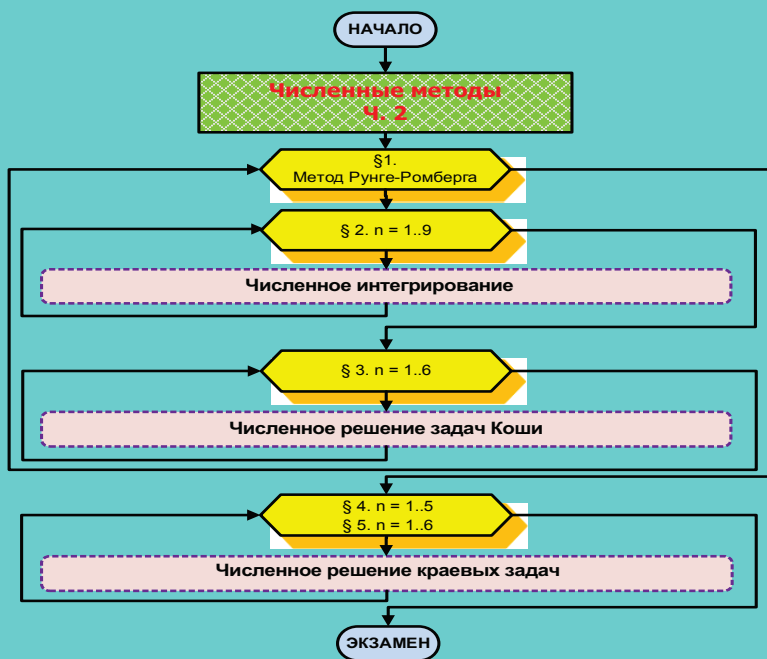


А. С. БАННИКОВ, И. Г. КИМ, Н. В. ЛАТЫПОВА

ЧИСЛЕННЫЕ МЕТОДЫ

Учебное пособие

Часть 2



Ижевск
2020

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Удмуртский государственный университет»
Институт математики, информационных технологий и физики

А. С. Банников, И. Г. Ким, Н. В. Латыпова

ЧИСЛЕННЫЕ МЕТОДЫ

Учебное пособие

Часть 2



Ижевск
2020

УДК 519.6(075)

ББК 22.19я7

Б 232

Рекомендовано к изданию Учебно-методическим советом УдГУ

Рецензент: к. ф.-м. н., с. н. с. ФГБУН Удмуртский федеральный исследовательский центр УрО РАН ФТИ ОФХН Немцова О. М.

Банников А. С., Ким И. Г., Латыпова Н. В.

Б 232 Численные методы: учеб. пособие. Ч. 2. — 2-е изд., испр. и доп. — Ижевск: Издательский центр «Удмуртский университет», 2020. — 88 с.

ISBN 978-5-4312-0642-9

ISBN 978-5-4312-0840-9 (Часть 2)

В данном пособии излагаются основные понятия, формулы и алгоритмы курса «Численные методы». Во второй части пособия рассматриваются задачи численного интегрирования. Излагаются основные методы численного решения дифференциальных уравнений. Даны темы для домашнего задания и варианты работ лабораторного практикума с рекомендациями по их выполнению в пакете *Mathematica*.

Пособие предназначено для студентов всех направлений института математики, информационных технологий и физики.

УДК 519.6(075)

ББК 22.19.1я7

ISBN 978-5-4312-0642-9

ISBN 978-5-4312-0840-9 (Часть 2)

©А. С. Банников, И. Г. Ким,

Н. В. Латыпова, 2020

©ФГБОУ ВО «Удмуртский государственный университет», 2020

Оглавление

Предисловие	5
1. Метод Рунге–Ромберга	6
2. Численное интегрирование	7
2.1. Формулы прямоугольников	8
2.2. Формула трапеций	11
2.3. Формула Симпсона	12
2.4. Вычисления с заданной точностью	13
2.5. Формулы интерполяционного типа	15
2.6. Сплаины в задаче численного интегрирования	16
2.7. Квадратурные формулы Гаусса	17
2.8. Некоторые весовые функции и соответствующие им ортогональные многочлены	19
2.9. Несобственные интегралы	21
3. Численное решение задачи Коши	22
3.1. Метод последовательных приближений	23
3.2. Метод Эйлера, его модификации	23
3.3. Семейство схем второго порядка точности	25
3.4. Семейство методов Рунге–Кутты	25
3.5. Метод Рунге–Ромберга	27
3.6. Метод Адамса	27
4. Численное решение краевых задач с одной независимой переменной	28
4.1. Метод стрельбы для линейных краевых задач	28
4.2. Метод стрельбы для нелинейных краевых задач	30
4.3. Метод коллокации	31
4.4. Метод Галеркина	33
4.5. Разностный метод	34
5. Численное решение уравнений в частных производных	36
5.1. Основные понятия теории разностных схем	36
5.2. Основные теоремы	38
5.3. Разностный метод для уравнений эллиптического типа	39
5.4. Разностный метод для уравнений параболического типа	41

5.5. Разностный метод для уравнений гиперболического типа .	44
5.6. Метод конечных элементов (МКЭ)	46
6. Лабораторные работы	52
6.1. Численное интегрирование	52
6.2. Численное решение задачи Коши	53
6.3. Численное решение краевых задач	55
6.4. Примерные темы домашних заданий	63
Список рекомендуемой литературы	64
Рекомендации по выполнению лабораторных работ	66
Приложение	87

Предисловие

Курс «Численные методы» традиционно считается сложным у студентов, так как в нем требуется сочетать умения и навыки программирования со знаниями базовых математических курсов. Поэтому данное учебное пособие призвано помочь студентам в освоении данной дисциплины. Основные цели и задачи курса: ознакомить с методами получения аналитических зависимостей при обработке результатов экспериментов; научить правильно оценивать результаты вычислений, выполненных на компьютере средствами языков программирования; ознакомить с признаками плохо обусловленных и некорректных задач и подходами к их решению; сформировать культуру критичного отношения к полученным результатам.

Предлагаемое пособие является вторым изданием к данному курсу, исправленным и дополненным.

Пособие снабжено теоретической частью, в которой кратко излагаются формулы и алгоритмы классических методов. При оформлении справочной части были использованы материалы [5, 6, 12]. Для самостоятельной работы студентов предложен лабораторный практикум с индивидуальными заданиями и тематика домашних работ. Лабораторные работы курса направлены на углубление и расширение лекционного материала, приобретение вычислительного опыта при решении поставленных задач. Домашнее задание предполагает составление математической модели, разработку метода решения, написание программы на языке программирования и проведение вычислительного эксперимента.

Дополнениями первого издания являются методические рекомендации по выполнению лабораторных работ в пакете Wolfram Mathematica, а также теоретического материала по методу конечных элементов. Были исправлены некоторые неточности и опечатки.

Издание предназначено для студентов бакалавриата направлений по укрупненным группам: 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки». Кроме того, пособие может быть использовано и студентами инженерных направлений, а также будет полезно всем, кто интересуется вычислительной математикой.

1. Метод Рунге–Ромберга

Пусть решается задача определения Y и есть алгоритм нахождения приближенного значения в зависимости от параметра h такой, что $\|Y - Y_h\| = O(\|h\|^m)$, т. е. остаточный член приближенной формулы Y_h имеет вид

$$Y - Y_h = R \cdot h^m + O(h^{m+1}). \quad (1)$$

Выполним теперь вычисления по той же приближенной формуле, но с другим шагом (параметром сетки) rh :

$$Y - Y_{rh} = R \cdot (rh)^m + O(h^{m+1}), \quad (2)$$

где r — константа. Вычтем почленно из (1) формулу (2) и преобразуем, выразив $R \cdot h^m$:

$$R \cdot h^m = \frac{Y_{rh} - Y_h}{r^m - 1} + O(h^{m+1}). \quad (3)$$

Формулу (3) называют *первой формулой Рунге*. Первое слагаемое справа есть главный член погрешности, который позволяет оценить погрешность уже найденного значения Y_h . Подставим его в (1) и выразим Y :

$$Y = Y_h + \frac{Y_{rh} - Y_h}{r^m - 1} + O(h^{m+1}). \quad (4)$$

Получили *вторую формулу Рунге* для нахождения Y с более высокой точностью.

В частном случае, при $r = \frac{1}{2}$ формула принимает вид

$$Y = Y_{\frac{h}{2}} + \frac{Y_{\frac{h}{2}} - Y_h}{2^m - 1} + O(h^{m+1}). \quad (5)$$

Если мы хотим получить значение Y с заданной точностью ε , то вычисляем значения с шагами h и $\frac{h}{2}$ и проверяем выполнение неравенства

$$|\delta| < \varepsilon, \text{ где } \delta = \frac{Y_{\frac{h}{2}} - Y_h}{2^m - 1}.$$

Если неравенство выполняется, то счет можно прекращать, взяв в качестве приближенного значения, найденное с заданной точностью $Y \approx Y_{\frac{h}{2}} + \delta$. Если же неравенство не выполняется, т. е. $\delta \geq \varepsilon$, то снова

уменьшаем шаг в два раза, вычисляем новое значение и сравниваем два последних приближения и т. д.

Отсюда видно, что для получения высокой точности не обязательно производить вычисления по формулам высокого порядка. Достаточно произвести вычисления по простым формулам низкой точности на разных сетках и уточнить результат методом Рунге–Ромберга.

Замечание. При реализации метода, кроме ошибок связанных с величиной h , на результат оказывает влияние и округление.

2. Численное интегрирование

Основная задача численного интегрирования — получить простые, однотипные формулы или алгоритмы, которые легко реализуются в виде программ на ЭВМ и пригодны для вычисления определенных интегралов для достаточно широкого класса функций. *Общий способ построения* формул численного интегрирования состоит в том, что подынтегральная функция приближенно заменяется на другую, более простую, от которой интеграл легко «берется».

Будем изучать формулы, основанные на замене интеграла конечной суммой,

$$\int_a^b f(x) dx \approx \sum_{k=0}^n C_k f(x_k) \doteq I_n. \quad (6)$$

Приближенное равенство (6) называется *квадратурной формулой*, точки x_k — *узлами* квадратурной формулы, числа C_k — *коэффициентами*, а разность

$$R = \int_a^b f(x) dx - \sum_{k=0}^n C_k f(x_k)$$

называется *погрешностью квадратурной формулы*.

Разобьем отрезок $[a, b]$ на узлы, не обязательно равномерно,

$$a = x_0 < x_1 < \dots < x_N = b,$$

и представим интеграл в виде суммы интегралов:

$$\int_a^b f(x) dx = \sum_{i=1}^N \int_{x_{i-1}}^{x_i} f(x) dx. \quad (7)$$

Ясно, что нам достаточно построить квадратурную формулу для одного конкретного интервала $\int_{x_{i-1}}^{x_i} f(x) dx$ и воспользоваться (7).

2.1. Формулы прямоугольников

Самый простой способ получения приближенных формул численного интегрирования — это методы прямоугольников.

1. Если криволинейную площадь под графиком подынтегральной функции на участке $[x_{i-1}, x_i]$ заменить площадью прямоугольника высотой, равной значению функции в левом конце отрезка $f(x_{i-1})$, то получим *формулу левых прямоугольников*

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx (x_i - x_{i-1}) \cdot f(x_{i-1}). \quad (8)$$

2. Если криволинейную площадь под графиком подынтегральной функции на участке $[x_{i-1}, x_i]$ заменить площадью прямоугольника высотой, равной значению функции в правом конце $f(x_i)$, то получим *формулу правых прямоугольников*

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx (x_i - x_{i-1}) \cdot f(x_i). \quad (9)$$

3. Если площадь криволинейной трапеции на участке $[x_{i-1}, x_i]$ заменить площадью прямоугольника высотой, равной значению функции в середине отрезка $f\left(x_{i-\frac{1}{2}}\right)$, где $x_{i-\frac{1}{2}} = \frac{x_i + x_{i-1}}{2}$, то получим *формулу средних прямоугольников*

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx (x_i - x_{i-1}) \cdot f\left(x_{i-\frac{1}{2}}\right). \quad (10)$$

Подставив формулы (8)–(10) в формулу (7), получим так называемые обобщенные формулы левых прямоугольников

$$\int_a^b f(x) dx \approx \sum_{i=1}^N (x_i - x_{i-1}) \cdot f(x_{i-1}), \quad (11)$$

правых прямоугольников

$$\int_a^b f(x) dx \approx \sum_{i=1}^N (x_i - x_{i-1}) \cdot f(x_i) \quad (12)$$

и обобщенную формулу средних прямоугольников (см. рис. 1)

$$\int_a^b f(x) dx \approx \sum_{i=1}^N (x_i - x_{i-1}) \cdot f\left(x_{i-\frac{1}{2}}\right). \quad (13)$$

В случае, когда узлы расположены равномерно ($x_i - x_{i-1} = h$), формулы примут вид соответственно:

$$\begin{aligned} \int_a^b f(x) dx &\approx h \sum_{i=0}^{N-1} f(x_i), & \int_a^b f(x) dx &\approx h \sum_{i=1}^N f(x_i), \\ \int_a^b f(x) dx &\approx h \sum_{i=1}^N f\left(x_{i-\frac{1}{2}}\right). \end{aligned} \quad (14)$$

Обозначим $M_{1i} = \max_{[x_{i-1}, x_i]} |f'(x)|$, $r_i = \int_{x_{i-1}}^{x_i} f(x) dx - hf(x_{i-1})$. Оценка

погрешности формул левых (и правых) прямоугольников на каждом маленьком отрезке разбиения имеет вид

$$|r_i| \leq M_{1i} \frac{h^2}{2} = O(h^2). \quad (15)$$

Погрешность для обобщенных формул левых и правых прямоугольников равна суммарной погрешности $R = \sum_{i=1}^N r_i$, и справедлива оценка

$$|R| \leq \frac{(b-a)h}{2} M_1 = O(h), \text{ где } M_1 = \max_{[a,b]} |f'(x)|. \quad (16)$$

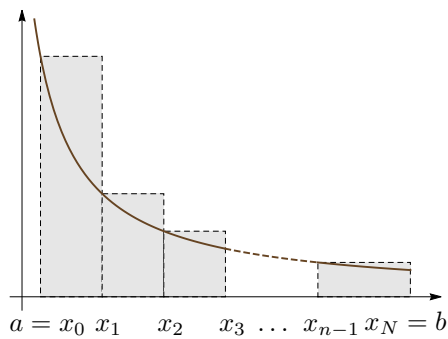


Схема для метода левых прямоугольников

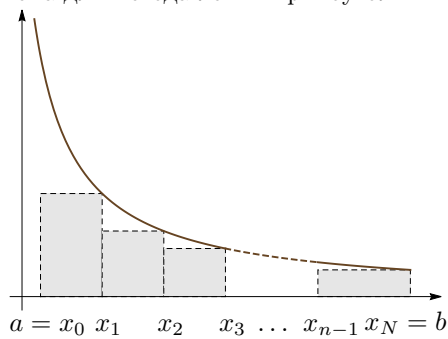


Схема для метода правых прямоугольников

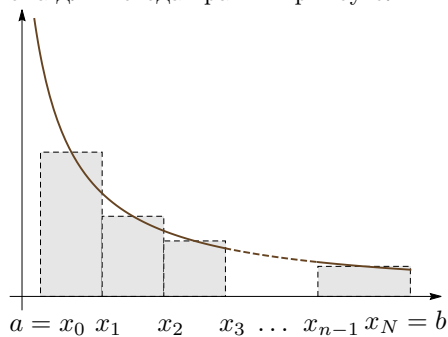


Схема для метода средних прямоугольников

Рис. 1. Иллюстрация методов прямоугольников

Обозначим $M_{2i} = \max_{[x_{i-1}, x_i]} |f''(x)|$, $r_i = \int_{x_{i-1}}^{x_i} f(x) dx - hf(x_{i-\frac{1}{2}})$. Для

формулы средних имеем $|r_i| \leq \frac{h^3}{24} M_{2i}$. Для формулы обобщенных средних получается оценка

$$|R| = \left| \sum_{i=1}^N r_i \right| \leq \frac{b-a}{24} h^2 M_2 = O(h^2), \text{ где } M_2 = \max_{[a,b]} |f''(x)|. \quad (17)$$

Замечание. Формулы (17) для оценки погрешности неуллучшаемы, так как можно указать функцию, для которой такая погрешность достигается, например, $f(x) = \left(x - x_{i-\frac{1}{2}}\right)^2$.

2.2. Формула трапеций

Если определенный интеграл на отрезке $[x_{i-1}, x_i]$ заменить площадью трапеции с длинами оснований $f(x_{i-1})$ и $f(x_i)$, высотой $h_i = x_i - x_{i-1}$, то получится формула трапеций

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx (x_i - x_{i-1}) \cdot \frac{f(x_{i-1}) + f(x_i)}{2}. \quad (18)$$

Обобщенная формула трапеций имеет вид (см. рис. 2)

$$\int_a^b f(x) dx \approx \sum_{i=1}^N (x_i - x_{i-1}) \cdot \frac{f(x_{i-1}) + f(x_i)}{2},$$

которая в случае равномерной сетки определяется формулой (для краткости $f_i \doteq f(x_i)$)

$$\int_a^b f(x) dx \approx \frac{h}{2} (f_0 + 2f_1 + \dots + 2f_{N-1} + f_N).$$

Оценка погрешности формулы трапеций на каждом шаге имеет вид $|r_i| \leq \frac{M_{2i}h^3}{12}$, где $M_{2i} = \max_{[x_{i-1}, x_i]} |f''(x)|$, а обобщенная формула дает

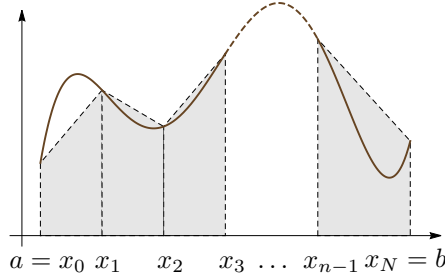


Рис. 2. Схема для метода трапеций

погрешность порядка h^2 :

$$|R| = \left| \sum_{i=1}^N r_i \right| \leq \frac{b-a}{12} h^2 M_2 = O(h^2), \text{ где } M_2 = \max_{[a,b]} |f''(x)|. \quad (19)$$

Замечание. Формула трапеций имеет погрешность в два раза больше, чем погрешность для формулы средних (сравните формулы (17) и (19)).

2.3. Формула Симпсона

Если подынтегральную функцию $f(x)$ заменить квадратичной функцией, то вычислив определенный интеграл от нее на отрезке $[x_{i-1}, x_i]$, получим формулу Симпсона

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx \frac{h}{6} \left(f_{i-1} + 4f_{i-\frac{1}{2}} + f_i \right). \quad (20)$$

Обобщенная формула Симпсона будет иметь вид

$$\int_a^b f(x) dx \approx \frac{h}{6} \left[f_0 + f_N + 2(f_1 + f_2 + \dots + f_{N-1}) + \right. \\ \left. + 4 \left(f_{\frac{1}{2}} + f_{\frac{3}{2}} + \dots + f_{N-\frac{1}{2}} \right) \right].$$

Замечание. Если взять четное число N , то можно формулу Симпсона записать в виде $\int_{x_{i-1}}^{x_{i+1}} f(x) dx \approx \frac{h}{3} (f_{i-1} + 4f_i + f_{i+1})$, а обобщенную формулу Симпсона в виде

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[f_0 + f_N + 2(f_2 + f_4 + \dots + f_{N-2}) + 4(f_1 + f_3 + \dots + f_{N-1}) \right]. \quad (21)$$

Теорема. Формула Симпсона является точной для любого многочлена третьей степени, т. е. имеет место равенство

$$\int_{x_{i-1}}^{x_i} f(x) dx = \frac{h}{6} \left(f_{i-1} + 4f_{i-\frac{1}{2}} + f_i \right), \text{ где } f(x) = a_0 + a_1x + a_2x^2 + a_3x^3.$$

Оценка погрешности формулы Симпсона на отрезке $[x_{i-1}, x_i]$ имеет вид

$$|r_i| \leq \frac{M_{4i} h^5}{2880}, \text{ где } M_{4i} = \max_{[x_{i-1}, x_i]} |f^{(4)}(x)|.$$

Оценка погрешности для обобщенной формулы Симпсона получается, если просуммировать погрешности по всем отрезкам:

$$|R| = \left| \sum_{i=1}^N r_i \right| \leq \frac{b-a}{2880} h^4 M_4 = O(h^4), \text{ где } M_4 = \max_{[a,b]} |f^{(4)}(x)|. \quad (22)$$

Формулу Симпсона можно получить как линейную комбинацию формулы средних прямоугольников и формулы трапеций.

2.4. Вычисления с заданной точностью

Имеется два подхода для вычисления $\int_a^b f(x) dx$ с точностью $\varepsilon > 0$.

- Используя оценки для погрешностей (17), (19), (22), выписываем неравенство для нахождения шага h . Например, для приближенного вычисления интеграла с заданной точностью $\varepsilon > 0$ по формуле Симпсона решаем неравенство

$$|R| \leq \frac{M_4(b-a)h^4}{2880} < \varepsilon \Rightarrow h < \sqrt[4]{\frac{2880\varepsilon}{M_4(b-a)}},$$

и с этим шагом находим приближенное значение интеграла по формуле (21). При этом будет гарантирована точность ε . Но у этого метода есть недостаток: нужно иметь производную четвертого порядка от подынтегральной функции, оценить ее по модулю и получить M_4 . Такой метод называется априори (заранее), ибо мы заранее знаем, что если будем использовать найденный шаг, то получим необходимую точность.

- Пользуясь тем, что имеется формула оценки погрешности, можно применить метод Рунге–Ромберга. Обозначим через $Y = \int_a^b f(x)dx$

точное значение интеграла, а приближенное значение интеграла, полученное по какой-либо квадратурной формуле при шаге h через Y_h . Пусть квадратурная формула имеет погрешность порядка m , т.е. $Y - Y_h = O(h^m)$. Например, для обобщенной формулы Симпсона $m = 4$. Для вычисления с заданной точностью $\varepsilon > 0$ используем двойной пересчет: сначала вычисляем приближенное значение интеграла с каким-то шагом h , получаем Y_h , затем интеграл вычисляется при $\frac{h}{2}$, получаем $Y_{\frac{h}{2}}$, далее оцениваем

$\delta = \frac{Y_{\frac{h}{2}} - Y_h}{2^m - 1}$. Если $|\delta| < \varepsilon$, то счет можно прекращать — правило

Рунге–Ромберга дает приближенное значение $Y \approx Y_{\frac{h}{2}} + \delta$ с заданной точностью. Если же $|\delta| \geq \varepsilon$, то снова уменьшаем шаг в два раза, вычисляем новое значение интеграла и сравниваем два последних приближения и т.д. Таким образом, обеспечивается автоматический подбор шага интегрирования, при котором будет найдено приближенное значение интеграла с любой наперед заданной точностью. Шаг определяется не заранее, а по ходу вычислений, поэтому такая оценка называется апостериорной.

