}(\mathbf{A})$$

Остаточный член принимает вид

$$\mathbf{R}_{m-1}(\mathbf{A}) = \frac{f^{(m)}(\xi)}{m!}\omega(\mathbf{A})$$

где $\omega(\mathbf{A}) = (\mathbf{A}-\lambda_1\mathbf{E})\dots(\mathbf{A}-\lambda_m\mathbf{E})$. По теореме Кели-Гамильтона $\omega(\mathbf{A}) = \mathbf{O}$ и, следовательно, $\mathbf{f}(\mathbf{A}) = \mathbf{L}_{m-1}(\mathbf{A})$

Вопрос 18. Решение систем линейных дифференциальных и разностных уравнений с постоянной матрицей.

18.1 Дифференциальные уравнения.

Рассмотрим систему обыкновенных дифференциальных уравнений первого порядка, разрешенную относительно производных

$$\frac{dx^{(i)}(t)}{dt} = f^{(i)}\left(t, x^{(1)}(t), x^{(2)}(t), \ldots, x^{(m)}(t)\right), \qquad i = 1,2,\ldots,m$$

где t – независимая переменная, $x^{(i)}(t)$ – искомые функции, $f^{(i)}$ – функции, определенные на некотором множестве $(m+1)$-мерного евклидова пространства переменных $t, x^{(i)}(t)$. Номер компоненты вектора здесь везде будем писать, как верхний индекс в скобках. Перейдя к векторно-матричным обозначениям

$$\mathbf{x}(t) = \begin{pmatrix} x^{(1)}(t) \\ x^{(2)}(t) \\ \ldots \\ x^{(m)}(t) \end{pmatrix}, \qquad \mathbf{f}(t, \mathbf{x}) = \begin{pmatrix} f^{(1)}(t, x) \\ f^{(2)}(t, x) \\ \ldots \\ f^{(m)}(t, x) \end{pmatrix}$$

Исходную систему перепишем в виде

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}) \tag{79}$$

При этом требуется найти решение $\mathbf{x}(t)$, удовлетворяющее начальным условиям $\mathbf{x}(t_0) = \mathbf{x}_0$. Такая задача называется начальной задачей или *задачей Коши.*
Важным классом дифференциальных систем являются линейные системы с постоянной матрицей или постоянными коэффициентами

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{g}(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{80}$$

Сначала обратимся к однородной системе

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t)$$

Ее решением является функция $\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{c}$, где $\mathbf{c}$ – вектор произвольных постоянных. Убедиться в этом можно непосредственной подстановкой решения в уравнение.

Неоднородная система решается методом Лагранжа вариации произвольных постоянных. При этом полагаем, что элементы вектора $\mathbf{c}$ являются функциями независимой переменной $\mathbf{c} = \mathbf{c}(t)$. Подставляем искомый вид решения в уравнение

$$\mathbf{A}e^{\mathbf{A}t}\mathbf{c}(t) + e^{\mathbf{A}t}\frac{d\mathbf{c}(t)}{dt} = \mathbf{A}e^{\mathbf{A}t}\mathbf{c}(t) + \mathbf{g}(t)$$

Отсюда $e^{\mathbf{A}t}\dfrac{d\mathbf{c}(t)}{dt} = \mathbf{g}(t)$, и после умножения обеих частей равенства на $e^{-\mathbf{A}t}$ получаем:

$$\frac{d\mathbf{c}(t)}{dt} = e^{-\mathbf{A}t}\mathbf{g}(t)$$

Интегрируем это уравнение от t_0 до t

$$\mathbf{c}(t) - \mathbf{c}(t_0) = \int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{g}(\tau)d\tau$$

и, подставив $\mathbf{c}(t)$ в искомый вид решения, определяем общее решение линейной неоднородной дифференциальной системы

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{c}(t_0) + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{g}(\tau)d\tau$$

Учитывая начальные условия, находим вектор $c(t_0)$: $\mathbf{x}(t_0) = \mathbf{x}_0 = e^{\mathbf{A}t_0}\mathbf{c}(t_0)$ или $\mathbf{c}(t_0) = e^{-\mathbf{A}t_0}\mathbf{x}_0$ и окончательно получаем

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau$$

Без нарушения общности можно считать, что начальным значением независимой переменной является $t_0 = 0$. Тогда, используя теорему о свертке,

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_{0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{x}_0 + \int_{0}^{t} e^{\mathbf{A}\tau}\mathbf{g}(t-\tau)d\tau \tag{81}$$

Считая вектор $\mathbf{g}(t)$ постоянным, упростим полученное равенство:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_{0}^{t} e^{\mathbf{A}(t-\tau)}d\tau \cdot \mathbf{g} = e^{\mathbf{A}t}\mathbf{x}_0 + \left(e^{\mathbf{A}t} - \mathbf{E}\right)\mathbf{A}^{-1}\mathbf{g}$$

18.2 Разностные уравнения.

Рассмотрим линейную систему разностных уравнений с постоянной матрицей, где k – независимая целочисленная переменная

$$\mathbf{y}(k+1) = \mathbf{B}\mathbf{y}(k) + \mathbf{g}(k) \tag{82}$$

Будем ее решать так называемым пошаговым методом, последовательно назначая в (82) значения k равным 0,1,2,... и обозначая $\mathbf{y}_k = \mathbf{y}(k)$

$$\begin{aligned}
&\mathbf{y}_1 = \mathbf{B}\mathbf{y}_0 + \mathbf{g}_0\\
&\mathbf{y}_2 = \mathbf{B}\mathbf{y}_1 + \mathbf{g}_1 = \mathbf{B}^2\mathbf{y}_0 + \mathbf{B}\mathbf{g}_0 + \mathbf{g}_1\\
&\mathbf{y}_3 = \mathbf{B}\mathbf{y}_2 + \mathbf{g}_2 = \mathbf{B}^3\mathbf{y}_0 + \mathbf{B}^2\mathbf{g}_0 + \mathbf{B}\mathbf{g}_1 + \mathbf{g}_2\\
&\dots
\end{aligned}$$

$$\mathbf{y}_k = \mathbf{B}^k\mathbf{y}_0 + \sum_{i=0}^{k-1}\mathbf{B}^{k-i-1}\mathbf{g}_i \tag{83}$$

В частном случае, когда $\mathbf{g}(k) = const = \mathbf{g}$, этот вектор можно вынести за знак суммы

$$\mathbf{y}_k = \mathbf{B}^k\mathbf{y}_0 + \left(\sum_{i=0}^{k-1}\mathbf{B}^i\right)\mathbf{g} = \mathbf{B}^k\mathbf{y}_0 + \left(\mathbf{B}^k - \mathbf{E}\right)(\mathbf{B} - \mathbf{E})^{-1}\mathbf{g} \tag{84}$$

Вопрос 19. Устойчивость решений дифференциальных и разностных уравнений.

Обратимся к системе нелинейных дифференциальных уравнений

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t \in [a, b] \tag{85}$$

где t – независимая переменная, $\mathbf{x}$ – вектор решения; $\mathbf{f}(t, \mathbf{x})$ – вектор-функция, непрерывная по t и имеющая непрерывные частные производные первого порядка по компонентам вектора $\mathbf{x}$.

Большой интерес представляет исследование зависимости решения задачи Коши от начальных условий. Если незначительные изменения в $\mathbf{x}_0$ могут существенно изменить решение, то в прикладном отношении такое решение часто неприемлемо. На конечном промежутке $[a, b]$ для систем (85) с непрерывной функцией $\mathbf{f}(t, \mathbf{x})$ и свойством единственности решения имеет место *интегральная непрерывность решений*. Иными словами,

$$\forall \varepsilon > 0 \quad \exists\, \delta > 0 : \quad \|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \delta \Rightarrow \|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon$$

Иначе обстоит дело при $t \to \infty$. Изучением этих вопросов занимается теория устойчивости.

Definition 18

Решение $\mathbf{x}(t)$ называется *устойчивым по Ляпунову*, если

$$\forall \varepsilon > 0 \quad \exists\, \delta > 0 : \quad \forall \mathbf{z}(t) \quad \forall t \in [t_0; \infty) \quad \|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \delta \Rightarrow \|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon$$

Иными словами, решение $\mathbf{x}(t)$ называется устойчивым, если другие достаточно близкие к нему в момент времени t_0 решения $\mathbf{z}(t)$ целиком находятся в узкой ε-трубке, построенной вокруг $\mathbf{x}(t)$.

Definition 19

Решение $\mathbf{x}(t)$ системы (85) называется *асимптотически устойчивым* по Ляпунову, если

1. Оно устойчиво
2. Выполняется условие

$$\exists \Delta > 0 : \quad \|\mathbf{x}(t_0) - \mathbf{z}(t_0)\| < \Delta \Rightarrow \lim_{t \to \infty} \|\mathbf{x}(t) - \mathbf{z}(t)\| = 0$$

В случае асимптотической устойчивости близкие решения не только остаются близкими друг к другу, но и неограниченно сближаются при возрастании t.

Для систем разностных уравнений

$$\mathbf{y}(n+1) = \mathbf{g}\left(n, \mathbf{y}(n)\right), \quad \mathbf{y}(n_0) = \mathbf{y}_0, \quad n \in [n_0; \infty) \tag{86}$$

понятие устойчивости вводится аналогичным образом.

Definition 20

Решение $\mathbf{y}(n)$ называется *устойчивым*, если

$$\forall \varepsilon > 0 \quad \exists\, \delta > 0 : \forall \mathbf{w}(n)\, \forall n \in [n_0; \infty) \quad \|\mathbf{y}(n_0) - \mathbf{w}(n_0)\| < \delta \Rightarrow \|\mathbf{y}(n) - \mathbf{w}(n)\| < \varepsilon$$

Сформулированные определения позволяют сделать суждение об устойчивости после анализа уже полученных решений. С практической точки зрения важно судить об устойчивости, не решая систему. Это возможно, в частности, для линейных систем с постоянной матрицей

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{g}(t), \qquad \mathbf{x}(t_0) = \mathbf{x}_0$$

Будем называть их устойчивыми (асимптотически устойчивыми, неустойчивыми), если все их решения устойчивы (асимптотически устойчивы, неустойчивы).
Пусть $\mathbf{x}(t)$ и $\mathbf{z}(t)$ – два различных решения (85), отличающиеся начальными условиями. В соответствии с (81) они имеют вид

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}\tau}\mathbf{g}(t-\tau)d\tau$$
$$\mathbf{z}(t) = e^{\mathbf{A}t}\mathbf{z}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{g}(\tau)d\tau = e^{\mathbf{A}t}\mathbf{z}_0 + \int_0^t e^{\mathbf{A}\tau}\mathbf{g}(t-\tau)d\tau$$

Вычтем из первой формулы вторую. После сокращения интегралов получаем

$$\mathbf{x}(t) - \mathbf{z}(t) = e^{\mathbf{A}t}\left(\mathbf{x}_0 - \mathbf{z}_0\right)$$

Пусть первоначально собственные значения матрицы $\mathbf{A}$ различны. Тогда, используя для матричной экспоненты формулу Лагранжа-Сильвестра, имеем

$$\mathbf{x}(t) - \mathbf{z}(t) = e^{\mathbf{A}t}\left(\mathbf{x}_0 - \mathbf{z}_0\right) = \sum_{k=1}^{m} e^{\lambda_k t}\mathbf{T}_k\left(\mathbf{x}_0 - \mathbf{z}_0\right)$$

Обращаясь к определениям устойчивости, приходим к выводу о том, что для обеспечения неравенства ($\|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon$) элементы матричной экспоненты при $t \to \infty$ должны быть ограничены. А это, в свою очередь, требует, чтобы вещественные части $\Re(\lambda_k)$ собственных значений были бы неположительные. Для асимптотической устойчивости условие $\lim\limits_{t\to\infty} \|\mathbf{x}(t) - \mathbf{z}(t)\| = 0$ выполняется, когда элементы матричной экспоненты при $t \to \infty$ стремятся к нулю, а вещественные части собственных значений соответственно отрицательные.
Если среди собственных значений есть кратные, условия несколько корректируются. Пусть, например, собственное значение λ_k имеет кратность s. Тогда в решении этой группе собственных значений отвечает слагаемое $P_{s-1}(t)e^{\lambda_k t}$. Если для $\Re(\lambda_k) < 0$ асимптотическая устойчивость обеспечивается независимо от кратности корня

$$P_{s-1}(t)e^{\lambda_k t} \to 0 \text{ при } t \to \infty$$

то при нулевой вещественной части $P_{s-1}(t) \to \pm\infty$ при $t \to \infty$ и не выполняется условие ($\|\mathbf{x}(t) - \mathbf{z}(t)\| < \varepsilon$).
Подведем итоги.

