

ФГБОУ ВО «Воронежский государственный
технический университет»

Кафедра высшей математики
и физико-математического моделирования

**ПРИБЛИЖЕНИЕ ФУНКЦИЙ И
ЧИСЛЕННЫЕ МЕТОДЫ РЕШЕНИЯ
ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для лабораторных работ по курсу «Спецглавы математики»
для студентов направления 22.03.02 «Металлургия»
очной формы обучения

Воронеж 2021

Составители: канд. техн. наук С.А. Кострюков,
канд. техн. наук В.В. Пешков,
канд. физ.-мат. наук Г.Е. Шунин

Векторный анализ и элементы теории поля: Методические указания для практических занятий по курсу «Спецглавы математики» для студентов направления 22.03.02 «Металлургия» очной формы обучения / Воронеж. ФГБОУ ВО «Воронежский государственный технический университет»; сост.: С.А. Кострюков, В.В. Пешков, Г.Е. Шунин. Воронеж, 2021. 92 с. Электронный ресурс кафедры высшей математики и физико-математического моделирования.

Настоящие методические указания предназначены в качестве руководства при выполнении лабораторных работ для студентов очной формы обучения. Приведены темы лабораторных занятий, основные теоретические сведения, большое количество задач и примеры их решения с помощью универсальной системы компьютерной математики Maple. Методические указания подготовлены в электронной форме и содержатся в файле ПФ и ЧМР.pdf.

Ил. 14. Библиогр.: 3 назв.

Ответственный за выпуск зав. кафедрой д-р физ.-мат. наук,
проф. И.Л. Батаронов

Публикуется по решению методического семинара кафедры высшей математики и физико-математического моделирования.

СОДЕРЖАНИЕ РАЗДЕЛА РАБОЧЕЙ ПРОГРАММЫ

Понятие функционала и его вариации. Экстремум функционала. Уравнение Эйлера-Лагранжа. Методы конечных разностей, Рунге, Галёркина и конечных элементов.

Перечень лабораторных работ

1. Аппроксимация функций.
2. Численное решение задачи Коши для основного динамического уравнения.
3. Решение одномерных краевых задач методом конечных разностей.
4. Решение одномерных краевых задач методом Рунге.
5. Решение одномерных краевых задач методом Галёркина.
6. Решение одномерных краевых задач методом конечных элементов.

1. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

1.1. Основные понятия

Обыкновенным дифференциальным уравнением порядка r называется уравнение

$$F(x, y(x), y'(x), \dots, y^{(r)}(x)) = 0, \quad (1.1)$$

которое связывает независимую переменную x , искомую функцию $y = y(x)$ и ее производные $y'(x)$, ..., $y^{(r)}(x)$.

Решение (интегрирование) дифференциального уравнения (1.1) заключается в отыскании функций (решений) $y(x)$, которые удовлетворяют этому уравнению для всех значений x в определенном конечном или бесконечном интервале (a, b) .

Общее решение обыкновенного дифференциального уравнения порядка r имеет вид

$$y = y(x, C_1, \dots, C_r),$$

где C_1, \dots, C_r – произвольные постоянные, частный выбор которых дает частное решение.

В задаче Коши (начальной задаче) требуется найти частное решение, удовлетворяющее начальным условиям

$$y(x_0) = y_0, \quad y'(x_0) = y'_0, \quad \dots, \quad y^{(r-1)}(x_0) = y_0^{(r-1)},$$

по которым определяются r постоянных C_1, \dots, C_r .

Задача Коши для нормальной системы дифференциальных уравнений

$$y'_1 = f_1(x, y_1, \dots, y_n),$$

.....

$$y'_n = f_n(x, y_1, \dots, y_n)$$

заключается в отыскании функций $y_1(x), \dots, y_n(x)$, удовлетворяющих этой системе и начальным условиям

$$y_1(x_0) = y_{10}, \quad \dots, \quad y_n(x_0) = y_{n0}.$$

Двухточечная граничная (краевая) задача для уравнения (1.1) ставится следующим образом: найти функцию $y(x)$, которая на отрезке $[a, b]$ удовлетворяет (1.1), а на концах отрезка – граничным условиям

$$\varphi_i(y(a), y'(a), \dots, y^{(r-1)}(a)) = 0, \quad i=1, 2, \dots, L,$$

$$\psi_j(y(b), y'(b), \dots, y^{(r-1)}(b)) = 0, \quad j=L+1, L+2, \dots, r.$$

В важном частном случае линейной граничной задачи при $r=2$ дифференциальное уравнение и условия записываются в виде

$$\begin{aligned} y''(x) + p(x)y'(x) + q(x)y(x) &= f(x), \quad a \leq x \leq b; \\ \alpha_0 y'(a) + \beta_0 y(a) &= \gamma_0; \\ \alpha_1 y'(b) + \beta_1 y(b) &= \gamma_1 \quad (\alpha_0^2 + \beta_0^2 \neq 0, \alpha_1^2 + \beta_1^2 \neq 0). \end{aligned} \tag{1.2}$$

Методы решения обыкновенных дифференциальных уравнений можно разделить на графические, аналитические, приближенные и численные. Графические методы используют геометрические построения. В частности, метод изоклин, применяемый для решения уравнений 1-го порядка вида $y' = f(x, y)$, позволяет построить интегральные кривые по полю направлений, определенному изоклинами.

С помощью аналитических методов для некоторых классов дифференциальных уравнений удастся получить решение в виде конечных формул и выражений. С некоторыми этими методами студенты знакомятся в курсе дифференциальных уравнений, изучая решение уравнений первого порядка – с разделяющимися переменными, однородные, линейные, в полных дифференциалах и др., а также ряд уравнений высших порядков (например, линейные с постоянными коэффициентами). Решение, записанное в виде бесконечной суммы (функционального ряда), также принято считать аналитическим.

Многие приближенные методы используют подходы, позволяющие находить приближенные решения путем специального выбора классов функций. Например, широко известен метод степенных рядов, который аппроксимирует решение отрезком ряда Тейлора. К приближенным методам также относят различные методы линеаризации, разложение решения по некоторому малому параметру, содержащемуся в задаче, асимптотические методы, метод последовательных приближений и т.д.

Наиболее значимыми являются численные методы решения дифференциальных уравнений, которые в настоящее время служат основным инструментом при исследовании научных технических задач. Особенно эффективны эти методы в сочетании с использованием современных компьютеров. Суть большинства численных методов (так называемых конечно-разностных методов) состоит в следующем: область непрерывного изменения аргумента x (например, отрезок) заменяется дискретным множеством точек $\{x_i\}$, называемых узлами. Эти узлы составляют разностную сетку. Искомая функция не-

прерывного аргумента $y(x)$ приближенно заменяется узловыми значениями $\{y_i\}$ – функцией дискретного аргумента на заданной сетке. Эта функция называется сеточной. Исходное дифференциальное уравнение заменяется разностным уравнением относительно сеточной функции. При этом для входящих в уравнение производных используются соответствующие конечно-разностные соотношения. Такая замена дифференциального уравнения разностным называется разностной аппроксимацией. Таким образом, решение дифференциального уравнения сводится к получению числовой таблицы приближенных значений $\{y_i\}$ на сетке $\{x_i\} \in [x_0, b]$.

1.2. Численное решение задачи Коши

Многие дифференциальные уравнения не имеют аналитического решения. Кроме того, часто аналитическое решение и не нужно, но требуется получить результат в виде графических построений, либо в табулированном виде. В таких случаях решение дифференциальных уравнений ищется в численном виде, для чего Maple-функция **dsolve** используется с параметром **numeric** или **type=numeric**.

Пример 1. Решить задачу Коши

$$y' + y = \sin xy, \quad y(0) = 0,5.$$

Решение.

В Maple 14 численное решение уравнения осуществляется так:

```
> p:=dsolve({D(y)(x)+y(x)=sin(x*y(x)), y(0)=0.5}, y(x),
type=numeric, range=0..4): # в range указывается диапазон изменения x
```

```
p := proc(x_rkf45) ... end proc
```

```
> plots[odeplot](p); # построение графика решения
```

Следует отметить, что в более ранних версиях, в частности, Maple 5.4, вызов этих функций отличается тем, что диапазон изменения указывается в **plots**, а не в **dsolve**:

```
> p:=dsolve({D(y)(x)+y(x)=sin(x*y(x)),y(0)=0.5},y(x),
type=numeric);
```

```
p := proc(x_rkf45) ... end proc
```

```
> plots[odeplot](p,0..4); # построение графика решения
```

Помимо построения графика можно вычислить значение функции в любой заданной точке.

```
> for t from 0 by 0.4 to 4 do print(p(t)); od; # табули-
рование функции
```

```
[x=0., y(x)=0.5000000000000000]
```

```
[x=0.4, y(x)=0.363017603454018012]
```

```
[x=0.8, y(x)=0.308828294005778160]
```

```
[x=1.2, y(x)=0.306867805078362366]
```

```
[x=1.6, y(x)=0.352983256331245887]
```

```
[x=2.0, y(x)=0.456367192785067156]
```

```
[x=2.4, y(x)=0.608172116288803854]
```

```
[x=2.8, y(x)=0.725942775091025338]
```

```
[x=3.2, y(x)=0.743994997189059371]
```

```
[x=3.6, y(x)=0.703144650928495874]
```

```
[x=4.0, y(x)=0.649493152675693964]
```

Обратите внимание на особый вывод значений функции.