2.5. Формулы интерполяционного типа

На отрезке $[a, b]$ рассмотрим множество (не обязательно равноотстоящих) узлов x_0, x_1, \dots, x_N и заменим подынтегральную функцию интерполяционным многочленом N -ой степени сразу на всем отрезке.

Тогда можно положить $\int_a^b f(x)dx \approx \int_a^b L_N(x)dx$. Полученная квад-

ратурная формула имеет вид $\int_a^b f(x)dx \approx \sum_{i=0}^N C_i f(x_i)$, т.к. $L_N(x) =$

$= \sum_{i=0}^N f(x_i) \frac{\omega(x)}{(x - x_i)\omega'(x_i)}$, где $\omega(x) = \prod_{k=0}^N (x - x_k)$, (см. [1, с. 32]) и

$C_i = \int_a^b \frac{\omega(x)}{(x - x_i)\omega'(x_i)} dx$. Такие формулы называются *квадратурными*

формулами интерполяционного типа. Рассмотренные ранее квадратурные формулы прямоугольников, трапеций, Симпсона на отрезке $[x_{i-1}, x_i]$ являются частным случаем интерполяционной формулы для $N = 0, 1, 2$ соответственно.

Обобщая этот подход, можно найти квадратурную формулу для приближенного вычисления интегралов с весами

$$\int_a^b \rho(x) f(x) dx \approx \sum_{i=0}^N C_i f(x_i), \quad (23)$$

где $\rho(x) > 0$ — некоторая весовая функция, причем

$$C_i = \int_a^b \frac{\rho(x)\omega(x)}{(x - x_i)\omega'(x_i)} dx, \quad i = 0, 1, \dots, N. \quad (24)$$

Формула (23) будет квадратурной формулой интерполяционного типа тогда и только тогда, когда коэффициенты C_i вычисляются по формулам (24).

Погрешность квадратурной формулы (23) определяется как

$$|R| \leq \frac{M_{N+1}}{(N+1)!} \int_a^b \rho(x) |\omega(x)| dx, \quad \text{где } M_{N+1} = \max_{[a,b]} |f^{(N+1)}(x)|.$$

Следовательно, интерполяционная формула (23) точна для всех многочленов до N -ой степени включительно. Верно и обратное утверждение.

Теорема. *Если квадратурная формула (23) точна для всех многочленов до N -ой степени, то это квадратурная формула интерполяционного типа.*

★**Упр.** Докажите сформулированную теорему.

Если в качестве крайних узлов используются a и b , то квадратурная формула называется *формулой замкнутого типа*, если же хотя бы один из концов отрезка $[a, b]$ не является узлом, то формула называется *формулой открытого типа*. Например, формула средних прямоугольников — формула открытого типа, а формула трапеций — замкнутого типа.

2.6. Сплайны в задаче численного интегрирования

Пусть дан интеграл $\int_a^b f(x)dx$. Функцию $f(x)$ заменим интерполяционным сплайном $S(x)$ и в качестве приближенного значения искомого интеграла возьмем $\int_a^b S(x)dx$, т. е. $\int_a^b f(x)dx \approx \int_a^b S(x)dx$. Погрешность вычисления можно оценить следующим образом:

$$\left| \int_a^b f(x)dx - \int_a^b S(x)dx \right| \leq \int_a^b |f(x) - S(x)|dx \leq (b-a) \|f - S\|_{C[a,b]}.$$

Другими словами, для оценки погрешности приближения определенного интеграла достаточно иметь оценку погрешности приближения функции $f(x)$ сплайном $S(x)$.

Ограничимся рассмотрением только кубических сплайнов. Пусть на $[a, b]$ задана сетка $\Delta : a = x_0 < x_1 < \dots < x_n = b$. Кубический сплайн представим в виде (см. [1, с. 37])

$$S_3(x) = M_i \frac{(x_{i+1} - x)^3}{6h_i} + M_{i+1} \frac{(x - x_i)^3}{6h_i} + \left[f(x_i) - \frac{M_i}{6} h_i^2 \right] \frac{x_{i+1} - x}{h_i} + \left[f(x_{i+1}) - \frac{M_{i+1}}{6} h_i^2 \right] \frac{x - x_i}{h_i}, \quad x \in [x_i, x_{i+1}], \quad (25)$$

где $M_i = S_3''(x_i)$, $h_i = x_{i+1} - x_i$ для всех $i = 0, 1, \dots, n-1$.

Подставляя $S_3(x)$ в интеграл и вычисляя его, имеем

$$\int_a^b f(x)dx \approx \frac{1}{2} \sum_{i=0}^{n-1} h_i(f_i + f_{i+1}) - \frac{1}{24} \sum_{i=0}^{n-1} h_i^3(M_i + M_{i+1}). \quad (26)$$

Аналог формулы (26) на равномерной сетке имеет вид

$$\begin{aligned} \int_a^b f(x)dx &\approx \frac{5h}{12}(f_0 + f_n) + \frac{13h}{12}(f_1 + f_{n-1}) + \\ &+ h \sum_{i=2}^{n-2} f_i - \frac{h^3}{72}(2M_0 + M_1 + M_{n-1} + 2M_n). \end{aligned} \quad (27)$$

★Упр. Докажите формулу (27).

Пусть требуется вычислить интегралы от сильно осцилирующих

функций вида $\int_a^b \cos \alpha x f(x)dx$, $\int_a^b \sin \alpha x f(x)dx$. При больших значениях

параметра α здесь невыгодно использовать квадратурные формулы, основанные на замене сплайном всей подынтегральной функции, т. к. это потребует большого числа узлов. Более удобные формулы получаются, когда функции $\cos \alpha x$, $\sin \alpha x$ рассматриваются как весовые, а сплайном приближается только $f(x)$.

2.7. Квадратурные формулы Гаусса

В квадратурной формуле

$$\int_a^b \rho(x)f(x)dx \approx \sum_{i=1}^n C_i f(x_i), \quad (28)$$

где $\rho(x) > 0$ — весовая функция, подберем коэффициенты C_1, C_2, \dots, C_n и узлы x_1, x_2, \dots, x_n так, чтобы она была точна для многочленов наибольшей возможной степени. Такие формулы называются *квадратурными формулами Гаусса*. Наивысшая степень многочлена, для которых формула будет точна, это $m = 2n - 1$.

Теорема 1. Квадратурная формула (28) точна для многочленов степени $m = 2n - 1$ тогда и только тогда, когда

1) многочлен $\omega(x) = \prod_{i=1}^n (x - x_i)$ степени n ортогонален с весом $\rho(x)$ на отрезке $[a, b]$ любому многочлену $q(x)$ степени меньше n , т. е.

$$\int_a^b \rho(x) \omega(x) q(x) dx = 0; \quad (29)$$

2) формула (28) является квадратурной формулой интерполяционного типа, т. е.

$$C_i = \int_a^b \frac{\rho(x) \omega(x)}{(x - x_i) \omega'(x_i)} dx, \quad i = 1, 2, \dots, n. \quad (30)$$

Использование теоремы упрощает построение формул Гаусса. Условия (29) эквивалентны требованиям

$$\int_a^b \rho(x) \omega(x) x^k dx = 0, \quad k = 0, 1, \dots, n - 1, \quad (31)$$

и представляют из себя систему с n неизвестными относительно узлов квадратурной формулы x_1, x_2, \dots, x_n . Решая ее, затем при найденных узлах определяем коэффициенты C_i по формулам (30).

★**Упр.** Получите квадратурные формулы Гаусса при $n = 1, 2$, используя систему (31) и (30).

Теорема 1 не гарантирует существования и единственности решения системы (31). Тем более она не гарантирует, что все найденные узлы, т. е. корни многочлена $\omega(x)$, принадлежат отрезку $[a, b]$. Ответы на эти важные вопросы дает следующая теорема.

Теорема 2. Многочлен $\omega(x)$ степени n , ортогональный с весом $\rho(x) > 0$ на отрезке $[a, b]$ любому многочлену $q(x)$ степени меньше n , определяется однозначно, все корни этого многочлена различны и находятся на отрезке $[a, b]$.

Свойства квадратурных формул Гаусса

1. Наивысший порядок многочлена, для которых формула Гаусса с n узлами точна, — это $2n - 1$, т. е. существует многочлен степени $2n$, для которого формула уже не будет точна.

2. Все коэффициенты формулы Гаусса положительны, т. е. $C_i > 0$.

3. Для любой квадратурной формулы интерполяционного типа $\sum_{k=1}^n C_k$ ограничена.

★**Упр.** Докажите свойства квадратурных формул Гаусса.

Если $\rho(x)$ сохраняет знак на (a, b) и f имеет непрерывную $2n$ -ю производную на (a, b) , то справедлива оценка погрешности

$$|R| \leq \frac{1}{(2n)!} M_{2n} \int_a^b \rho(x) w^2(x) dx, \quad M_{2n} = \max_{[a,b]} |f^{(2n)}(x)|. \quad (32)$$

2.8. Некоторые весовые функции и соответствующие им ортогональные многочлены

Весовые функции $\rho(x)$ обычно используются, чтобы заранее учесть наиболее часто встречающиеся особенности подынтегральных функций $F(x)$ путем представления их в виде произведения $F(x) = \rho(x)f(x)$. Теорема 1 для формулы Гаусса утверждает, что в качестве узлов x_1, \dots, x_n нужно взять корни такого многочлена $\omega(x)$, который ортогонален с весом $\rho(x)$ всем многочленам $q(x)$ степени меньше n . В качестве $\rho(x)$ применяются следующие классические весовые функции:

- Если $\rho(x) \equiv 1$, то в качестве многочлена $\omega(x)$ используются ортогональные многочлены Лежандра, которые определяются как $P_n(x) = \frac{1}{2^n n!} ((x^2 - 1)^n)^{(n)}$, $n = 0, 1, \dots$ и являются ортогональными на отрезке $[-1, 1]$, т. е. $\int_{-1}^1 P_n(x) \cdot P_m(x) dx = 0$ при $n \neq m$.

В частности, $P_0(x) = 1$, $P_1(x) = x$, $P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}$, $P_3(x) = \frac{1}{2}(5x^3 - 3x), \dots$ Имеет место рекуррентная формула

$$(n+1)P_{n+1}(x) - x(2n+1)P_n(x) + nP_{n-1}(x) = 0, \quad n = 1, 2, \dots$$

Справедлива оценка погрешности

$$|R| \leq \frac{(b-a)^{2n+1} (n!)^4}{(2n+1)((2n!))^3} M_{2n}, \quad M_{2n} = \max_{[a,b]} |f^{(2n)}(x)|.$$

- Если $\rho(x) = (x-a)^\alpha(b-x)^\beta$, то в качестве многочлена $\omega(x)$ используются ортогональные многочлены Якоби. Данная весовая функция применяется, если подынтегральная функция имеет степенные особенности на концах отрезка.

Справедлива оценка погрешности

$$|R| \leq \frac{M_{2n}}{(2n)!} \frac{(b-a)^{\alpha+\beta+2n+1} n!}{(\alpha+\beta+2n+1)} \times \frac{\Gamma(\alpha+n+1)\Gamma(\beta+n+1)\Gamma(\alpha+\beta+n+1)}{\Gamma^2(\alpha+\beta+2n+1)},$$

$$M_{2n} = \max_{[a,b]} |f^{(2n)}(x)|,$$

где $\Gamma(x)$ — гамма функция Эйлера (см., например, [10]).

- Если $\rho(x) = \frac{1}{\sqrt{1-x^2}}$, то в качестве многочлена $\omega(x)$ используются ортогональные многочлены Чебышева, которые определяются как $T_n(x) = \cos(n \arccos x)$, $x \in [-1, 1]$. В частности, $T_0(x) = 1$, $T_1(x) = x$, $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x, \dots$ Имеет место рекуррентная формула

$$T_{n+1}(x) = 2x \cdot T_n(x) - T_{n-1}(x), \quad n = 1, 2, \dots$$

Корни x_k многочлена Чебышева и коэффициенты C_k квадратной формулы вычисляются в явном виде: $x_k = \cos \frac{2k-1}{2n} \pi$, $C_k = \pi/n$, $k = 1, 2, \dots, n$. Тогда формула Гаусса для такого типа интегралов имеет вид

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n} \sum_{k=1}^n f(x_k).$$

Справедлива оценка погрешности

$$|R| \leq \frac{\pi}{2^{2n-1}(2n)!} M_{2n}, \quad M_{2n} = \max_{[-1,1]} |f^{(2n)}(x)|.$$

- Если $\rho(x) = e^{-x^2}$, то в качестве многочлена $\omega(x)$ используются ортогональные многочлены Эрмита, которые имеют следующий вид: $H_n(x) = (-1)^n e^{x^2} (e^{-x^2})^{(n)}$. В частности, $H_0(x) = 1$, $H_1(x) = 2x$, $H_2(x) = 4x^2 - 2$, $H_3(x) = 8x^3 - 12x, \dots$ Имеет место рекуррентная формула

$$H_{n+1}(x) = 2x \cdot H_n(x) - 2n \cdot H_{n-1}(x), \quad n = 1, 2, \dots$$

Справедлива оценка погрешности

$$|R| \leq \frac{n! \pi}{2^n (2n)!} M_{2n}, \quad M_{2n} = \max_{x \in \mathbb{R}} |f^{(2n)}(x)|.$$

2.9. Несобственные интегралы

Чтобы вычислить сходящийся несобственный интеграл первого рода $\int_a^{+\infty} f(x) dx$ с точностью ε , необходимо: аналитически найти такое b ,

чтобы $\left| \int_b^{+\infty} f(x) dx \right| < \frac{\varepsilon}{2}$; вычислить определенный интеграл $\int_a^b f(x) dx$ по

приближенным формулам с точностью $\frac{\varepsilon}{2}$. Тогда это полученное прибли-

женное значение будет давать значение исходного интеграла $\int_a^{+\infty} f(x) dx$ с точностью ε .

Аналогично можно поступить в случае $\int_{-\infty}^b f(x) dx$.

Чтобы вычислить сходящийся несобственный интеграл второго рода $\int_a^b f(x) dx$ с точностью ε , где функция $f(x)$ имеет особенность в некоторой точке $c \in (a, b)$, необходимо: аналитически найти такое $\delta > 0$,

чтобы $\left| \int_{c-\delta}^{c+\delta} f(x) dx \right| < \frac{\varepsilon}{3}$; вычислить два определенных интеграла $I_1 =$

$= \int_a^{c-\delta} f(x)dx$ и $I_2 = \int_{c+\delta}^b f(x)dx$ по приближенным формулам с точностью $\frac{\varepsilon}{3}$ каждый. Тогда приближенное равенство $\int_a^b f(x)dx \approx I_1 + I_2$

имеет погрешность не более ε .

Комбинируя эти методы, можно вычислять интеграл, например, с бесконечными пределами при наличии нескольких особых точек, при этом отделяем каждую особую точку и весь интеграл заменяем суммой собственных интегралов на конечном отрезке без особенностей.

3. Численное решение задачи Коши

Пусть дана задача Коши

$$\begin{cases} u'(x) = f(x, u), \\ u(x_0) = u_0. \end{cases} \quad (33)$$

Будем предполагать, что правая часть уравнения удовлетворяет условиям, при которых в некоторой окрестности точки $M_0(x_0, u_0)$ задача имеет единственное решение. Задачу Коши в эквивалентном виде можно записать как

$$u(x) = u_0 + \int_{x_0}^x f(s, u)ds. \quad (34)$$

Для численного решения таких задач используют *метод дискретизации задачи*. Это означает, что вместо функции $y(x)$ непрерывного аргумента $x \in [a, b]$ ищут таблицу значений этой функции для некоторого набора узлов $x_0, x_1, \dots, x_n \in [a, b]$. Узлы расположены не обязательно равномерно. Чаще всего узлы задают заранее, что связано с естественной задачей определить значения функции в конкретных, заранее перечисленных точках.

Многие методы решения одного уравнения можно применить к системе уравнений. Задачу Коши для линейных дифференциальных уравнений высокого порядка можно записать в виде задачи для системы дифференциальных уравнений первого порядка, поэтому мы ограничимся численными методами решения задачи Коши для уравнений первого порядка.

В дальнейшем $u(x)$ будет означать точное решение дифференциального уравнения, а $y(x)$ — приближенное решение. В частности, через y_i будем обозначать таблицу приближенных значений искомой функции.

3.1. Метод последовательных приближений

Рассмотрим задачу Коши (33) в эквивалентном виде (34). Пусть $y^{(0)}(x) = u(x_0) = u_0$ — начальное приближение. Тогда процесс

$$y^{(n)}(x) = u_0 + \int_{x_0}^x f(s, y^{(n-1)}(s)) ds \quad (n = 1, 2, \dots) \quad (35)$$

называют *методом Пикара* или *методом последовательных приближений*.

Теорема (сходимости метода Пикара). *Если правая часть задачи Коши удовлетворяет условию Липшица по второму аргументу, то последовательные приближения $y^{(n)}(x)$ на некотором достаточно малом отрезке $[x_0, x_0 + h]$ имеют смысл и равномерно сходятся к $u(x)$, причем предельная функция является решением задачи (33).*

Замечание. Метод Пикара легко переносится на системы дифференциальных уравнений первого порядка, причем будет справедлива аналогичная теорема сходимости метода Пикара.

Метод Пикара имеет один существенный недостаток: на каждом шаге приходится вычислять интегралы, которые трудно запрограммировать. Т. к. на практике обычно требуется не аналитическое выражение для решения, а таблица значений, то итерационные процессы по формулам (35) реализуют в виде программ, в которых вычисление определенных интегралов заменяют квадратурными формулами.

3.2. Метод Эйлера, его модификации

Рассмотрим задачу Коши (33) на отрезке $[a, b]$, где $a = x_0$. Разобьем отрезок точками $x_i = a + ih$, $i = 0, 1, \dots, N$. Метод Эйлера (метод ломаных) определяется формулой

$$\begin{cases} y_{i+1} = y_i + h \cdot f(x_i, y_i), & i = 0, 1, \dots, N-1, \\ y_0 = u_0. \end{cases} \quad (36)$$

Заметим, что погрешность метода на одном шаге $[x_i, x_{i+1}]$ имеет порядок $O(h^2)$, а на всем отрезке $[a, b]$ составит величину порядка $O(h)$.

Поэтому этот метод на практике используется редко из-за большой погрешности.

Обычно используют другие варианты метода Эйлера. Процесс

$$\begin{cases} y_{i+1} = y_i + h \cdot f(x_i, y_i) + \frac{h^2}{2} \cdot (f'_x(x_i, y_i) + f'_u(x_i, y_i) \cdot f(x_i, y_i)), \\ i = 0, 1, \dots, N-1, \quad y_0 = u_0, \end{cases}$$

называют *исправленным методом Эйлера*. На каждом отрезке $[x_i, x_{i+1}]$, погрешность имеет порядок $O(h^3)$, на всем отрезке $[a, b]$ — погрешность $O(h^2)$. Главный недостаток метода в том, что на каждом шаге приходится вычислять частные производные f'_x и f'_u , поэтому он тоже используется редко.

Существует другая идея получения точных формул — это *усовершенствованные формулы Эйлера*:

$$\begin{cases} y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x_i, y_i), \\ y_{i+1} = y_i + h f\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right), \quad i = 0, 1, \dots, N-1, \end{cases} \quad (37)$$

где $y_0 = u_0$, $x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}$. Формулы (37) определяют *первый усовершенствованный метод Эйлера*. *Второй усовершенствованный метод Эйлера (метод трапеций Эйлера)* имеет вид

$$\begin{cases} \bar{y}_{i+1} = y_i + h f(x_i, y_i), \\ y_{i+1} = y_i + \frac{h}{2} (f(x_i, y_i) + f(x_{i+1}, \bar{y}_{i+1})), \quad i = 0, 1, \dots \end{cases} \quad (38)$$

Этот метод можно комбинировать с итерационным методом:

$$y_{i+1}^{(k+1)} = y_i + \frac{h}{2} \left(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k)}) \right), \quad k = 0, 1, \dots,$$

где в качестве начального приближения берем $y_{i+1}^{(0)} = y_i + h f(x_i, y_i)$ и на каждом шаге i решаем нелинейное уравнение

$$y_{i+1} = y_i + \frac{h}{2} (f(x_i, y_i) + f(x_{i+1}, y_{i+1}))$$

относительно y_{i+1} итерационным способом, причем продолжаем процесс, пока не будет выполнено условие $\left| y_{i+1}^{(k+1)} - y_{i+1}^{(k)} \right| < \varepsilon$, где ε — точность решения нелинейного уравнения. Если неравенство выполнено, то берем в качестве $y_{i+1} \approx y_{i+1}^{(k+1)}$.

Замечание. Нет смысла решать нелинейное уравнение, используя этот итерационный процесс, с большей точностью, чем погрешность самого усовершенствованного метода.

Все эти методы без труда переносятся на решение систем дифференциальных уравнений.

В формулах Эйлера можно при необходимости менять шаг на ходу, т.е. если в процессе вычислений обнаружится, что функция y_i стала очень медленно (быстро) меняться, то дальше вычисления лучше вести с более крупным (мелким) шагом.

3.3. Семейство схем второго порядка точности

Семейство формул вида

$$y_{i+1} = y_i + h \left[(1 - \beta)f(x_i, y_i) + \beta f \left(x_i + \frac{h}{2\beta}, y_i + \frac{hf(x_i, y_i)}{2\beta} \right) \right] \quad (39)$$

имеют третий порядок точности на одном шаге и второй порядок точности на всем отрезке $[a, b]$. При $\beta = 1$ получаем первую усовершенствованную формулу Эйлера (37). При $\beta = \frac{1}{2}$ имеем вторую усовершенствованную формулу Эйлера (38). Значит, усовершенствованные формулы Эйлера имеют точность второго порядка. Используя другие значения коэффициента β , можно получить различные формулы этого семейства.

3.4. Семейство методов Рунге–Кутты

Правую часть уравнения (33) вычислим p раз:

$$\begin{cases} \varphi_1 = hf(x_k, y_k), \\ \varphi_2 = hf(x_k + \alpha_1 h, y_k + \beta_{11}\varphi_1), \\ \varphi_3 = hf(x_k + \alpha_2 h, y_k + \beta_{21}\varphi_1 + \beta_{22}\varphi_2), \\ \vdots \\ \varphi_p = hf \left(x_k + \alpha_{p-1} h, y_k + \sum_{i=1}^{p-1} \beta_{p-1,i} \varphi_i \right). \end{cases} \quad (40)$$

Тогда общая формула семейства методов Рунге–Кутты имеет вид

$$y_{k+1} = y_k + \sum_{i=1}^p A_i \varphi_i, \quad k = 0, 1, \dots \quad (41)$$

Метод выбирается заданием p и набором параметров $\alpha_i, \beta_{ij}, A_i$.