1. Для асимптотической устойчивости необходимо и достаточно, чтобы для всех собственных значений выполнялись условия $\Re(\lambda_k) < 0$.
2. Для устойчивости необходимо, чтобы $\Re(\lambda_k) \leq 0$. При этом достаточно, чтобы среди собственных значений с нулевой вещественной частью не было бы кратных.
3. Для неустойчивости необходимо наличие хотя бы одного собственного значения с $\Re(\lambda_k) > 0$ или кратных собственных значений с $\Re(\lambda_k) = 0$.

Теперь обратимся к системе разностных уравнений с постоянной матрицей

$$\mathbf{y}(n+1) = \mathbf{B}\mathbf{y}(n) + \mathbf{g}(n)$$

Пусть $\mathbf{y}$(n) и $\mathbf{w}$(n) – ее два различных решения, отличающиеся начальными условиями. Они имеют вид

$$\mathbf{y}(n) = \mathbf{B}^n\mathbf{y}(0) + \sum_{k=0}^{n-1}\mathbf{B}^{n-k-1}\mathbf{g}(k) = \mathbf{B}^n\mathbf{y}(0) + \sum_{k=0}^{n-1}\mathbf{B}^{k}\mathbf{g}(n-k-1)$$
$$\mathbf{w}(n) = \mathbf{B}^n\mathbf{w}(0) + \sum_{k=0}^{n-1}\mathbf{B}^{n-k-1}\mathbf{g}(k) = \mathbf{B}^n\mathbf{w}(0) + \sum_{k=0}^{n-1}\mathbf{B}^{k}\mathbf{g}(n-k-1)$$

Вычитая из первой формулы вторую, после сокращения сумм получаем

$$\mathbf{y}(n) - \mathbf{w}(n) = \mathbf{B}^n\left(\mathbf{y}_0 - \mathbf{w}_0\right)$$

Если все собственные значения μ_k матрицы $\mathbf{B}$ различны, то, воспользовавшись формулой Лагранжа-Сильвестра для $\mathbf{B}^n$, имеем

$$\mathbf{y}(n) - \mathbf{w}(n) = \sum_{k=1}^{m}\mu_k^n\mathbf{T}_k\left(\mathbf{y}_0 - \mathbf{w}_0\right)$$

Аналогично предыдущему для обеспечения неравенства ($\|\mathbf{y}(n) - \mathbf{w}(n)\| < \varepsilon$) элементы матрицы $\mathbf{B}^n$ при $n \to \infty$ должны быть ограничены. А это, в свою очередь, требует выполнения условий $|\mu_k| \leq 1$ для всех собственных значений. Для асимптотической устойчивости неравенства должны быть строгими: $|\mu_k| < 1$. Если собственное значение μ_k имеет кратность s, то, как и для дифференциальных уравнений, в решении появляется слагаемое $P_{s-1}(n) \cdot \mu_k^n$. Для $|\mu_k| < 1$ этот факт не оказывает влияния на условие устойчивости, но для $|\mu_k| = 1$ условие устойчивости нарушается, если $P_{s-1}(n) \to \pm\infty$ при $n \to \infty$.

Как результат, сформулируем условия устойчивости.

1. Для асимптотической устойчивости необходимо и достаточно, чтобы для всех собственных значений выполнялись условия $|\mu_k| < 1$.
2. Для устойчивости необходимо, чтобы $|\mu_k| \leq 1$. При этом достаточно, чтобы среди собственных значений с единичными модулями не было бы кратных.
3. Для неустойчивости необходимо наличие хотя бы одного собственного значения с $|\mu_k| > 1$ или кратных собственных значений с $|\mu_k| = 1$.

Вопрос 20. Метод Гаусса и явление плохой обусловленности. LU-разложение матрицы. Подпрограммы DECOMP и SOLVE.

20.1 Плохая обусловленность матрицы.

Рассмотрим систему нелинейных алгебраических уравнений

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad \det(\mathbf{A}) \neq 0 \tag{87}$$

Так как матрица $\mathbf{A}$ неособенная, ее единственным решением является вектор

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{88}$$

Как сильно оно изменится при малой вариации исходных данных (элементов $\mathbf{A}$ и вектора $\mathbf{b}$)?
Численное решение линейных алгебраических систем подвержено влиянию нескольких источников ошибок. Два из них традиционны и очевидны: ограниченность разрядной сетки компьютера и погрешность представления исходных данных.

Definition 21

Матрица и вместе с ней система (87) называются *плохо обусловленными*, если малым изменениям элементов $\mathbf{A}$ отвечают большие изменения элементов $\mathbf{A}^{-1}$ и, следовательно, сильные изменения вектора решения.

Получим количественную характеристику этого явления. Первоначально будем считать, что матрица $\mathbf{A}$ известна точно, а вектор $\mathbf{b}$ – с некоторой погрешностью $\Delta\mathbf{b}$. Тогда система приобретает вид

$$\mathbf{A}\left(\mathbf{x} + \Delta\mathbf{x}\right) = \mathbf{b} + \Delta\mathbf{b}$$

или после вычитания (87) и обращения матрицы:

$$\mathbf{A}\Delta\mathbf{x} = \Delta\mathbf{b} \qquad \Delta\mathbf{x} = \mathbf{A}^{-1}\Delta\mathbf{b}$$

Далее при использовании любой ранее рассмотренной нормы матрицы, получаем

$$\|\Delta\mathbf{x}\| \leq \left\|\mathbf{A}^{-1}\right\| \|\Delta\mathbf{b}\| \qquad \|\mathbf{b}\| \leq \|\mathbf{A}\| \|\mathbf{x}\|$$

Перемножение этих двух неравенств в предположении, что $\mathbf{b} \neq \mathbf{O}$, и деление на $\|\mathbf{b}\| \cdot \|\mathbf{x}\|$ дает

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq cond(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}, \qquad cond(\mathbf{A}) = \|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\| \tag{89}$$

Definition 22

Число $cond(\mathbf{A}) = \|\mathbf{A}\| \cdot \left\|\mathbf{A}^{-1}\right\|$ называется *стандартным числом обусловленности.*

Вычисляя норму от обоих частей равенства $\mathbf{A}\cdot\mathbf{A}^{-1} = \mathbf{E}$, имеем $\|\mathbf{A}\|\cdot\left\|\mathbf{A}^{-1}\right\| \geq \|\mathbf{E}\|$, т.е. $cond(\mathbf{A}) \geq 1$. Равенство (89) допускает простую интерпретацию для практики. Число обусловленности матрицы $\mathbf{A}$ является верхней границей «усиления» относительной ошибки вектора $\mathbf{b}$, т.е. относительное изменение вектора $\mathbf{b}$ влечет за собой относительное изменение в решении не более чем в $cond(\mathbf{A})$ раз. Если величина $cond(\mathbf{A})$ невелика, то говорят о *хорошей* обусловленности матрицы $\mathbf{A}$, в противном случае – о *плохой.*

Теперь рассмотрим ситуацию, когда вектор $\mathbf{b}$ известен точно, а коэффициенты матрицы $\mathbf{A}$ заданы с погрешностью $\Delta\mathbf{A}$:

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b}$$

$$\mathbf{A}\Delta\mathbf{x} = -\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})$$
$$\|\Delta\mathbf{x}\| \leq \left\|\mathbf{A}^{-1}\right\| \cdot \|\Delta\mathbf{A}\| \cdot \|\mathbf{x} + \Delta\mathbf{x}\|$$
$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x} + \Delta\mathbf{x}\|} \leq cond(\mathbf{A})\frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|} \tag{90}$$

И в этом случае $cond(\mathbf{A})$ ограничивает сверху увеличение относительной ошибки решения по сравнению с относительной ошибкой исходных данных.

Существуют и другие количественные характеристики плохой обусловленности. Например, таким числом является величина, отражающая разброс спектра собственных значений $\mathbf{A}$:

$$k(\mathbf{A}) = \frac{|\lambda_k|_{max}}{|\lambda_k|_{min}}$$

Заметим, что $k(\mathbf{A}) \leq cond(\mathbf{A})$. Действительно, если λ_k собственные значения $\mathbf{A}$, то $\dfrac{1}{\lambda_k}$ собственные значения $\mathbf{A}^{-1}$. Поэтому $\max|\lambda_k| \leq \|\mathbf{A}\|$, а также $\dfrac{1}{|\lambda_k|_{min}} \leq \left\|\mathbf{A}^{-1}\right\|$, и для плохо обусловленных матриц имеем:

$$cond(\mathbf{A}) = \|\mathbf{A}\|\left\|\mathbf{A}^{-1}\right\| \geq k(\mathbf{A}) = \frac{\max|\lambda_k|}{\min|\lambda_k|} \gg 1$$

20.2 Метод Гаусса. LU-разложение матрицы.

Различают два больших класса методов решения системы (87): *точные* и *итерационные*. Точные методы за конечное число арифметических операций при отсутствии ошибок округления (что эквивалентно бесконечной разрядной сетке) дают точное решение задачи. В ходе применения итерационных методов рождается последовательность векторов, сходящаяся к решению.
В качестве наиболее популярного представителя методов первой группы рассмотрим метод Гаусса исключения неизвестных. Одна из его примитивных модификаций предполагает на первом шаге исключение $x^{(1)}$ с помощью первого уравнения из

остальных уравнений. С этой целью первое уравнение умножается на $m_{k1} = -a_{k1}/a_{11}$ и складывается с k-м уравнением и т.д. На втором шаге с помощью преобразованного второго уравнения исключается $x^{(2)}$ из последующих уравнений. После исключения $x^{(n-1)}$ завершается так называемый *прямой ход* метода Гаусса, результатом которого является треугольная матрица. *Обратный ход* метода Гаусса (гораздо менее трудоемкий) сводится к последовательному получению неизвестных, начиная с последнего уравнения.
Алгоритм в таком виде нуждается в существенном замечании. Нельзя заранее предвидеть, что элемент, стоящий в левом верхнем углу обрабатываемой матрицы, всегда будет отличен от нуля. Если ситуация с нулевым элементом возникнет, то, чтобы избежать деления на нуль, необходимо переставить строки, сделав элемент в этой позиции (ведущий элемент) ненулевым. Более того, желательно избегать не только нулевых, но и относительно малых ведущих элементов.
Наиболее известны следующие две стратегии выбора ведущего элемента:

1. *Полный* выбор.
   Здесь на k-м шаге в качестве ведущего берется наибольший по модулю элемент в неприведенной части матрицы. Затем строки и столбцы переставляются так, чтобы этот элемент поменялся местами с a_{kk}. В этом случае каждый раз осуществляется деление на максимальный по модулю элемент, но перестановка столбцов фактически сводится к перенумерации компонент вектора x.
2. *Частичный* выбор.
   Здесь на k-м шаге в качестве ведущего используют наибольший по модулю элемент первого столбца неприведенной части. Затем этот элемент меняют местами с a_{kk}, для чего переставляют только строки, избегая перенумерации компонент вектора $\mathbf{x}$.

С современной точки зрения метод Гаусса интерпретируется как разложение матрицы системы (87) в произведение двух треугольных матриц (**LU**-разложение). Этот факт отражает следующая теорема, приводимая без доказательства.