Пример 2. Решить задачу Коши для системы

$$y' = z, \quad z' = 0.1z - \sin y, \quad y(0) = 0, \quad z(0) = 1.$$

Решение.

```
> sys:=diff(y(x),x)=z(x),diff(z(x),x)=0.1*z(x)-sin(y(x));
```

$$\text{sys} := \frac{d}{dx} y(x) = z(x), \frac{d}{dx} z(x) = 0.1 z(x) - \sin(y(x))$$

```
> fcns:={y(x),z(x)};
```

```
fcns := {y(x), z(x)}
```

```
> F:=dsolve({sys,y(0)=0,z(0)=1},fcns,numeric); # чис-
ленное решение
```

```
F := proc(x_rkf45) ... end proc
```

```
> F(2); # вывод значений функций при x=2
```

```
[x=2., y(x) = 1.12663272094226685 z(x) = -0.21079505078752783]
> plots[odeplot](F, [[x, y(x)], [x, z(x)]], 0..2.5,
color=black); # графики функций
```

Ниже приводятся формулы итерационных процессов наиболее известных одношаговых методов решения задачи Коши.

Метод Эйлера:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad i = 0, 1, 2, \dots$$

Метод Эйлера с пересчетом:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))], \quad i = 0, 1, 2, \dots$$

Усовершенствованный метод Эйлера:

$$\begin{aligned} \tilde{y} &= y_i + (h/2)f(x_i, y_i), \\ y_{i+1} &= y_i + hf(x_i, \tilde{y}), \quad i = 0, 1, 2, \dots \end{aligned}$$

Метод Рунге-Кутты 4-го порядка:

$$y_{i+1} = y_i + \frac{h}{6}(k_0 + 2k_1 + 2k_2 + k_3), \quad i = 0, 1, 2, \dots,$$

где

$$\begin{aligned} k_0 &= f(x_i, y_i), \quad k_1 = f(x_i + h/2, y_i + (h/2)k_0), \\ k_2 &= f(x_i + h/2, y_i + (h/2)k_1), \quad k_3 = f(x_i + h, y_i + hk_2). \end{aligned}$$

Пример 3. Решить задачу Коши $y' = x + y + \sin xy$, $y(0) = 0$.

Решение.

Составим программу, в которую заложены сразу все четыре приведенных выше метода. Выбор метода решения задачи определяется значением переменной **_Meth** (1 соответствует методу Эйлера, 2 – методу Эйлера с пересчетом, 3 – усовершенствованному методу Эйлера и 4 – метод Рунге–Кутты).

```
> restart;
```

> **f:=(x,y)->x+y+sin(x*y); #** задана функция $f(x, y)$ из правой части уравнения

$$f := (x, y) \rightarrow x + y + 1. \sin(xy)$$

> **u:=array(0..100,[]); #** выделение памяти под массив

> **Y1:=vector(101);X:=vector(101); #** выделение памяти под массивы

u := array(0 .. 100, [])

Y1 := array(1 .. 101, [])

X := array(1 .. 101, [])

Присвоение начальных значений: **a** – начальная точка отрезка (берется из начального условия), **b** – конечная точка (выбирается произвольно), **h** – шаг, **n** – число отсчетов, **y0** – начальное значение.

> **a:=0.; b:=2.; h:=0.1; n:=(b-a)/h;**

> **y0:=0; #** число **y0** берется из начального условия $u(a)=y_0$

> **u[0]:=y0; #** инициализация нулевого элемента массива

> **x:=a; k:='k';**

> **_Meth:=4; #** значение этой переменной определяет один из четырех реализованных методов

> **if _Meth =1 then #** если метод Эйлера ...

for i from 0 to n-1 do

u[i+1]:=evalf((u[i]+h*f(x,u[i]));

x:=x+h;

od:

fi;

> **if _Meth =2 then #** если метод Эйлера с пересчетом...

for i from 0 to n-1 do

ut:=u[i]+h*f(x,u[i]);

u[i+1]:=evalf(u[i]+h/2*(f(x,u[i])+f(x+h,ut)));

x:=x+h;

od:

fi;

> **if _Meth =3 then #** если усоверш. метод Эйлера ...

for i from 0 to n-1 do

ut:=u[i]+h/2*f(x,u[i]);

u[i+1]:=evalf(u[i]+h*f(x+h/2,ut));

```

x:=x+h;
od:
fi;
> if _Meth =4 then # если метод Рунге–Кутты ...
for i from 0 to n-1 do
k0:=f(x,u[i]); k1:=f(x+h/2,u[i]+h*k0/2);
k2:=f(x+h/2,u[i]+h*k1/2); k3:=f(x+h,u[i]+h*k2);
u[i+1]:=evalf(u[i]+h/6*(k0+2*k1+2*k2+k3));
x:=x+h;
od:
fi;
> Y1:=evalf([seq(u[k],k=0..n)]); # формирование век-
тора узловых значений
Y1 := [0., 0.005183979167, 0.02162386603, 0.05104378161, 0.09580415509
0.1590857034, 0.2451322918, 0.3595432837, 0.5095314988, 0.7038353178
0.9514154636, 1.257209511, 1.613830532, 1.995140009, 2.367069445, 2.715459640
3.064688190, 3.481232674, 4.057674745, 4.767235270, 5.428150597]

```

```

> X:=evalf([seq(a+k*h,k=0..n)]); # формирование век-
тора узлов
X := [0., 0.1000000000, 0.2000000000, 0.3000000000, 0.4000000000, 0.5000000000
0.6000000000, 0.7000000000, 0.8000000000, 0.9000000000, 1.000000000
1.100000000, 1.200000000, 1.300000000, 1.400000000, 1.500000000, 1.600000000
1.700000000, 1.800000000, 1.900000000, 2.000000000]

```

Нанесём на координатной плоскости точки, изображающие найденное решение:

```

> g1:=plot([ [X[m], Y1[m]] $m=1..n+1 ], style=POINT,
symbol=CIRCLE, color=black) :

```

Для сравнения получим численное решение данной задачи встроенными средствами Maple:

```

> x:='x':y:='y':
p:=dsolve({diff(y(x),x)=f(x,y(x)),y(a)=y0),y(x),
type=numeric);
> g2:=plots[odeplot](p,a..b): # график этого решения
> plots[display](g1,g2); # совмещение двух полученных
решений

```

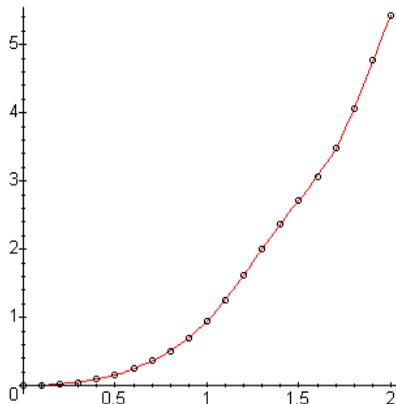


Рис. 1.1

Упражнения

1. Решить задачу Коши для дифференциального уравнения $y' = f(x, y)$ на отрезке $[a, b]$ при начальном заданном условии $y(a) = y_0$ и шаге интегрирования h :

- 1) методом Эйлера и усовершенствованным (модифицированным) методом Эйлера с шагом h и $h/2$,
- 2) методом Рунге-Кутты с шагом h и $2h$,
- 3) методом Адамса.

№	Функция $f(x, y)$	Интервал $[a, b]$	y_0	Шаг h
1	$f(x, y) = 3x^2 + 0,1xy$	$[0; 1]$	$y(0) = 0,2$	0,1
2	$f(x, y) = 0,185(x^2 + \cos(0,7x)) + 1,843y$	$[0,2; 1,2]$	$y(0,2) = 0,25$	0,1
3	$f(x, y) = x + \cos\left(\frac{y}{3}\right)$	$[1,6; 2,6]$	$y(1,6) = 4,6$	0,1
4	$f(x, y) = x + \sin\left(\frac{y+1}{\sqrt{13}}\right)$	$[0,2; 1,2]$	$y(0,2) = 1,1$	0,1

№	Функция $f(x, y)$	Интервал [a, b]	y_0	Шаг h
5	$f(x, y) = x + \cos\left(\frac{y}{e}\right)$	[1,4; 2,4]	$y(1,4) = 2,5$	0,1
6	$f(x, y) = x + \cos\left(\frac{y}{\pi}\right)$	[1,7; 2,7]	$y(1,7) = 5,3$	0,1
7	$f(x, y) = -3y^2 + \sqrt{4x^2 + 1}$	[2,6; 4,6]	$y(2,6) = 3,5$	0,2
8	$f(x, y) = 2 - \sin^2(x + y)$	[2; 3]	$y(2) = 2,3$	0,1
9	$f(x, y) = 1,6x + 0,5y^2$	[0; 1]	$y(0) = 0,3$	0,1
10	$f(x, y) = x - \cos\left(\frac{y}{\sqrt{5}}\right)$	[1,8; 2,8]	$y(1,8) = 2,6$	0,1
11	$f(x, y) = x + \cos\left(\frac{y}{\sqrt{11}}\right)$	[2,1; 3,1]	$y(2,1) = 2,5$	0,1
12	$f(x, y) = e^{2x} + 0,25y^2$	[0; 0,5]	$y(0) = 2,6$	0,05
13	$f(x, y) = x + \cos\left(\frac{y}{\sqrt{2}}\right)$	[-2; -1]	$y(-2) = 3$	0,1
14	$f(x, y) = 0,133 \cdot (x^2 + \sin(2x)) + 1 + 0,872y$	[0,2; 1,2]	$y(0,2) = 0,25$	0,1
15	$f(x, y) = \sin(x + y) + 1,5$	[1,5; 2,5]	$y(1,5) = 4,5$	0,1
16	$f(x, y) = x + \cos\left(\frac{y}{1,25}\right)$	[0,4; 1,4]	$y(0,4) = 0,8$	0,1
17	$f(x, y) = 2,5x + \cos(y + 0,6)$	[1; 3]	$y(1) = 1,5$	0,2
18	$f(x, y) = \cos(1,5y + x)^2 + 1,4$	[1; 2]	$y(1) = 1,5$	0,1
19	$f(x, y) = \frac{1}{1 + x^3 y} + 2y$	[1,5; 2]	$y(1,5) = 2,1$	0,05