Наиболее часто используется метод при $p = 4$, т.е. имеем формулы Рунге–Кутты 4-го порядка точности

$$\begin{cases} y_{k+1} = y_k + \frac{1}{6} (\varphi_1 + 2\varphi_2 + 2\varphi_3 + \varphi_4), \\ \varphi_1 = hf(x_k, y_k), \\ \varphi_2 = hf\left(x_k + \frac{h}{2}, y_k + \frac{\varphi_1}{2}\right), \\ \varphi_3 = hf\left(x_k + \frac{h}{2}, y_k + \frac{\varphi_2}{2}\right), \\ \varphi_4 = hf(x_k + h, y_k + \varphi_3). \end{cases} \quad (42)$$

Основные достоинства метода:

- 1) хорошая точность при простом алгоритме;
- 2) явная схема;
- 3) шаг h можно менять на ходу, т.е. в процессе вычислений;
- 4) для начала расчета нужно только то, что дано в самой постановке задачи Коши;

5) легко переносится на систему дифференциальных уравнений первого порядка. Например, для системы

$$\begin{cases} y' = f(x, y, z), \\ z' = g(x, y, z), \end{cases} \quad \begin{cases} y(x_0) = y_0, \\ z(x_0) = z_0. \end{cases} \quad (43)$$

Формулы Рунге–Кутты для нахождения численного решения имеют вид:

$$\begin{aligned} \varphi_1 &= hf(x_i, y_i, z_i), \\ \psi_1 &= hg(x_i, y_i, z_i), \\ \varphi_2 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{\varphi_1}{2}, z_i + \frac{\psi_1}{2}\right), \\ \psi_2 &= hg\left(x_i + \frac{h}{2}, y_i + \frac{\varphi_1}{2}, z_i + \frac{\psi_1}{2}\right), \\ \varphi_3 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{\varphi_2}{2}, z_i + \frac{\psi_2}{2}\right), \\ \psi_3 &= hg\left(x_i + \frac{h}{2}, y_i + \frac{\varphi_2}{2}, z_i + \frac{\psi_2}{2}\right), \\ \varphi_4 &= hf(x_i + h, y_i + \varphi_3, z_i + \psi_3), \\ \psi_4 &= hg(x_i + h, y_i + \varphi_3, z_i + \psi_3), \end{aligned} \quad (44)$$

$$\begin{aligned} y_{i+1} &= y_i + \frac{1}{6}(\varphi_1 + 2\varphi_2 + 2\varphi_3 + \varphi_4), \\ z_{i+1} &= z_i + \frac{1}{6}(\psi_1 + 2\psi_2 + 2\psi_3 + \psi_4). \end{aligned} \quad (45)$$

3.5. Метод Рунге–Ромберга

Используя формулу Рунге–Ромберга (5), можно получить приближенное решение задачи Коши (33) в точке b с любой заранее заданной точностью $\varepsilon > 0$.

В соответствии с методикой Рунге–Ромберга будем находить приближенные решения в этой точке с разными шагами и сравнивать их. Находим $Y_h(b)$, например, по формулам Рунге–Кутты с $m = 4$. Уменьшаем шаг в два раза и вычисляем значение $Y_{\frac{h}{2}}(b)$. Находим

$$\delta = \frac{Y_{\frac{h}{2}}(b) - Y_h(b)}{2^m - 1}.$$

Если $|\delta| < \varepsilon$, то полагаем $u(b) \approx Y_{\frac{h}{2}}(b) + \delta$, и на этом вычисления заканчиваются. Если $|\delta| \geq \varepsilon$, то уменьшаем шаг вдвое, находим новое приближение в точке b , сравниваем с тем, что было найдено на предыдущем этапе и т. д. Описанная методика дает возможность получить приближенное значение решения $u(x)$ задачи Коши с наперед заданной точностью в любой точке $b > a$.

Для получения таблицы приближенных значений решения $u(x)$ в точках x_i с заданной точностью ε , тоже можно пользоваться методикой Рунге–Ромберга. Сначала решаем задачу Коши на $[x_0, x_1]$ с заданной точностью ε , получим $u(x_1) \approx y(x_1)$ с точностью ε . Далее берем другой отрезок $[x_1, x_2]$ и находим $u(x_2) \approx y(x_2)$ с точностью ε и т. д.

Получим таблицу

x_1	x_2	\dots	x_N
y_1	y_2	\dots	y_N

 с требуемой точностью ε .

3.6. Метод Адамса

Пусть решается задача Коши (33) на $[a, b]$. Используя для нахождения y_{i+1} четыре предыдущих значения $y_i, y_{i-1}, y_{i-2}, y_{i-3}$, получаем

4-шаговый разностный метод Адамса ($h = x_{i+1} - x_i$)

$$y_{i+1} = y_i + \frac{h}{24} \left[55f(x_i, y_i) - 59f(x_{i-1}, y_{i-1}) + 37f(x_{i-2}, y_{i-2}) - 9f(x_{i-3}, y_{i-3}) \right], \quad i = 3, 4, \dots, \quad (46)$$

который имеет четвертый порядок точности.

★**Упр.** Докажите, что формулы Адамса имеют четвертый порядок точности.

Вычисления по формулам Адамса можно начать только после того, как найдено 4 «начальных» значения y_0, y_1, y_2, y_3 . Их находят другим методом, обычно методом Рунге–Кутты.

В одношаговых методах можно было легко менять шаг на ходу. В методе Адамса, как видно из формулы (46), это трудно сделать в процессе вычисления. При необходимости сначала нужно получить значение y_i для четырех узлов с новым шагом с помощью одношагового метода, а потом продолжить вычисление с новым шагом по формулам Адамса.

4. Численное решение краевых задач с одной независимой переменной

Краевая задача называется *корректной*, если в некотором классе входных данных ее решение существует, единственно и непрерывным образом зависит от входных данных.

4.1. Метод стрельбы для линейных краевых задач

Рассмотрим простейшую краевую задачу, удовлетворяющую следующим условиям:

$$L[u] \equiv u''(x) + p(x)u'(x) + q(x)u(x) = f(x), \quad (47)$$

$$l_a[u] \equiv \alpha u'(a) + \beta u(a) = A, \quad (48)$$

$$l_b[u] \equiv \delta u'(b) + \gamma u(b) = B, \quad (49)$$

где $|\alpha| + |\beta| \neq 0$, $|\gamma| + |\delta| \neq 0$, а на функции f, p, q накладываются такие условия, при которых задача (47)–(49) имеет единственное решение. Метод основан на том, что линейную краевую задачу (47)–(49) можно свести к решению двух задач Коши.

Пусть $v(x)$ — решение задачи Коши (47) с краевыми условиями

$$v'(a) = \frac{A}{\alpha}, \quad v(a) = 0 \quad \text{при } \alpha \neq 0 \quad (50)$$

или

$$v'(a) = 0, \quad v(a) = \frac{A}{\beta} \quad \text{при } \beta \neq 0. \quad (51)$$

Пусть $w(x)$ — решение однородного уравнения

$$L[w] = 0 \quad (52)$$

с однородным условием на левом конце

$$l_a[w] = 0. \quad (53)$$

Для определенности начальные условия можно взять в виде

$$w'(a) = -\frac{\beta}{\alpha}, \quad w(a) = 1 \quad \text{при } \alpha \neq 0$$

или

$$w'(a) = 1, \quad w(a) = -\frac{\alpha}{\beta} \quad \text{при } \beta \neq 0.$$

Функция

$$u(x) = v(x) + cw(x) \quad (54)$$

с константой c , определяемой равенством

$$c = \frac{B - \delta v'(b) - \gamma v(b)}{\delta w'(b) + \gamma w(b)}, \quad (55)$$

дает искомое решение $u(x)$ исходной краевой задачи (47)–(49).

Таким образом, метод стрельбы для линейной задачи (47)–(49) сводится к решению двух задач Коши для функций $v(x)$ и $w(x)$ — двум пробным пристрелкам и построению функции $y(x)$ по формуле (54), где константа c определяется из (55). Задачи Коши (47), (50) (или (47), (51)) и (52)–(53) решаются, как правило, численными методами, например, методом Рунге–Кутты. Поэтому метод стрельбы относится к численным методам. Погрешность метода определяется погрешностью решения задачи Коши для функций $v(x)$ и $w(x)$.

Если знаменатель выражения (55) окажется равным нулю, т.е. $\delta w'(b) + \gamma w(b) = 0$, то $l_b[w] = 0$, что значит, наша задача имеет нетривиальное решение, а это служит признаком вырожденности исходной задачи (47)–(49).

Целесообразно строить численные решения обеих задач Коши на одном и том же дискретном множестве. Если требуется найти решение краевой задачи с заданной точностью ε , то можно использовать метод Рунге–Ромберга.

4.2. Метод стрельбы для нелинейных краевых задач

Если оператор $L[u]$ — нелинейный, то рассмотренный вариант метода стрельбы не будет работать. Но метод стрельбы можно использовать и для решения нелинейных краевых задач, если применить следующий несколько общий подход, который продемонстрируем на примере решения простейшей краевой задачи

$$u'' = f(x, u, u'), \quad (56)$$

$$u(a) = u_a, \quad (57)$$

$$u(b) = u_b, \quad (58)$$

где функция f — нелинейная.

Возьмем некоторую константу ξ и рассмотрим задачу Коши для уравнения (56) с начальными условиями

$$u(a) = u_a, \quad u'(a) = \xi, \quad (59)$$

где ξ — произвольное число, пока неизвестное. Предположим, что мы нашли решение $u(x, \xi)$ этой задачи при некотором $\xi = \xi_1$. Тогда в точке b значение $u(b, \xi_1) \neq u_b$, т.е. $u(x, \xi_1)$ не является решением (56)–(58). Значит, будем подбирать параметр ξ таким образом, чтобы функция $u(x, \xi)$ удовлетворяла не только условиям (56)–(57), но и условию (58):

$$u(b, \xi) = u_b, \quad (60)$$

где $u(b, \xi)$ — решение задачи Коши (56), (59).

Уравнение (60) — нелинейное уравнение относительно ξ , значит, его решение в явном виде не выписывается по результатам двух пристрелок. Следовательно, его решение ищем приближенным методом, например, методом половинного деления. Введем функцию $F(\xi) = u(b, \xi) - u_b$

и решим уравнение $F(\xi) = 0$. Надо иметь в виду, что эта функция не задана аналитическим выражением, а задается в виде алгоритма — при каждом заданном значении аргумента для нахождения $F(\xi)$ надо решать задачу Коши (56), (59). Для решения уравнения методом половинного деления определяем отрезок $[c, d]$, на концах которого функция принимала бы значения разных знаков: $F(c) \cdot F(d) < 0$. Найдем точку $\xi_* : F(\xi_*) = 0$. Для найденного значения $\xi = \xi_*$ еще раз решаем задачу Коши (56), (59). Полученное решение удовлетворяет нелинейной краевой задаче (56)–(58).

4.3. Метод коллокации

Другие названия метода коллокации — интерполяционный метод или метод совпадений. Будем искать приближенное решение линейной краевой задачи (47)–(49) в виде функции

$$y(x) = \varphi_0(x) + \sum_{i=1}^n c_i \varphi_i(x), \quad (61)$$

где определяемые на отрезке $[a, b]$ базисные функции $\{\varphi_i(x)\}_{i=1}^n$ и дополнительная функция $\varphi_0(x)$ должны быть дважды дифференцируемы и попарно линейно независимыми. Кроме того, функция $\varphi_0(x)$ должна удовлетворять краевым условиям (48)–(49), а функции $\varphi_i(x)$, $i = 1, 2, \dots, n$, — соответствующим однородным краевым условиям, т. е. должны выполняться равенства $\forall i = 1, \dots, n$

$$l_a[\varphi_i] = \alpha \varphi'_i(a) + \beta \varphi_i(a) = 0, \quad l_b[\varphi_i] = \delta \varphi'_i(b) + \gamma \varphi_i(b) = 0. \quad (62)$$

В таком случае функция $y(x)$, определяемая выражением (61), при любых значениях коэффициентов c_i гарантировано удовлетворяет краевым условиям (48)–(49).

В методе коллокации коэффициенты c_i в представлении (61) приближенного решения $y(x)$ подбираются так, чтобы в узлах коллокации x_i таких, что $a < x_1 < x_2 < \dots < x_n < b$, значения $y(x_i)$ приближенного решения были согласованы с точными значениями $u(x_i)$. Узлы коллокации должны быть не обязательно равноотстоящими, но строго внутренними точками отрезка $[a, b]$.

Поскольку точное решение $u(x)$ задачи (47)–(49) неизвестно, то согласование $y(x)$ и $u(x)$ в узлах коллокации x_i проводят подстановкой

$y(x)$ в уравнение (47). Полагая

$$a_{ij} = \varphi_j''(x_i) + p(x_i)\varphi_j'(x_i) + q(x_i)\varphi_j(x_i), \quad (63)$$

$$b_i = f(x_i) - \varphi_0''(x_i) - p(x_i)\varphi_0'(x_i) - q(x_i)\varphi_0(x_i), \quad (64)$$

получим СЛАУ относительно коэффициентов c_1, c_2, \dots, c_n

$$\sum_{j=1}^n c_j a_{ij} = b_i, \quad i = 1, \dots, n. \quad (65)$$

Решив эту систему каким-нибудь стандартным методом (см. [1, гл. 1]) и подставив найденные коэффициенты c_i в (61), получим приближенное решение краевой задачи (47)–(49) методом коллокации.

Успех применения метода коллокации к задаче (47)–(49) во многом зависит от удачного выбора базисных функций φ_i . В конкретных задачах выбор таких функций, по возможности, должен опираться на априорные и эмпирические сведения о решении. В отсутствие таковых, для смешанной краевой задачи можно предложить следующий набор базисных функций.

Функцию $\varphi_0(x)$ подберем так, чтобы она удовлетворяла неоднородным краевым условиям (48)–(49). Обычно берут линейную функцию $\varphi_0(x) = cx + d$, для которой находят коэффициенты c и d из системы

$$\begin{cases} \alpha c + \beta(ca + d) = A, \\ \delta c + \gamma(cb + d) = B. \end{cases} \quad (66)$$

Функции $\varphi_i(x)$ можно взять однопараметрическими вида

$$\varphi_i(x) = p_i(x - a)^i + (x - a)^{i+1}, \quad (67)$$

где $p_i = -\frac{(b-a)(\delta(i+1) + \gamma(b-a))}{\delta i + \gamma(b-a)}$, если $\alpha = 0$, или вида

$$\varphi_i(x) = p_i(x - a)^{i+1} + (x - a)^{i+2}, \quad (68)$$

где $p_i = -\frac{(b-a)(\delta(i+2) + \gamma(b-a))}{\delta(i+1) + \gamma(b-a)}$ в самом общем случае. Тогда функция (61), где $\varphi_0(x)$ и $\varphi_i(x)$ определены с помощью (66) и (67)–(68), может служить приближенным решением краевой задачи (47)–(49).

Приближенное решение $y(x)$ в точках коллокации $x_i \in (a, b)$ должно удовлетворять дифференциальному уравнению (47), но оно не обязано совпадать в этих точках с точным решением $u(x)$.

Метод коллокации можно использовать и для решения нелинейных краевых задач. Трудности здесь ожидают с подбором базисных функций и в необходимости решать нелинейные системы при отыскании коэффициентов c_i .

4.4. Метод Галеркина

Метод Галеркина относится к проекционным методам. Пусть необходимо решить краевую задачу (47)–(49). В качестве гильбертова пространства рассмотрим $L_2[a, b]$ — пространство функций, интегрируемых с квадратом, где скалярное произведение определяется как

$$(u, v) = \int_a^b u(x)v(x)dx.$$

Пусть $\{\varphi_i\}_{i=1}^{\infty}$ — полная замкнутая система линейно независимых элементов в $L_2[a, b]$. Зафиксируем первые n элементов системы $\{\varphi_i\}_{i=1}^n$. Тогда они образуют базис некоторого конечномерного подпространства $H_n \subset L_2[a, b]$. Приближенное решение $y(x)$ будем искать в виде $y(x) = \varphi_0(x) + y_n(x)$, где $y_n(x) \in H_n$, $\varphi_0(x) \perp H_n$, и любой элемент раскладывается по базису $y_n(x) = \sum_{i=1}^n c_i \varphi_i(x)$, т.е. $y(x)$ ищется в виде (61).

Успех приближенного метода во многом зависит от удачного выбора базисных функций φ_i . Сначала выбирают функцию $\varphi_0(x)$, которая удовлетворяла бы краевым условиям (48)–(49). Обычно берут линейную функцию $\varphi_0(x) = cx + d$, для которой находят коэффициенты c и d из системы (66). Далее выбирают функции $\varphi_i(x)$ так, чтобы они удовлетворяли однородным краевым условиям (62) при любых коэффициентах c_i . К примеру, в качестве функции $\varphi_i(x)$ можно взять однопараметрический набор функций вида (67) или (68).

Неизвестные c_i подбираем так, чтобы решение $y(x)$ было наилучшим среди всех возможных таких решений (61). Для этого подставим (61) в дифференциальное уравнение (47) и рассмотрим разность, которую называют *невязкой* уравнения $L[u] = f$,

$$R(c_1, c_2, \dots, c_n) = L[\varphi_0] + \sum_{i=1}^n c_i L[\varphi_i] - f. \quad (69)$$

Если $u(x)$ — точное решение, то $R(c_i) = 0$. Поэтому для приближенного решения коэффициенты c_i выбираем так, чтобы невязка была в каком-то смысле мала. Согласно методу Галеркина требуется, чтобы невязка R была ортогональна всем базисным функциям φ_j , т. е.

$$(R, \varphi_j) = 0, \quad \forall j = 1, 2, \dots, n.$$

Используя свойства скалярного произведения, получим СЛАУ (65), где $a_{ij} = (L[\varphi_i], \varphi_j)$ и $b_j = (f - L[\varphi_0], \varphi_j)$. Решив СЛАУ и подставив найденные коэффициенты c_i в (61), получим решение краевой задачи (47)–(49) методом Галеркина.

4.5. Разностный метод

Один из наиболее простых и универсальных методов приближенного решения краевых задач является конечно-разностный метод.

Идея метода заключается в следующем.

1. Область непрерывного изменения аргумента заменяется конечным множеством точек, которое называется *сеткой*.

2. На этом конечном множестве определяют сеточную функцию, поставив в соответствие каждому элементу множества вполне определенное число. Вместо функции непрерывного аргумента будем искать сеточную функцию, которая в каком-то смысле является решением исходной краевой задачи. Другими словами, мы хотим найти таблицу значений искомой функции в некоторых точках.

3. Понятие производной к сеточным функциям в обычном классическом смысле неприменимо, т. к. есть минимальное расстояние между узлами сетки, и нет смысла рассматривать предел при стремлении приращения аргумента к нулю. Вместо производных в исходном дифференциальном уравнении и краевых условиях используются их приближенные конечно-разностные аппроксимации, т. е. линейные комбинации значений функции в некоторых точках заданной сетки.

4. Заменяя производные конечно-разностными отношениями формул численного дифференцирования, получим систему уравнений. Причем количество уравнений равно количеству узлов сетки. Таким образом, вместо данной краевой задачи будем иметь систему алгебраических уравнений относительно значений искомой функции в узлах выбранной сетки. Полученная система называется *разностной схемой*.

Рассмотрим реализацию метода на примере решения линейной краевой задачи (47)–(49). На отрезке $[a, b]$ введем сетку с постоянным шагом $h = \frac{b-a}{n}$, $x_i = a + ih$, $i = 0, 1, \dots, n$. Для замены дифференциального уравнения разностным надо ввести сеточные функции. Положим $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$, где $i = 0, 1, \dots, n$. Подставляя формулы численного дифференцирования (см. [1, с. 42]) для производных в дифференциальное уравнение (47) и граничные условия (48)–(49), получим разностную схему

$$\begin{cases} \alpha \frac{y_1 - y_0}{h} + \beta y_0 = A, \\ \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i, \quad i = 1, \dots, n-1, \\ \delta \frac{y_n - y_{n-1}}{h} + \gamma y_n = B, \end{cases}$$

которая представляет из себя СЛАУ относительно неизвестных значений сеточной функции y_0, \dots, y_n .

Дифференциальное уравнение аппроксимировали с точностью $O(h^2)$, краевые условия — с точностью $O(h)$, поэтому разностная схема аппроксимирует исходную краевую задачу с точностью $O(h)$.

Вводя обозначения $\beta h - \alpha = c_0$, $\alpha = d_0$, $Ah = r_0$, $b_i = 2 - p_i h$, $c_i = -4 + 2h^2 q_i$, $d_i = 2 + hp_i$, $r_i = 2h^2 f_i$, $b_n = -\delta$, $c_n = \gamma h + \delta$, $r_n = Bh$, систему перепишем следующим образом:

$$\begin{cases} c_0 y_0 + d_0 y_1 = r_0, \\ b_i y_{i-1} + c_i y_i + d_i y_{i+1} = r_i, \quad i = 1, \dots, n-1, \\ b_n y_{n-1} + c_n y_n = r_n. \end{cases}$$

Это СЛАУ с трехдиагональной матрицей, которая решается высокоэффективным методом прогонки (см. [1, с. 11]). Достаточные условия устойчивости и корректности метода прогонки для данной системы имеют вид (см. [1, с. 12])

$$h \max_{x \in [a, b]} |p(x)| \leq 2, \quad q(x) < 0. \quad (70)$$

Если имеем краевые условия первого рода, то достаточно выполнения условий (70). В случае краевых условий третьего рода требуется выполнение (70), $\alpha\beta < 0$ и $\gamma\delta > 0$.

5. Численное решение уравнений в частных производных

5.1. Основные понятия теории разностных схем

Суть разностного метода была изложена в п. 4.5. Применение метода приводит к решению СЛАУ. Получив вместо исходной задачи СЛАУ, недостаточно считать, что решение задачи закончено, необходимо еще изучить возможные методы решения полученных систем. Заметим, что получаемые системы будут иметь, как правило, некоторые особенности, которые облегчают решение.

Каждое уравнение системы будет связывать значения искомой функции в нескольких соседних точках, называемых *шаблоном*, т.е. в каждое уравнение входит только несколько неизвестных (в зависимости от количества точек в шаблоне), а значит, получим матрицу, хоть и большого порядка, но очень сильно разреженную. Это позволяет подбирать экономичные методы решения таких систем.

Пусть имеем краевую задачу

$$Lu = f(x), \quad x \in G, \quad G \subset \mathbb{R}^m. \quad (71)$$

Будем считать, что все дополнительные условия (начальные и краевые) включены в линейный оператор L и в функцию f .