Theorem 28

Пусть $\mathbf{A}^{(k)}$ – главные миноры квадратной матрицы $\mathbf{A}$ порядка $m \times m$ ($k = 1,2,\ldots,m-1$). Предположим, что $\det(\mathbf{A}^{(k)}) \neq 0$. Тогда существует единственная нижняя треугольная матрица $\mathbf{L} = (l_{ij})$, где $l_{11} = l_{22} = \cdots = l_{nn} = 1$, и единственная верхняя треугольная матрица $\mathbf{U} = (u_{ij})$, такие, что $\mathbf{L} \cdot \mathbf{U} = \mathbf{A}$. Более того, $\det(\mathbf{A}) = u_{11} \cdot u_{22} \ldots u_{nn}$.

Эта теорема позволяет представить решение (87)

$$\mathbf{Ax} = \mathbf{b} \Longrightarrow (\mathbf{LU})\,\mathbf{x} = \mathbf{b} \Longrightarrow \mathbf{L}\,(\mathbf{Ux}) = \mathbf{b}$$

как решение двух систем с треугольными матрицами $\mathbf{L}$ и $\mathbf{U}$: $\mathbf{Ly} = \mathbf{b}$ и $\mathbf{Ux} = \mathbf{y}$. Решение первой системы с одновременным вычислением $\mathbf{L}$ и $\mathbf{U}$ соответсвует прямому ходу метода Гаусса, а решение второй системы – обратному ходу. Технологию **LU**-разложения проиллюстрируем на примере системы четвертого порядка без выбора

ведущего элемента. Пусть $m_{k1} = -a_{k1}/a_{11}, (k = 2, 3, 4)$. Первый шаг прямого хода эквивалентен умножению матрицы $\mathbf{A}$ и вектора $\mathbf{b}$ слева на матрицу $\mathbf{M}_1$:

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & 0 & 1 & 0 \\ m_{41} & 0 & 0 & 1 \end{pmatrix}, \mathbf{A}_2 = \mathbf{M}_1\mathbf{A},\ \mathbf{b}_2 = \mathbf{M}_1\mathbf{b}$$

На втором шаге матрица $\mathbf{A}_2$ и вектор $\mathbf{b}_2$ умножаются на матрицу $\mathbf{M}_2$, а на третьем шаге матрица $\mathbf{A}_3 = \mathbf{M}_2\mathbf{A}_2$ и вектор $\mathbf{b}_3 = \mathbf{M}_2\mathbf{b}_2$ умножаются на матрицу $\mathbf{M}_3$. $\mathbf{A}_4 = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1 \cdot \mathbf{A}$

$$\mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & m_{32} & 1 & 0 \\ 0 & m_{33} & 0 & 1 \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & m_{43} & 1 \end{pmatrix}$$

Согласно построению $\mathbf{A}_4$ есть верхняя треугольная матрица $\mathbf{U}$:

$$\mathbf{M} = \mathbf{M}_3\mathbf{M}_2\mathbf{M}_1, \quad \mathbf{MA} = \mathbf{U}, \quad \mathbf{L} = \mathbf{M}^{-1}, \quad \mathbf{A} = \mathbf{LU}, \text{ где}$$

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & -m_{32} & 1 & 0 \\ -m_{41} & -m_{42} & -m_{43} & 1 \end{pmatrix}$$

20.3 Подпрограммы DECOMP и SOLVE.

Реализованные в большинстве пакетов по линейной алгебре программы представляют собой набор из двух программ. В первой осуществляется $\mathbf{LU}$-разложение, а во второй решаются две системы с треугольными матрицами $\mathbf{L}$ и $\mathbf{U}$ ($\mathbf{Ly} = \mathbf{b}$ и $\mathbf{Ux} = \mathbf{y}$). Примером являются написанные на фортране программы `DECOMP` и `SOLVE`. Они имеют следующие параметры

```
DECOMP(NDIM, N, A, COND, IPVT, WORK)
SOLVE(NDIM, N, A, B, IPVT)
```

NDIM – объявленная в описании строчная размерность массива, в котором располагается матрица $\mathbf{A}$
N – порядок системы уравнений
A – матрица, подвергающаяся разложению (по окончании работы программы на ее месте располагаются матрицы $\mathbf{L}$ и $\mathbf{U}$)
COND – оценка числа обусловленности
IPVT – вектор индексов ведущих элементов (размерность $\mathbf{N}$)
WORK – рабочий одномерный массив (размерность $\mathbf{N}$)
B – вектор правых частей системы (87), где по окончании работы программы `SOLVE` размещается вектор решения $\mathbf{x}$.
В заключение оценим число арифметических операций в методе Гаусса. На каждом шаге исключения мы встречаемся с операциями деления и умножения-вычитания. Возьмем за единицу измерения операцию именно такого типа. На k-м шаге в одной

строке выполняется одно деление и k умножений-вычитаний. Тогда для всех $k-1$ строк имеем: $(k+1)(k-1) = k^2 - 1$ операций. В прямом ходе Гаусса таких шагов m. В итоге получаем:

$$\sum_{k=1}^{m} \left(k^2 - 1\right) = \sum_{k=1}^{m} k^2 - m = \frac{2m^3 + 3m^2 - 5m}{6}$$

При больших значениях m хорошим приближением для числа операций будет $m^3/3$. Для обратного хода нужно на порядок меньше операций (одно деление и $k-1$ умножение-вычитание при вычислении $x^{(k)}$, что для всех компонент дает величину $\sum_{k=1}^{m} \frac{m^2+m}{2}$).

Вопрос 21. Метод последовательных приближений для решения линейных систем.

Итерационные методы (еще одно название – *методы последовательных приближений*) дают возможность для системы (87) строить последовательность векторов $\mathbf{x}_0, \mathbf{x}_1, \ldots$, пределом которой должно быть точное решение $\mathbf{x}^*$

$$\mathbf{x}^* = \lim_{n\to\infty} \mathbf{x}_n$$

На практике построение последовательности обрывается как только достигается желаемая точность. Чаще всего для достаточно малого значения $\varepsilon > 0$ контролируется выполнение оценки $|\mathbf{x}^* - \mathbf{x}_n| < \varepsilon$. Метод последовательных приближений может быть построен, например, по следующей схеме. Эквивалентными преобразованиями приведем систему (87) к виду

$$\mathbf{x} = \mathbf{C}\mathbf{x} + \mathbf{d} \tag{91}$$

Под эквивалентными преобразованиями будем понимать преобразования, сохраняющие решение системы (т.е. решения (87) и (91) совпадают). Точное решение $\mathbf{x}^*$ системы имеет вид

$$\mathbf{x}^* = (\mathbf{E} - \mathbf{C})^{-1}\,\mathbf{d}$$

Вместо (91) будем решать следующую систему разностных уравнений

$$\mathbf{x}_{n+1} = \mathbf{C}\mathbf{x}_n + \mathbf{d} \tag{92}$$

пошаговым методом. При этом необходимо решить целый ряд вопросов. Сходится ли итерационный процесс (92)? Если сходится, что является пределом последовательности, и какова скорость сходимости?
Ранее было показано, что решение системы (92) записывается в виде

$$\mathbf{x}_n = \mathbf{C}^n\mathbf{x}_0 + (\mathbf{E} - \mathbf{C}^n)\,(\mathbf{E} - \mathbf{C})^{-1}\,\mathbf{d}$$

Вычитания из него точное решение, получаем

$$\mathbf{x}_n - \mathbf{x}^* = \mathbf{C}^n\mathbf{x}_0 - \mathbf{C}^n\,(\mathbf{E} - \mathbf{C})^{-1}\,\mathbf{d} = \mathbf{C}^n\,(\mathbf{x}_0 - \mathbf{x}^*)$$

Чтобы обеспечить условие сходимости, все элементы матрицы $\mathbf{C}^n$ должны стремиться к нулю при $n \to \infty$. Для этого, в свою очередь, необходимо и достаточно, чтобы *все* собственные значения матрицы $\mathbf{C}$ были по модулю меньше единицы

$$|\lambda_k| < 1$$

Поскольку нахождение всех собственных значений доставляет значительные трудности, вместо этого условия можно использовать достаточное условие сходимости

$$\|\mathbf{C}\| < 1$$

которое справедливо для любой канонической нормы.
Количество итераций по формуле (92) будет тем меньше, чем меньше по модулю собственные значения матрицы $\mathbf{C}$ и чем ближе к $\mathbf{x}^*$ выбрано начальное приближение $\mathbf{x}_0$.

На практике при реализации на компьютере процесс (92) прерывается либо заданием максимального числа итераций, либо условием $\|\mathbf{x}_{n+1} - \mathbf{x}_n\| < \varepsilon$. Таким образом, основным неформальным моментом является такое приведение системы (87) к виду (91), чтобы выполнялось условие ограниченности собственных значений. В общем случае универсальный способ такого перехода с малой трудоемкостью отсутствует, и поэтому часто используется специфика решаемой задачи. Рассмотрим следующий пример.
Пусть диагональные элементы матрицы $\mathbf{A}$ в (87) значительно превышают по модулю остальные элементы в соответствующих строках. Разделим каждое уравнение на соответствующий диагональный элемент и получим

$$\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}, \qquad \mathbf{x} = \left(\mathbf{E} - \tilde{\mathbf{A}}\right)\mathbf{x} + \tilde{\mathbf{b}}$$

На главной диагонали у матрицы $\tilde{\mathbf{A}}$ стоят единицы, а у матрицы $\left(\mathbf{E} - \tilde{\mathbf{A}}\right)$ расположены нули. Вне главной диагонали у обеих матриц находятся малые по модулю элементы, что позволяет, выбрав $\mathbf{C} = \mathbf{E} - \tilde{\mathbf{A}}$, легко обеспечить условие ограниченности нормы и быструю сходимость итерационного процесса.

Вопрос 22. Методы бисекции, секущих, обратной параболической интерполяции для решения нелинейных уравнений. Подпрограмма ZEROIN.

Большинство методов, предназначенных для нахождения корней нелинейного уравнения $f(x) = 0$ предполагает, что заранее определены некоторые промежутки, где это уравнение имеет только один корень. Поэтому задаче нахождения решения с заданной точностью предшествует этап *отделения корней*, связанный с исследованием количества, характера расположения корней и нахождением их грубого приближения. Уравнение может вообще не иметь решений, а может встретиться ситуация, когда корней бесконечно много. Этот этап формализуется лишь частично и чаще относится к области математического искусства. Универсального эффективного метода в общем случае нет. На практике в большинстве случаев ограничиваются приближенным построением графика $y = f(x)$ или составлением таблицы для $f(x)$ с некоторым шагом и нахождением участков, где функция меняет знак. При этом шаг не должен быть слишком крупным (должна быть уверенность в не более, чем одном нуле между узлами), а, с другой стороны, он не должен быть излишне мал (иначе резко возрастает объем вычислений). Иногда удобно преобразовать уравнение к виду $\varphi(x) = \mu(x)$, а затем искать точку пересечения графиков $y = \varphi(x)$ и $y = \mu(x)$, что и будет начальным приближением.
В рамках раздела будем использовать следующие обозначения: $x^* = x_n + \varepsilon_n$, где x^* – точное решение, x_n – очередное приближение к решению, ε_n – погрешность.
Остановимся лишь на вещественных корнях уравнения $f(x) = 0$, считая, что функция $f(x)$ нужное число раз непрерывно дифференцируема для выбранного метода и установлен промежуток $[a, b]$, где находится единственный корень. При этом $f(a) \cdot f(b) < 0$. Тогда наиболее простым и абсолютно надежным способом является *метод бисекции* (или метод *дихотомии*, *половинного деления*). Его алгоритм представим следующим образом:

1. Вычислить $f(a)$ и $f(b)$.
2. Положить $c = \dfrac{a+b}{2}$ и вычислить $f(c)$.
3. Если $\operatorname{sign}(f(a)) == \operatorname{sign}(f(c))$, заменить a на c. Иначе заменить b на c.
4. Если $|b - a| > \varepsilon$, перейти к шагу 2. Иначе закончить вычисления.