№	Функция $f(x, y)$	Интервал [a, b]	y_0	Шаг h
20	$f(x, y) = \cos(y) + 3x$	[0; 2]	$y(0) = 1,3$	0,1
21	$f(x, y) = \cos(1,5x - y^2) - 1,3$	[-1; 1]	$y(-1) = 0,2$	0,2
22	$f(x, y) = x + \sin\left(\frac{y}{3}\right)$	[1,6; 2,6]	$y(1,6) = 4,6$	0,1
23	$f(x, y) = e^{-(y-1)} + 2x$	[0; 0,5]	$y(0) = 0,3$	0,05
24	$f(x, y) = 1 + 2y \cdot \sin(x) - y^2$	[1; 2]	$y(1) = 0$	0,1
25	$f(x, y) = \frac{\cos(y)}{x+2} - 0,3y^2$	[0; 1]	$y(0) = 0$	0,1
26	$f(x, y) = 0,166 \cdot (x^2 + \sin(1,1 \cdot x)) + 0,883 \cdot y$	[0,2; 1,2]	$y(0,2) = 0,25$	0,1
27	$f(x, y) = x + \cos\left(\frac{y+x}{\pi}\right)$	[1,7; 2,7]	$y(1,7) = 5,6$	0,1
28	$f(x, y) = x + \cos\left(\frac{y}{e}\right)$	[1,4; 2,4]	$y(1,4) = 2,5$	0,1
29	$f(x, y) = x + \sin\left(\frac{y}{\sqrt{10}}\right)$	[0,6; 1,6]	$y(0,6) = 0,8$	0,1
30	$f(x, y) = y + \cos\left(\frac{x}{\pi}\right)$	[1; 2]	$y(1) = 5,9$	0,1
31	$f(x, y) = 1 + 0,8y \cdot \sin(x) - 2y^2$	[0; 1]	$y(0) = 0$	0,1
32	$f(x, y) = x + \sin\left(\frac{y}{2,5}\right)$	[0,5; 1,5]	$y(0,5) = 1,8$	0,1
33	$f(x, y) = x + \cos\left(\frac{y}{\sqrt{1,3}}\right)$	[1,2; 2,2]	$y(1,2) = 1,8$	0,1

№	Функция $f(x, y)$	Интервал $[a, b]$	y_0	Шаг h
34	$f(x, y) = 1 + 2,2 \cdot \sin(x) + 1,5y^2$	[0; 1]	$y(0) = 0$	0,1
35	$f(x, y) = \frac{\cos(y)}{x+2} - 0,3y^2$	[0; 1]	$y(0) = 0$	0,1
36	$f(x, y) = \frac{\cos(y)}{1,5+x} - 1,2y^2$	[0; 1]	$y(0) = 0$	0,1
37	$f(x, y) = \cos(x-y) + \frac{1,25y}{1,5+x}$	[0; 1]	$y(0) = 0$	0,1
38	$f(x, y) = 0,2x^2 + y^2$	[0; 1]	$y(0) = 0,8$	0,1
39	$f(x, y) = x^2 + y$	[0; 1]	$y(0) = 0,4$	0,1
40	$f(x, y) = xy + 0,1y^2$	[0; 1]	$y(0) = 0,5$	0,1

2. Методом пристрелки найти решение граничных задач в точках $x_k = 0,1k$.

Вариант 1. $y''(x) + 2y'(x) - \frac{4}{x}y(x) = 1,$

$$0,5 \leq x \leq 1, \quad y'(0,5) = 1,5, \quad y(1) + y'(1) = 4.$$

Вариант 2. $y''(x) - \frac{6x}{3x^2 - 0,5}y'(x) - \frac{1}{x}y(x) = 0,5 - x^2,$

$$0,5 \leq x \leq 1, \quad y'(0,5) = 0,25, \quad 2y(1) + y'(1) = 3,5.$$

Вариант 3. $y''(x) + \frac{1}{x}y'(x) - \frac{1}{x^2}y(x) = -\frac{2}{x^3},$

$$0,5 \leq x \leq 1, \quad y(0,5) = -2\ln 2, \quad y(1) = 0.$$

Вариант 4. $y''(x) + \frac{1}{x}y'(x) = \frac{1}{x^2}, \quad 0,5 \leq x \leq 1, \quad y'(0,5) = 3, \quad y(1) = 1.$

Вариант 5. $y''(x) + 2xy'(x) - y(x) = 2(x^2 + 1)\cos x,$

$$0 \leq x \leq 0.5, \quad y(0) = 0, \quad y(0.5) = 0.5 \sin 0.5.$$

Вариант 7. $x^2 \ln x \cdot y''(x) - xy'(x) + y(x) = 0,$
 $y(1) = 0, \quad y(e) = e - 2.$

Вариант 8. $y''(x) - \ln x \cdot y'(x) - 2y(x) = 1, \quad y(0,5) + y'(0,5) = 1,$
 $y(1,5) - y'(1,5) = 1.$

Вариант 9. $y''(x) + y'(x) - \frac{2}{\cos^2 x} y(x) = \frac{2}{\cos^2 x}, \quad y(0) = 0,$
 $y'(0,5) = 1/\cos^2 0,5.$

Вариант 10. $y''(x) - \frac{2}{x} y'(x) - \frac{4}{x^2 + 2} y(x) = 8, \quad y'(0,5) = 0,5,$
 $y(1) + y'(1) = 1.$

2. ПРИБЛИЖЕНИЕ (АППРОКСИМАЦИЯ) ФУНКЦИЙ

2.1. Введение

Пусть величина y является функцией аргумента x . Это означает, что любому значению x из области определения поставлено в соответствие число y . Вместе с тем на практике часто неизвестна явная связь между y и x , т.е. невозможно записать эту связь в виде некоторой зависимости $y = f(x)$. Иногда даже известная зависимость $y = f(x)$ оказывается настолько громоздкой (например, содержит трудно вычисляемые выражения, сложные интегралы и т.п.), что ее использование в практических расчетах требует слишком много времени.

В этой связи и возникает задача о приближении, или *аппроксимации*, функции, состоящая в том, чтобы данную функцию $f(x)$ приближенно заменить (аппроксимировать) некоторой функцией $g(x)$ так, чтобы отклонение (в некотором смысле) $g(x)$ от $f(x)$ в заданной области было наименьшим. Функция $g(x)$ при этом называется аппроксимирующей. Что ка-

сается самого понятия «наименьшее отклонение», то оно уточняется при рассмотрении конкретных способов аппроксимации.

Аппроксимация, при которой приближение строится на дискретном множестве точек, называется *точечной* (дискретной). Это наиболее распространенный и практически важный случай, когда точный вид связи между величинами x и y неизвестен, но эта связь задается в виде таблицы $\{x_i, y_i\}$. Это означает, что дискретному множеству значений аргумента $\{x_i\}$ поставлено в соответствие множество значений функции $\{y_i\}$ ($i = 1, 2, \dots, n$). Эти значения – либо результаты расчетов, либо экспериментальные данные. На практике могут понадобиться значения величины и в других точках, отличных от x_i . Однако зачастую получить эти значения можно лишь путем очень сложных расчетов или проведением дорогостоящих экспериментов. Вот здесь и приходит на помощь точечная аппроксимация. К ней относятся интерполирование, среднеквадратическое приближение и др.

При построении приближения на непрерывном множестве точек (например, на отрезке) аппроксимация называется *непрерывной*, или интегральной. Такая аппроксимация возникает, когда точная связь известна, но ей по каким-то причинам пользоваться нецелесообразно. К непрерывной аппроксимации относится, например, приближение на основе метода взвешенных невязок.

Отметим, что аппроксимация функций – это традиционно один из основных разделов численных методов. Разработанные давно, методы аппроксимации при вычислениях вручную могли использоваться лишь для решения не слишком трудоемких задач. С появлением компьютеров начался период их широкого и эффективного внедрения в практику. В настоящее время они имеют не только самостоятельное применение, но и являются весьма важной частью многих других методов вычислительной математики (например, методов решения дифференциальных краевых задач).

2.2. Интерполирование

Линейную комбинацию

$$g(x) = c_1\varphi_1(x) + c_2\varphi_2(x) + \dots + c_n\varphi_n(x) = \sum_{k=1}^n c_k\varphi_k(x) \quad (2.1)$$

с действительными коэффициентами c_k называют обобщенным многочленом по системе функций φ_k ($k = 1, \dots, n$).

Задача интерполирования функции $f(x)$ состоит в том, чтобы найти значение $f(x)$ при $x \neq x_m$, $m = 1, 2, \dots, n$, если известны узлы интерполирования x_1, x_2, \dots, x_n и значения функции $f(x)$ в этих узлах. Решается эта задача следующим образом: выбирается система функций $\{\varphi_k\}$, строится обобщенный многочлен (6.1), а коэффициенты c_k задаются таким образом, чтобы в узлах интерполирования значения обобщенного многочлена совпадали со значениями данной функции $f(x)$:

$$g(x_m) = \sum_{k=1}^n c_k\varphi_k(x_m) = f(x_m), \quad m = 1, 2, \dots, n. \quad (2.2)$$

Обобщенный многочлен, обладающий свойством (2.2), называется обобщенным интерполяционным многочленом. За приближенное значение $f(x)$ принимают значение *интерполирующей* функции – интерполяционного многочлена $g(x)$.