В области G выберем сетку G_h , где $h = (h_1, h_2, \dots, h_m)$ задает шаг сетки по каждой координате. На сетке G_h будем рассматривать сеточные функции $y_h(x)$, $x \in G_h$. Эти функции образуют некоторое пространство H_h . Функция $u(x)$, $x \in G$, непрерывного аргумента является элементом некоторого функционального пространства $u(x) \in H_0$. Заменим дифференциальный оператор L разностным оператором L_h , а $f(x)$ — сеточной функцией $\varphi_h(x)$. Тогда (71) заменится разностной схемой

$$L_h y_h(x) = \varphi_h(x), \quad x \in G_h. \quad (72)$$

Пусть $u(x)$ — решение исходной задачи (71), $y_h(x)$ — решение разностной задачи (72). Каждой функции $u(x) \in H_0$ ставим в соответствие некоторую сеточную функцию $u_h(x) \in H_h$, $x \in G_h$, с помощью оператора проектирования $P_h: H_0 \rightarrow H_h$, т.е. $P_h u(x) = u_h(x)$.

Рассмотрим разность $y_h - u_h$, которая является элементом пространства H_h , и близость этих функций будем оценивать по норме этого пространства $\|y_h - u_h\|_h$. При этом нужно, чтобы выполнялось условие

согласования норм $\|\cdot\|_h$ и $\|\cdot\|_0$ в смысле предела

$$\lim_{h \rightarrow 0} \|u_h\|_h = \|u\|_0, \quad \forall u \in H_0.$$

Погрешностью аппроксимации оператора L разностным оператором L_h называется сеточная функция $\psi_h = (Lv)_h - L_h v_h$, где $v \in H_0$, $v_h = P_h v$, $(Lv)_h = P_h(Lv)$. Если $\|\psi_h\|_h \rightarrow 0$ при $\|h\| \rightarrow 0$, то говорят, что разностный оператор L_h аппроксимирует дифференциальный оператор L . Оператор L_h аппроксимирует дифференциальный оператор L с порядком $k > 0$, если выполнено $\|\psi\|_h = O(\|h\|^k)$ ($\|h\| \rightarrow 0$).

Разностная схема (72) называется *корректной*, если 1) решение разностной схемы существует и единственно для любой правой части $\varphi_h \in H_h$; 2) существует константа $M_1 > 0$, не зависящая от h : $\forall \varphi_h \in H_h$ выполнено неравенство $\|y_h\|_h \leq M_1 \|\varphi_h\|_h$. Свойство 2) означает, что решение разностной задачи непрерывно зависит от правой части, причем имеем равномерную непрерывность относительно h . Свойство 2) называется *устойчивостью разностной схемы к малым изменениям правой части*.

Сеточная функция $z_h(x) = y_h(x) - P_h u(x)$, $x \in G_h$ называется *погрешностью разностной схемы*. Говорят, что решение разностной задачи (72) *сходится к решению дифференциальной задачи (71)*, если погрешность $\|z_h\|_h \rightarrow 0$ при $\|h\| \rightarrow 0$. Разностная схема имеет k -й порядок точности, если $\|z_h\|_h = O(\|h\|^k)$ ($\|h\| \rightarrow 0$).

Ответ на вопрос об оценке точности получаемых результатов и сходимости дает следующая теорема.

Теорема (о сходимости в теории разностных схем). Пусть дифференциальная задача (71) поставлена корректно, разностная схема (72) является корректной и аппроксимирует исходную задачу. Тогда решение задачи (72) сходится к решению исходной задачи (71), причем порядок точности совпадает с порядком аппроксимации.

Введем следующие обозначения: P — центральная точка (x_i, y_j) шаблона, $I(P)$ — множество всех точек этого шаблона, $I'(P)$ — проколотая окрестность точки P .

Каноническая запись разностного уравнения имеет вид

$$A(P)u(P) = \sum_{Q \in I'(P)} B(P, Q)u(Q) + F(P), \quad P \in \Omega, \quad (73)$$

где Ω — сеточное множество.

Отметим, что коэффициенты $A(P)$ и $B(P, Q)$, вообще говоря, зависят от выбранной точки P сетки и обладают важными свойствами:

$$A(P) > 0, \quad B(P, Q) > 0, \quad A(P) = \sum_{Q \in I'(P)} B(P, Q). \quad (74)$$

★**Упр.** Докажите свойства (74).

Для граничных узлов $P \in G_h$ полагают выполнение условий:

$$A(P) = 1, \quad I'(P) = \emptyset, \quad F(P) = \varphi(P). \quad (75)$$

Введем сеточный оператор

$$Lu(P) = A(P)u(P) - \sum_{Q \in I'(P)} B(P, Q)u(Q), \quad (76)$$

тогда разностное уравнение (73) запишется в виде

$$Lu(P) = F(P), \quad P \in \Omega. \quad (77)$$

Обозначив $D(P) = A(P) - \sum_{Q \in I'(P)} B(P, Q)$, получим

$$Lu(P) = D(P)u(P) + \sum_{Q \in I'(P)} B(P, Q)[u(P) - u(Q)].$$

Говорят, что в точке $P \in \Omega$ выполнены условия положительности коэффициентов разностной схемы (73), если

$$A(P) > 0, \quad B(P, Q) > 0 \quad \forall Q \in I'(P), \quad D(P) \geq 0. \quad (78)$$

Если точка сетки $P_0 \in \Omega$ является граничной, то $D(P_0) = A(P_0) > 0$.

5.2. Основные теоремы

Теорема 1. Пусть при всех $P \in \Omega$ выполнено условие положительности коэффициентов (78), $Lu(P) \leq 0$ ($Lu(P) \geq 0$) и найдется хотя бы один узел $P_0 \in \Omega$: $D(P_0) > 0$. Тогда $u(P) \leq 0$ ($u(P) \geq 0$) $\forall P \in \Omega$.

Теорема 2 (о существовании и единственности решения разностной схемы). Если выполнено условие положительности коэффициентов (78) и $\exists P_0 \in \Omega$: $D(P_0) > 0$, то разностная задача (77) имеет единственное решение.

Пусть наряду с уравнением (77) дано уравнение

$$Lv(P) = \tilde{F}(P), \quad P \in \Omega, \quad (79)$$

которое отличается от (77) только правой частью. Пусть Ω — сеточное множество.

Теорема 3 (сравнения). *Пусть на сетке Ω выполнены условия положительности коэффициентов (78). Если выполнено $|F(P)| \leq |\tilde{F}(P)|$, $\forall P \in \Omega$, то $|u(P)| \leq |v(P)|$, $\forall P \in \Omega$.*

Обозначим через ω — множество внутренних точек, γ — множество граничных точек, причем $\Omega = \omega \cup \gamma$.

Переформулируем теорему сравнения на случай краевой задачи первого типа. Пусть даны две задачи:

$$Lu(P) = F(P), \quad P \in \omega, \quad u(P) = \mu(P), \quad P \in \gamma, \quad (80)$$

$$Lv(P) = \tilde{F}(P), \quad P \in \omega, \quad v(P) = \tilde{\mu}(P), \quad P \in \gamma. \quad (81)$$

Теорема 4 (сравнения для краевых задач I типа). *Пусть при $P \in \gamma$ выполнены условия положительности коэффициентов (78) и $|F(P)| \leq |\tilde{F}(P)|$, $\forall P \in \omega$; $|\mu(P)| \leq |\tilde{\mu}(P)|$, $\forall P \in \gamma$. Тогда $|u(P)| \leq |v(P)|$, $\forall P \in \Omega = \omega \cup \gamma$.*

Используя теорему сравнения, легко получить устойчивость решения краевой задачи первого типа по граничным условиям.

Теорема 5 (устойчивости решения). *Для решения однородного уравнения $Lu(P) = 0$, $\forall P \in \omega$, с неоднородными краевыми условиями $u(P) = \mu(P)$, $\forall P \in \gamma$, при выполнении условий положительности коэффициентов (78) имеет место следующая оценка*

$$\max_{P \in \omega} |u(P)| \leq \max_{P \in \gamma} |\mu(P)|.$$

5.3. Разностный метод для уравнений эллиптического типа

Пусть дано уравнение Пуассона

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x, y), \quad (x, y) \in G, \quad (82)$$

с условиями Дирихле на границе области G

$$u|_{\Gamma} = \varphi(x, y), \quad (x, y) \in \Gamma = \partial G. \quad (83)$$

Ограничимся рассмотрением случая, когда область $G = [0, a] \times [0, b]$ есть прямоугольник. Введем сетку на области G с шагами $h_1 = \frac{a}{n+1}$ и $h_2 = \frac{b}{m+1}$ по переменным $x_i = h_1 \cdot i$ и $y_j = h_2 \cdot j$:

$$G_h = \{(x_i, y_j), i = 0, 1, \dots, n+1, j = 0, 1, \dots, m+1\}.$$

Введем сеточные функции $f(x_i, y_j) = f_{ij}$, $\varphi(x_i, y_j) = \varphi_{ij}$.

Уравнение Пуассона заменится разностным уравнением для внутренних узлов

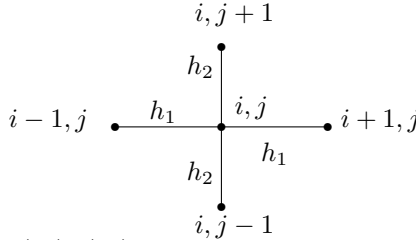
$$\frac{u_{i-1,j} - 2u_{ij} + u_{i+1,j}}{h_1^2} + \frac{u_{i,j-1} - 2u_{ij} + u_{i,j+1}}{h_2^2} = -f_{ij}, \quad (84)$$

$$i = 1, 2, \dots, n, j = 1, 2, \dots, m,$$

а условие Дирихле (83) для граничных узлов переписется в виде

$$u_{ij} = \varphi_{ij} \quad (85)$$

при $i = 0$ и $i = n+1$, $j = 0, 1, \dots, m+1$; при $j = 0$ и $j = m+1$, $i = 1, 2, \dots, n$. Разностное уравнение (84) в узле (x_i, y_j) связывает значения функции в пяти точках шаблона (x_i, y_j) , (x_{i-1}, y_j) , (x_{i+1}, y_j) , (x_i, y_{j-1}) , (x_i, y_{j+1}) .



Получили СЛАУ (84), (85) для неизвестных значений u_{ij} сеточной функции в узлах сетки G_h . Количество уравнений и неизвестных этой функции в общем случае равно $(n+2) \cdot (m+2)$. Учитывая граничные условия, которые определяются сразу, имеем количество уравнений и неизвестных — $n \cdot m$.

Заметим, что при $h_1 = h_2 = h$ уравнение (84) можно переписать в виде

$$u_{i-1,j} - 4u_{ij} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} = -h^2 f_{ij}. \quad (86)$$

Из уравнения (86) видно, что удобно использовать метод Якоби или метод Зейделя. Перепишем в виде

$$u_{ij} = \frac{1}{4} (u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) + \frac{h^2}{4} f_{ij}.$$

Возьмем $u_{ij}^{(0)}$ — начальное приближение. Тогда итерационный процесс с помощью метода Якоби задается как

$$u_{ij}^{(k+1)} = \frac{1}{4} \left(u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)} \right) + \frac{h^2}{4} f_{ij}, \quad k = 0, 1, \dots$$

Для заданной точности ε итерационный процесс останавливается, если будет выполнено неравенство $\|u^{(k+1)} - u^{(k)}\| < \varepsilon$.

Для метода Зейделя итерационный процесс принимает вид

$$u_{ij}^{(k+1)} = \frac{1}{4} \left(u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} \right) + \frac{h^2}{4} \cdot f_{ij}, \quad k = 0, 1, \dots$$

Замечание. В методе Якоби порядок, в котором обрабатываются уравнения, не имеет значения, а для метода Зейделя каждому способу упорядочения соответствует свой итерационный процесс.

★**Упр.** Исследуйте вопросы сходимости рассмотренных итерационных методов. *Указание.* Используйте необходимое и достаточное условие сходимости метода простых итераций для СЛАУ (см. [1, с. 13]).

5.4. Разностный метод для уравнений параболического типа

Рассмотрим на области $G = \left\{ (x, t) \middle| 0 < x < l, 0 < t \leq T \right\}$ первую краевую задачу для уравнения теплопроводности

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + g(x, t), \quad (87)$$

$$u(x, 0) = f(x), \quad \forall x \in (0, l), \quad (88)$$

$$u(0, t) = \mu_1(t), \quad u(l, t) = \mu_2(t), \quad \forall t \in [0, T], \quad (89)$$

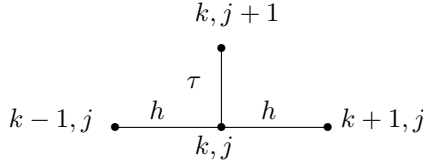
где $g(x, t)$, $f(x)$, $\mu_1(t)$, $\mu_2(t)$ — заданные функции. Введем сеточное множество

$$G_{h,\tau} = \left\{ (x_k, t_j) \middle| x_k = kh, \quad k = 0, 1, \dots, n; \quad t_j = j\tau, \quad j = 0, 1, \dots, m \right\},$$

где $h = \frac{l}{n}$ — шаг по пространственной координате, $\tau = \frac{T}{m}$ — шаг по временной координате. Введем сеточные функции $g_{kj} = g(x_k, t_j)$, $f_k = f(x_k)$, $\mu_{1j} = \mu_1(t_j)$, $\mu_{2j} = \mu_2(t_j)$, $k = 0, 1, \dots, n$, $j = 0, 1, \dots, m$. Получим схему

$$\begin{cases} \frac{y_{k,j+1} - y_{k,j}}{\tau} = \frac{y_{k-1,j} - 2y_{k,j} + y_{k+1,j}}{h^2} + g_{kj}, \\ k = 1, 2, \dots, n-1; j = 0, 1, \dots, m-1; \\ y_{k,0} = f_k, k = 1, 2, \dots, n-1; \\ y_{0,j} = \mu_{1j}, y_{n,j} = \mu_{2j}, j = 0, 1, \dots, m, \end{cases}$$

шаблон которой имеет вид



Получим СЛАУ относительно $y_{k,j}$. Выразим $y_{k,j+1}$:

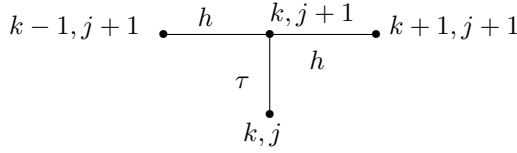
$$\begin{aligned} y_{k,j+1} &= \frac{\tau}{h^2} y_{k-1,j} + \left(1 - \frac{2\tau}{h^2}\right) y_{k,j} + \frac{\tau}{h^2} y_{k+1,j} + \tau g_{kj}, \\ k &= 1, 2, \dots, n-1; j = 0, 1, \dots, m-1. \end{aligned} \quad (90)$$

Равенство (90) позволяет вычислять все значения $y_{k,j}$ в явном виде, не решая системы. Взяв $j = 0$, можно вычислить $y_{k,1}$; взяв $j = 1$, можем вычислить $y_{k,2}$ и т. д. В силу записи (90), полученная разностная схема называется *явной разностной схемой* решения уравнения теплопроводности (87)–(89). Погрешность аппроксимации полученной разностной схемы имеет величину порядка $O(\tau + h^2)$. Недостаток явной схемы заключается в том, что решение задачи является неустойчивым к погрешностям в начальных данных. Условие устойчивости имеет вид

$$\tau \leq \frac{h^2}{2}. \quad (91)$$

Так как устойчивость возможна только при (91), то разностная схема называется *условно устойчивой*. Условие (91) является достаточно сильным ограничением. Поэтому условно устойчивые схемы на практике используются редко.

Используя шаблон



получим неявную разностную схему

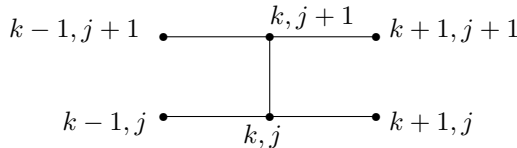
$$\left\{ \begin{array}{l} \frac{y_{k,j+1} - y_{k,j}}{\tau} = \frac{y_{k+1,j+1} - 2y_{k,j+1} + y_{k-1,j+1}}{h^2} + g_{kj}, \\ k = 1, 2, \dots, n-1, j = 0, 1, \dots, m-1; \\ y_{k,0} = f_k, k = 1, 2, \dots, n-1; \\ y_{0,j} = \mu_{1j}, j = 0, 1, \dots, m; \\ y_{n,j} = \mu_{2j}, j = 0, 1, \dots, m. \end{array} \right. \quad (92)$$

Преобразуя (92) и вводя обозначение $\frac{\tau}{h^2} = \sigma$, получим систему для нахождения неизвестных:

$$\left\{ \begin{array}{l} \sigma y_{k-1,j+1} - (1 + 2\sigma)y_{k,j+1} + \sigma y_{k+1,j+1} = -\tau g_{kj} - y_{k,j}, \\ k = 1, 2, \dots, n-1, j = 0, 1, \dots, m-1, \\ y_{k,0} = f_k, k = 1, 2, \dots, n-1; \\ y_{0,j} = \mu_{1j}, j = 0, 1, \dots, m; \\ y_{n,j} = \mu_{2j}, j = 0, 1, \dots, m. \end{array} \right.$$

При каждом фиксированном j имеем СЛАУ с трехдиагональной матрицей, следовательно, можно применить метод прогонки. Неявная схема лишена главного недостатка, присущего явным схемам, — она *безусловно устойчива*.

Рассмотрим для исходной задачи (87)–(89) разностную схему с весами — объединенный вариант явной и неявной схемы, которая использует следующий шаблон



Введем параметр $\rho \geq 0$, который будем называть *весом*. Для аппроксимации второй производной по x в правой части уравнения (87)

будем использовать и j слой, как в явной схеме, и $j+1$ слой, как в неявной

$$\left\{ \begin{array}{l} \frac{y_{k,j+1} - y_{k,j}}{\tau} = \rho \frac{y_{k+1,j+1} - 2y_{k,j+1} + y_{k-1,j+1}}{h^2} + \\ + (1 - \rho) \frac{y_{k+1,j} - 2y_{k,j} + y_{k-1,j}}{h^2} + g_{kj}, \\ k = 1, 2, \dots, n-1, \quad j = 0, 1, \dots, m-1; \\ y_{k,0} = f_k, \quad k = 1, 2, \dots, n-1; \\ y_{0,j} = \mu_{1j}, \quad j = 0, 1, \dots, m; \\ y_{n,j} = \mu_{2j}, \quad j = 0, 1, \dots, m. \end{array} \right.$$

При $\rho = 0$ получим явную схему, при $\rho = 1$ — неявную схему, а при $\rho = \frac{1}{2}$ — симметричную шеститочечную разностную схему. Разностная схема с весами будет устойчивой, если выполнены условия:

$$0 < \rho < 1, \quad \rho > 1 - \frac{h^2}{2\tau}.$$

5.5. Разностный метод для уравнений гиперболического типа

Рассмотрим уравнение колебания однородной струны единичной длины:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + g(x, t), \quad 0 < x < 1, \quad 0 < t \leq T, \quad (93)$$

с начальными условиями

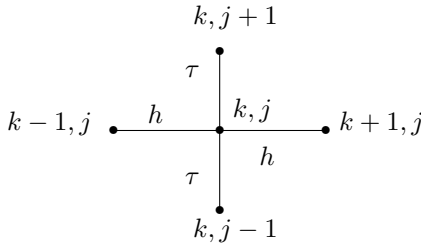
$$u(x, 0) = \varphi(x), \quad u'_t(x, 0) = \psi(x), \quad x \in (0, 1), \quad (94)$$

и краевыми условиями первого типа

$$u(0, t) = \mu_1(t), \quad u(1, t) = \mu_2(t), \quad t \in [0, T], \quad (95)$$

где $g(x, t)$, $\varphi(x)$, $\psi(x)$, $\mu_1(x)$, $\mu_2(x)$ — заданные функции.

На пятиточечном шаблоне



получим простейшее разностное уравнение

$$\frac{y_{k,j+1} - 2y_{k,j} + y_{k,j-1}}{\tau^2} = \frac{y_{k-1,j} - 2y_{k,j} + y_{k+1,j}}{h^2} + g_{k,j}, \quad (96)$$

$$j = 1, 2, \dots, m-1; \quad k = 1, 2, \dots, n-1,$$

которое имеет второй порядок погрешности аппроксимации по τ и h , т. е. $O(\tau^2 + h^2)$. Вводя обозначение $\frac{\tau^2}{h^2} = \sigma$, перепишем разностное уравнение (96) в виде

$$y_{k,j+1} = \sigma y_{k-1,j} + 2(1 - \sigma)y_{k,j} + \sigma y_{k+1,j} - y_{k,j-1} + \tau^2 g_{k,j}, \quad (97)$$

$$j = 1, 2, \dots, m-1, \quad k = 1, 2, \dots, n-1.$$

Дополним его граничными условиями

$$y_{0,j} = \mu_{1j}, \quad y_{n,j} = \mu_{2j}, \quad j = 0, 1, \dots, m, \quad (98)$$

$$y_{k,0} = \varphi_k, \quad k = 1, 2, \dots, n-1, \quad (99)$$

$$y_{k,1} = y_{k,0} + \tau \cdot \psi_k + \frac{\tau^2}{2} \cdot (\varphi_k'' + g_{k0}), \quad k = 1, 2, \dots, n-1. \quad (100)$$

Дифференциальная задача (93)–(95) заменяется разностной задачей (97)–(100), порядок аппроксимации полученной разностной схемы будет второй. Данная разностная схема решается в явном виде следующим образом. По (99)–(100) определим значения на нулевом и первом слоях сетки и добавим значения $y_{0,0}$, $y_{n,0}$, $y_{0,1}$, $y_{n,1}$, найденные по формулам (98). Из (97) при $j = 1$ вычисляем значения на втором слое $y_{k,2}$ ($k = 1, 2, \dots, n-1$). Эти значения дополняем значениями $y_{0,2}$ и $y_{n,2}$, полученными из (98). Вычисляем, используя (97) при $j = 2$, значения $y_{k,3}$ ($k = 1, 2, \dots, n-1$) и дополняем $y_{0,3}$ и $y_{n,3}$, найденными по (98) и т. д.

Устойчивость полученной разностной схемы определяется соотношением

$$\tau \leq h, \quad (101)$$

которое называют *условием Куранта*. Снова имеем ограничение на шаг по времени. Но это ограничение не такое жесткое, как в случае явных схем для уравнений параболического типа.

Используя неявные схемы для гиперболических уравнений, также можно получить безусловно устойчивые схемы. Однако они применяются редко, т. к. условие Куранта не является обременительным для расчетов, т. е. у неявных схем нет особых преимуществ перед явными.

★**Упр.** Получите неявную схему для уравнений колебаний и исследуйте ее на устойчивость.

5.6. Метод конечных элементов (МКЭ)

Пусть дана краевая задача в операторном виде

$$Au = f. \quad (102)$$

Задача построения обобщенного решения этого уравнения равносильна задаче об отыскании элемента энергетического пространства H_A , реализующего минимум функционала

$$J(u) = (Au, u) - 2(f, u), \quad u \in H_A. \quad (103)$$

Для отыскания минимума функционала обычно строят минимизирующую последовательность.