Одна итерация алгоритма позволяет *гарантированно* сократить исходный промежуток в два раза независимо от вида функции.
Другой алгоритм, называемый *методом секущих*, можно построить, используя интерполяционный полином Лагранжа первой степени для $f(x)$ по двум узлам a и b. Тогда нуль этого полинома принимается в качестве очередного приближения к корню уравнения.

$$Q_1(x) = \frac{x-b}{a-b}f(a) + \frac{x-a}{b-a}f(b), \qquad c = a - \frac{b-a}{f(b)-f(a)}f(a)$$

Новый промежуток будет $[c, b]$ или $[a, c]$ в зависимости от знака $f(x)$ в точке c. Скорость сходимости метода секущих определяется неравенством $|x^* - x_{k+1}| \le |x^* - x_k|^{1,618}$.
Следует отметить, что замедление сходимости этого алгоритма часто наблюдается,

когда очередное приближение получается слишком близко к одному из концов промежутка.
Если функция вычислена более, чем в двух точках, то эта информация может быть использована в дальнейшем. Так, в *методе обратной квадратичной интерполяции* строится интерполяционный полином второй степени по точкам x_k, x_{k-1}, x_{k-2}. Для обратной функции с выполнением условий $x_i = g(f_i), i = k, k-1, k-2$. В качестве следующего приближения берется $x_{k+1} = g(0)$. Одна из предыдущих точек удаляется. Важно, чтобы три значения f_i были различными, тогда исключается деление на нуль, и сходимость метода определяется неравенством $|x^* - x_{k+1}| \leq |x^* - x_k|^{1,839}$.

Сочетание методов бисекции и обратной квадратичной интерполяции реализовано в процедуре-функции `ZEROIN(A, B, F, EPS)`.
A, **B** – концы интервала.
F – имя процедуры-функции, имеющей лишь один аргумент, для которого вычисляется $f(x)$.
EPS – граница погрешности, допустимой в результате.

Основным алгоритмом является метод обратной квадратической интерполяции (если узлы не являются различными, то используется метод секущих). Если очередное приближение получается слишком близким к одному из краев промежутка, то осуществляется переключение на метод бисекции.

Вопрос 23. Методы последовательных приближений и Ньютона для решения нелинейных уравнений и систем.

23.1 Метод последовательных приближений для решения нелинейных уравнений.

Эквивалентными преобразованиями приведем $f(x) = 0$ к виду

$$x = \varphi(x) \tag{93}$$

Вместо этого уравнения предлагается решить разностное

$$x_{n+1} = \varphi(x_n) \tag{94}$$

пошаговым методом. Для оценки сходимости запишем равенство (93) в точке x^* и вычтем из него равенство (94):

$$\varepsilon_{n+1} = x^* - x_{n+1} = \varphi(x^*) - \varphi(x_n) = \varphi(x_n + \varepsilon_n) - \varphi(x_n)$$

Раскладывая $\varphi(x_n + \varepsilon_n)$ в ряд по степеням ε_n и ограничиваясь в остаточном члене первой производной, получаем уравнение погрешности

$$\varepsilon_{n+1} = \varphi(x_n + \varepsilon_n) - \varphi(x_n) = \varphi(x_n) + \varepsilon_n\varphi'(\eta) - \varphi(x_n) = \varepsilon_n\varphi'(\eta)$$

Отсюда непосредственно следует, что для убывания погрешности необходимо потребовать выполнение условия

$$|\varphi'(\eta)| < 1$$

Искусство пользователя, таким образом, заключается в приведении уравнения $f(x) = 0$ к виду (93) так, чтобы имело место это неравенство. При этом, чем меньше по модулю значение производной, тем быстрее достигается желаемая точность.

23.2 Метод Ньютона для решения нелинейных уравнений.

Высокой скоростью сходимости в ряде случаев обладает *метод Ньютона* (или *метод касательных*). Подставляя в уравнение $f(x) = 0$ его корень x^*, и раскладывая в ряд по степеням ε_n

$$0 = f(x^*) = f(x_n + \varepsilon_n) = f(x_n) + \varepsilon_n f'(x_n) + \frac{\varepsilon_n^2}{2!}f''(\eta)$$

пренебрежем последним слагаемым и для ε_n получим

$$\varepsilon_n \approx -\frac{f(x_n)}{f'(x_n)}$$

Тогда рабочая формула Ньютона принимает вид:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{95}$$

В отличие от методов биссекции, секущих и обратной квадратической интерполяции сходимость (95) обеспечивается далеко не всегда. В перечне достаточных условий сходимости фигурирует не только существование ненулевой производной $f'(x)$ в точках x_n, но и ее знакопостоянство. Если, тем не менее, методу Ньютона обеспечено хорошее начальное приближение, то в дальнейшем убывание погрешности носит квадратичный характер. Для доказательства этого факта из очевидного неравенства $x^* = x^*$ вычтем уравнение (95):

$$\varepsilon_{n+1} = \varepsilon_n + \frac{f(x_n)}{f'(x_n)} = \frac{f(x_n) + \varepsilon_n f'(x_n)}{f'(x_n)} \tag{96}$$

Упрощая числитель при помощи выполненного разложения в ряд, получаем

$$\varepsilon_{n+1} = -\frac{f''(\eta)}{2f'(x_n)}\varepsilon_n^2$$

$$|\varepsilon_{n+1}| < C\varepsilon_n^2, \qquad \left|\frac{f''(\eta)}{2f'(x_n)}\right| < C \tag{97}$$

Для расширения области сходимости можно использовать метод Ньютона с регулировкой шага

$$x_{n+1} = x_n - \alpha_n \frac{f(x_n)}{f'(x_n)}, \qquad 0 < \alpha_n \leq 1 \tag{98}$$

Первоначально, когда начальное приближение x_0 еще далеко от x^*, параметр α_n выбирают меньше 1 (часто на практике это примерно $1/3$), а по мере приближения x_n к x^* значение $\alpha_n \to 1$, превращая уравнение в обычный метод Ньютона. В некоторых случаях это и позволяет расширить область сходимости.
Широкое распространение получил и модифицированный метод Ньютона с постоянной производной

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \tag{99}$$

Сходимость в этом случае несколько замедляется, но заметно уменьшается трудоемкость отдельной итерации, не требующей теперь вычисления производной.

23.3 Метод Ньютона для решения нелинейных систем.

Решение систем нелинейных уравнений доставляет очень большие трудности, так как нет универсальных алгоритмов решения этих задач, особенно для больших m.
Достоинством метода Ньютона является то, что он обобщается на системы уравнений. С этой целью обратимся к уравнению $f(x) = 0$, полагая $\mathbf{x}$ и $\mathbf{f}$ векторами:

$$\mathbf{x} = \left(x^{(1)}, x^{(2)}, \ldots, x^{(m)}\right)^T, \quad \mathbf{f} = \left(f^{(1)}, f^{(2)}, \ldots, f^{(m)}\right)^T$$

Формула, являющаяся аналогом (95) может быть получена таким же образом, как и для скалярного случая, на основе разложения в ряд. При этом ε_n представляет собой вектор, и разложение в ряд необходимо проводить по всем компонентам этого вектора для $\mathbf{f}$, как функции многих переменных.

В качестве иллюстрации остановимся на системе двух нелинейных уравнений с прежними обозначениями $\mathbf{x}_* = \mathbf{x}_n + \boldsymbol{\varepsilon}_n$

$$\mathbf{x}_* = \begin{pmatrix} x_*^{(1)} \\ x_*^{(2)} \end{pmatrix}, \quad \mathbf{x}_n = \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \end{pmatrix}, \quad \boldsymbol{\varepsilon}_n = \begin{pmatrix} \varepsilon_n^{(1)} \\ \varepsilon_n^{(2)} \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f^{(1)} \\ f^{(2)} \end{pmatrix}$$

Подставим в первое уравнение значение решения $\mathbf{x}_*$ и разложим в ряд по $\varepsilon_n^{(1)}$ и $\varepsilon_n^{(2)}$, пренебрегая малыми второго порядка и выше

$$\begin{aligned} 0 = f^{(1)}\left(x_*^{(1)}, x_*^{(2)}\right) = f^{(1)}\left(x_n^{(1)} + \varepsilon_n^{(1)}, x_n^{(2)} + \varepsilon_n^{(2)}\right) = f^{(1)}\left(x_n^{(1)}, x_n^{(2)} + \varepsilon_n^{(2)}\right) + \\ + \frac{\partial f^{(1)}}{\partial x^{(1)}}\left(x_n^{(1)}, x_n^{(2)} + \varepsilon_n^{(2)}\right) \cdot \varepsilon_n^{(1)} + (**) = f^{(1)}\left(x_n^{(1)}, x_n^{(2)}\right) + \frac{\partial f^{(1)}}{\partial x^{(2)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(2)} \\ + \frac{\partial f^{(1)}}{\partial x^{(1)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(1)} + (**) \end{aligned}$$

Аналогичное разложение можем записать для второго уравнения и суммарно получаем

$$\begin{aligned} \frac{\partial f^{(1)}}{\partial x_n^{(1)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(1)} + \frac{\partial f^{(1)}}{\partial x_n^{(2)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(2)} = -f^{(1)}\left(x_n^{(1)}, x_n^{(2)}\right) \\ \frac{\partial f^{(2)}}{\partial x_n^{(1)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(1)} + \frac{\partial f^{(2)}}{\partial x_n^{(2)}}\left(x_n^{(1)}, x_n^{(2)}\right) \cdot \varepsilon_n^{(2)} = -f^{(2)}\left(x_n^{(1)}, x_n^{(2)}\right) \end{aligned}$$

Эта система может быть записана в матричной форме

$$\frac{\partial \mathbf{f}(\mathbf{x}_n)}{\partial \mathbf{x}}\left(\mathbf{x}_{n+1} - \mathbf{x}_n\right) = -\mathbf{f}(\mathbf{x}_n) \tag{100}$$

или

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \left(\frac{\partial \mathbf{f}(\mathbf{x}_n)}{\partial \mathbf{x}_n}\right)^{-1} \mathbf{f}(\mathbf{x}_n) \tag{101}$$

где $\dfrac{\partial \mathbf{f}(\mathbf{x}_n)}{\partial \mathbf{x}}$ – матрица Якоби решаемой системы

$$\frac{\partial \mathbf{f}(\mathbf{x}_n)}{\partial \mathbf{x}} = \begin{pmatrix} \dfrac{\partial f^{(1)}}{\partial x^{(1)}}\left(x_n^{(1)}, x_n^{(2)}\right) & \dfrac{\partial f^{(1)}}{\partial x^{(2)}}\left(x_n^{(1)}, x_n^{(2)}\right) \\ \dfrac{\partial f^{(2)}}{\partial x^{(1)}}\left(x_n^{(1)}, x_n^{(2)}\right) & \dfrac{\partial f^{(2)}}{\partial x^{(2)}}\left(x_n^{(1)}, x_n^{(2)}\right) \end{pmatrix}$$

Представление метода в виде (101) позволяет уменьшить вычислительные затраты, поскольку не требует обращения матрицы Якоби, а сводится к решению линейной алгебраической системы на каждом шаге итерационного процесса. Как и в скалярном случае, в достаточно малой окрестности корня итерации сходятся и скорость сходимости квадратичная. Значительно уменьшается объем вычислений в модифицированном варианте метода Ньютона, когда матрица Якоби вычисляется однократно, раскладывается в произведение треугольных матриц программой `DECOMP`, а затем для получения очередного приближения используется только программа `SOLVE`.

23.4 Метод последовательных приближений для решения нелинейных систем.

Формула (94) сохраняет прежний вид, только $\mathbf{x}$ и $\boldsymbol{\varphi}(\mathbf{x})$ являются векторами. Достаточным условием сходимости является выполнение неравенства $\left\|\dfrac{\partial \boldsymbol{\varphi}}{\partial \mathbf{x}}\right\| < 1$, где $\dfrac{\partial \boldsymbol{\varphi}}{\partial \mathbf{x}}$ – матрица Якоби.

Вопрос 24. Задача Коши решения обыкновенных дифференциальных уравнений. Явный и неявный методы ломаных Эйлера, метод трапеций.

Как известно, в практических приложениях решения дифференциального уравнения или системы уравнений описывают динамику разнообразных явлений и процессов (например, движение совокупности взаимодействующих материальных точек, химическую кинетику, процессы в электрических цепях и т.п.). Однако интегрируемых в явном виде дифференциальных уравнений крайне мало. Поэтому столь важны численные методы.