Не для всякой системы $\{\varphi_k\}$ функций задача интерполирования решается однозначно. Общие условия существования и единственности обобщенного интерполяционного многочлена требуют, чтобы данная система функций была системой Чебышева на заданном отрезке $[a, b]$. Не вдаваясь в детали, скажем, что таким свойством обладают следующие системы, чаще всего используемые на практике:

1) $1, x, x^2, \dots, x^n, \dots$

2) $1, \sin \frac{2\pi x}{l}, \cos \frac{2\pi x}{l}, \dots, \sin \frac{2\pi n x}{l}, \cos \frac{2\pi n x}{l}, \dots$

3) $e^{\alpha_1 x}, e^{\alpha_2 x}, \dots, e^{\alpha_n x}, \dots$, где $\{\alpha_k\}$ – некоторая числовая последовательность попарно различных действительных чисел.

$f(x)$	0.3	1	2	2	1
--------	-----	---	---	---	---

1-й способ: на основе прямого решения системы (2.3).

Зададим два массива \mathbf{X} и \mathbf{Y} – соответственно абсциссы и ординаты отдельных точек функции:

```
> X:=[0.5, 2, 3, 4, 5]; Y:=[0.3, 1, 2, 2, 1];
      X := [0.5, 2, 3, 4, 5]
      Y := [0.3, 1, 2, 2, 1]
```

```
> n:=5; # число узлов
```

```
      n := 5
```

Определим диапазон $[a, b]$ изменения x , который потребуется в дальнейшем для построения графика:

```
> a:=min(X[i]$i=1..n); b:=max(X[i]$i=1..n);
      a := 0.5
      b := 5
```

Может оказаться полезным слегка расширить этот диапазон, взяв его в виде $[a - \delta, b + \delta]$, где δ достаточно мало, например, положить

```
> delta:=(b-a)/n/5.;
      delta := 0.4500000000
```

и тем самым

```
> a:=a-delta; b:=b+delta;
```

Поскольку число узлов 5, искомым интерполяционный полином будет степени на единицу меньше, т.е. 4. Введем его в общем виде с пока неопределенными коэффициентами:

```
> g:=t->c0+c1*t+c2*t^2+c3*t^3+c4*t^4;
      g := t -> c0 + c1 t + c2 t^2 + c3 t^3 + c4 t^4
```

Составим систему уравнений (2.3) и присвоим её переменной **eqns**:

```
> eqns:={g(X[1])=Y[1], g(X[2])=Y[2], g(X[3])=Y[3],
g(X[4])=Y[4], g(X[5])=Y[5]};
```

```
eqns := {c0 + 0.5 c1 + 0.25 c2 + 0.125 c3 + 0.0625 c4 = 0.3,
c0 + 2 c1 + 4 c2 + 8 c3 + 16 c4 = 1, c0 + 3 c1 + 9 c2 + 27 c3 + 81 c4 = 2,
c0 + 4 c1 + 16 c2 + 64 c3 + 256 c4 = 2, c0 + 5 c1 + 25 c2 + 125 c3 + 625 c4 = 1}
```

Заметим, что то же самое можно было сделать короче с помощью команды

```
> eqns := { (g(X[i])=Y[i]) $i=1..n };
```

Решим эту систему:

```
> r := solve(eqns, {c0, c1, c2, c3, c4});
```

```
r := {c4 = 0.045291005, c3 = -0.63407407, c2 = 2.7156614,  
      c1 = -3.4748148, c0 = 1.4349206}
```

Подставим эти значения коэффициентов в интерполяционный полином g :

```
> subs(r, g(t));
```

```
r1 := 1.434921 - 3.474815t + 2.715661t2 - 0.6340741t3 + 0.04529101t4
```

Это и есть искомый интерполяционный многочлен, записанный в виде простого выражения. Чтобы его использовать в дальнейших вычислениях как Maple-функцию, целесообразно воспользоваться строкой

```
> fu := unapply(subs(r, g(t)), t);
```

Вычислим эту функцию в ряде точек:

```
> { [1, fu(1)], [2, fu(2)], [3.5, fu(3.5)], [4.82, fu(4.82)] };  
{ [3.5, 2.1504764], [4.82, 1.219491], [1, 0.086984135], [2, 1.0000005] }
```

Построим график (интервал, на котором он строится, должен содержать все узлы):

```
> g1 := plot(fu(x), x=a..b) : # вывод изображения будет позже
```

Для того чтобы наглядно убедиться в прохождении интерполирующей функции через заданные точки функции (те, что из таблицы), полезно на график нанести эти точки. Это можно сделать так:

```
> g2 := plot([ [X[i], Y[i]] $i=1..n ], x=a..b, style=POINT,  
            symbol=CIRCLE, color=black);
```

```
g2 := PLOT(CURVES(  
  [[0.5000000000000000, 0.3000000000000000], [2., 1.], [3., 2.], [4., 2.], [5., 1.]],  
  SYMBOL(CIRCLE), COLOUR(RGB, 0, 0, 0), AXESLABELS(x, ),  
  STYLE(POINT), VIEW(0..5., DEFAULT))
```

(Результат сохраняется в переменной $g2$, но графически пока не отображается).

Для графического вывода построений (как по отдельности, так и совместно) используется функция **display** из пакета графических построений **plots**. В частности, следующая команда совмещает два последних объекта **g1** и **g2**:

```
> plots[display] (g1, g2) ;
```

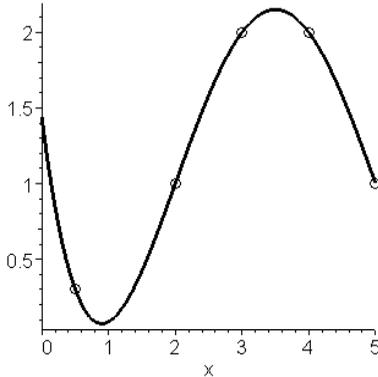


Рис. 2.1

2-й способ: на основе формулы Лагранжа (2.4).

```
> X:=[0.5,2,3,4,5]: Y:=[0.3,1,2,2,1]: n:=5:
> a:=min(X[i]$i=1..n): b:=max(X[i]$i=1..n):
> r1:=sum(Y[k]/product(piecewise(i=k,1,(X[k]-X[i])),
i=1..n)/(x-X[k]),k=1..n)*product((x-X[k]),k=1..n);
r1 := ( 0.005079365080 / (x-0.5) - 0.111111111111 / (x-2) + 0.400000000000 / (x-3) - 0.2857142857 / (x-4)
+ 0.03703703704 / (x-5) ) (x-0.5)(x-2)(x-3)(x-4)(x-5)
```

Смысл использования функции **piecewise**, стоящей внутри первого произведения, – отсеять сомножитель $(x_k - x_i)$ при $i = k$.

```
> simplify(r1) ;
-3.474815 x + 0.04529100 x^4 - 0.6340740 x^3 + 2.715661 x^2 + 1.434921
```

Команды, отвечающие за построение графика, – почти такие же, как в 1-м способе.

График интерполирующей функции сохраним в **g1**:

```
> g1:=plot(r1,x=a..b):
```

Нанесём также заданные точки функции, заданные координатами в массивах \mathbf{X} и \mathbf{Y} :

```
> g2:=plot([X[i],Y[i]]$i=1..n),x=a..b,style=POINT,  
symbol=CIRCLE,color=black):
```

Совместим два последних изображения:

```
> plots[display](g1,g2);
```

Картинка не отличается от приведенной на рис. 2.1.

2.3. Локальная интерполяция

В алгебраическом интерполировании при росте числа узлов увеличивается, как правило, степень интерполяционного многочлена. Кроме того, когда интерполирование выполняется для функций, не являющихся достаточно гладкими, интерполирование высокого порядка нецелесообразно. В этих случаях лучший результат может дать локальная (кусочная) интерполяция.

В отличие от глобальной интерполяции, когда интерполирующая функция $g(x)$ строится сразу для всего рассматриваемого интервала изменения x , при локальной интерполяции такая функция ищется отдельно для разных частей этого интервала, а затем как бы сшивается из соответствующих отдельных кусков. При локальной интерполяции должно сохраняться основное условие интерполирования: в узлах x_i значения интерполирующей функции должны быть равны значениям данной функции $y_i = f(x_i)$.

Простейшим и часто используемым видом локальной интерполяции является линейная (*кусочно-линейная*) интерполяция. Она состоит в том, что заданные точки (x_k, y_k) ($k=1, 2, \dots, n$) соединяются прямолинейными отрезками, и функция $f(x)$ приближается ломаной с вершинами в этих точках.

Уравнения каждого отрезка ломаной в общем случае разные. Поскольку имеется $n-1$ интервалов (x_k, x_{k+1}) , то для каждого из них в качестве уравнения интерполяционного многочлена используется уравнение прямой, проходящей через две

заданные точки. В частности для k -го интервала можно написать уравнение прямой, проходящей через две точки (x_k, y_k) и (x_{k+1}, y_{k+1}) в виде

$$y = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} (x - x_k) + y_k, \quad x_k \leq x \leq x_{k+1}. \quad (2.5)$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента, а затем подставить его в формулу (2.5) и найти приближенное значение в данной точке.

Приведем реализацию этого алгоритма в среде Maple для таблично заданной функции из предыдущего примера.