Последовательность функций $\{u_n\}$, принадлежащих области определения функционала D_A , называют *минимизирующей* для этого функционала, если $\lim_{n \rightarrow \infty} J(u_n) = \mu$. Предел минимизирующей последовательности $\lim_{n \rightarrow \infty} u_n = u$ дает искомое решение.

Одним из способов построения минимизирующей последовательности является процесс Ритца. В классическом варианте **метод Ритца** заключается в следующем. Выберем последовательность элементов $\{\varphi_i\}$, удовлетворяющую трем условиям:

- 1) все элементы $\varphi_i \in H_A$ из энергетического пространства;
- 2) $\forall n$ элементы $\varphi_1, \varphi_2, \dots, \varphi_n$ линейно не зависимы;
- 3) последовательность $\{\varphi_i\}$ полна в H_A , т.е. $\forall u \in H_A \quad \forall \varepsilon > 0 \quad \exists N$

$$\forall n > N \quad \exists u_n = \sum_{i=1}^n a_i \varphi_i, \quad a_i = \text{const} : \|u - u_n\|_{H_A} < \varepsilon.$$

Тогда n -е приближение (или n -й элемент минимизирующей последовательности) ищется в виде $u_n = \sum_{i=1}^n a_i \varphi_i$, где числовые коэффициенты a_i выбираются из условия минимума функционала $J(u_n)$.

Но нам потребуется несколько более **общая формулировка метода Ритца**. Введем последовательность конечномерных подпространств $V_1, V_2, \dots, V_n, \dots$ энергетического пространства H_A . Будем говорить, что эта последовательность $\{V_i\}$ полна в H_A , если $\forall u \in H_A \quad \forall \varepsilon > 0$

$\exists N \forall n > N: \inf_{v \in V_n} \|u - v\|_{H_A} < \varepsilon$. Другими словами, полнота последовательности пространств $\{V_i\}$ означает, что каждый элемент $u \in H_A$ с любой степенью точности может быть приближен элементом $u_n \in V_n$, если n достаточно велико. Тогда n -м приближением (или n -м элементом минимизирующей последовательности) назовем элемент $u_n \in V_n$, который минимизирует функционал $J(u)$ на этом подпространстве.

В данной формулировке метод Ритца реализуется следующим образом. Пусть $\{\varphi_i^{(n)}\}$ — базис подпространства V_n , т. е. любой элемент $u_n \in V_n$ может быть записан в виде

$$u_n = \sum_{i=1}^n a_i^{(n)} \varphi_i^{(n)}. \quad (104)$$

Подставим u_n вместо u в функционал (103). Тогда получим функцию n независимых переменных $a_1^{(n)}, a_2^{(n)}, \dots, a_n^{(n)}$. Коэффициенты $a_i^{(n)}$ выберем таким образом, чтобы полученная функция принимала минимальное значение, для чего приравняем нулю ее первые частные производные по этим переменным. Получаем систему Ритца

$$\sum_{j=1}^n (A\varphi_j^{(n)}, \varphi_i^{(n)}) a_j^{(n)} = (f, \varphi_i^{(n)}), \quad i = 1, 2, \dots, n. \quad (105)$$

Матрица этой системы симметрична, т. к. $(Au, v) = (u, Av)$. Определитель СЛАУ (105) есть определитель Грамма линейно не зависящих элементов и поэтому отличен от нуля. Найдя коэффициенты $a_1^{(n)}, a_2^{(n)}, \dots, a_n^{(n)}$ и подставив их в (104), получим элемент u_n , являющийся приближенным решением уравнения (102) по Ритцу.

Приближенное по Ритцу решение тем ближе к точному, чем больше n . Как следствие, для достижения высокой точности приходится решать СЛАУ (105) с матрицей большой размерности.

Метод Ритца основан на вариационной формулировке краевой задачи. Более общим вариационным методом, который непосредственно применим к уравнению (102), является метод Галеркина (называемый так же проекционный метод Бубнова–Галеркина). Теперь оператор A может быть произвольным линейным оператором, определенным на множестве D_A , всюду плотном в гильбертовом пространстве H_A . Запишем уравнение (102) в виде уравнения Галеркина

$$(Au, v) = (f, v), \quad v \in H_A. \quad (106)$$

Как и в методе Ритца выбирается последовательность конечномерных подпространств $V_n \subset H_A$ с базисами $\{\varphi_i^{(n)}\}$. Приближение по Галеркину по-прежнему ищется в виде (104). Но теперь коэффициенты $a_j^{(n)}$ определяются не из условия минимизации функционала, а из условия ортогональности невязки $Au_n - f$ элементам базиса $\{\varphi_i^{(n)}\}$, т. е.

$$(Au_n - f, \varphi_i^{(n)}) = 0, \quad i = 1, 2, \dots, n.$$

Это снова даст СЛАУ вида (105). Полученное совпадение в общем случае носит формальный характер, а по существу имеет место только в случае, когда оператор A обладает свойством самосопряженности, т. е. $(Au, v) = (u, Av)$. Метод Галеркина применим к более широкому классу задач, в частности, к нестационарным задачам.

Метод конечных элементов (далее МКЭ) является специальным вариантом метода Ритца–Галеркина. Классические методы Ритца и Галеркина обладают двумя существенными *недостатками*:

- 1) на практике построение базисных функций φ_i легко осуществить только для некоторых специальных областей;
- 2) соответствующие матрицы получаемых систем являются полными и очень часто, даже в случае простых задач, плохо обусловлены.

Принципиальное различие МКЭ состоит в построении базисных функций. В МКЭ базисные функции выбираются в виде локальных сплайнов и для областей общего вида могут быть вычислены весьма просто. Главная особенность таких функций в том, что они финитны, т. е. обращаются в нуль всюду, кроме фиксированного числа элементарных подобластей, на которые делится данная область. Отсюда следует, что матрица системы будет сильно разреженной и имеет ленточную структуру. Это дает устойчивость процесса численного решения системы.

Этапы МКЭ:

1. Заданная область Ω разбивается на конечное число элементарных подобластей, называемых *конечными элементами* так, чтобы выполнялись следующие свойства:

- а) $\Omega = \cup_{K \in T_n} K$, где K — конечный элемент;
- б) K не вырождаются (треугольники не переходят в отрезки, отрезки — в точки, четырехугольники — в треугольники и т. п.);
- в) соседние конечные элементы между собой не перекрываются.

2. Строится конечномерное подпространство функций V_n таких, что на каждой области K , выбранного разбиения, эти функции являются многочленами не выше заданной степени. Можно использовать рациональные функции или другие легко вычисляемые функции.

3. В пространстве V_n должен существовать хотя бы один базис из функций с минимальными носителями. Предполагается, что этот базис может быть легко описан. Тогда приближение по Галеркину ищется по этому базису. Существование такого базиса обеспечивает большое число нулей в матрице системы (105), что облегчает нахождение решения такой системы.

Успех в использовании МКЭ в значительной степени зависит от того, насколько удачно осуществлено указанное построение применительно к условиям конкретной задачи.

Пример решения краевой задачи для обыкновенного дифференциального уравнения методом конечных элементов.

Рассмотрим краевую задачу

$$-y'' = q(x), \quad y(0) = y(1) = 0. \quad (107)$$

Согласно схеме метода конечных элементов разобьем равномерно интервал $(0, 1)$ с шагом h , $x_0 = a, x_{n+1} = b$. Будем искать приближенное решение (107) в виде

$$y(x) = \sum_{i=1}^n a_i \varphi_i. \quad (108)$$

Чтобы $y(x_i) = a_i$ во всех узлах, функции φ_i должны удовлетворять условиям $\varphi_i(x_i) = 1$ и $\varphi_i(x_j) = 0$ при $i \neq j$. В качестве базисных функций будем использовать кусочно-линейные функции

$$\varphi_i = \begin{cases} \frac{x - x_{i-1}}{h}, & x \in [x_{i-1}, x_i], \\ \frac{x - x_{i+1}}{h}, & x \in [x_i, x_{i+1}], \\ 0, & x \notin [x_{i-1}, x_{i+1}]. \end{cases}$$

Требуется определить a_i в (108).

Обозначим невязку дифференциального уравнения из (107) через

$$R(x) = y''(x) + q(x).$$

Тогда по методу взвешенных невязок (см. [11])

$$\int_0^1 (y''(x) + q(x))W_j(x)dx = 0, \quad j = 1, \dots, n. \quad (109)$$

Используя метод Галеркина, приравняем весовые функции $W(x)$ к базисным, то есть $W_j(x) = \varphi_j(x)$. Проинтегрируем (109) по частям, учитывая $\varphi_j(0) = \varphi_j(1) = 0$, и получим

$$\int_0^1 (-y'(x)\varphi'_j + q(x)\varphi_j)dx = 0, \quad j = 1, \dots, n. \quad (110)$$

Подставим (108) в (110), получим

$$\int_0^1 \left[- \left(\sum_{i=1}^n (a_i \varphi_i)' \right) \varphi'_j + q(x)\varphi_j \right] dx = 0.$$

В силу того, что $a_i = \text{const}$,

$$\sum_{i=1}^n a_i K_{ij} = P_j, \quad (111)$$

$$K_{ij} = \int_0^1 \varphi'_i \varphi'_j dx, \quad P_j = \int_0^1 q(x) \varphi_j dx. \quad (112)$$

Матрица $K = \{K_{ij}\}$ называется матрицей жесткости, при этом является симметричной. Продифференцируем функции φ_i :

$$\varphi'_i = \begin{cases} \frac{1}{h}, & x \in (x_{i-1}, x_i), \\ -\frac{1}{h}, & x \in (x_i, x_{i+1}), \\ 0, & x \notin (x_{i-1}, x_{i+1}). \end{cases}$$

Для нахождения диагональных элементов матрицы K ($i = j$) получим

формулу

$$\begin{aligned}
 K_{ii} &= \int_{x_{i-1}}^{x_{i+1}} (\varphi'_i)^2 dx = \int_{x_{i-1}}^{x_i} \left(\frac{1}{h}\right)^2 dx + \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h}\right)^2 dx = \\
 &= \frac{1}{h^2} \left(x \Big|_{x_{i-1}}^{x_i} + x \Big|_{x_i}^{x_{i+1}} \right) = \frac{1}{h^2} (x_i - x_{i-1} + x_{i+1} - x_i) = \\
 &= \frac{1}{h^2} (x_{i+1} - x_{i-1}) = \frac{1}{h^2} \cdot 2h = \frac{2}{h};
 \end{aligned}$$

при $j = i + 1$ имеем

$$\begin{aligned}
 K_{i,i+1} &= \int_{x_{i-1}}^{x_{i+1}} (\varphi'_i \varphi'_{i+1}) dx = \int_{x_{i-1}}^{x_i} (\varphi'_i \varphi'_{i+1}) dx + \int_{x_i}^{x_{i+1}} (\varphi'_i \varphi'_{i+1}) dx = \\
 &= \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h} \cdot \frac{1}{h} \right) dx = -\frac{1}{h}.
 \end{aligned}$$

при $j = i - 1$ аналогично $K_{i,i-1} = -\frac{1}{h}$; при $i \neq j$, $j \neq i \pm 1$, очевидно $K_{ij} = 0$. Таким образом, матрица K имеет трехдиагональный вид

$$K = \frac{1}{h} \begin{bmatrix} 2 & -1 & 0 & & 0 & \dots & 0 \\ -1 & 2 & -1 & & 0 & \dots & 0 \\ 0 & -1 & 2 & & 0 & \dots & 0 \\ & & & \ddots & & & \\ 0 & \dots & 0 & & 2 & -1 & 0 \\ 0 & \dots & 0 & & -1 & 2 & -1 \\ 0 & \dots & 0 & & 0 & -1 & 2 \end{bmatrix}.$$

Решая систему (111), найдем a_1, \dots, a_n и подставим их в приближенное решение (108).

6. Лабораторные работы

6.1. Численное интегрирование

Дан интеграл I_1 (n — номер Вашего варианта):

$$I_1 = \int_{0.1(n+3)}^{0.25(n+3)} x^3 e^{-0.01nx^{\frac{3}{2}}} dx \text{ — для вариантов с четными номерами,}$$

$$I_1 = \int_{0.2(n+6)}^{0.3(n+6)} \frac{1}{x + \sin 0.1nx} dx \text{ — для вариантов с нечетными номерами.}$$

1. Определите число узлов для нахождения значения интеграла I_1 с точностью $\varepsilon_0 = 10^{-3}$ по формулам прямоугольников. Вычислите I_1 с данной точностью.

2. Определите число узлов для нахождения значения интеграла I_1 с точностью $\varepsilon_0 = 10^{-3}$ по формулам трапеций. Вычислите I_1 с точностью ε_0 .

3. Определите число узлов для нахождения значения интеграла I_1 с точностью $\varepsilon_0 = 10^{-3}$ по формулам Симпсона. Вычислите I_1 с заданной точностью.

4. Используя формулы интерполяционного типа, вычислите интеграл I_1 . Оцените погрешность.

5. Вычислите I_1 по формуле Гаусса с k узлами ($k \geq 4$ выбираете самостоятельно).

6. Вычислите первый интеграл с помощью сплайнов.

7. С точностью $\varepsilon_1 = 10^{-8}$ найдите значение интеграла I_1 методом Рунге–Ромберга.

8. Подбирая подходящие методы вычисления определенных и несобственных интегралов, с точностью $\varepsilon_2 = 10^{-3}$ найдите значение интеграла:

$$I_2 = \int_{0.1n}^{+\infty} x^{k+2} e^{-x^2} dx \text{ — для вариантов с нечетными номерами,}$$

$$I_2 = \int_{-0.1n}^{0.1n} \frac{\sin(x + 0.1n)}{\sqrt[3]{(x + 0.1n)^2(x - n)^2}} dx \text{ — для вариантов с четными номерами,}$$

где n — номер Вашего варианта, а k вычисляется: $k = 2 \left[\frac{n}{10} \right]$.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Реализуйте метод решения задачи на ЭВМ, используя системы компьютерной математики (Maple, Wolfram Mathematica, Mathcad и пр.) или языки программирования (по выбору студента).

Выведите на экран полученный результат, вычислите невязку или выполните проверку, используя встроенные функции. Округлите результат до p знаков после запятой при заданной точности вычислений $\varepsilon = 10^{-p}$.

Для успешной сдачи задания преподаватель может потребовать от студента:

- 1) сформулировать постановку задачи;
- 2) определить входные данные, выходные данные, их тип;
- 3) дать основные определения, относящиеся к изучаемому разделу;
- 4) изложить суть используемого метода;
- 5) пояснить алгоритм своей программы;
- 6) дать описание встроенных функций и подключаемых библиотек, используемых в программе.

6.2. Численное решение задачи Коши

Дана задача Коши

$$\begin{cases} y' = f(x, y), \\ y(a) = y_0, \end{cases}$$

где функция $f(x, y)$, отрезок $[a, b]$, начальное значение y_0 , а n — номер Вашего варианта.

n	$f(x, y)$	$[a, b]$	y_0	n	$f(x, y)$	$[a, b]$	y_0
1	$\cos^4 x - 2y \operatorname{tg} x$	$[0; 1]$	1	16	$\frac{x + y^2}{xy}$	$[1; 2]$	1
2	$y^2 e^{2x} - 3y$	$[0; 1]$	1	17	$2xy + e^{x^2}$	$[0; 1]$	1
3	$\sin x \cos x - y \cos x$	$[0; 1]$	1	18	$2xe^{-x^2} - 2xy$	$[0; 1]$	1
4	$-e^{-x^2} - 2xy$	$[1; 2]$	0	19	$3x^3 + \frac{y}{x}$	$[1; 2]$	1
5	$x \sin x e^{-x^2} - 2xy$	$[0; 1]$	1	20	$\frac{1}{\cos^3 x} + y \operatorname{tg} x$	$[0; 1]$	0
6	$-xe^{e^x} + ye^x$	$[0; 1]$	e	21	$e^{(1-x)e^x} - xye^x$	$[1; 2]$	1
7	$\cos\left(\frac{5x}{2} - \frac{y}{2}\right)$	$[0; 1]$	0	22	$\frac{2x^4 + 2y}{x}$	$[1; 2]$	1
8	$\frac{xy + e^x}{x}$	$[1; 2]$	1	23	$\frac{4x + 2y}{2x + 1}$	$[0; 1]$	1

n	$f(x, y)$	$[a, b]$	y_0	n	$f(x, y)$	$[a, b]$	y_0
9	$-\frac{1+xy}{x^2}$	$[1; 2]$	1	24	$\frac{y}{x} + x \cos x$	$[1; 2]$	1
10	$\frac{x}{3} + 2y$	$[0; 1]$	0	25	$2x(x^2 + y)$	$[0; 1]$	1
11	$\frac{2y}{x \ln x} + \frac{1}{x}$	$[2; 4]$	0	26	$\frac{3x^2 e^{-x} - (x+1)y}{x}$	$[1; 2]$	1
12	$\frac{x^2 + y}{x}$	$[1; 2]$	1	27	$2e^x - y$	$[0; 1]$	1
13	$\frac{2y + x - 4 \ln x}{x}$	$[1; 3]$	1	28	$\frac{3y - x^2}{x}$	$[1; 2]$	1
14	$\cos(x + y)$	$[0; \pi]$	0	29	$\frac{1}{x + 2y}$	$[0; 2]$	-1
15	$\frac{y}{3x - y^2}$	$[1; 3]$	1	30	$\frac{x}{y} + \frac{xy}{x^2 - 1}$	$[2; 3]$	1

Решите задачу Коши

1. методом последовательных приближений (Пикара);
2. методом Эйлера;
3. одним из усовершенствованных методов Эйлера;
4. по формулам Рунге–Кутты;
5. методом Адамса.
6. Сравните с точным решением, полученным методами дифференциальных уравнений, или сравните результаты, построив графики полученных решений. Оцените погрешность.
7. Методом Рунге–Ромберга вычислите значение $y(x)$ в правом конце отрезка с заданной точностью 10^{-5} .

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Отрезок $[a, b]$ разбейте равномерно на $N \geq 10$ частей. Выведите на экран полученный результат в виде таблицы с полями — узлы, значения численного решения в узлах, значения точного решения в узлах или в виде графиков численного и точного решений, построенных в одной координатной плоскости.

Для успешной сдачи задания преподаватель может потребовать от студента:

- 1) сформулировать постановку задачи;
- 2) определить входные данные, выходные данные, их тип;

- 3) дать основные определения, относящиеся к изучаемому разделу;
- 4) изложить суть используемого метода;
- 5) пояснить алгоритм своей программы;
- 6) дать описание встроенных функций и подключаемых библиотек, используемых в программе.

6.3. Численное решение краевых задач

Дана линейная краевая задача

$$\begin{cases} u'' + p(x)u' + q(x)u = f(x), & x \in [a, b], \\ \alpha u'(a) + \beta u(a) = A, \\ \delta u'(b) + \gamma u(b) = B, \end{cases}$$

где функция и краевые условия заданы в таблице 2, а n — номер Вашего варианта.

Таблица 2

n	$p(x)$	$q(x)$	$f(x)$	a	b	α	β	γ	δ	A	B
1	$-\frac{x}{1-x}$	$\frac{1}{x-1}$	$(x-1)e^x$	-1	0	1	-1	-3	2	1	-2
2	$\frac{2}{x(x-1)}$	$\frac{2}{x^2(x-1)}$	$\frac{xe^x}{x-1}$	-2	-1	11	-5	1	0	0	-3
3	$-\frac{x+3}{x}$	$-\frac{2x+3}{x}$	x^2	1	2	1	-3	2	1	1	-1
4	$\frac{4x+1}{2x}$	$-\frac{2x+1}{2x}$	$\frac{e^{-x}}{2x}$	1	2	2	0	1	0	1	1
5	$\frac{2x}{x-1}$	$\frac{2x}{(x-1)^2}$	$(x-1)^2$	-1	0	0	-2	3	1	1	-1
6	x^3	$-x$	$\frac{6}{x^4} - \frac{3}{x}$	1	2	0	2	3	1	1	0.5
7	$\frac{4x-2}{x(x-1)}$	$-\frac{2}{x(x-1)}$	$\frac{e^{-x}}{x(x-1)}$	2	3	1	-5	1	1	1	-0.5
8	$-\frac{2(x+1)}{x}$	$\frac{2(x+1)}{x^2}$	$2xe^{2x}$	-2	-1	1	-5	5	3	1	3
9	$-\sqrt[3]{x} + 1$	$-\sqrt[3]{x^3}$	x	1	2	0	-6	1	0	1	1
10	$-\frac{x+4}{2x}$	$-\frac{x+4}{2x^2}$	$-\frac{x^2}{2}$	1	2	1	-3	1	2	-1	0.5
11	\sqrt{x}	$-\sqrt[3]{x^2}$	$x-2$	1	2	1	-6	1	0	1	1
12	$\frac{2}{\sqrt{x}}$	$-x$	$\frac{1}{x}$	1	2	1	-5	2	1	-1	4
13	$\sin x$	$\sin 2x$	$2 \cos x$	$-\frac{\pi}{3}$	$-\frac{\pi}{6}$	1	2	0	1	-1	-2
14	0	$-\operatorname{ch} x$	e^x	0	1	1	3	1	1	$-e$	$-3e$

n	$p(x)$	$q(x)$	$f(x)$	a	b	α	β	γ	δ	A	B
15	x^2	$-x$	$\frac{6-3x^2}{x^4}$	1	2	3	-8	1	3	1	0.5
16	3	$-\frac{2}{(x+1)^2}$	$-\frac{1}{(1+x)^3}$	0	1	0	1	0	1	0	0
17	x^3	$\frac{1-x}{x}$	$\frac{5}{x^2}$	1	2	1	-0.5	3	1	1.25	0.5
18	$\text{sh } 2x$	$-7e^{-x^2}$	$\text{ch } 2x$	0	2	1	-3	e	2	1	-1
19	$\frac{2-4x}{x}$	$-\frac{4(x-1)}{x}$	$\frac{1}{x}$	2	3	0	-1	1	1	1	1
20	$\frac{2-4x}{x-1}$	$\frac{4x}{1-x}$	$\frac{6}{1-x}$	2	3	-4	1	1	1	-1	1
21	$\frac{2}{x}$	-1	$\frac{1}{x^3}$	1	2	1	-6	1	3	2	0.5
22	$2-x^2$	$3x+1$	$4x^3$	-2	-1	1	0	8	0	-10	1
23	$2\cos x$	$\sin x$	2	-1	0	3	-6	1	0	1	3
24	$\frac{1}{x^2+1}$	$-\frac{x^2}{x^2+1}$	-3	-1	0	1	-10	2	1	-1	-12
25	e^{-x}	xe^{-x}	1	-1	0	0	-2	4	1	1	-1
26	$2x^3$	$-x$	$-x^2$	0	2	-1	8	1	0	-3	1
27	$\frac{1}{x}$	$-\frac{1}{x^2}$	$\frac{2}{x^3}$	1	2	1	-2	1	0	1	$4\ln 2$
28	$\frac{2}{x}$	x	$\frac{1}{x^2}$	-2	-1	1	0	1	0	2	0
29	$\text{sh } x$	$-e^{-2x}$	e^{2x}	0	1	1	-3	e	2	0	-1
30	$\frac{x}{1+x^2}$	$-\frac{1}{1+x^2}$	$\frac{x^2}{(1+x^2)^2}$	-1	0	1	-2	1	1	1	0

I. Решите краевую задачу, используя

- 1) метод стрельбы для линейной краевой задачи;
- 2) метод коллокации;
- 3) метод Галеркина;
- 4) разностный метод.