Задача Коши (или задача с начальными условиями) из множества решений для системы

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}(t)) \tag{102}$$

где t – независимая переменная, $\mathbf{x}(t) = \left(x^{(1)}, \ldots, x^{(m)}\right)^T$ – вектор искомых функций, удовлетворяющих уравнению, и $\mathbf{f}(t, \mathbf{x})$ – вектор заданных, нужное число раз дифференцируемых функций, выделяет одно решение, проходящее через начальную точку $(t_0, \mathbf{x}_0)$. Аналогично ставится задача и для дифференциального уравнения m-го порядка, разрешенного относительно старшей производной, которое сводится к системе (102) из m уравнений первого порядка.

Если правые части $\mathbf{f}(t, \mathbf{x})$, а также элементы матрицы Якоби $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ непрерывны и ограничены в некоторой окрестности точки $t_0, \mathbf{x}_0$, то задача Коши имеет единственное решение. Первоначально, исключительно для простоты рассуждений, будем полагать, что (102) представляет собой одно уравнение. Вместе с тем, абсолютно все излагаемые в настоящем разделе методы сохраняют свой внешний вид и для случая, когда $\mathbf{x}$ и $\mathbf{f}$ являются векторами и (102) является системой уравнений.

Общий подход к решению (102) заключается в приближенном сведении дифференциального уравнения к некоторому разностному уравнению, которое, в свою очередь, решается затем пошаговым методом. С этой целью выполним дискретизацию независимой переменной: $t_n = t_0 + nh$, где h – шаг интегрирования (шаг дискретности), а значения решения и его производной в этих точках кратко обозначим как $\mathbf{x}_n = \mathbf{x}(t_n)$ и $\mathbf{f}_n = \mathbf{f}(t, \mathbf{x}_n)$. Интегрируя (102) на промежутке $[t_n, t_{n+1}]$, получаем формулу

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \int_{t_n}^{t_{n+1}} \mathbf{f}(\tau, \mathbf{x}(\tau))d\tau \tag{103}$$

которую можно считать базовой для построения большей части известных разностных схем. Различные методы при этом отличаются способом вычисления интеграла в равенстве (103).

Использование квадратурных формул левых и правых прямоугольников, а также формулы трапеций, приводит соответственно к следующим численным методам:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) \tag{104}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_{n+1}, \mathbf{x}_{n+1}) \tag{105}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{f}(t_n, \mathbf{x}_n) + \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1})\right) \tag{106}$$

которые получили название *явного метода ломаных Эйлера*, *неявного метода ломаных Эйлера* и *неявного метода трапеций*. Разностные уравнения (105) и (106) неявно задают значения $\mathbf{x}_{n+1}$ и требуют решения нелинейных уравнений на каждом шаге интегрирования.

Заметим, что для остальных известных нам квадратурных формул требуется вычисление значения функции в точках между узлами, в то время как решение $\mathbf{x}(t)$ в этих точках не определено. То есть, подобный способ «превращения» квадратурных формул в формулы для решения дифференциальных уравнений здесь не пройдет.

Вопрос 25. Методы Адамса. Локальная и глобальная погрешности, степень метода.

25.1 Методы Адамса.

Рассмотрим другой подход решения задачи Коши (102). Для построения решения в точке t_{n+1} будем использовать информацию в ранее полученных точках $t_n, t_{n-1}, \ldots$. Так, по двум предыдущим точкам t_n и t_{n-1} построим интерполяционный полином первой степени для функции $\mathbf{f}(t, \mathbf{x})$

$$\mathbf{f}(\tau, \mathbf{x}(\tau)) \approx \frac{\tau - t_{n-1}}{t_n - t_{n-1}}\mathbf{f}_n + \frac{\tau - t_n}{t_{n-1} - t_n}\mathbf{f}_{n-1}$$

и подставим его в формулу (103). Попутно *заметим, что* полином используется вне промежутка интерполирования, т.е. проводится *экстраполяция*. Получаем следующий численный метод:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}(3\mathbf{f}_n - \mathbf{f}_{n-1}) \tag{107}$$

Использование трех точек t_n, t_{n-1}, t_{n-2} и полинома второй степени приведет к формуле

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{12}(23\mathbf{f}_n - 16\mathbf{f}_{n-1} + 5\mathbf{f}_{n-2}) \tag{108}$$

а для четырех точек разностная схема алгоритма принимает вид

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{24}(55\mathbf{f}_n - 59\mathbf{f}_{n-1} + 37\mathbf{f}_{n-2} - 9\mathbf{f}_{n-3}) \tag{109}$$

Все эти методы получили название *методов Адамса*. Они принадлежат семейству ~~Адамс~~ многошаговых алгоритмов, разностные уравнения которых имеют порядок выше первого. Методы (107) – (109) являются *явными* методами Адамса. Если в состав точек, по которым строится полином, включить t_{n+1}, то возникают *неявные* методы Адамса. Для двух точек t_{n+1}, t_n получается метод трапеций (106), а для трех точек t_{n+1}, t_n, t_{n-1} и четырех точек $t_{n+1}, t_n, t_{n-1}, t_{n-2}$ – следующие два метода

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{12}(5\mathbf{f}_{n+1} + 8\mathbf{f}_n - \mathbf{f}_{n-1}) \tag{110}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{24}(9\mathbf{f}_{n+1} + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2}) \tag{111}$$

Несомненным достоинством явных методов Адамса является тот факт, что все они независимо от своей точности требуют лишь однократного вычисления функции $\mathbf{f}(t, \mathbf{x})$ на одном шаге и конкурировать с ними в этом плане весьма трудно. Остальные значения производной решения берутся с предыдущих шагов. Вместе с тем, методы Адамса, как и другие многошаговые алгоритмы, не являются самостартующими, т.е. они требуют для начала интегрирования специальных стартовых алгоритмов для расчета дополнительных начальных условий. В качестве этих стартовых методов могут быть использованы любые другие методы, например методы Рунге-Кутты, или специально разработанные для этих целей алгоритмы.

Неявные методы Адамса могут использоваться как сами по себе (тогда на каждом шаге решаются нелинейные уравнения относительно $\mathbf{x}_{n+1}$), так и в паре с явными методами. В последнем случае значение $\mathbf{x}_{n+1}$ сначала оценивается явным методом ($\mathbf{x}^{\text{Э}}_{n+1}$), а затем уточняется неявным алгоритмом.
Например, такую пару методов образуют методы (109) и (111):

$$\mathbf{x}^{\text{Э}}_{n+1} = \mathbf{x}_n + \frac{h}{24}\left(55\mathbf{f}_n - 59\mathbf{f}_{n-1} + 37\mathbf{f}_{n-2} - 9\mathbf{f}_{n-3}\right) \tag{112}$$

$$\mathbf{x}^{\text{Э}}_{n+1} = \mathbf{x}_n + \frac{h}{24}\left(9\mathbf{f}(t_{n+1}, \mathbf{x}^{\text{Э}}_{n+1}) + 19\mathbf{f}_n - 5\mathbf{f}_{n-1} + \mathbf{f}_{n-2}\right) \tag{113}$$

В зарубежной литературе совместное использование явного и неявного методов называют *методами прогноза-коррекции*. В нашей литературе часто используют термин *экстраполяционные* методы для (112) и *интерполяционные* методы для (113).

25.2 Локальная и глобальная погрешности, степень метода.

Теперь обратимся к анализу погрешности численных методов и начнем с самого простого алгоритма – явного метода ломаных Эйлера. Рассмотрим частный случай формулы (104), когда функция $\mathbf{f}(t, \mathbf{x})$ не зависит от $\mathbf{x}$. Тогда явный метод ломаных Эйлера превращается в квадратурную формулу левых прямоугольников:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}\,(t_n) = \mathbf{x}_0 + h\sum_{k=0}^{n}\mathbf{f}_k \tag{114}$$

При этом общая погрешность в точке t_n является точной суммой погрешностей, допущенных на каждом отдельном шаге.
Иная ситуация складывается, когда $\mathbf{f}(t, \mathbf{x})$ зависит от $\mathbf{x}$. Только погрешность первого шага формулы (104) при $n = 0$ вычисляется аналогично (114).

$$\mathbf{x}_1 = \mathbf{x}_0 + h\mathbf{f}(t_0, \mathbf{x}_0)$$

Уже на втором шаге при $n = 1$ эта погрешность сложным образом зависит от погрешности первого шага, так как при вычислении $\mathbf{f}(t_1, \mathbf{x}_1)$ используется приближенное значение $\mathbf{x}_1$

$$\mathbf{x}_2 = \mathbf{x}_1 + h\mathbf{f}(t_1, \mathbf{x}_1)$$

В общем случае на n-м шаге погрешность очень сложно зависит от всех погрешностей, допущенных на предыдущих шагах. Разностное уравнение метода может оказаться неустойчивым, и тогда происходит неприемлемый рост погрешности.
Устойчивость разностной схемы связана с выбранным методом, шагом интегрирования и видом функции $\mathbf{f}(t, \mathbf{x})$. Важно так выбрать сам метод и шаг для него, чтобы погрешность решения была бы приемлемой. В соответствии со сказанным вводятся погрешности двух видов.

Definition 23: Локальная погрешность

Это погрешность, допущенная на одном шаге при условии, что решение во всех предыдущих точках вычислено точно.

Definition 24: Глобальная погрешность

Это разность между точным и приближенным решением на n-м шаге.

Именно глобальная погрешность является истинной погрешностью. Локальная погрешность совпадает с ней лишь на первом шаге. Однако, в общем случае оценка глобальной погрешности крайне затруднена, а чаще невозможна, и поэтому оценивают локальную погрешность на каждом шаге.
Малая величина локальной погрешности вовсе не гарантирует малую величину глобальной, но если есть уверенность, что устойчивость разностного уравнения метода обеспечена, то из малой величины локальной погрешности следует, что глобальная не будет слишком велика. Будем пока полагать, что устойчивость обеспечена, и рассмотрим подробнее характеристики локальной погрешности.
Важной характеристикой является *степень* (или *порядок точности*) метода. Все ранее рассмотренные методы могут быть записаны в следующем виде:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{F}\left(t_n, h, \mathbf{x}_{n+1}, \mathbf{x}_n, \mathbf{x}_{n-1}, \ldots, \mathbf{x}_{n-k}\right) \tag{115}$$

Разложим выражение в правой части равенства в ряд Тейлора по степеням h в точке t_n:

$$\mathbf{x}_{n+1} = \mathbf{x}\left(t_n + h\right) = \mathbf{x}_n + \sum_{k=1}^{\infty} \alpha_k(t_n) h^k \frac{d^k \mathbf{x}(t_n)}{dt^k} \tag{116}$$

где коэффициенты $\alpha_k(t_n)$ определяются выбранным методом.
С другой стороны, значение $\mathbf{x}_{n+1} = \mathbf{x}(t_n + h)$ может быть представлено, в свою очередь, точным разложением в ряд

$$\mathbf{x}_{n+1} = \mathbf{x}(t_n + h) = \mathbf{x}_n + \sum_{k=1}^{\infty} \frac{h^k}{k!} \frac{d^k \mathbf{x}(t_n)}{dt^k} \tag{117}$$

Метод имеет степень s, если коэффициенты разложения (116) совпадают с соответствующими коэффициентами (117) до h^s включительно. В качестве примера определим степень некоторых ранее полученных методов.
Для явного метода ломаных Эйлера:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) = \mathbf{x}_n + h\mathbf{x}'(t_n)$$