> restart;

> x:=[0.5, 2, 3, 4, 5]; Y:=[0.3, 1, 2, 2, 1]; n:=5; # исходные данные

Программируем формулу (2.5), причем величину y из этой формулы определяем как функцию (с именем h) не только аргумента x , но и номера интервала k :

> h:=(x, k)->(Y[k+1]-Y[k])*(x-X[k])/ (X[k+1]-X[k])+Y[k];

Интерполирующую функцию на всем числовом промежутке задаем кусочным образом:

> w:=x->piecewise(x<X[2], h(x, 1), x<X[3], h(x, 2), x<X[4], h(x, 3), h(x, 4));

w := x → piecewise(x < X₂, h(x, 1), x < X₃, h(x, 2), x < X₄, h(x, 3), h(x, 4))

> simplify(w(x));

$$\left\{ \begin{array}{ll} 0.466667x + 0.066666 & x < 2. \\ x - 1. & x < 3. \\ 2. & x < 4. \\ -1. x + 6. & 4. \leq x \end{array} \right.$$

Итак, искомая интерполирующая функция есть $w(x)$. Далее – графический вывод результата интерполирования:

> g3:=plot(w(x), x=X[1]..X[n]):

> g2:=plot([[X[i], Y[i]]\$i=1..n], x=X[1]-0.5..X[n], style=POINT, symbol=CIRCLE, color=black);

```
> plots[display](g3,g2);
```

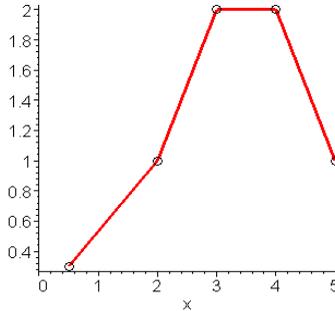


Рис. 2.2

Рассмотрим теперь случай *квадратичной* локальной интерполяции. В качестве интерполяционной функции на отрезке $[x_k, x_{k+2}]$ принимается квадратичный трехчлен. Такую интерполяцию называют также параболической. Если отвлечься от того, что отрезок $[x_k, x_{k+2}]$ является подмножеством отрезка $[x_1, x_n]$, и рассмотреть его отдельно, то квадратичную интерполяцию можно трактовать как глобальную с тремя узлами. Интерполирующую функцию на этом отрезке можно выразить явно как многочлен Лагранжа $L_2(x)$:

$$y = \frac{(x - x_{k+1})(x - x_{k+2})}{(x_k - x_{k+1})(x_k - x_{k+2})} y_k + \frac{(x - x_k)(x - x_{k+2})}{(x_{k+1} - x_k)(x_{k+1} - x_{k+2})} y_{k+1} + \frac{(x - x_k)(x - x_{k+1})}{(x_{k+2} - x_k)(x_{k+2} - x_{k+1})} y_{k+2}, \quad x_k \leq x \leq x_{k+2}.$$

Ниже приводится фрагмент программы на языке Maple, реализующий кусочно-квадратичную интерполяцию (подразумевается, что переменные **X**, **Y**, **n**, **g2**, введенные ранее, остались в оперативной памяти).

```
> h:=(x,k)->Y[k]*(x-X[k+1])*(x-X[k+2])/(X[k]-X[k+1])/(X[k]-X[k+2])+Y[k+1]*(x-X[k])*(x-X[k+2])/(X[k+1]-X[k])/(X[k+1]-X[k+2])+Y[k+2]*(x-X[k])*(x-X[k+1])/(X[k+2]-X[k])/(X[k+2]-X[k+1]);
> w:=x->piecewise(x<X[3],h(x,1),x<=X[5],h(x,3));
```

$w := x \rightarrow \text{piecewise}(x < X_3, h(x, 1), x \leq X_5, h(x, 3))$

> **simplify(w(x)) ;**

$$\begin{cases} 0.21333x^2 - 0.066655x + 0.28000 & x < 3. \\ -0.50000x^2 + 3.5000x - 4. & x \leq 5. \\ 0. & 5. < x \end{cases}$$

> **g3:=plot(w(x), x=X[1]..X[5]) :**

> **plots[display](g3, g2) ;**

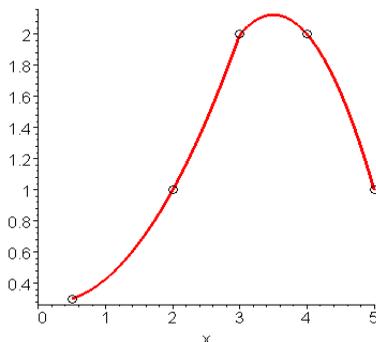


Рис. 2.3

Если присмотреться, из последнего графика можно увидеть главный недостаток данного способа кусочной интерполяции: в граничных точках отдельных участков, где осуществляется интерполирование, производная аппроксимирующей функции терпит разрыв. От этого недостатка свободны другие виды локальной интерполяции, в частности, сплайн-интерполяция и кусочная интерполяция многочленами Эрмита.

2.4. Интерполирование сплайнами

Пусть функция $f(x)$ определена на отрезке $[a, b]$ и известны её значения в системе узлов $a = x_1 < x_2 < \dots < x_n = b$. Функция $S_m(x)$ называется интерполяционным сплайном порядка m для функции $f(x)$, если выполнены следующие условия:

1) на каждом из отрезков $[x_k, x_{k+1}]$ ($k = 1, \dots, n-1$) $S_m(x)$ является полиномом степени m ;

2) на всем отрезке $[a, b]$ $S_m(x)$ имеет непрерывные производные до порядка $m - 1$ включительно;

$$3) S_m(x_k) = f(x_k), k = 1, \dots, n.$$

Если $m \geq 2$, то для единственности $S_m(x)$ следует задать дополнительно еще $m - 1$ условий, которые обычно задаются на концах отрезка, либо произвольно, либо из дополнительной информации о поведении $f(x)$.

При $m = 1$ получаем известный метод ломаных – кусочно-линейную интерполяцию. Сплайны демонстрируют ряд замечательных свойств, особенно в плане равномерной сходимости и гладкости. Этим объясняется их широкое применение в различных задачах численного анализа. Чаще всего используются квадратичный сплайн $S_2(x)$ и кубический сплайн $S_3(x)$.

Рассмотрим построение кубического сплайна $S_3(x)$. Пусть на отрезке $[a, b]$ заданы значения функции $f(x)$ в узлах x_i ($i = 1, \dots, n$), $x_1 = a$, $x_n = b$. По определению, $S_3(x)$ на i -м отрезке $[x_i, x_{i+1}]$ является кубическим многочленом. Обозначим его $p_i(x)$. Будем $p_i(x)$ искать по формуле

$$p_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, \quad i = 1, \dots, n - 1.$$

Таким образом, для построения кубического сплайна нужно найти $4 \times (n - 1)$ чисел – коэффициентов $\{a_i\}$, $\{b_i\}$, $\{c_i\}$, $\{d_i\}$. По определению $S_3(x)$ должны выполняться условия

$$p_i(x_i) = f(x_i), \quad i = 1, \dots, n - 1, \quad (2.6a)$$

$$p_i(x_{i+1}) = f(x_{i+1}), \quad i = 1, \dots, n - 1, \quad (2.6b)$$

$$p'_i(x_{i+1}) = p'_{i+1}(x_{i+1}), \quad i = 1, \dots, n - 2, \quad (2.6c)$$

$$p''_i(x_{i+1}) = p''_{i+1}(x_{i+1}), \quad i = 1, \dots, n - 2. \quad (2.6d)$$

Дополнительные два условия можно взять в виде

$$p''_1(x_1) = 0, \quad p''_{n-1}(x_n) = 0, \quad (2.7)$$

что соответствует нулевой кривизне линии в граничных точках отрезка $[a, b]$. Получаемая с такими условиями функция $S_3(x)$ называется свободным кубическим сплайном.

Итак, имеем замкнутую систему линейных алгебраических уравнений (2.6) – (2.7) относительно $\{a_i\}$, $\{b_i\}$, $\{c_i\}$, $\{d_i\}$, решение которой и даст требуемые числовые коэффициенты.

Осуществим сплайн-интерполяцию в системе Maple для функции, заданной таблично, непосредственно программируя приведенные выше формулы.

> **restart;**

> **x:=[0.5, 2, 3, 4, 5]; y:=[0.3, 1, 2, 2, 1]; n:=5;**

Определим кубический многочлен $p_k(x)$ как функцию двух переменных \mathbf{x} и \mathbf{k} (номер частичного отрезка):

> **p:=(x, k) -> a[k] + b[k] * (x - X[k]) + c[k] * (x - X[k]) ^ 2 + d[k] * (x - X[k]) ^ 3;**

$$p := (x, k) \rightarrow a_k + b_k (x - X_k) + c_k (x - X_k)^2 + d_k (x - X_k)^3$$

Запишем в отдельные переменные группы уравнений – последовательно соотношения (2.6a)-(2.6d):

> **eqns1:=(p(X[i], i)=Y[i]) \$i=1..n-1;**

$$eqns1 := a_1 = 0.3, a_2 = 1, a_3 = 2, a_4 = 2$$

> **eqns2:=(p(X[i+1], i)=Y[i+1]) \$i=1..n-1;**

$$eqns2 := a_1 + 1.5 b_1 + 2.25 c_1 + 3.375 d_1 = 1, a_2 + b_2 + c_2 + d_2 = 2, a_3 + b_3 + c_3 + d_3 = 2, \\ a_4 + b_4 + c_4 + d_4 = 1$$

> **eqns3:=(D[1](p)(X[i+1], i)=D[1](p)(X[i+1], i+1)) \$i=1..n-2;**

$$eqns3 := b_1 + 3.0 c_1 + 6.75 d_1 = b_2, b_2 + 2 c_2 + 3 d_2 = b_3, b_3 + 2 c_3 + 3 d_3 = b_4$$