II. Сравните результаты, построив графики полученных решений. Оцените погрешность.

III. Используя разностный метод, решите краевую задачу для уравнения эллиптического типа:

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x, y), & (x, y) \in G, \\ u|_{\Gamma} = \varphi(x, y), \end{cases}$$

где $G = [0, a] \times [0, a]$, Γ — граница области G , а соответствующие функции и параметры заданы в таблице, n — номер Вашего варианта.

n	$f(x, y)$	a	$\varphi(x, y)$
1	$2x^2 + y^2$	2	$3x - 5y + 2$
2	$x^2 - y$	3	$(x - y)^2 + 2x$
3	e^{x+y}	1	$3e^x + y^2(1 - x)$
4	$\cos(x + \pi y)$	$\pi/2$	$3 - 2(1 - y) + 2x/\pi$
5	$\sin x - 2y$	$\pi/2$	$3 - 2(1 - y) + \cos 2x$
6	$\operatorname{ch} x - \operatorname{sh} y$	1	$3x + (2 - y)\operatorname{sh} x + 1$
7	$\sqrt{x^2 + y^2}$	1	$(1 - y)y + 2x\sqrt{y + x}$
8	$x^3 - 2(x + 1)y^2$	2	$4x(y - 1) - y(x + 1) + 2$
9	$x - \operatorname{sh} y$	2	$x - (2 - y)\operatorname{ch} x$
10	$(x - 1)^3 - (1 - y)^2$	1	$x - (1 - y)(x - 2)$
11	$\sqrt[3]{x^2} - \sqrt{1 + y}$	1	$2x - 2y(x - 1)$
12	$x^2 - y^2$	1	$2(x - 1)^2y + (3 - y)x$
13	$\operatorname{tg} x - 2\cos y$	$\pi/4$	$\sin 2x - \sin^2 y$
14	$3(x^2 + y^2)$	1	$x^3y^3 + 1 - (x^3 - 1)(y^3 - 1)$
15	0	1	$e^x \cos y$
16	0	π	$e^{\frac{y}{\pi}} \sin x$
17	$-y$	$\pi/2$	$\frac{2}{\pi}x \cos y$
18	$-y$	$\pi/2$	$\frac{2}{\pi}y \sin x$
19	$-2y$	$\pi/2$	$\cos x \cos y$
20	$2e^{-x} \cos x \sin y - 3y$	$\pi/2$	$e^{-x} \cos x \cos y$
21	$2e^{-y} \sin x \cos y - 3y$	$\pi/2$	$e^{-y} \cos x \cos y$
22	$-2e^{-(x-y)} \sin(x + y) - 4y$	$\pi/2$	$e^{-(x+y)} \cos x \cos y$
23	$-2\sin(x - y)$	$\pi/2$	$-\sin(x + y)$
24	$-5\sin x \sin 2y$	$\pi/2$	$\sin x \sin 2y$
25	$12x$	1	$3x^2 - y$
26	0	1	$xy + 1 + (x - 1)(y - 1)$
27	xe^y	1	xe^y
28	$e^{xy}(x^2 + y^2)$	1	$e^{(x+y)}$

n	$f(x, y)$	a	$\varphi(x, y)$
29	$\sqrt[3]{x+y}$	2	$\sqrt[3]{x}(y-2)^3 - \sqrt[3]{(x-2)^2}y$
30	x^2e^{x-y}	1	$2ye^x - x^2e^y$

7. Используя разностный метод, решите краевую задачу для уравнения теплопроводности:

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + g(x, t), & (x, t) \in G, \\ u(x, 0) = f(x), & x \in [0, l], \\ u(0, t) = \mu_1(t), \\ u(l, t) = \mu_2(t), & t \in [0, T], \end{cases}$$

где $G = [0, l] \times [0, T]$, соответствующие функции и параметры заданы в таблице, а n — номер Вашего варианта.

n	$g(t, x)$	$f(x)$	$\mu_1(t)$	$\mu_2(t)$	l	T
1	$\ln(tx+1)$	$x-1$	$t-1$	1	2	2
2	$x^3\sqrt{t}$	x^3	$-t$	t^2-8t+1	1	2
3	$\sqrt{tx+1}$	0	t^2	$-t$	1	1
4	xe^t	$-x$	$\operatorname{sh}(t)$	$-e$	e	e
5	$2x^2(1-t)-t(1-x^2)$	$1-x$	$1-t$	t	1	1
6	t^2-e^x	$x+1$	$(t-1)^2$	e^t+1	1	2
7	$12\cos 4t\cos 2x$	$\cos x+\pi-2x$	$\cos \frac{t}{2}+\pi$	0	$\pi/2$	π
8	$\sin x^2+\cos t$	$\sin(\sin x)$	$\sin(2t)$	$-\cos t+\sin 1+1$	$\pi/2$	π
9	t^2-2x+1	$-x$	$(t-3)t^3$	$-t^2-2$	2	2
10	t^3-7x	$-x^2+1$	$\frac{1}{t^2+1}$	$t-3$	2	3
11	$(xt)^2-10t$	$x-4x^2$	$t(t-1)$	-3	1	2
12	$-\operatorname{sh}(t-7)$	$-e^{x-1}$	$-e^{-1}$	$-e^{t+1}$	2	2
13	$x(xt+1)$	0	t^2	t	1	2
14	e^{t-x+1}	-1	e^t-2	$-e^{-2t}$	2	2
15	$e^{-t}\cos 2x$	$\sin 2x$	0	$\sin t$	π	π
16	x^2-3t	x	t	$2t+2$	2	2
17	$2x^2+t^3$	$x+1$	$(t-1)^2$	t^4+3	2	1

n	$g(t, x)$	$f(x)$	$\mu_1(t)$	$\mu_2(t)$	l	T
18	$\operatorname{sh} x - 2t$	e^x	$t^2 + 1$	e^{t+2}	2	1
19	xe^{-t}	$-x$	$\operatorname{sh} t$	$-e$	e	e
20	$\sin x - \cos 2t$	$\cos x$	1	$\sin t - 1$	π	$\pi/2$
21	$\operatorname{ch} x + \operatorname{sh} t$	e^{-x}	$\operatorname{sh} t + 1$	e^{t^2-t-1}	1	e
22	$-2x^2 + t^3$	$x + 4$	$(t + 2)^2$	$\frac{t}{2} + 6$	2	1
23	$xt - 3 + t$	$x - 2$	$-2 - t$	t^3	2	1
24	$\sin(x - 2t)$	$\cos 2x$	$\cos t$	$\sin t + 1$	π	$\pi/2$
25	$\cos^2 x + 2 \cos t$	$\sin x$	$\sin t$	1	$\pi/2$	$\pi/2$
26	$x - \cos^2 t$	$2x$	0	$\sin t + 2$	1	$\pi/4$
27	$\sqrt{x} - 2t^2 + 1$	x	t	$1 - t$	1	2
28	$x^3 - \sqrt{2 - t}$	$4x + 1$	$t + 1$	$t + 5$	1	2
29	$x - 2t$	$-x$	$\frac{t}{2}$	$-t - 1$	1	2
30	e^{x+t}	$\operatorname{sh} x$	$e^t - 1$	$-\operatorname{ch} t + 1 + \operatorname{sh} 2$	2	2

8. Используя разностный метод, решите краевую задачу для уравнения колебаний:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + g(x, t), & (x, t) \in G, \\ u(x, 0) = \varphi(x), & x \in [0, 1], \\ u'_t(x, 0) = \psi(x), & x \in [0, 1], \\ u(0, t) = \mu_1(t), & t \in [0, T], \\ u(1, t) = \mu_2(t), & t \in [0, T], \end{cases}$$

где $G = [0, 1] \times [0, T]$, n — номер Вашего варианта, а функции и параметры заданы в таблице.

n	$g(t, x)$	$\varphi(x)$	$\psi(x)$	$\mu_1(t)$	$\mu_2(t)$	T
1	$-x \cos t$	$2x - \cos x$	$1 - \sin x$	$t - \cos t$	$1 + t - \cos(t-1) + \cos t$	$\frac{\pi}{2}$
2	$8x \cos t \sin t$	$-2x$	$4 - 4x$	$4t$	$2t - 2 - \sin(2t)$	$\frac{\pi}{2}$
3	$2x^2 \operatorname{sh} t$	$2 \operatorname{sh} x$	$2x^2 + 4$	$4 \operatorname{sh} t$	$\operatorname{sh}(t+1) - \operatorname{sh}(t-1) + 6 \operatorname{sh} t$	1
4	$3 \operatorname{ch} t - \operatorname{sh} x$	3	$\operatorname{ch} x$	$\operatorname{sh} t + 3 \operatorname{ch} t$	$\operatorname{sh}(t-1) + 3 \operatorname{ch} t + \operatorname{sh} 1$	2
5	$-xt^2$	$x^2 + \frac{1}{60}x^5$	$-2x$	t^2	$\frac{7}{6}t^2 - 2t + \frac{61}{60}$	3
6	$xt + 1$	$\sin x - \frac{1}{2}x^2$	$\cos x - \frac{1}{6}x^3$	$\sin t$	$\sin(t+1) - \frac{1}{6}t - \frac{1}{2}$	1
7	$x(t-2)$	$\sin x - x + \frac{1}{3}x^3$	$\cos x + 1 - \frac{1}{6}x^3$	$\sin t + t$	$\sin(t+1) + \frac{5}{6}t - \frac{2}{3}$	1
8	$xt + 2$	$-x^2$	$2 - \frac{1}{6}x^3$	$2t$	$\frac{11}{6}t - 1$	1
9	$\cos(t-x)$	$-\frac{1}{2}x(2 + \sin x)$	$3 + \frac{1}{2}x \cos x$	$3t$	$-1 + 3t + \frac{1}{2}\sin(t-1)$	$\frac{\pi}{3}$
10	$\sin x + 3 \cos t$	$\cos x + \sin x - 3$	$-\sin x$	$-2 \cos t$	$\cos(t+1) - 3 \cos t + \sin 1$	$\frac{\pi}{3}$
11	$e^{-t} \sin x$	$e^x + \frac{1}{2} \sin x$	$e^x - \frac{1}{2} \sin x$	e^t	$e^{t+1} + \frac{1}{2}e^{-t} \sin 1$	2
12	$4e^{-t}x^2$	$e^x - x + 4x^2 + 8$	$e^x - 7 - 4x^2$	$e^t + t + 8e^{-t}$	$e^{t+1} + t - 1 + 12e^{-t}$	2
13	$e^x(t+1)$	$-x$	1	$e^t - 1$	$-et + e^{t+1} - e + t - 1$	1
14	$\operatorname{ch}(2t) + x^3$	$\frac{1}{4} - \frac{1}{20}x^5$	2	$2t + \frac{1}{4}\operatorname{ch}(2t)$	$2t + \frac{1}{4}\operatorname{ch}(2t) - \frac{1}{20}$	2
15	$2 \operatorname{sh} x + t - 1$	$-\operatorname{sh} x - x + \frac{1}{2}x^2$	$\operatorname{ch} x + 1 - \frac{1}{2}x^2$	$\operatorname{sh} t + t$	$\operatorname{sh}(t+1) + \frac{1}{2}t - \frac{1}{2} - 2 \operatorname{sh} 1$	2

n	$g(t, x)$	$\varphi(x)$	$\psi(x)$	$\mu_1(t)$	$\mu_2(t)$	T
16	$6x^2t^3 + 2$	x^2	$-\frac{1}{10}x^6$	$2t^2$	$2t^2 - \frac{1}{10}t + 1 - \frac{1}{2}t^3$	2
17	$4x^2 - 3t$	$-\frac{1}{3}x^4$	$2 + \frac{3}{2}x^2$	$2t$	$\frac{7}{2}t - \frac{1}{3}$	3
18	$58e^{-3t} \cos(7x)$	$2e^{3x} + \cos(7x)$	$-3 \cos(7x)$	$e^{3t} + 2e^{-3t}$	$e^{3+3t} + e^{3-3t} + e^{-3t} \cos(7)$	3
19	$8e^t \sin x$	$4 \sin x$	$2 + 4 \sin x$	$2t$	$2t + 4 \sin 1e^t$	π
20	$8 \cos t \cos x$	$-4x \sin x$	0	0	$2 \sin(t-1) - 2 \sin(t+1)$	$\pi/2$
21	$x - 32t^2$	$2x - \frac{1}{6}x^3$	0	$-\frac{8}{3}t^4$	$\frac{11}{6} - \frac{8}{3}t^4$	3
22	$x^2 - t^2$	$-2x - \frac{1}{12}x^4$	0	$-\frac{1}{12}t^4$	$-\frac{25}{12} - \frac{1}{12}t^4$	1
23	$3t^2x - 2x^2t$	$-\frac{1}{20}x^5$	$4x + \frac{1}{6}x^4$	0	$-\frac{1}{2}t^2 + \frac{25}{6}t - \frac{1}{20}$	2
24	$\operatorname{sh} x + t^2$	$-\operatorname{sh} x - 2x$	0	$\frac{1}{12}t^4$	$-2 - \operatorname{sh} 1 + \frac{1}{12}t^4$	2
25	$3x - \operatorname{sh} t$	$-2 \operatorname{sh} x - \frac{1}{2}x^3$	-1	$-\operatorname{sh} t$	$-\operatorname{sh}(t+1) + \operatorname{sh}(t-1) - \frac{1}{2} - \operatorname{sh} t$	1
26	$\sin(x+2t)$	$\frac{2}{3} \sin x$	$\frac{1}{3} \cos x$	$\sin t - \frac{1}{3} \sin(2t)$	$\sin(t+1) - \frac{1}{3} \sin(2t+1)$	π
27	$\cos x + 2t \sin x$	$\cos x$	$2 + 2 \sin x$	$2t + 1$	$2t + \cos 1 + 2t \sin 1$	2
28	$e^{-t^2} \cos x$	$4 \sin x$	$2 + 4 \sin x$	$2t$	$2t + 4 \sin 1e^t$	π
29	$(x+2t)t + 2x^4$	$2x - \frac{1}{15}x^6$	$-\frac{1}{6}x^3$	$\frac{1}{6}t^4$	$\frac{1}{6}t^4 - \frac{1}{6}t + \frac{29}{15}$	2
30	$5x^3 + 8t^2$	$x^3 + x^2 - \frac{1}{4}x^5$	$3x^2 - 2x$	$\frac{2}{3}t^4 + t^3 + t^2$	$\frac{2}{3}t^4 + t^3 + 4t^2 + t + \frac{7}{4}$	1

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

Разбиение для множества G выбирайте, исходя из устойчивости используемой разностной схемы. Выведите на экран полученный результат в виде таблицы с полями — узлы, значения численного решения в узлах, значения решения, найденного встроенными функциями, в узлах или в виде графиков численного и точного решений, построенных в одной координатной плоскости.

Для успешной сдачи задания преподаватель может потребовать от студента:

- 1) сформулировать постановку задачи;
- 2) определить входные данные, выходные данные, их тип;
- 3) дать основные определения, относящиеся к изучаемому разделу;
- 4) изложить суть используемого метода;
- 5) пояснить алгоритм своей программы;
- 6) дать описание встроенных функций и подключаемых библиотек, используемых в программе.

6.4. Примерные темы домашних заданий

Домашнее задание состоит в самостоятельном изучении выбранного метода, написании программы и подготовке отчета (в печатном виде). Отчет должен содержать: постановку задачи, описание используемого метода, код программы, результаты ее работы на тестовых примерах, список литературы.

1. Семейство схем Эйлера второго порядка точности (случаи, отличные от рассмотренных на лекциях и в лабораторных работах).
2. Метод Рунге–Кутты порядка не равного четырем.
3. Метод конечных элементов.
4. Метод граничных элементов.
5. Методы оптимизации:
 - Метод покоординатного спуска;
 - Методы случайного поиска (без обучения и с обучением);
 - Градиентные методы;
 - Метод Ньютона;
 - Прямой симплекс-метод;
 - Модифицированный симплекс-метод;
 - Метод искусственного базиса;
 - Метод возможных направлений;
 - Метод Келли (или секущих плоскостей);
 - Метод ветвей и границ;
 - Методы штрафа.
6. Решение краевых задач с помощью сплайнов.
7. Решение нелинейной краевой задачи методом стрельбы.
8. Решение задачи Коши с помощью теории сплайнов.
9. Решение интегральных уравнений.
10. Задача приближения неявных функций.
11. Решение систем дифференциальных уравнений первого порядка методами Эйлера.
12. Решение систем дифференциальных уравнений первого порядка методом Рунге–Кутты.
13. Решение систем дифференциальных уравнений.
14. Многошаговые разностные методы решения задачи Коши.
15. Многошаговые разностные методы решения краевой задачи.

Список рекомендуемой литературы

1. Банников А. С., Ким И. Г., Латыпова Н. В. Численные методы. Ч. 1. Ижевск: Изд-во «Удм. ун-т», 2018. 78 с.
2. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: БИНОМ. Лаборатория знаний, 2008. 636 с.
3. Бахвалов Н. С., Лапин А. В., Чижонков Е. В. Численные методы в задачах и упражнениях. М.: БИНОМ. Лаборатория знаний, 2013. 240 с.
4. Березин И. С., Жидков Н. П. Методы вычислений. Т. 1. М.: ФИЗМАТЛИТ, 1962. 464 с.
5. Вержбицкий В. М. Численные методы (математический анализ и обыкновенные дифференциальные уравнения). М.: ОНИКС 21 век, 2005. 400 с.
6. Воробьёва Г. Н., Данилова А. Н. Практикум по вычислительной математике. М.: Высшая школа, 1990. 208 с.
7. Завьялов Ю. С., Квасов Б. И., Мирошниченко В. Л. Методы сплайн-функций. М.: Наука, 1980. 352 с.
8. Каханер Д., Моулер К., Нэш С. Численные методы и программное обеспечение. М.: Мир, 2001. 575 с.
9. Латыпова Н. В. Приложения сплайнов. Ижевск: Изд-во «Удм. ун-т», 2005. 52 с.
10. Латыпова Н. В. Интегралы, зависящие от параметра. Ижевск: Изд-во «Удм. ун-т», 2007. 57 с.
11. Макарьянц Г. М., Прокофьев А. Б. Основы метода конечных элементов. Самара: Изд-во СГАУ, 2013. 80 с.
12. Махмутов М. М. Лекции по численным методам. М.; Ижевск: НИЦ «Регулярная и хаотическая динамика», 2007. 238 с.
13. Рябенский В. С. Введение в вычислительную математику. М.: ФИЗМАТЛИТ, 2008. 288 с.

14. Самарский А. А. Введение в численные методы. СПб.: Лань, 2009. 288 с.
15. Самарский А. А., Гулин А. В. Численные методы. М.: Наука, 1989. 432 с.
16. Образовательный математический сайт exponenta.ru.
URL: <http://www.exponenta.ru>
17. EqWorld мир математических уравнений.
URL: <http://eqworld.ipmnet.ru/ru/library/mathematics/numerics.htm>
18. Научно-образовательный Интернет-ресурс НИВЦ МГУ по численному анализу. URL: <http://num-anal.srcc.msu.su>
19. Numerical Recipes Online.
URL: http://www.numerical=recipes.com/nronline_switcher.html

Рекомендации по выполнению лабораторных работ

Wolfram — весьма общий мультипарадигмальный язык программирования, разработанный компанией Wolfram Research, который служит основным связующим языком для системы Mathematica. Он был спроектирован как максимально универсальный язык, с акцентом на символьные вычисления, функциональное и логическое программирование.

Язык является достаточно большим и касается многих, часто специализированных, сфер знаний. Например, он обладает встроенными функциями для создания и приведения в действие машины Тьюринга, создания графики и аудио, анализа трехмерных моделей и решения дифференциальных уравнений. В частности, имеется большое количество функций, в которых уже реализованы некоторые алгоритмы, рассматриваемые в данном пособии. Возникает естественный вопрос: зачем самостоятельно реализовывать те или иные алгоритмы, если они уже есть в языке? Но чтобы рационально использовать встроенные функции, их нужно понимать. Во-первых, функции и библиотеки могут иметь побочные эффекты или поведение, которые сложно предусмотреть (предвидеть) без понимания алгоритмов. Получив баг в таком случае, можно долго и упорно пытаться его поймать и исправить, когда этого можно было избежать. Во-вторых, различные инструменты и библиотеки часто нужно «настраивать» — указывать какие алгоритмы, структуры данных и технологии использовать внутри. Без элементарных знаний придется либо пользоваться значениями по умолчанию (порой неэффективными именно для вашей задачи), либо выбирать наугад.

В этом разделе мы рассмотрим «всего» две функции: NIntegrate и NDSolve, но каждая из них — большой универсальный фреймворк, содержащий внутри себя множество алгоритмов, с возможностью написания собственных расширений, и эффективное их использование — залог успешного решения задач, где в качестве промежуточных используются вычисления интегралов и/или решение дифференциальных уравнений.

NIntegrate на верхнем уровне использует метаалгоритмы, называемые *стратегиями* («integration strategies»), чтобы вычислить значение интеграла, удовлетворяющее заданным пользователем показателям точности (AccuracyGoal и PrecisionGoal). На уровне ниже стратегии используют алгоритмы интегрирования, так называемые *правила* инте-

грирования («integration rules»), которые (в большинстве своем) определяют значение интеграла в виде некоторой взвешенной суммы значений подынтегральной функции в отдельно взятых точках («sampling points»). Также дополнительно может использоваться предварительный символьный анализ («symbolic preprocessing») подынтегрального выражения, который упрощает дальнейшие вычисления, например, в случаях кусочно-заданных функций, или функций с определенной четностью. В одномерном случае могут быть определены сильно осциллирующие функции, для которых будут использоваться специальные алгоритмы.