Вычитая эту формулу из (117), видим, что коэффициенты совпадают лишь при h^1, и для локальной погрешности этого метода первой степени справедлива оценка: $\boldsymbol{\varepsilon}_{n+1} = \dfrac{h^2 \mathbf{x}''(\eta)}{2}$. Первую степень имеет и неявный метод ломаных Эйлера (105) с локальной погрешностью $\boldsymbol{\varepsilon}_{n+1} = -\dfrac{h^2 \mathbf{x}''(\eta)}{2}$.
Аналогично убеждаемся, что метод Адамса имеет вторую степень (совпадают члены разложения при h^1 и h^2)

$$\begin{aligned}
&\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(3\mathbf{x}'(t_n) - \mathbf{x}'(t_n - h)\right) = \\
&= \mathbf{x}_n + \frac{h}{2}\left(3\mathbf{x}'(t_n) - \mathbf{x}'(t_n) + h\mathbf{x}''(t_n) - \frac{h^2}{2}\mathbf{x}'''(t_n) + \ldots\right) = \\
&= \mathbf{x}_n + h\mathbf{x}'(t_n) + \frac{h^2}{2}\mathbf{x}''(t_n) - \frac{h^3}{4}\mathbf{x}'''(t_n) + \ldots
\end{aligned}$$

а после вычитания этого выражения из (117) получаем оценку локальной погрешности: $\varepsilon_{n+1} = \dfrac{5h^3\mathbf{x}'''(\eta)}{12}$. Также второй порядок точности будет и у метода трапеций (106)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{f}(t_n, \mathbf{x}_n) + \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1})\right) = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{x}'(t_n) + \mathbf{x}'(t_n + h)\right) =$$
$$= \mathbf{x}_n + \frac{h}{2}\left(\mathbf{x}'(t_n) + \mathbf{x}'(t_n) + h\mathbf{x}''(t_n) + \frac{h^2}{2}\mathbf{x}'''(t_n) + \dots\right) =$$
$$= \mathbf{x}_n + h\mathbf{x}'(t_n) + \frac{h^2}{2}\mathbf{x}''(t_n) + \frac{h^3}{4}\mathbf{x}'''(t_n) + \dots$$

Можно убедиться, что методы Адамса имеют третью степень, а методы (109) и (111) – четвертую степень соответственно. Главный член погрешности метода s-й степени содержит, как множитель, величину h^{s+1}

Вопрос 26. Методы Рунге-Кутты. Подпрограмма RKF45.

Метод трапеций является неявным. Что произойдет, если вычислить $\mathbf{x}_{n+1}$ сначала по формуле (104), а затем уточнить по формуле (106)?

$$\mathbf{x}^*_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n), \tag{118}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{f}(t_n, \mathbf{x}_n) + \mathbf{f}(t_{n+1}, \mathbf{x}_{n+1})\right)$$

Полученный одношаговый метод, называемый *методом Эйлера-Коши*, является уже явным. Как будет показано, он имеет вторую степень, которая достигается ценой двух вычислений функции $\mathbf{f}(t, \mathbf{x})$ на каждом шаге.
Приведем еще один пример. Сделаем полшага с помощью явного метода ломаных Эйлера, а затем используем полученное значение в квадратурной формуле средних прямоугольников, примененной к интегралу (103).

$$\mathbf{x}^*_{n+1/2} = \mathbf{x}_n + \frac{h}{2}\mathbf{f}(t_n, \mathbf{x}_n) \tag{119}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}^*_{n+1/2}\right)$$

Этот метод, называемый *усовершенствованным методом ломаных Эйлера*, также имеет вторую степень и требует двукратного вычисления $\mathbf{f}(t, \mathbf{x})$. Приведенные примеры укладываются в следующую схему. Вычислим $\mathbf{f}(t, \mathbf{x})$ дважды в некоторых точках и их линейную комбинацию используем для получения $\mathbf{x}_{n+1}$

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n) \tag{120}$$

$$\mathbf{k}_2 = h\mathbf{f}\left(t_n + \alpha_2 h, \mathbf{x}_n + \beta_{21}\mathbf{k}_1\right)$$
$$\mathbf{x}_{n+1} = \mathbf{x}_n + p_1\mathbf{k}_1 + p_2\mathbf{k}_2$$

Параметры $p_1, p_2, \alpha_2, \beta_{21}$ будем выбирать так, чтобы разложение формулы метода (120) в ряд максимальным образом совпадало с разложением точного решения (117). С этой целью отметим, что согласно (102)

$$\mathbf{x}' = \mathbf{f}(t, \mathbf{x}) \qquad \mathbf{x}'' = \frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{f}$$

и (117) имеет вид

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) + \frac{h^2}{2}\left(\frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right) + \dots$$

Раскладывая (120) в ряд и приравнивая коэффициенты при соответствующих степенях h, добиваемся того, что формула (120) задает методы второй степени

$$\mathbf{x}_{n+1} = \mathbf{x}_n + p_1 h\mathbf{f}_n + p_2 h\left(\mathbf{f}_n + \alpha_2 h\frac{\partial \mathbf{f}}{\partial t} + \beta_{21} h\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\mathbf{f} + \dots\right)$$

$$p_1 + p_2 = 1, \quad p_2\alpha_2 = 1/2, \quad p_2\beta_{21} = 1/2 \tag{121}$$

Условия (121) представляют собой три уравнения с четырьмя неизвестными и, следовательно, методов второй степени вида (120) бесконечно много. В частности, определяя параметры $p_1 = p_2 = 1/2$, $\alpha_2 = \beta_{21} = 1$, получаем метод Эйлера-Коши, а набор $p_1 = 0$, $p_2 = 1, \alpha_2 = \beta_{21} = 1/2$ задает усовершенствованный метод ломаных Эйлера. В то же время построить метод третьей степени с двумя вычислениями $\mathbf{f}(t, \mathbf{x})$ не удается.

Увеличивая число вычислений функции $\mathbf{f}(t, \mathbf{x})$ на одном шаге, получаем семейство методов Рунге-Кутты в виде

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n),\ \mathbf{k}_r = h\mathbf{f}\left(t_n + \alpha_r h, \mathbf{x}_n + \sum_{i=1}^{r-1} \beta_{ri}\mathbf{k}_i\right), \quad r = 1,2,\ldots,s$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{r=1}^{s} p_r \mathbf{k}_r$$

Коэффициенты методов вычисляем аналогично тому, как это было выполнено для методов второй степени. При этом, если для метода второй степени достаточно рассчитать $\mathbf{f}(t, \mathbf{x})$ два раза на одном шаге, то метод третьей степени требует трех вычислений $\mathbf{f}(t, \mathbf{x})$, а метод четвертой степени – четырех таких вычислений. Все эти методы, как и методы второй степени, образуют семейства. Среди них наиболее популярными являются следующие методы третьей степени:

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n), \qquad \mathbf{k}_2 = h\mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{\mathbf{k}_1}{2}\right)$$

$$\mathbf{k}_3 = h\mathbf{f}\left(t_n + h, \mathbf{x}_n - \mathbf{k}_1 + 2\mathbf{k}_2\right), \qquad \mathbf{x}_{n+1} = \mathbf{x}_n + \left(\frac{\mathbf{k}_1 + 4\mathbf{k}_2 + \mathbf{k}_3}{6}\right) \tag{122}$$

и четвертой степени:

$$\mathbf{k}_1 = h\mathbf{f}(t_n, \mathbf{x}_n), \qquad \mathbf{k}_2 = h\mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{\mathbf{k}_1}{2}\right)$$

$$\mathbf{k}_3 = h\mathbf{f}\left(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{\mathbf{k}_2}{2}\right), \qquad \mathbf{k}_4 = h\mathbf{f}\left(t_n + h, \mathbf{x}_n + \mathbf{k}_3\right)$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \left(\frac{\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4}{6}\right) \tag{123}$$

Как уже отмечалось, если функция $\mathbf{f}(t, \mathbf{x})$ не зависит от $\mathbf{x}$, то все методы интегрирования дифференциальных уравнений превращаются в соответствующие им квадратурные формулы.

С увеличением степени метода резко возрастает число параметров p_r, α_r, β_{ri}, а также число нелинейных уравнений для их определения. Оказывается, что метод Рунге-Кутты четвертой степени является последним методом, у которого количество вычислений $\mathbf{f}(t, \mathbf{x})$ на одном шаге совпадает со степенью метода. Уже метод Рунге-Кутты пятой степени требует вычислять функцию $\mathbf{f}(t, \mathbf{x})$ шесть раз, шестой степени – 7 раз, седьмой степени – 9, восьмой – 11. С дальнейшим ростом степени методов трудности их построения растут по экспоненте.

Теперь обратимся к такому важному моменту, как контроль погрешности метода в процессе интегрирования. Представляется весьма желательным использование переменного шага интегрирования подобно тому, как это делается в программах, реализующих адаптивные квадратурные формулы (например, программа `QUANC8`). Хотелось бы выбирать маленький шаг там, где решение меняется быстро, и большой, где оно меняется относительно медленно. Оценивать погрешность по отбрасываемому члену разложения чрезвычайно неудобно. Поэтому на практике используются различные другие подходы для контроля локальной погрешности методов. Один их них состоит в сравнении на каждом шаге интегрирования решений, получаемых по формулам методов различных степеней. Этот подход реализован в программе `RKF45`, построенной на методах Рунге-Кутты-Фельберга четвертой и пятой степени. Фельбергу удалось так подобрать параметры методов, что одни и те же шесть вычислений $\mathbf{k}_r$ функции $\mathbf{f}(t, \mathbf{x})$ с различными весами p_r используются для получения решения методами и четвертой, и пятой степени

$$\mathbf{x}_{n+1}^{(4)} = \mathbf{x}_n + \sum_{r=1}^{6} p_r \mathbf{k}_r, \quad \mathbf{x}_{n+1}^{(5)} = \mathbf{x}_n + \sum_{r=1}^{6} p_r^* \mathbf{k}_r, \quad \mathbf{x}_{n+1}^{(5)} - \mathbf{x}_{n+1}^{(4)} = \sum_{r=1}^{6} (p_r^* - p_r)\, \mathbf{k}_r$$

Тогда разность между этими решениями может использоваться для контроля величины шага дискретности.

26.1 Подпрограмма RKF45.

```
RKF45(F, N, X, T, TOUT, RE, AE, IFLAG, WORK, IWORK)
```

F – имя процедуры, написанной пользователем для вычисления правых частей системы (102). Эта программа, в свою очередь, должна иметь следующие параметры: `F(T, X, DX)` (**X** – вектор решения в точке **T**, а **DX** – вектор производных)
N – количество интегрируемых уравнений;
X – вектор решения размерностью **N** в точке **T** на входе в программу и в точке **TOUT** при выходе из нее;
T – начальное значение независимой переменной на входе в программу (при нормальном выходе это **TOUT**);
TOUT – точка выхода по независимой переменной;
RE, AE – границы относительной и абсолютной погрешностей;
WORK – рабочий вещественный массив размерности $6\mathbf{N} + 3$;
IWORK – рабочий целый массив размерности не менее 5;
IFLAG – указатель режима интегрирования. Обычно при первом обращении на входе **IFLAG** = 1, а при последующих обращениях на входе **IFLAG** = 2. Нормальное выходное значение **IFLAG** = 2. Другие выходные значения указывают на возникшие отклонения от нормального процесса:

= 3 – заданное значение **RE** оказалось слишком малым и требуется его увеличить;
= 4 – потребовалось более 3000 вычислений $\mathbf{f}(t, \mathbf{x})$ (это отвечает приблизительно 500 шагам). Можно, не изменяя **IFLAG**, снова обратиться к программе или, если система является жесткой, применить специальные алгоритмы решения жестких систем.
= 5 – решение обратилось в нуль, а **AE** равно нулю. Требуется задать ненулевое значение **AE**.

= 6 – требуемая точность не достигнута даже при наименьшей допустимой величине шага и требуется увеличить **AE** и **RE**.

= 7 – слишком большое число требуемых выходных точек препятствует выбору естественной величины шага (он может быть значительно увеличен при заданной точности). Нужно или увеличить **TOUT-T** или задать значение **IFLAG** = 2 и продолжить работу программы.

= 8 – неправильное задание параметров процедуры (например, $\mathbf{N} < 0$)

Вопрос 27. Глобальная погрешность. Устойчивость метода. Ограничение на шаг. Явление жесткости и методы решения жестких систем.