> **eqns4:=(D[1]@@2)(p)(X[i+1], i)=(D[1]@@2)(p)(X[i+1], i+1)) \$i=1..n-2;**

$$eqns4 := 2 c_1 + 9.0 d_1 = 2 c_2, 2 c_2 + 6 d_2 = 2 c_3, 2 c_3 + 6 d_3 = 2 c_4$$

Последнюю группу составят дополнительные условия (2.7):

> **eqns5:=(D[1]@@2)(p)(X[1], 1)=0, (D[1]@@2)(p)(X[n], n-1)=0;**

$$eqns5 := 2 c_1 = 0, 2 c_4 + 6 d_4 = 0$$

Сведем воедино все указанные группы уравнений, а также неизвестные коэффициенты:

> **eqns:={eqns1, eqns2, eqns3, eqns4, eqns5};**

> **var:={a[i] \$i=1..n-1, b[i] \$i=1..n-1, c[i] \$i=1..n-1, d[i] \$i=1..n-1};**

$var := \{a_2, a_3, a_4, a_1, b_1, c_1, d_1, b_2, c_2, d_2, b_3, c_3, d_3, b_4, c_4, d_4\}$

Решим полученную систему:

> rez:=solve(eqns, var) ;

$rez := \{c_1 = 0., a_2 = 1., a_3 = 2., a_4 = 2., a_1 = 0.3000000000, d_4 = 0.1896713615,$
 $d_1 = 0.1032863850, d_2 = -0.3962441315, d_3 = 0.05164319249, b_4 = -0.6206572770,$
 $b_1 = 0.2342723005, b_3 = 0.6723004695, c_3 = -0.7239436620, b_2 = 0.9314553991,$
 $c_2 = 0.4647887324, c_4 = -0.5690140845\}$

Сплайн построим как кусочную функцию, отождествляя с каждым частичным интервалом соответствующий полином $p_k(x)$:

> w:=x->piecewise(x<X[2], subs(rez, p(x, 1)), x<X[3],
subs(rez, p(x, 2)), x<X[4], subs(rez, p(x, 3)), subs(rez, p(x, 4)));
 $w := x \rightarrow \text{piecewise}(x < X_2, \text{subs}(rez, p(x, 1)), x < X_3, \text{subs}(rez, p(x, 2)), x < X_4,$
 $\text{subs}(rez, p(x, 3)), \text{subs}(rez, p(x, 4)))$

Здесь функция **subs** обеспечивает постановку найденных значений коэффициентов в $p_k(x)$.

Выведем результат в более наглядном виде:

> simplify(w(x)): evalf(", 5) ;

$$\begin{cases} 0.16995 + 0.31174x + 0.10329x^3 - 0.15493x^2 & x < 2. \\ 4.1662 - 5.6826x + 2.8423x^2 - 0.39624x^3 & x < 3. \\ -7.9268 + 6.4103x - 1.1887x^2 + 0.051643x^3 & x < 4. \\ -16.761 + 13.036x - 2.8451x^2 + 0.18967x^3 & 4. \leq x \end{cases}$$

И далее – привычные графические построения:

> g4:=plot(w(x), x=X[1]..X[n], color=black) :

> g2:=plot([[X[i], Y[i]] \$i=1..n], style=POINT,
symbol=CIRCLE) :

> plots[display](g2, g4) ;

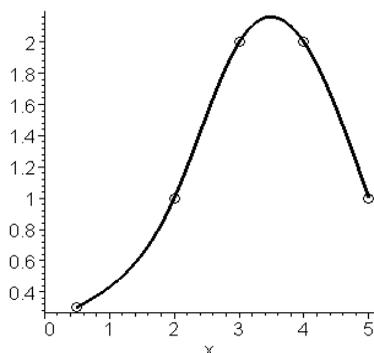


Рис. 2.4

2.5. Среднеквадратическое приближение

При интерполировании основным условием является прохождение графика интерполирующей функции через данные значения функции в узлах интерполяции. Однако в ряде случаев выполнение этого условия затруднительно или даже нецелесообразно. Например, как отмечалось ранее, при большом числе узлов, когда используется глобальная интерполяция, получается высокая степень многочлена, что часто оборачивается увеличением погрешности аппроксимации вплоть до полной несостоятельности приближения. Кроме того, табличные данные могли быть получены путем измерений и содержать ошибки. Построение аппроксимирующей функции с условием обязательного её графика через эти экспериментальные точки означало бы тщательное повторение допущенных при измерениях ошибок. Выход из этого положения может быть найден выбором такой функции, график которой проходит близко от данных точек. Понятие «близко» уточняется для конкретных видов аппроксимации.

Одним из таких видов является среднеквадратическое приближение, для которого мерой отклонения аппроксимирующей функции $g(x)$ от заданной функции $f(x)$ на множестве

точек (x_k, y_k) ($k = 1, 2, \dots, n$) является величина, равная сумме квадратов разностей между значениями в данных точках:

$$S = \sum_{k=1}^n [g(x_k) - y_k]^2.$$

Аппроксимирующую функцию нужно подобрать так, чтобы величина S была наименьшей. В этом состоит метод наименьших квадратов.

Пусть вид зависимости $g(x)$ однозначно определяется выбором параметров $\alpha_1, \alpha_2, \dots, \alpha_m$. Тогда величину S можно рассматривать как функцию многих независимых переменных, в роли которых выступают эти самые параметры:

$$S(\alpha_1, \alpha_2, \dots, \alpha_m) = \sum_{k=1}^n [g(x_k, \alpha_1, \alpha_2, \dots, \alpha_m) - y_k]^2. \quad (2.8)$$

Условия минимума функции $S = S(\alpha_1, \alpha_2, \dots, \alpha_m)$ требуют, чтобы её частные производные по всем переменным равнялись нулю:

$$\frac{\partial S}{\partial \alpha_1} = 0, \quad \frac{\partial S}{\partial \alpha_2} = 0, \quad \dots, \quad \frac{\partial S}{\partial \alpha_m} = 0.$$

Записанные соотношения – система уравнений для определения параметров $\alpha_1, \alpha_2, \dots, \alpha_m$.

Наиболее часто при применении метода наименьших квадратов функция $g(x)$ является линейной по неизвестным параметрам $\alpha_1, \alpha_2, \dots, \alpha_m$:

$$g(x, \alpha_1, \alpha_2, \dots, \alpha_m) = \alpha_1 \varphi_1(x) + \alpha_2 \varphi_2(x) + \dots + \alpha_m \varphi_m(x) = \sum_{i=1}^m \alpha_i \varphi_i(x).$$

В этом случае система алгебраических уравнений тоже является линейной, что, конечно же, важно при реализации вычислительного процесса. Часто в качестве такой аппроксимирующей функции используется полином степени $m - 1$:

$$g(x, \alpha_1, \alpha_2, \dots, \alpha_m) = \alpha_1 + \alpha_2 x + \dots + \alpha_m x^{m-1} = \sum_{i=1}^m \alpha_i x^{i-1},$$

причем $m < n$; случай $m = n$ соответствует глобальной интерполяции. На практике стараются подобрать аппроксимирующую функцию как можно более простого вида, например многочлен степени 1, 2, 3, 4.

Пример. Аппроксимировать по методу наименьших квадратов функцию, заданную в виде таблицы, линейной и квадратичной функцией.

x	0.5	2	3	4	5
$f(x)$	0.6	1	1.4	1.6	1.55

Решение.

> **restart;**

Задаём абсциссы и ординаты известных точек, а также их количество:

> **X:=[0.5,2,3,4,5]: Y:=[0.6,1,1.4,1.6,1.55]: n:=5:**

Построим приближение квадратичной функцией: $f(x) \approx a + bx + cx^2$. Для этого вводим аппроксимирующую функцию в виде многочлена 2-й степени, имеющего три параметра a , b и c :

> **p:=a+b*x+c*x^2; # (1)**

$$p := cx^2 + bx + a$$

Ставится задача определить эти параметры. Составляем величину S по формуле (2.8):

> **S:=sum((subs(x=X['k'],p)-Y['k'])^2,'k'=1..n);**

$$S := (a + 0.5b + 0.25c - 0.6)^2 + (a + 2b + 4c - 1)^2 + (a + 3b + 9c - 1.4)^2 + (a + 4b + 16c - 1.6)^2 + (a + 5b + 25c - 1.55)^2$$

Решаем систему $\partial S / \partial a = 0$, $\partial S / \partial b = 0$, $\partial S / \partial c = 0$ относительно параметров a , b и c :

> **r:=solve({diff(S,a)=0,diff(S,b)=0,diff(S,c)=0},{a,b,c}); # (2)**

$$r := \{a = 0.3399584847, b = 0.4712636222, c = -0.04392838609\}$$

Окончательно формируем аппроксимирующую функцию, подставляя в многочлен полученные значения параметров:

> **w:=unapply(subs(r,p),x);**

```

w := x → -0.04392838609 x2 + 0.4712636222 x + 0.3399584847
> xmin:=min(X[i]$i=1..n) : xmax:=max(X[i]$i=1..n) :
> d:=(xmax-xmin)/n/3.;
> g8:=plot(w(x), x=xmin-d..xmax+d,
color=black) : # построение графика и заданных точек
> g2:=plot([[X[i],Y[i]]$i=1..n], style=POINT,
symbol=CIRCLE) :
> plots[display](g2,g8) ;

```

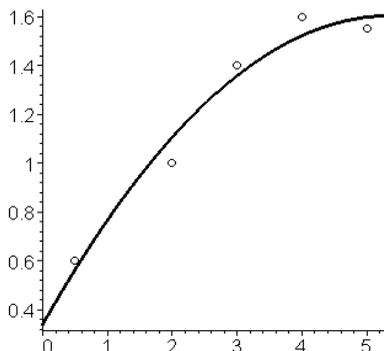


Рис. 2.7

Для аппроксимации в виде линейной функции $y = a + bx$ в приведенном блоке команд достаточно заменить строку (1) на

```
> p:=a+b*x;
```

Соответственно команда под номером (2) хотя и должна получить вид

```
> r:=solve({diff(S,a)=0,diff(S,b)=0},{a,b});
```

тем не менее, в силу избыточности может оставаться в старом варианте – результат окажется верным.