Все имеющиеся стратегии NIntegrate могут быть разделены на две общие группы: детерминистические и вероятностные. Каждая группа стратегий может быть разбита на три категории: адаптивные, неадаптивные и специализированные стратегии. Первая категория стратегий производит дополнительные вычисления лишь в тех областях, на которые разбит интервал интегрирования, в которых получающаяся оценка погрешности превосходит заданную. Неадаптивные стратегии улучшают получающуюся оценку значения интеграла увеличением узлов по всем областям разбиения исходного интервала (количество разбиений контролируется опциями MaxRecursion и MinRecursion). Процесс интегрирования останавливается, когда достигается условие останова

$$globalError \leq globalIntegral \cdot 10^{-pg} \text{ or } globalError \leq 10^{-ag}$$

(*pg* — значение опции PrecisionGoal, по умолчанию значение опции равно WorkingPrecision/2, *ag* — значение опции AccuracyGoal, по умолчанию значение опции равно Infinity, что соответствует вычислениям с 7–8 верными знаками), либо когда все рекурсивные разбиения проведены, но заданной точности достигнуть не удалось (в этом случае полученный ответ сопровождается специальным сообщением с указанием найденного значения интеграла и полученной оценкой для достигнутой погрешности). Специализированные стратегии применяются для конкретных типов подынтегральных выражений, например, для произведений осциллирующих и неосциллирующих функций, или многомерных функций, обладающий определенной сферической симметрией.

При использовании стратегий «GlobalAdaptive» и «LocalAdaptive» можно дополнительно уточнить применяемые правила интегрирования и используемые этими правилами дополнительные параметры. Обе эти стратегии применимы как для одномерных, так и многомерных пра-

Группы стратегий NIntegrate

Стратегии	Детерминистические	Вероятностные
адаптивные	«GlobalAdaptive» «LocalAdaptive»	«AdaptiveMonteCarlo» «AdaptiveQuasiMonteCarlo»
неадаптивные	«DoubleExponential» «Trapezoidal»	«MonteCarlo»
специализированные	«DuffyCoordinates» «Oscillatory» «PrincipalValue»	

вил интегрирования. Вызов NIntegrate при явном указании стратегий и правил в опции Method выглядит следующим образом:

`NIntegrate[f[x],{x,a,b},Method->{«стратегия»,Method->«правило»}]`

Встроенные правила интегрирования, используемые в стратегиях «GlobalAdaptive» и «LocalAdaptive»:

	Правила интегрирования
одномерные	«BooleRule» «ClenshawCurtisRule» «GaussBerntsenEspelidRule» «GaussKronrodRule» «LobattoKronrodRule» «LobattoPeanoRule» «MultiPanelRule» «NewtonCotesRule» «PattersonRule» «SimpsonThreeEightsRule» «TrapezoidalRule»
многомерные	«CartesianRule» «MultiDimensionalRule»

Пример численного интегрирования с указанием стратегии, правила интегрирования, дополнительных опций и сравнение с точным результатом:

```
res = NIntegrate[x^6, {x, 0, 1}
, Method -> {"LocalAdaptive"}
```

```

    , Method -> {"NewtonCotesRule"
    , "Points" -> 4
    , "Type" -> "Open"
    , "SymbolicProcessing" -> 0
    }
    , "SingularityHandler" -> None
  }
  , MaxRecursion -> 0]
exact = Integrate[x^6, {x, 0, 1}];
Abs[res - exact]
0.142857
2.77556*10^-16

```

В этом примере явно указывается тип стратегии — "LocalAdaptive", применяется правило интегрирования "NewtonCotesRule" — т.е. используются формулы интерполяционного типа (см. с. 15). Отрезок интегрирования разбивается узлами на равные части, концы отрезка в разбиение не входят, так как указывается открытый тип формул ("Type" -> "Open"). Вычисляются соответствующие коэффициенты C_i (см. (24)) для определения интеграла взвешенной суммой (см. (23)). Анализ подынтегрального выражения на наличие особых точек (сингулярностей) и возможные упрощения отключен ("SingularityHandler" -> None, "SymbolicProcessing" -> 0). Данная процедура проводится однократно, так как глубина рекурсии ограничена опцией MaxRecursion->0. Несмотря на это, итоговый результат отличается от точного всего на $2.77556 \cdot 10^{-16}$.

Метод прямоугольников

В Wolfram Language метод левых и правых прямоугольников не реализован в виду малого порядка сходимости методов, а метод средних прямоугольников, хоть и имеет тот же порядок точности, что и метод трапеций, уступает последнему в плане переиспользования уже вычисленных значений подынтегральной функции при рекурсивном разбиении исходного отрезка для достижения заданной точности. Поэтому приведем возможную реализацию метода левых прямоугольников, на основе которой можно с очевидными изменениями получить реализацию оставшихся методов. Для объединения нескольких действий в рамках алгоритма используется обертка Module, состоящая из двух частей:

списка локальных переменных и одного или нескольких выражений, объединенных в одно при помощи CompoundExpression («;»). Ответом является последняя строка (после которой символ «;» не ставится), либо выражение в Return (оператор выхода из процедуры).

```
ClearAll[leftRect, leftRectTol, f];
leftRect[f_, a_, b_, n_] :=
  Module[{h = (b - a)/n}
    , Return[h Total[f /@ Range[a, b - h, h]]]
  ];
leftRectTol[f_, a_, b_, tol_] :=
  Module[{h = (b - a)/2, ans1, ans2}
    , ans1 = h Total[f /@ Range[a, b - h, h]];
    h = h/2;
    ans2 = h Total[f /@ Range[a, b - h, h]];
    While[Abs[ans1 - ans2] > tol
      , ans1 = ans2;
        h = h/2;
        ans2 = h Total[f /@ Range[a, b - h, h]]
    ];
    Return[ans2]
  ];
f[x_] = Exp[x] Sin[2 x];
leftRect[f, 0., 1., 1000]
leftRectTol[f, 0., 1., 10^-3]
N[Integrate[f[x], {x, 0, 1}]]
NIntegrate[f[x], {x, 0, 1}]
1.34559
1.34622
1.34683
1.34683
```

Метод Гаусса

В Wolfram Language метод Гаусса представлен двумя модификациями: "GaussBerntsenEspelidRule" (основное отличие — нечетное количество точек разбиения и связанная с этим оценка погрешности) и "GaussKronrodRule" (к оптимальным точкам разбиения добавляются еще узлы, позволяющие повысить порядок метода). Нахождение

точек разбиения по известным формулам (нули многочлена Лежандра) можно пропустить, если обратиться к встроенным функциям вида `NIntegrate[NameRuleData]`, где `Name` — название правила интегрирования. Они возвращают три списка: координаты точек разбиения на стандартном отрезке $[0, 1]$, весовые коэффициенты (30) для вычисления интеграла в виде взвешенной суммы (28) и (если это определено правилом интегрирования) весовые коэффициенты для оценивания ошибки интегрирования взвешенной суммой вида (28). Поэтому для вычисления интеграла на произвольном отрезке $[a, b]$ необходимо произвести перенормировку координат с отрезка $[0, 1]$ на отрезок $[a, b]$ и домножить взвешенную сумму значений подынтегральной функции на длину отрезка $b - a$:

```
ClearAll["Global`*"];
f[x_] = Exp[x] Sin[2 x];
{a, b} = {-2, 4};
n = 9;
{abcs, weights, errweights} =
  NIntegrate[GaussRuleData[n, MachinePrecision];
xknots = Rescale[abcs, {0, 1}, {a, b}];
yknots = f /@ xknots;
res = (b - a) weights.yknots
NIntegrate[f[x], {x, a, b},
  Method -> {"GaussBerntsenEspelidRule", "Points" -> 4,
    "SymbolicProcessing" -> 0}, MaxRecursion -> 0]
13.9252
13.9252
```

Встроенные алгоритмы для решения дифференциальных уравнений представлены семействами функций `DSolve` (символьное решение), `NDSolve` (численное решение) и `ParametricNDSolve` (численное решение параметризованного семейства дифференциальных уравнений). Поддерживается решение

- обыкновенных дифференциальных уравнений,
`NDSolve[eqns,u,{x,xmin,xmax}]`
- дифференциальных уравнений в частных производных,
`NDSolve[eqns,u,{x,xmin,xmax},{y,ymin,ymax}]`

- систем дифференциально-алгебраических уравнений,
- дифференциальных уравнений с запаздыванием,
- интегро-дифференциальных уравнений,
- гибридных дифференциальных уравнений.

Для успешного нахождения численного решения дифференциальные уравнения должны содержать достаточное количество начальных или граничных условий. Как обычно, опция `AccuracyGoal` отвечает за абсолютную величину желаемой погрешности, а опция `PrecisionGoal` — за относительную. Поэтому для решений, близких к нулю, значение опции `AccuracyGoal` необходимо увеличивать, вплоть до `Infinity`. По умолчанию же обе опции равны значению `WorkingPrecision/2`. Чтобы удовлетворить заданные пределы погрешности, `Mathematica` может менять шаг интегрирования, если это поддерживается выбранным методом решения. Настройка выбора шага осуществляется опциями `StartingStepSize` (начальное значение шага), `MaxStepSize` (максимально допустимый размер шага), `MaxSteps` (максимальное количество шагов, после которого процесс интегрирования дифференциального уравнения останавливается), `MaxStepFraction` (определяет параметры рекурсивного разбиения отрезка интегрирования на более мелкие).

В процессе поиска решения `NDSolve` проходит несколько стадий, в зависимости от типа решаемой задачи. При использовании опции `Method` в виде `Method->{s1->m1,s2->m2,...}` можно явно указать, какой метод `mi` будет применяться на стадии `si`. Список возможных стадий состоит из

- `"TimeIntegration"` (интегрирование по независимой переменной системы дифференциальных уравнений);
- `"BoundaryValues"` (краевые задачи);
- `"DiscontinuityProcessing"` (анализ дифференциальных уравнений с разрывными частями);
- `"EquationSimplification"` (символьные преобразования для упрощения численных вычислений);
- `"IndexReduction"` (редукция (понижение порядка) систем дифференциально-алгебраических уравнений);

- "DAEInitialization" (проверка согласованности начальных условий для систем дифференциально-алгебраических уравнений);
- "PDEDiscretization" (дискретизация (построение сетки) для дифференциальных уравнений в частных производных).

Если явно указывается только метод интегрирования по независимой переменной, то указание стадии можно опускать, поэтому запись `Method->m1` эквивалентна записи `Method->{"TimeIntegration" -> m1}`. Список имеющихся семейств методов интегрирования, которые можно явно указать, включает в себя:

- "Adams" (предикт-корректорные семейства методов Адамса порядков от 1 до 12);
- "BDF" (неявные методы интегрирования в обратном времени порядков от 1 до 5);
- "ExplicitRungeKutta" (адаптивные семейства методов Рунге-Кутты порядков от 2 до 9);
- "IDA" (семейство неявных методов для систем дифференциально-алгебраических уравнений);
- "ImplicitRungeKutta" (семейство неявных методов Рунге-Кутты произвольного порядка).

При помощи конструкции `Method->{"controller", Method->{m1, m2, ...}}` можно контролировать иерархию вызываемых методов. Например, для использования методов Адамса необходимо получить первые несколько значений решения, обычно для этого используются методы Рунге-Кутты, но можно выбрать из дополнительного списка

- "ExplicitEuler" (прямой метод Эйлера);
- "ExplicitMidpoint" (метод средней точки);
- "LinearlyImplicitEuler" (неявный метод Эйлера);
- "LocallyExact" (численная аппроксимация для локально известного точного символического решения).

Реализация классического метода Рунге–Кутты

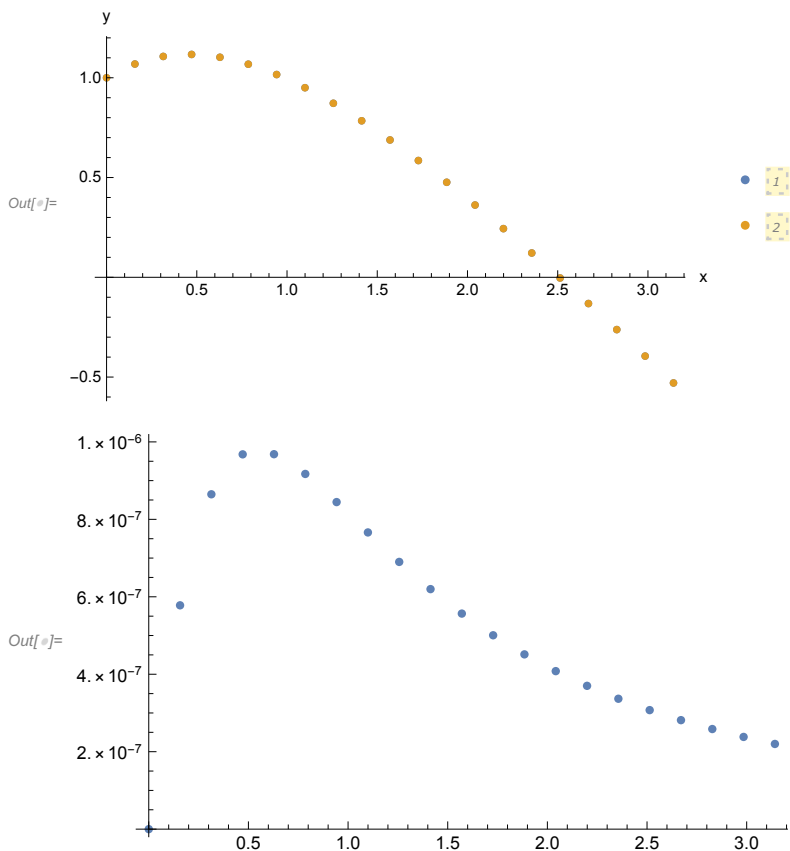
Для реализации классического метода Рунге–Кутты (42) воспользуемся Module. В переменных x и y будем накапливать очередные значения координат узлов и найденных значений решения в этих узлах. Иницилируем y начальным значением y_0 . В цикле До последовательно вычисляем промежуточные величины φ_i и очередное значение y_{k+1} . На выходе получаем двумерный массив. Транспонированием получаем координаты точек на плоскости (табличную функцию, являющуюся решением задачи Коши). Сравниваем полученное решение с решением, полученным NDSolve встроенным методом Рунге–Кутты такого же порядка (четвертого) с фиксированным шагом. Как видим, разница двух подходов составляет не более 10^{-6} .

```
ClearAll["Global`*"]
f[x_, y_] = Cos[x + y];
{a, b} = {0., \[Pi]};
y0 = 1.;
n = 20;
CRK[f_, {a_, b_}, y0_, n_] := Module[
  {h = (b - a)/n, x, y = ConstantArray[0, n + 1]
  , \[CurlyPhi]1, \[CurlyPhi]2, \[CurlyPhi]3, \[CurlyPhi]4},
  x = Range[a, b, h];
  y[[1]] = y0;
  Do[
    \[CurlyPhi]1 = h f[x[[k]], y[[k]]];
    \[CurlyPhi]2 = h f[x[[k]] + h/2, y[[k]] + \[CurlyPhi]1/2];
    \[CurlyPhi]3 = h f[x[[k]] + h/2, y[[k]] + \[CurlyPhi]2/2];
    \[CurlyPhi]4 = h f[x[[k]] + h, y[[k]] + \[CurlyPhi]3];
    y[[k + 1]] =
      y[[k]] + (\[CurlyPhi]1 + 2 \[CurlyPhi]2 +
        2 \[CurlyPhi]3 + \[CurlyPhi]4)/6
    , {k, n}];
  Transpose[{x, y}]
]
ndsol = NDSolveValue[
  {y'[x] == f[x, y[x]], y[a] == y0}, y, {x, a, b}
  , Method -> {"FixedStep"
  , Method -> {"ExplicitRungeKutta"}}
```

```

, "DifferenceOrder" -> 4}},
StartingStepSize -> (b - a)/n];
ndsolcoords = First@ndsol["Coordinates"];
ndsolvalues = ndsol["ValuesOnGrid"];
crksol = CRK[f, {a, b}, y0, n];
{crksolcoords, crksolvalues} = Transpose@crksol;
ListPlot[{ndsol, crksol}
, PlotLegends -> Automatic
, AxesLabel -> {"x", "y"}]
ListPlot[ndsolvalues - crksolvalues, DataRange -> {a, b}]

```

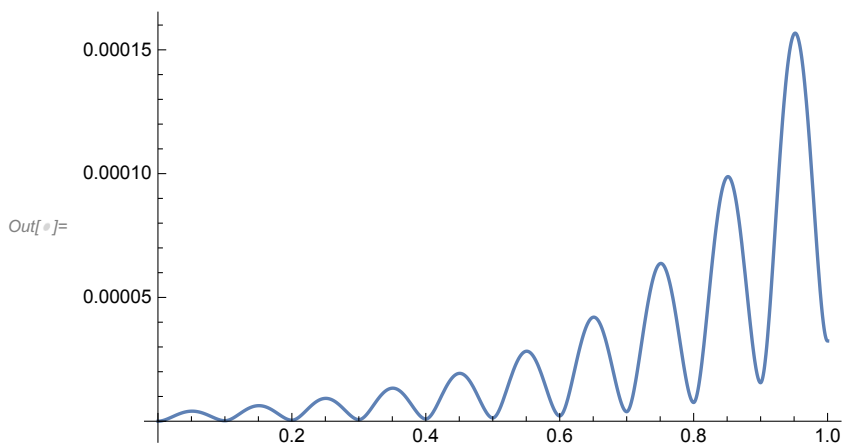
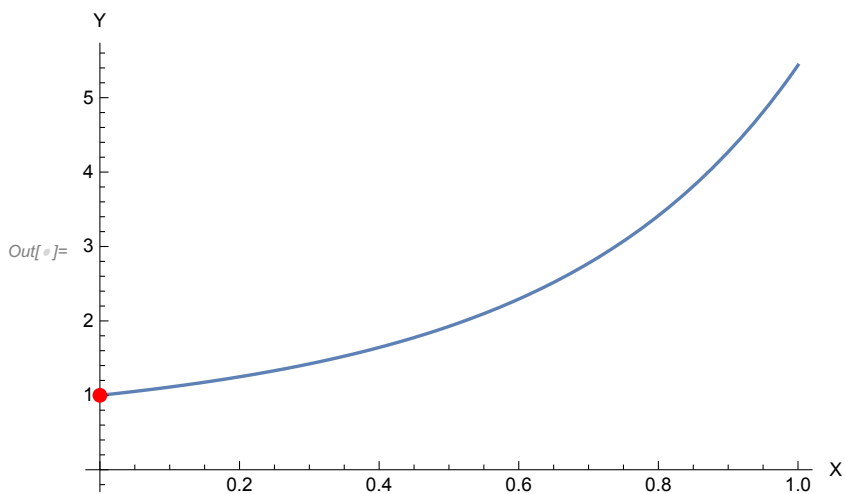


Реализация метода Рунге–Кутты по коэффициентам таблицы Бутчера

Mathematica поддерживает пользовательские расширения встроенных методов. В частности, поскольку любой метод из семейства методов Рунге–Кутты однозначно определяется своими наборами коэффициентов (таблицей Бутчера), то достаточно просто переопределить опцию "Coefficients" метода "ExplicitRungeKutta". Для этого нужно указать последовательно порядок метода (опция "DifferenceOrder"), матрицу β_{ij} (переменная `crkamat`) и векторы α_i (переменная `crkbvec`) и A_i (переменная `crkcvec`) (см. (40), (41)), которые будут вычислены при вызове функции `ClassicalRungeKuttaCoefficients`. Так как решаемое уравнение имеет точное (символьное) решение, мы можем сравнить его с полученным численным решением. Видим, что погрешность нарастает к концу интервала, но не превосходит 0.0002.

```
ClearAll["Global`*"]
f[x_, y_] = 2 x y + Exp[x^2];
{a, b} = {0, 1};
y0 = 1;
sys = {y'[x] == f[x, y[x]]};
ics = {y[a] == y0};
crkamat = {{1/2}, {0, 1/2}, {0, 0, 1}};
crkbvec = {1/6, 1/3, 1/3, 1/6};
crkcvec = {1/2, 1/2, 1};
ClassicalRungeKuttaCoefficients[4, p_] :=
  N[{crkamat, crkbvec, crkcvec}, p];
dsol = DSolveValue[{sys, ics}, y, {x, a, b}];
ndsol = NDSolveValue[{sys, ics}
  , y, {x, a, b},
  Method -> {"ExplicitRungeKutta"
    , "DifferenceOrder" -> 4
    , "Coefficients" -> ClassicalRungeKuttaCoefficients}
  , StartingStepSize -> 1/10];
Plot[ndsol[x], {x, a, b}
  , Epilog -> {Red, PointSize[Large], Point[{a, y0}]}
  , AxesOrigin -> {0, 0}
  , AxesLabel -> {"X", "Y"}
]
```

```
Plot[RealAbs[ndsol[x] - dsol[x]], {x, a, b},
, PlotRange -> All]
```



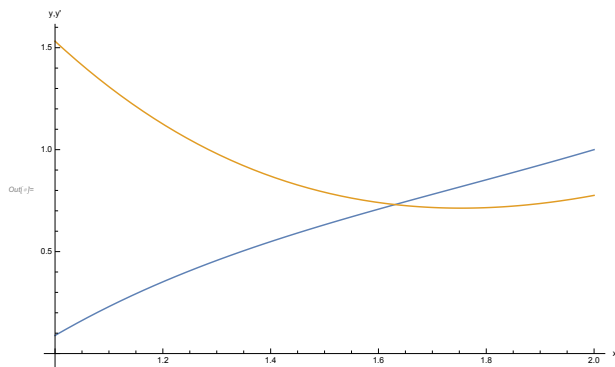
Метод стрельбы для краевых задач

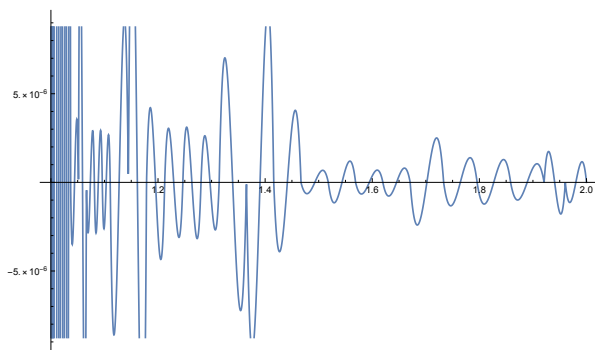
Для решения краевых задач в Mathematica предусмотрено два метода: "Shooting" и "Chasing". В качестве опций метода стрельбы можно

указать параметры, которые будут переданы для определения начального значения (см. (60)) в решатель FindRoot. Это "ImplicitSolver" (метод решения, который будет использован в FindRoot, по умолчанию — это метод Ньютона) и "MaxIterations" — максимальное количество итераций, которое будет использовано для нахождения ответа.

```
ClearAll["Global`*"]
p[x_] = Sqrt[x]; q[x_] = -Power[x^2, (3)^-1]; f[x_] = x - 2;
{a, b} = {1, 2}; {\[Alpha], \[Beta]} = {1, -6};
{\[Gamma], \[Delta]} = {1, 0}; {A, B} = {1, 1};
eqn = D[y[x], {x, 2}] + p[x] D[y[x], x] + q[x] y[x] - f[x] == 0;
bcs = {\[Alpha] Derivative[1][y][a] + \[Beta] y[a] == A
, \[Delta] Derivative[1][y][b] + \[Gamma] y[b] == B};
ndsol = y /. First@NDSolve[{eqn, bcs}
, y, {x, a, b}
, Method -> {"BoundaryValues" -> {"Shooting"
, "StartingInitialConditions" ->
{y[a] == 0, y'[a] == A/\[Alpha]}
, "ImplicitSolver" -> "Newton"
, "MaxIterations" -> 5}}
, AccuracyGoal -> 8
, WorkingPrecision -> 20];
Chop[(First /@ bcs /. y -> ndsol) - {A, B}, 8]
Plot[Evaluate[First@eqn /. y -> ndsol], {x, a, b}]

{0, 0}
```