Подменяя анализ глобальной погрешности анализом локальной, мы высказывали непременное условие такой замены – обеспечение устойчивости решения разностного уравнения метода, которая, в свою очередь, зависит не только от формулы метода, но и от шага интегрирования. В отсутствие возможности оценить глобальную погрешность и устойчивость метода в общем случае появляется желание выбрать некоторую простую модельную («тестовую») систему уравнений и рассмотреть, как формируется глобальная погрешность для нее.

Такой тестовый пример должен удовлетворять двум требованиям. С одной стороны, он должен быть достаточно простым, чтобы можно было выполнить необходимый анализ, а с другой стороны, он должен быть достаточно «представительным» в том смысле, что сравнительные выводы о свойствах устойчивости различных методов должны носить относительно общий характер и распространяться на значительный круг задач. Этим требованиям удовлетворяет система линейных уравнений

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) \tag{124}$$

с постоянной матрицей. Она является достаточно представительной, так как нелинейная система может быть линеаризована в окрестности некоторой точки решения и заменена системой (124). Если какой-либо метод продемонстрирует негативные свойства в смысле устойчивости на примере системы (124), трудно ожидать от него хороших свойств на нелинейной задаче. Обратное верно не всегда.

Пусть все собственные значения λ_k матрицы $\mathbf{A}$ лежат в левой полуплоскости. Тогда решение системы дифференциальных уравнений (124) будет асимптотически устойчиво. Чтобы численный метод адекватно отражал реальность, необходимо потребовать, чтобы его разностное уравнение также обладало асимптотически устойчивым решением.

Первоначально обратимся к явному методу ломаных Эйлера. Его формула применительно к (124) записывается в виде

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}(t_n, \mathbf{x}_n) = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_n = (\mathbf{E} + h\mathbf{A})\,\mathbf{x}_n$$

Для асимптотической устойчивости решения разностного уравнения необходимо, чтобы собственные значения матрицы $\mathbf{E} + h\mathbf{A}$, равные $1 + h\lambda_k$, по модулю были меньше единицы, что для вещественных отрицательных λ_k приводит к выполнению неравенства

$$-1 < 1 + h\lambda_k < 1, \qquad h|\lambda_k| < 2 \tag{125}$$

Для комплексных значений $h\lambda_k = h\alpha_k + jh\omega_k$, условие асимптотической устойчивости

$$(1 + h\alpha_k)^2 + h^2\omega_k^2 < 1$$

требует, чтобы их значения находились на комплексной плоскости внутри круга с единичным радиусом и центром $(-1, 0)$, а ограничение на шаг имело вид:

$$h < \frac{-2\alpha_k}{\left(\alpha_k^2 + \omega_k^2\right)}$$

Definition 25

Множество решений $h\lambda_k$, удовлетворяющих условию устойчивости разностного уравнения метода, называют *областью устойчивости* данного метода.

Для явного метода ломаных Эйлера, который является одновременно методом Адамса первой степени и методом Рунге-Кутты первой степени, она представлена на рисунке кривой $s = 1$.

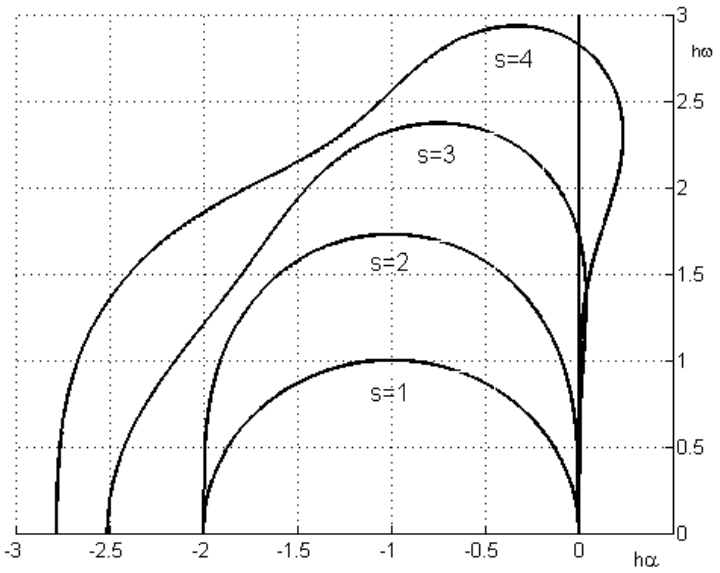


Рис. 27..1: Области устойчивости явных методов Рунге-Кутты

Является ли условие (125) недостатком только явного метода ломаных Эйлера, и как ведут себя другие численные методы для жестких систем? К сожалению, все рассмотренные явные методы Рунге-Кутты и Адамса непригодны для решения жестких систем. Так, последовательно применяя методы Рунге-Кутты второй, третьей и четвертой степени к системе (124), получаем следующие разностные уравнения:

$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2}\right)\mathbf{x}_n$$
$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2} + \frac{h^3\mathbf{A}^3}{6}\right)\mathbf{x}_n$$
$$\mathbf{x}_{n+1} = \left(\mathbf{E} + h\mathbf{A} + \frac{h^2\mathbf{A}^2}{2} + \frac{h^3\mathbf{A}^3}{6} + \frac{h^4\mathbf{A}^4}{24}\right)\mathbf{x}_n$$

и ограничения на шаг интегрирования, подобно (125) задающие области устойчиво-

сти:

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2}\right| < 1 \tag{126}$$

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6}\right| < 1 \tag{127}$$

$$\left|1 + h\lambda + \frac{h^2\lambda^2}{2} + \frac{h^3\lambda^3}{6} + \frac{h^4\lambda^4}{24}\right| < 1 \tag{128}$$

Для вещественных отрицательных λ_k эти ограничения принимают вид

$h|\lambda_k| < 2$– методы второй степени;
$h|\lambda_k| < 2.513$– метод третьей степени;
$h|\lambda_k| < 2.785$– метод Рунге-Кутты четвертой степени.

Для общего случая комплексных значений λ_k области устойчивости представлены на рисунке выше. Так как все области обладают свойством симметрии относительно действительной оси, то воспроизводится только часть границы областей, лежащая в верхей полуплоскости. Значения $h\lambda_k$ внутри этих областей удовлетворяют условиям ограниченности единицей. Хотя ограничения на шаг интегрирования незначительно ослабляются при увеличении степени метода, общий объем вычислений при этом даже возрастает, так как растет число значений $\mathbf{f}(t, \mathbf{x})$, требуемых на каждом шаге. Еще хуже обстоят дела с устойчивостью явных методов Адамса. Их разностные схемы приводят к еще более серьезным ограничениям на h по сравнению с методами Рунге-Кутты:

$h|\lambda_k| < 1.0$– метод Адамса второй степени;
$h|\lambda_k| < 6/11$– метод Адамса третьей степени;
$h|\lambda_k| < 0.3$– метод Адамса четвертой степени;

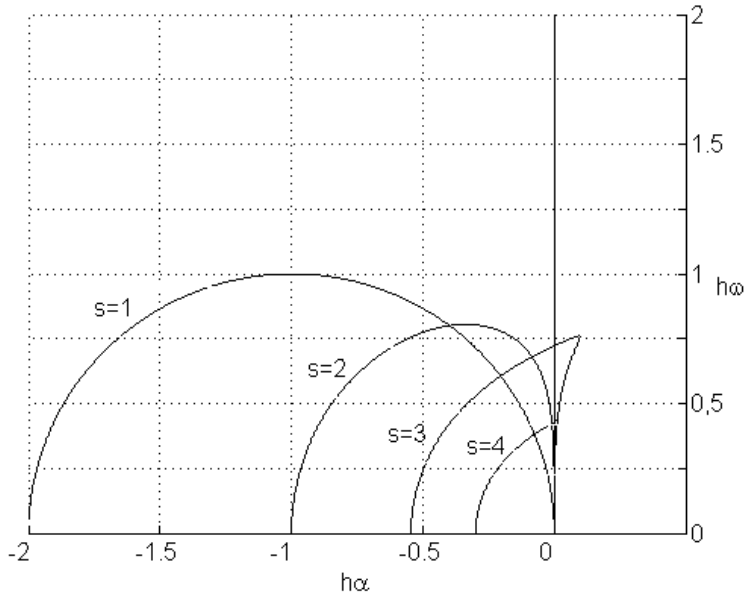


Рис. 27..2: Области устойчивости явных методов Адамса

Общая ситуация для вещественных отрицательных значений λ_k складывается следующим образом. Время наблюдения решения T определяется минимальным по модулю собственным значением матрицы $\mathbf{A}$, а шаг интегрирования – максимальным. Тогда число шагов N прямо пропорционально числу обусловленности, что и приводит к недопустимым затратам

$$T \sim \frac{1}{|\lambda_k|_{\min}}, \qquad h \sim \frac{1}{|\lambda_k|_{\max}}, \qquad N = \frac{T}{h} \sim \frac{|\lambda_k|_{\max}}{|\lambda_k|_{\min}} \gg 1$$

Чтобы изменить ситуацию, для методов, предназначенных интегрировать жесткие системы, следует потребовать, чтобы их область устойчивости включала в себя всю или почти всю левую полуплоскость, что позволит устранить ограничение на шаг типа (125) и увеличить шаг, когда пограничный слой уже пройден.
Используем для решения (124) неявный метод ломаных Эйлера. Соответствующее разностное уравнение и ограничение на шаг примут следующий вид:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{A}\mathbf{x}_{n+1} = (\mathbf{E} - h\mathbf{A})^{-1}\mathbf{x}_n, \qquad |1 - h\lambda_k| > 1$$

Полагая величину λ_k комплексной, $\lambda_k = \alpha_k + j\omega_k$, для области устойчивости метода (рис. 3) получим:

$$(1 - h\alpha_k)^2 + h^2\omega_k^2 > 1$$

Она включает в себя всю левую полуплоскость, и неустойчивость метода проявляется только в круге единичного радиуса с центром в точке $(1, 0)$.

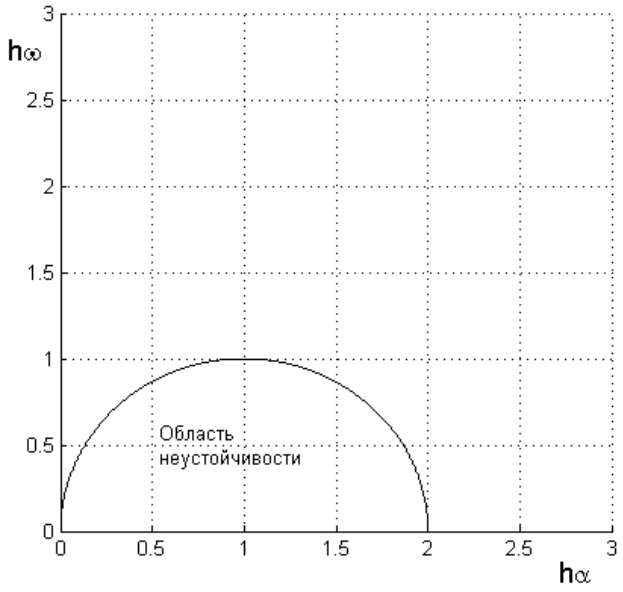


Рис. 27..3: Области устойчивости неявного метода ломаных Эйлера

Еще более интересный результат наблюдаем для неявного метода трапеций

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\left(\mathbf{A}\mathbf{x}_{n+1} + \mathbf{A}\mathbf{x}_n\right), \qquad \left(\mathbf{E} - \frac{h}{2}\mathbf{A}\right)\mathbf{x}_{n+1} = \left(\mathbf{E} + \frac{h}{2}\mathbf{A}\right)\mathbf{x}_n$$

$$\mathbf{x}_{n+1} = \left(\mathbf{E} - \frac{h}{2}\mathbf{A}\right)^{-1}\left(\mathbf{E} + \frac{h}{2}\mathbf{A}\right)\mathbf{x}_n, \qquad \left|\frac{1 + h\lambda_k/2}{1 - h\lambda_k/2}\right| < 1$$

Тогда для $\lambda_k = \alpha_k + j\omega_k$ получаем

$$\left(1 + \frac{h\alpha}{2}\right)^2 + \frac{h^2\omega^2}{4} < \left(1 - \frac{h\alpha}{2}\right)^2 + \frac{h^2\omega^2}{4}$$

что после упрощений приводит к выполнению неравенства $h\alpha < 0$, т.е. область устойчивости метода совпадает с областью, где устойчивость имеет место для решения дифференциального уравнения. Таким образом, оба алгоритма могут быть рекомендованы для решения жестких систем. Как будет видно ниже, типичное для неявных методов некоторое увеличение трудоемкости одного шага интегрирования с лихвой окупается большим выигрышем в величине шага для жестких систем.