В итоге после запуска всего модифицированного блока получим искомую функцию:

$$w := x \rightarrow 0.5608606557 + 0.2307377049x$$

и её график

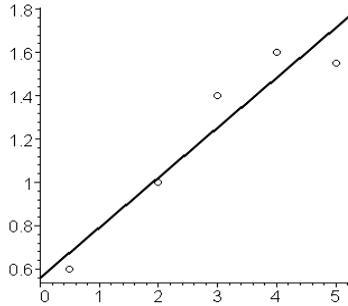


Рис. 2.8

Можно определить реальное числовое значение величины S (суммы квадратов отклонений) для полученных аппроксимаций с помощью команды

```
> S:=sum( (w(X['k'])-Y['k'])^2, 'k'=1..n) ;
```

В первом случае это значение равно $S_1 = 0.02275$, во втором – $S_2 = 0.06847$. Если S считать величиной, характеризующей погрешность аппроксимации, то можно сделать вывод, что данная зависимость лучше аппроксимируется параболой, нежели прямой. Но в общем случае к таким утверждениям следует подходить с осторожностью. Действительно, для глобальной интерполяции мы бы вообще получили величину S , равную нулю, но как отмечалось выше, такая аппроксимация не всегда правильно отражает поведение функции. Для адекватной оценки необходимо учитывать априорные сведения о функциональной зависимости, характер эмпирических данных, их вероятностное распределение и т.д.

Рассмотрим некоторые способы среднеквадратического приближения для нелинейной относительно параметров аппроксимирующей функции. Пусть $g(x) = ae^{bx}$. Действуя по обычной схеме, получим:

```
> p:=a*exp(b*x) ;
```

```
> S:=sum( (subs(x=X['k'],p)-Y['k'])^2, 'k'=1..n) ;
```

$$S := (a e^{(0.5b)} - 0.6)^2 + (a e^{(2b)} - 1)^2 + (a e^{(3b)} - 1.4)^2 + (a e^{(4b)} - 1.6)^2 + (a e^{(5b)} - 1.55)^2$$

> **r:=fsolve({diff(S,a)=0,diff(S,b)=0},{a=0..3,b=0..3}):**

> **w:=unapply(subs(r,p),x);**

$$w := x \rightarrow 0.7146300309 e^{(0.1759098190x)}$$

Обратите внимание, здесь из-за нелинейности системы приходится алгоритмы численного решения уравнений, для чего надо задавать интервалы изменения переменных, включающие решение. Для определения этих интервалов в случае двух параметров аппроксимации может помочь графическая визуализация. Для трёх и более параметров нужны дополнительные соображения о локализации решения. Однако, несмотря на все усилия, случается, что метод не сходится и не дает никакого результата. Это связано с тем, что сходимость решения систем нелинейных уравнений зависит слишком от многих факторов, которые не всегда удастся учесть с помощью универсального алгоритма. Так что среднеквадратическое приближение нелинейной функцией может столкнуться с серьезными трудностями вычислительного характера.

Поэтому на практике по возможности пытаются перейти к линейной аппроксимирующей зависимости. В нашем случае это достигается логарифмированием: $\ln g(x) = \ln a + bx$. Если ввести обозначения $A \equiv \ln a$, $\hat{g}(x) \equiv \ln g(x)$, получим новую функцию $\hat{g}(x) = A + bx$, аппроксимирующую табличные данные $(x_k, \ln y_k)$ ($k = 1, \dots, n$). Очевидно, эта функция уже линейна по своим параметрам A и b . Применив метод наименьших квадратов, найдем оптимальные значения этих параметров A^* и b^* , которые относятся к функции $\hat{g}(x)$, а исходную функцию $g(x)$ определим по формуле $g(x) = e^{A^*} \cdot e^{b^*x}$. Реализация на языке Maple (без вывода промежуточных результатов):

> **x:=[0.5,2,3,4,5]:y:=[0.6,1,1.4,1.6,1.55]:n:=5;**

```

> Y1:=[ln(Y[i])$i=1..n];
> p:=A+b*x;
> S:=sum((subs(x=X['k'],p)-Y1['k'])^2,'k'=1..n);
> r:=solve({diff(S,A)=0,diff(S,b)=0},{A,b});
           r:={b=0.2210629565,A=-0.4943015392}
> w1:=unapply(exp(subs(r,p)),x);
           w1:=x → e(-0.4943015392+0.2210629565x)
> xn:=min(X[i]$i=1..n)-0.5: xx:=max(X[i]$i=1..n)+0.3:
> g8:=plot(w1(x),x=xn..xx,color=black,thickness=3):
> g2:=plot([[X[i],Y[i]]$i=1..n],style=POINT,
symbol=BOX,color=black):
> plots[display](g2,g8);

```

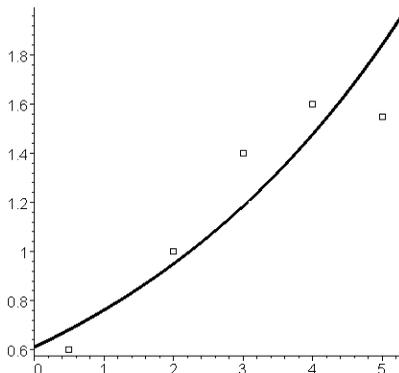


Рис. 2.9

Как видно из итоговых аналитических зависимостей, две последние аппроксимации с одной и той же функциональной зависимостью $g(x) = ae^{bx}$, но полученные разными способами, все же отличаются. Подумайте, почему?

Упражнения

1. По имеющимся данным провести глобальное интерполирование многочленом минимальной степени; локальное интерполирование полиномами 2-й степени, а также линейным и кубическим сплайном. Найти значение функции в точке x , ис-

пользуя полученные результаты. Построить графическую иллюстрацию интерполирования.

Варианты 1

x	0,43	0,48	0,55	0,62	0,70	0,75
$f(x)$	2,956	4,469	7,051	4,477	7,477	4,440

В точке $x = 0,702$; $x = 0,512$; $x = 0,645$; $x = 0,608$.

Варианты 2

x	0,02	0,08	0,12	0,17	0,23	0,30
$f(x)$	2,760	2,876	2,862	2,743	2,486	2,187

В точке $x = 0,203$; $x = 0,102$; $x = 0,114$; $x = 0,285$.

Варианты 3

x	0,35	0,41	0,47	0,51	0,56	0,64
$f(x)$	2,753	3,236	4,994	7,405	8,375	7,628

В точке $x = 0,482$; $x = 0,436$; $x = 0,552$; $x = 0,526$.

Варианты 4

x	0,41	0,46	0,52	0,60	0,65	0,72
$f(x)$	2,6049	2,2645	5,2273	7,0611	5,6399	2,4490

В точке $x = 0,616$; $x = 0,487$; $x = 0,665$; $x = 0,537$.

Варианты 5

x	0,68	0,73	0,80	0,88	0,93	0,99
$f(x)$	4,7947	5,3024	3,1089	1,1361	2,0616	4,5193

В точке $x = 0,774$; $x = 0,896$; $x = 0,715$; $x = 0,955$.

Варианты 6

x	0,11	0,15	0,21	0,29	0,35	0,40
$f(x)$	6,6635	7,2391	6,9275	3,9707	2,6940	4,1193

В точке $x = 0,275$; $x = 0,314$; $x = 0,235$; $x = 0,332$.

Варианты 7

x	0,05	0,10	0,17	0,25	0,30	0,36
$f(x)$	4,0175	2,7006	3,4000	6,3252	7,3091	5,6915

В точке $x = 0,263$; $x = 0,143$; $x = 0,325$; $x = 0,085$.

Варианты 8

x	0,51	0,55	0,61	0,69	0,75	0,80	0,84
$f(x)$	6,127	4,623	2,814	4,347	7,156	6,427	3,747

В точке $x = 0,749$; $x = 0,585$; $x = 0,67$; $x = 0,53$.

Варианты 9

x	0,115	0,159	0,218	0,294	0,353	0,408
$f(x)$	1,0352	1,0050	0,8945	0,6162	0,3327	0,1043

В точке $x = 0,256$; $x = 0,3$; $x = 0,188$; $x = 0,36$.

Варианты 10

x	2,0	2,3	2,5	3,0	3,5	3,8	4,0
$f(x)$	3,189	3,620	3,665	2,825	1,826	1,470	1,325

В точке $x = 2,22$; $x = 2,78$; $x = 3,34$; $x = 3,75$.