Уравнения эллиптического типа

В последних версиях Mathematica появилась поддержка символьного решения уравнений эллиптического типа. Ответ чаще всего представляется бесконечным рядом (`Inactivate[Sum]`), переменные суммирования имеют вид `K[i]`. Поэтому для дальнейшего практического использования бесконечный ряд подменяется суммой конечного числа слагаемых. Данный пример также иллюстрирует геометрический язык Mathematica, позволяющий описывать произвольные области (здесь — прямоугольную) и задавать начальные условия на части границы области (используя `DirichletCondition`). Если условие должно выполняться в каждой точке границы, то вместо предиката можно использовать `True` (см. последующие примеры).

```
ClearAll["Global`*"];
f[x_, y_] = 2 x^2 + y^2;
a = 2;
\[CurlyPhi][x_, y_] = 3 x - 5 y + 2;
G = Rectangle[{0, 0}, {a, a}];
dG = RegionBoundary[G];
ndsol = NDSolveValue[{
  Laplacian[u[x, y], {x, y}] == -f[x, y],
  DirichletCondition[
    u[x, y] == \[CurlyPhi][x, y], {x, y} \[Element] dG
  ], u, {x, y} \[Element] G];
numplot = Plot3D[ndsol[x, y],
  {x, y} \[Element] G]
```



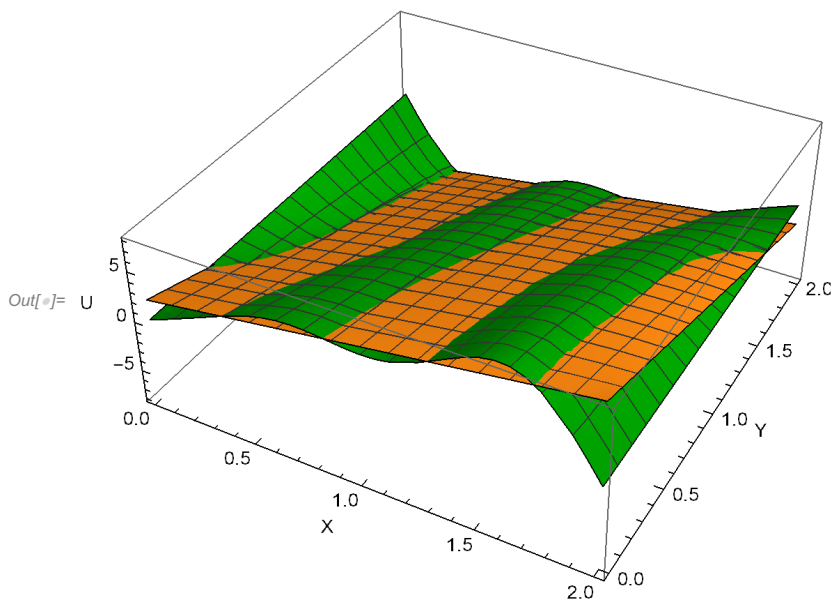
```

, AxesLabel -> {"X", "Y", "U"}];
dsol = DSolveValue[{Laplacian[u[x, y], {x, y}] == -f[x, y],
  DirichletCondition[u[x, y] == \[CurlyPhi][x, y], True]},
  u, {x, y} \[Element] G];
asol[n_] = dsol /. {\[Infinity] -> n}
Show[numplot,
  Plot3D[asol[4][x, y] // Activate // Evaluate
    , {x, y} \[Element] G
    , PlotRange -> All, PlotStyle -> Darker@Green],
  AxesLabel -> {"X", "Y", "U"}]

```

Out[]:= Function[{x, y},

$$\sum_{K[1]=1}^n \left(\frac{1}{\pi^5 K[1]^5} \right) 2 \sin\left[\left(\frac{1}{2}\right) \pi x K[1]\right] \left(\pi^4 \left(2 - 5y + (-1)^{K[1]} (-8 + 5y) \right) K[1]^4 + \right. \\
 \left. 4 \left(-8 + 8 (-1)^{K[1]} + \pi^2 \left(y^2 + (-1)^{1+K[1]} (8 + y^2) \right) K[1]^2 + \right. \right. \\
 \left. \left. 4 \operatorname{Csch}[\pi K[1]] \left(-2 \left(1 + (-1)^{K[1]} (-1 + \pi^2 K[1]^2) \right) \sinh\left[\left(\frac{1}{2}\right) \pi (-2 + y) K[1]\right] + \right. \right. \right. \\
 \left. \left. \left(2 - 2 (-1)^{K[1]} + (-1 + 3 (-1)^{K[1]}) \pi^2 K[1]^2 \right) \sinh\left[\left(\frac{1}{2}\right) \pi y K[1]\right] \right) \right) \left. \right) \left. \right)$$



Уравнения колебаний

Для численного решения уравнений в частных производных в Mathematica реализовано два метода: "MethodOfLines" и "FiniteElement". Для тонкой настройки можно указывать параметры построения расчетной сетки по фазовым переменным ("SpatialDiscretization"), такие как MinPoints, MaxPoints, MinStepSize и MaxStepSize.

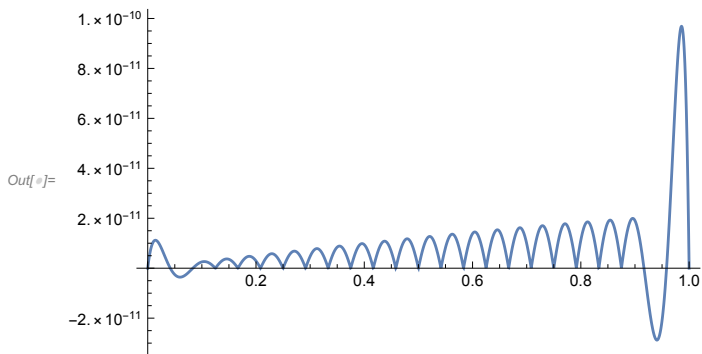
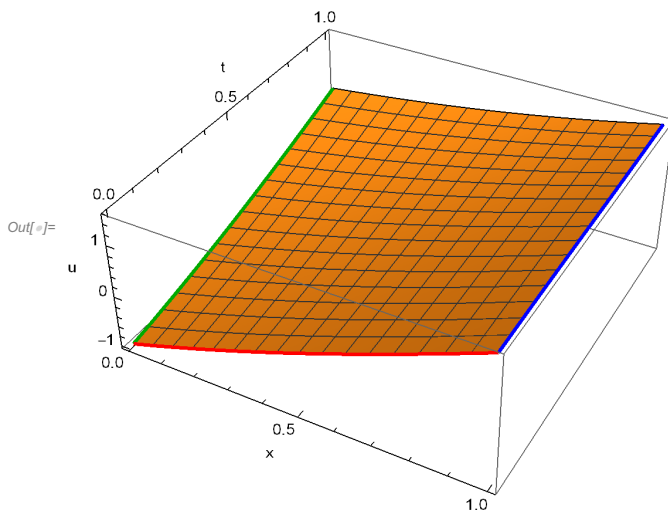
Как можно видеть из представленных иллюстраций, даже небольшое количество узлов (25 в данном примере) позволяет достигнуть точности 10^{-10} для начальных условий по производной, а подстановка найденного численного решения в исходное уравнение показывает достаточно малые отклонения (порядка нескольких процентов) только одной части границы (при $x = 0$).

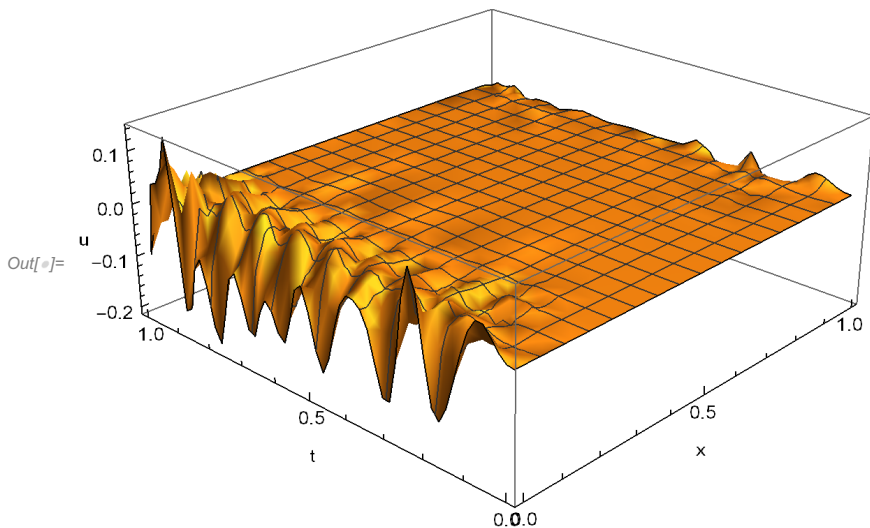
```
ClearAll["Global`*"]
T = 1;
g[x_, t_] = -x Cos[t];
\[CurlyPhi][x_] = 2 x - Cos[x];
\[Psi][x_] = 1 - Sin[x];
\[Mu]1[t_] = t - Cos[t];
\[Mu]2[t_] = 1 + t - Cos[t - 1] + Cos[t];
eqn = -D[u[x, t], {t, 2}] + D[u[x, t], {x, 2}] + g[x, t] == 0;
ics = {u[x, 0] == \[CurlyPhi][x],
  Derivative[0, 1][u][x, 0] == \[Psi][x]};
bcs = {u[0, t] == \[Mu]1[t], u[1, t] == \[Mu]2[t]};
ndsol = NDSolveValue[{eqn, ics, bcs}
  , u, {x, 0, 1}, {t, 0, T}
  , Method -> {"MethodOfLines"
    , "SpatialDiscretization" -> {"TensorProductGrid"
      , MinPoints -> 25}}];
Show[
  Plot3D[ndsol[x, t], {x, 0, 1}, {t, 0, T}]
  , ParametricPlot3D[{x, 0, \[CurlyPhi][x]}, {x, 0, 1}
    , PlotStyle -> Red]
  , ParametricPlot3D[{0, t, \[Mu]1[t]}, {t, 0, T}
    , PlotStyle -> Darker@Green]
  , ParametricPlot3D[{1, t, \[Mu]2[t]}, {t, 0, T}
    , PlotStyle -> Blue]
```

```

, SphericalRegion -> True
, AxesLabel -> {"x", "t", "u"}
]
Plot[Derivative[0, 1][ndsol][x, 0] - \[Psi][x], {x, 0, 1},
PlotRange -> All]
Plot3D[Evaluate[First@eqn /. u -> ndsol]
, {x, 0, 1}, {t, 0, T}, PlotRange -> All]

```





Уравнение теплопроводности

```

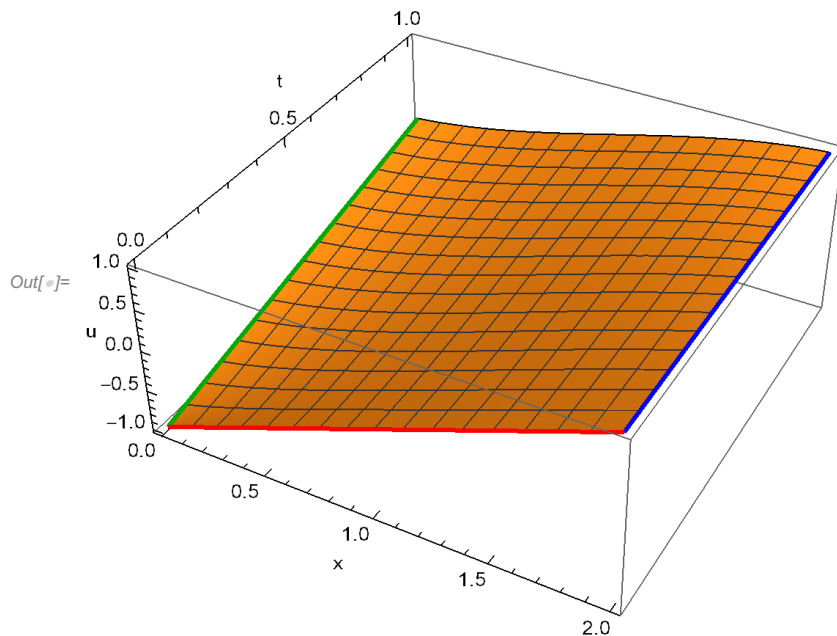
ClearAll["Global`*"]
l = 2;
T = 1;
g[x_, t_] = Log[t x + 1];
f[x_] = x - 1;
\[Mu]1[t_] = t - 1;
\[Mu]2[t_] = 1;
eqn = -D[u[x, t], t] + D[u[x, t], {x, 2}] + g[x, t] == 0;
ics = u[x, 0] == f[x];
bcs = {u[0, t] == \[Mu]1[t], u[l, t] == \[Mu]2[t]};
ndsol = NDSolveValue[{eqn, ics, bcs}
  , u, {x, 0, l}, {t, 0, T}
  , Method -> {"PDEDiscretization" -> {"MethodOfLines"
    , "SpatialDiscretization" -> "FiniteElement"}}];
Show[
  Plot3D[ndsol[x, t], {x, 0, l}, {t, 0, T}]
  , ParametricPlot3D[{x, 0, f[x]}, {x, 0, l}

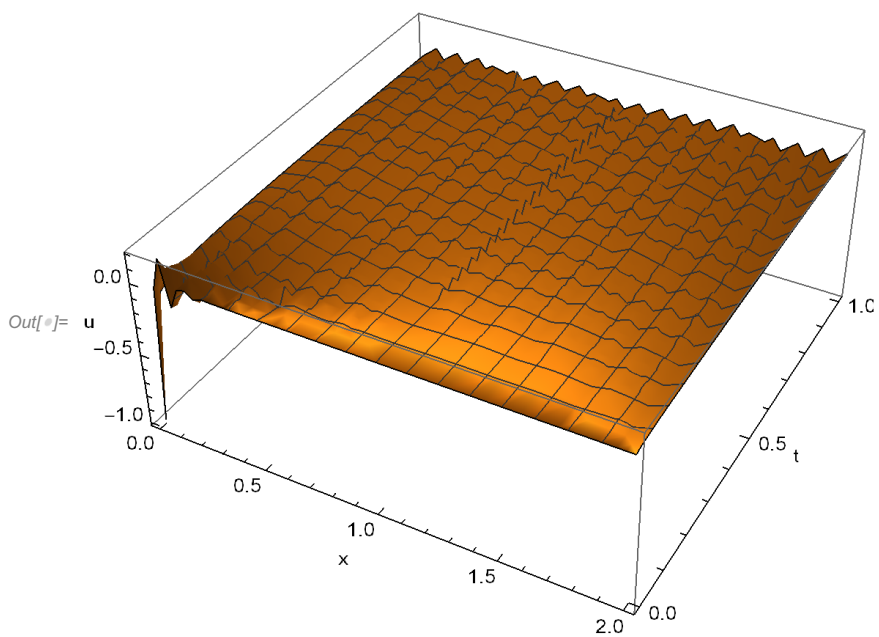
```

```

, PlotStyle -> Red]
, ParametricPlot3D[{0, t, \[Mu]1[t]}, {t, 0, T}
, PlotStyle -> Darker@Green]
, ParametricPlot3D[{1, t, \[Mu]2[t]}, {t, 0, T}
, PlotStyle -> Blue]
, SphericalRegion -> True
, AxesLabel -> {"x", "t", "u"}
]
Plot3D[Evaluate[First@eqn /. u -> ndsol]
, {x, 0, 1}, {t, 0, T}, PlotRange -> All
, SphericalRegion -> False
, AxesLabel -> {"x", "t", "u"}]]

```





Приложение

В качестве примера решения задачи рассмотрим задачу Коши: $y'' - y' - 2y = x$ с начальными условиями $y(0) = 0$ и $y'(0) = -1$. Соответствующая ей система имеет вид:

$$\begin{cases} y' = z, \\ z' = z + 2y + x, \end{cases}$$

с начальными условиями $y(0) = 0$, $z(0) = -1$.

Решение полученной системы разными методами представлено в таблицах П.1 и П.2 и на рис. П.1.

Таблица П.1

x_i	Метод Эйлера (Euler method)		Модифицированный метод Эйлера modified Euler method	
	$y(x_i)$	$z(x_i)$	$y(x_i)$	$z(x_i)$
0.0	0.0	-1.0	0.0	-1.0
0.1	-0.1	-1.1	-0.1050000000	-1.1100000000
0.2	-0.21	-1.22	-0.2221000000	-1.2442000000
0.3	-0.332	-1.364	-0.3539620000	-1.4079240000
0.4	-0.4684	-1.5368	-0.5038336400	-1.6076672800
0.5	-0.62208	-1.74416	-0.6756770408	-1.8513540820
0.6	-0.796496	-1.992992	-0.8743259898	-2.1486519800
0.7	-0.9957952	-2.2915904	-1.1056777708	-2.5113554160
0.8	-1.22495424	-2.64990848	-1.3769268040	-2.9538536080
0.9	-1.489945088	-3.079890176	-1.6968507010	-3.4937014020
1.0	-1.797934106	-3.595868211	-2.0761578550	-4.1523157100

Таблица П.2

x_i	Метод Рунге–Кутты 4 порядка (Runge–Kutta method)		Точное решение (exact value)	
	$y(x_i)$	$z(x_i)$	$y(x_i)$	$z(x_i)$
0.0	0.0	−1.0	0.0	−1.0
0.1	−0.1053500000	−1.110700000	−0.1053506895	−1.110701379
0.2	−0.2229544900	−1.245908980	−0.2229561745	−1.245912349
0.3	−0.3555266141	−1.411053228	−0.3555297000	−1.411059400
0.4	−0.5063802065	−1.612760413	−0.5063852320	−1.612770464
0.5	−0.6795627842	−1.859125568	−0.6795704570	−1.859140914
0.6	−0.8800179845	−2.160035969	−0.8800292308	−2.160058462
0.7	−1.113783966	−2.527567932	−1.113799992	−2.527599984
0.8	−1.388235736	−2.976471472	−1.388258106	−2.976516212
0.9	−1.712381128	−3.524762256	−1.712411866	−3.524823732
1.0	−2.097222310	−4.194444620	−2.097264025	−4.194528050

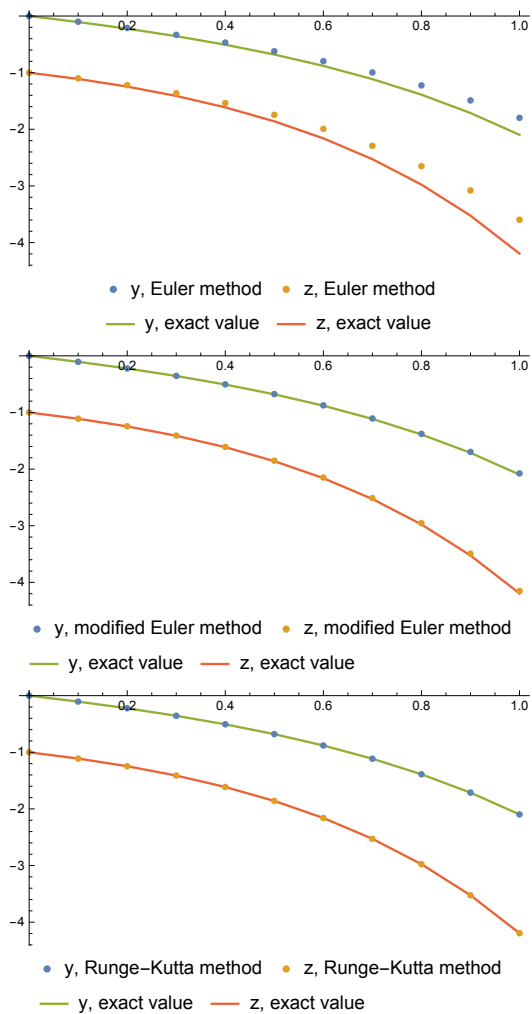


Рис. П.1: Графическое сравнение решений, полученных численными методами, с точным решением

Учебное издание

**Банников Александр Сергеевич
Ким Инна Геральдовна
Латыпова Наталья Владимировна**

ЧИСЛЕННЫЕ МЕТОДЫ

Учебное пособие

Часть 2

Авторская редакция

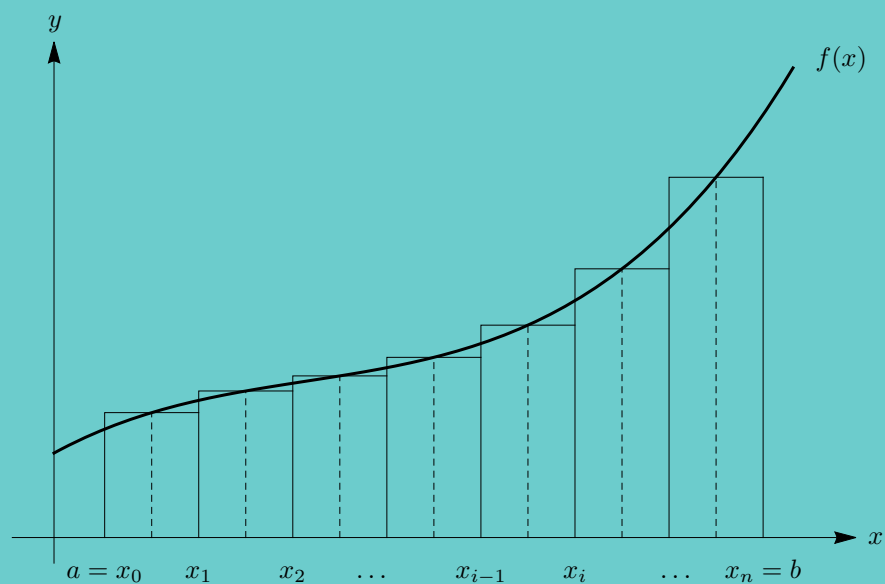
Подписано в печать 21.12.20. Формат 60х84 1/16.

Усл. печ. л. 5,1. Уч.-изд.л. 4,7.

Тираж 300 экз. Заказ № 1996

Издательский центр «Удмуртский университет»
426034, г. Ижевск, ул. Университетская, д. 1, корп. 4, каб. 207
Тел./факс: +7(3412) 500–295 E-mail: editorial@udsu.ru

Типография Издательского центра «Удмуртский университет»
426034, г. Ижевск, ул. Университетская, д. 1, корп. 2



ISBN 978-5-4312-0840-9