Вопрос 28. Метод Ньютона в неявных алгоритмах решения дифференциальных уравнений.

Рассмотрим, как решается проблема неявного задания $\mathbf{x}_{n+1}$, например, в неявном методе ломаных Эйлера. Решение этого уравнения относительно $\mathbf{x}_{n+1}$ может быть сведено к решению следующей системы

$$\mathbf{F}(\mathbf{z}) = \mathbf{z} - \mathbf{x}_n - h\mathbf{f}(t_{n+1}, \mathbf{z}) = 0 \tag{129}$$

методом Ньютона

$$\frac{\partial \mathbf{F}}{\partial \mathbf{z}}\left(\mathbf{z}^{(k)}\right)\left(\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\right) = -\mathbf{F}\left(\mathbf{z}^{(k)}\right); \qquad \frac{\partial \mathbf{F}}{\partial \mathbf{z}} = \mathbf{E} - h\frac{\partial \mathbf{f}}{\partial \mathbf{z}}$$

где $\mathbf{z}^{(k)}$ – k-е приближение к значению $\mathbf{x}_{n+1}$. Здесь весьма эффективен модифицированный метод Ньютона, когда матрица $\dfrac{\partial \mathbf{F}}{\partial \mathbf{z}}$ вычисляется в точке $\mathbf{x}_0$, раскладывается в произведение треугольных матриц программой `DECOMP`, и на последующих итерациях используется только программа `SOLVE`. Матрица вновь вычисляется только тогда, когда метод Ньютона перестает сходиться за три итерации. Даже если матрица Якоби $\dfrac{\partial \mathbf{f}}{\partial \mathbf{z}}$ исходной системы уравнений плохо обусловлена, обращение матрицы $\dfrac{\partial \mathbf{F}}{\partial \mathbf{z}}$, как правило, не вызывает затруднений, так как она значительно лучше обусловлена, чем $\dfrac{\partial \mathbf{f}}{\partial \mathbf{z}}$.
В итоге, применение метода Ньютона в неявных алгоритмах может быть описано по следующей схеме.

1. В некоторой точке вычисляем матрицу Якоби по аналитическим формулам для ее элементов или с помощью формул численного дифференцирования, а затем производим ее разложение с помощью программы `DECOMP`.
2. По начальному условию $\mathbf{z}^{(0)}$, рассчитанному с помощью явного метода Эйлера, выполняем итерации метода Ньютона для получения $\mathbf{x}_{n+1}$.
3. После одной-двух итераций по методу Ньютона при достижении сходимости переходим к шагу 2. Возвращение к шагу 1 проводится только в том случае, если метод Ньютона перестает сходиться за три итерации.

Такая организация вычислений приводит к малому объему работы на одном шаге, а сам метод Ньютона с хорошим начальным приближением сходится за одну-две итерации.
По аналогичной схеме для решения жестких систем используются и другие неявные методы. Методы с областью устойчивости, пригодной для решения жестких систем, почти всегда являются неявными, хотя, разумеется, далеко не все неявные методы такую область имеют.

Вопрос 29. Сведение дифференциального уравнения высокого порядка к системе уравнений первого порядка. Метод стрельбы для решения краевых задач.

29.1 Сведение дифференциального уравнения высокого порядка к системе уравнений первого порядка.

Эту процедуру рассмотрим на примере линейного дифференциального уравнения четвертого порядка с постоянными коэффициентами

$$\frac{d^4z(t)}{dt^4} + \alpha_1\frac{d^3z(t)}{dt^3} + \alpha_2\frac{d^2z(t)}{dz^2} + \alpha_3\frac{dz(t)}{dt} + \alpha_4 z(t) = \varphi(t) \tag{130}$$

Характеристическое уравнение для него имеет вид

$$\lambda^4 + \alpha_1\lambda^3 + \alpha_2\lambda^2 + \alpha_3\lambda + \alpha_4 = 0 \tag{131}$$

Если среди корней уравнения (131) нет кратных, то решение неоднородного уравнения (130) определяется линейной комбинацией экспонент, показателями которых являются корни уравнения (131), и частным решением уравнения (130), определяемым видом функции $\varphi(t)$.
Введем вектор $\mathbf{x}$ четвертого порядка со следующими компонентами

$$\mathbf{x} = \begin{pmatrix} x^{(1)} & x^{(2)} & x^{(3)} & x^{(4)} \end{pmatrix}^T = \begin{pmatrix} \frac{d^3z(t)}{dt^3} & \frac{d^2z(t)}{dt^2} & \frac{dz(t)}{dt} & z \end{pmatrix}^T$$

Тогда вместо уравнения (130) можно записать систему уравнений

$$\begin{aligned} \frac{dx^{(1)}}{dt} &= -\alpha_1 x^{(1)} - \alpha_2 x^{(2)} - \alpha_3 x^{(3)} - \alpha_4 x^{(4)} + \varphi(t) \\ \frac{dx^{(2)}}{dt} &= x^{(1)} \\ \frac{dx^{(3)}}{dt} &= x^{(2)} \\ \frac{dx^{(4)}}{dt} &= x^{(3)} \end{aligned}$$

Она же в векторно-матричной форме имеет вид

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\cdot\mathbf{x} + \mathbf{f}(t) \tag{132}$$

где

$$A = \begin{pmatrix} -\alpha_1 & -\alpha_2 & -\alpha_3 & -\alpha_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \qquad \mathbf{f}(t) = \begin{pmatrix} \varphi(t) & 0 & 0 & 0 \end{pmatrix}^T$$

Матрица $\mathbf{A}$ называется *матрицей Фробениуса*. При этом, решение системы (132) описывается линейной комбинацией экспонент, показателями которых являются собственные значения матрицы $\mathbf{A}$, и частным решением уравнения (132), определяемым видом функции $\mathbf{f}(t)$.
Таким образом, вместо поиска корней полинома (131) можно искать собственные значения матрицы Фробениуса, что на практике часто так и делается.
В случае, когда уравнение высокого порядка является нелинейным,

$$\frac{d^4 z(t)}{dt^4} = F\left(t, \frac{d^3 z(t)}{dt^3}, \frac{d^2 z(t)}{dt^2}, \frac{dz(t)}{dt}, z(t)\right)$$

описанная выше замена переменных сохраняется:

$$\frac{dx^{(1)}}{dt} = F\left(t, x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}\right),$$
$$\dots$$

Для решения этой системы можно использовать любую программу, реализующую ранее рассмотренные методы, в частности, программу `RKF45`.

29.2 Метод стрельбы для решения краевых задач.

В рассмотренной ранее задаче Коши одно из решений системы выделяется заданием начальных условий $(t_0, \mathbf{x}_0)$. Однако это не единственный способ. Задавая условия при двух или более значениях независимой переменной, приходим к краевой задаче. В общем случае краевые (граничные) условия выглядят следующим образом:

$$\Phi_i\left(x^{(1)}(t_k), \dots, x^{(m)}(t_k)\right) = 0, \qquad a \le t_k \le b, \qquad 1 \le i \le m$$

В зависимости от вида уравнения и краевых условий можно провести классификацию краевых задач, схожую с классификацией краевых задач Коши. Важным подклассом являются линейные краевые задачи, когда и система и краевые условия являются линейными. Эти условия имеют вид

$$\alpha_i x^{(1)}(t_k) + \beta_i x^{(2)}(t_k) + \dots + \omega_i x^{(m)}(t_k) = a_i, \qquad i \le m$$

Линейная краевая задача является однородной, если однородны уравнения и краевые условия. Такая задача всегда имеет тривиальное решение $\mathbf{x}(t) \equiv 0$, и в этом случае представляет интерес отыскание нетривиальных решений.
В свою очередь, из краевых задач выделяют двухточечные, когда условия задаются на левом и правом концах промежутка, т.е. при $t_k = a$ и $t_k = b$. Например, дифференциальное уравнение второго порядка

$$-\frac{d}{dt}\left(p(t)\frac{dx(t)}{dt}\right) + q(t)x(t) = f(t)$$
$$t \in [0,1], \qquad p(t) \ge p_0 > 0, \quad q(t) > 0$$

где краевые условия $x(0) = x(1) = 0$ определяют задачу, которая является моделью многих физических процессов: распределение тепла в неоднородном стержне, распределение концентрации вещества в процессах диффузии и др.
Несмотря на разнообразие форм краевых условий, краевые задачи в основном решаются одними и теми же численными методами. Выделяют два основных подхода:

1. Сведение к многократному решению задачи Коши;
2. Сведение к решению алгебраических систем.

Второй подход включает в себя как конечно-разностные, так и проекционные методы. К последним относятся, в свою очередь, давно применяющиеся методы коллокаций, Галеркина, Ритца, а также метод конечных элементов.

Многократное решение задачи Коши демонстрирует метод стрельбы, имеющий аналогию со стрельбой, когда, зафиксировав недолет или перелет, угол стрельбы изменяют так, чтобы следующий выстрел был ближе к цели. Рассмотрим систему из двух уравнений

$$\frac{du(t)}{dt} = f_1(t,u,v) \qquad \frac{dv(t)}{dt} = f_2(t,u,v), \qquad t \in [a,b]$$

с граничными условиями

$$v(a) = v_a, \quad u(b) = u_b$$

Выберем произвольное значение $u_a = u(a)$. С начальными условиями u_a и v_a проинтегрируем систему каким-либо методом. Результатом будут функции $u(t,u_a)$ и $v(t,u_a)$, зависящие от u_a как от параметра. При подстановке $u(b,u_a)$ в правое краевое условие получаем функцию относительно u_a. Задача свелась к нахождению решения уравнения $L(u_a) = 0$, где $L(u_a) = u(b,u_a) - u_b$.
Конкретный вид функции $L(u_a)$ неизвестен, но значения ее для любых значений u_0 легко вычисляются, и это дает возможность воспользоваться любым методом нахождения корней нелинейного уравнения. Так, в частности, можно воспользоваться уже известной процедурой-функцией `ZEROIN(A, B, F, EPS)`.

Основная программа вызывает `ZEROIN`. Предварительно подбираются два значения $u_a = A$ и $u_a = B$, для которых функция $L(u_a)$ имеет различный знак. Программа `ZEROIN` вызывает функцию $L(u_a)$, которую программирует пользователь. Эта функция, в свою очередь, обращается к программе `RKF45`, которая вызывает процедуру, вычисляющую $\mathbf{f}(t,\mathbf{x})$. Значение функции $L(u_a)$ для заданного u_a определяется выражением $L(u_a) = u(b,u_a) - u_b$.
Описанная ситуация резко упрощается, если задача является линейной двухточечной краевой

$$\frac{du}{dt} = p_1(t)u(t) + q_1(t)v(t) + s_1(t)$$
$$\frac{dv}{dt} = p_2(t)u(t) + q_2(t)v(t) + s_2(t)$$

с граничными условиями

$$v(a) = v_a, \quad u(b) = u_b$$

В силу линейности задачи решение будет зависеть от $u(a)$ линейно, и функция $L(u_a)$ также будет линейной. Отсюда следует, что для вычисления u_a^* – левого начального условия для функции $u(t)$, дающего решение краевой задачи, достаточно дважды проинтегрировать систему до $t = b$ с двумя различными начальными условиями (u_a^1, v_a^1) и (u_a^2, v_a^2), найти $L\left(u_a^1\right)$ и $L\left(u_a^2\right)$ и линейной интерполяцией определить u_a^*. Таким образом, u_a^* будет корнем уравнения

$$\frac{u_a - u_a^2}{u_a^1 - u_a^2} L\left(u_a^1\right) + \frac{u_a - u_a^1}{u_a^2 - u_a^1} L\left(u_a^2\right) = 0.$$