2. Набор экспериментальных данных значений x_i и y_i имеет вид таблицы:

1	x_i	1,20	1,57	1,94	2,31	2,68	3,05	3,42	3,79
	y_i	2,59	2,06	1,58	1,25	0,91	0,66	0,38	0,21
2	x_i	1,73	2,56	3,39	4,22	5,05	5,89	6,70	7,53
	y_i	0,63	1,11	1,42	1,94	2,30	2,89	3,29	3,87
3	x_i	-4,38	-3,84	-3,23	-2,76	-2,22	-1,67	-1,13	-0,60
	y_i	2,25	2,83	3,44	4,31	5,29	6,55	8,01	10,04
4	x_i	1,00	1,64	2,28	2,91	3,56	4,19	4,84	5,48
	y_i	0,28	0,19	0,15	0,11	0,09	0,08	0,07	0,06
5	x_i	5,84	3,82	6,19	9,22	7,87	6,29	4,43	8,91
	y_i	79,31	57,43	60,66	92,55	90,12	71,30	70,50	91,25
6	x_i	2,91	2,94	6,35	6,58	3,80	6,43	0,57	5,96
	y_i	82,16	61,02	44,56	82,52	99,17	70,24	63,23	66,48
7	x_i	5,46	2,73	6,49	4,26	2,39	6,46	0,86	2,05
	y_i	65,72	58,05	60,05	55,79	50,83	47,69	44,49	59,74
8	x_i	1,28	1,76	2,24	2,72	3,20	3,68	4,16	4,64
	y_i	2,10	2,62	3,21	3,96	4,98	6,06	7,47	9,25
9	x_i	-4,84	-4,30	-3,76	-3,22	-2,68	-2,14	-1,60	-1,06

	y_i	-0,09	-0,11	-0,13	-0,16	-0,19	-0,26	-0,39	-0,81
10	x_i	3,54	4,29	4,78	3,99	1,13	6,29	1,89	3,27
	y_i	22,81	28,42	24,95	26,96	8,78	33,55	15,77	22,89
11	x_i	4,08	4,42	2,52	-0,08	2,14	3,36	7,35	5,00
	y_i	18,31	21,85	16,93	8,23	10,90	17,18	36,45	24,11
12	x_i	1,16	1,88	2,60	3,32	4,04	4,76	5,48	6,20
	y_i	0,18	0,26	0,32	0,36	0,40	0,43	0,95	0,85
13	x_i	1,00	1,71	2,42	3,13	3,84	4,55	5,26	5,97
	y_i	12,49	4,76	2,55	1,60	1,11	0,82	0,63	0,50
14	x_i	-0,64	-0,36	-0,08	0,20	0,48	0,76	1,04	1,32
	y_i	29,51	18,86	12,05	7,70	4,92	3,14	2,01	1,28
15	x_i	-2,45	-1,94	-1,43	-0,92	-0,41	0,10	0,61	1,12
	y_i	0,87	1,19	1,68	2,23	3,04	4,15	5,66	7,72
16	x_i	1,54	1,91	2,28	2,65	3,02	3,39	3,76	4,13
	y_i	-2,52	-3,08	-3,54	-3,93	-4,27	-4,57	-4,84	-5,09
17	x_i	1,20	2,00	2,80	3,60	4,40	5,20	6,00	6,80
	y_i	-10,85	-6,15	-4,14	-3,02	-2,30	-1,81	-1,45	-1,17
18	x_i	-1,04	-0,67	-0,30	0,07	0,44	0,81	1,18	1,55
	y_i	10,80	8,08	5,97	4,44	3,31	2,46	1,83	1,36
19	x_i	0,41	0,97	1,53	2,09	2,65	3,21	3,77	4,33
	y_i	0,45	1,17	1,56	1,82	2,02	2,18	2,31	2,44
20	x_i	1	2	5	8	9	12	14	16
	y_i	6	7,45	8,24	12,46	13,09	14,56	25,89	29,91
21	x_i	0	2	4	5	8	10	12	15
	y_i	29,8	22,9	17,1	15,16	10,7	10,2	10,1	15,2
22	x_i	1,65	1,39	1,19	1,02	0,82	0,75	0,66	0,89
	y_i	1,034	1,232	1,432	1,752	2,056	2,37	2,76	3,198
23	x_i	-0,22	-3,05	-1,76	-1,25	-0,45	-0,80	-0,26	-3,07
	y_i	58,46	36,05	31,17	16,17	11,16	69,23	58,08	43,13
24	x_i	6,35	6,58	3,80	6,43	0,57	5,96	3,40	4,55

	y_i	44,56	82,52	99,17	70,2	63,23	66,48	48,35	40,24
25	x_i	2,95	2,60	2,69	3,01	2,44	2,51	3,37	2,98
	y_i	113,8	119,6	106,2	120	107,4	114,8	115,5	117,4
26	x_i	4,23	1,40	4,07	2,93	3,44	1,09	1,82	2,43
	y_i	65,72	58,05	60,05	55,79	50,83	47,69	44,49	59,74
27	x_i	4,07	2,93	3,44	1,09	1,82	2,43	3,85	0,97
	y_i	60,05	55,79	50,83	47,69	44,49	59,74	56,81	45,82
28	x_i	8,90	9,22	8,74	8,98	8,77	9,31	8,81	9,14
	y_i	105,2	85,02	52,76	56,86	72,19	61,09	70,44	51,67
29	x_i	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
	y_i	0,095	0,182	0,262	0,337	0,406	0,470	0,588	0,642
30	x_i	2,51	3,74	8,70	5,36	1,89	3,01	3,59	2,64
	y_i	55,65	67,68	105,2	85,02	52,76	56,86	72,19	61,09

Найти методом наименьших квадратов аппроксимирующие зависимости в виде линейных комбинаций функций следующих систем:

а) $1, x, x^2, x^3, x^4, \dots$

(т.е. в виде $g(x) = \alpha_1 + \alpha_2 x$, $g(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2$, $g(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$ и т.д.);

б) $1, \cos \frac{2\pi x}{l}, \sin \frac{2\pi x}{l}, \cos \frac{4\pi x}{l}, \sin \frac{4\pi x}{l}, \dots$ (здесь $l = b - a$; $b = \max_i x_i$; $a = \min_i x_i$);

в) $1, e^x, e^{2x}, e^{3x}, e^{4x}, \dots$;

Сравнить качество полученных приближений путем сравнения их отклонений. Построить графики получившихся зависимостей и табличных значений аргументов и функции. Сделать выводы о характере аппроксимирующей функции.

3. МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ

Предположим, что решается одномерная линейная краевая задача для уравнения 2-го порядка, т.е. требуется опреде-

лить функцию $y(x)$, удовлетворяющую заданному дифференциальному уравнению

$$y''(x) + p(x)y'(x) + q(x)y(x) = f(x) \quad (3.1)$$

на отрезке $a \leq x \leq b$ вместе с краевыми условиями при $x = a$ и $x = b$

$$\begin{aligned} \alpha_0 y'(a) + \beta_0 y(a) &= \gamma_0, \\ \alpha_1 y'(b) + \beta_1 y(b) &= \gamma_1 \end{aligned} \quad (3.2)$$

($\alpha_i^2 + \beta_i^2 \neq 0$, $i = 0, 1$). Для решения этой задачи методом конечных разностей (МКР) прежде всего производится дискретизация независимой переменной x , т.е. строится множество (или сетка) $n + 1$ точек (узлов) x_k ($k = 0, 1, 2, \dots, n$) на отрезке $a \leq x \leq b$, причем $x_0 = a$, $x_n = b$. Шаг сетки $h_k \equiv x_{k+1} - x_k$ из-за удобства программирования чаще всего берется постоянным ($h_1 = h_2 = \dots = h_n \equiv h$), но может быть и переменным (особенно когда заранее известен характер поведения решения).

Следующий этап состоит в замене в дифференциальном уравнении членов, содержащих дифференцирование, разностными отношениями. На основе формулы Тейлора можно получить выражения для производных функции

$$\left(\frac{dy}{dx}\right)_k = \frac{y_{k+1} - y_{k-1}}{2h} + O(h^2), \quad \left(\frac{d^2y}{dx^2}\right)_k = \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + O(h^2),$$

где $h = x_{k+1} - x_k = x_k - x_{k-1}$, $y_k \equiv y(x_k)$. Подставляя эти выражения в уравнение (3.1) для всех (может быть, за исключением граничных) узлов $\{x_k\}$, а также в уравнения граничных условий, получим систему линейных алгебраических уравнений относительно узловых значений $\{y_k\}$.

Для оценки точности получаемого решения необходимо провести расчеты для разных значений шага (не менее 3) и убедиться в том, что полученные значения функции в одних и тех же узлах близки между собой и разность их уменьшается, что говорит о стремлении решения к некоторому пределу при $h \rightarrow 0$.

Следует отметить, что при определенных наборах коэффициентов уравнения (3.1) $p(x)$, $q(x)$, $f(x)$ и граничных условий α_i , β_i , γ_i решение рассматриваемой краевой задачи может не существовать или быть не единственным. Кроме того, даже если оно и существует и единственно, то МКР может быть неустойчивым. Не касаясь деталей этого вопроса, ниже мы будем предполагать безусловный характер устойчивости как задачи, так и метода.

МКР может быть с успехом распространен на нелинейные дифференциальные уравнения, на системы уравнений, уравнения в частных производных, а также адаптирован к решению задач Коши для различных дифференциальных уравнений. В любом случае решение дифференциального уравнения конечно-разностным методом сводится к решению системы алгебраических уравнений относительно значений искомой функции на заданном множестве точек. Эта система является линейной или нелинейной в зависимости от того, линейно или нелинейно дифференциальное уравнение. Следует обратить внимание, что МКР непосредственно дает значения функции только на дискретном множестве узловых точек, а решение во всех остальных точках области задачи можно получить с помощью какого-нибудь интерполяционного метода. Из-за этого в точность решения будет вноситься дополнительная погрешность аппроксимации.

Предложенный ниже пример позволяет проследить процесс построения разностной схемы и выявить ключевые моменты реализации МКР.

Пример. Методом конечных разностей решить краевую задачу

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \frac{2}{x}; \quad \varphi(1) = 0, \quad \varphi'(2) = 1.$$

Решение. В соответствии с МКР в исходном уравнении

$$x^2 \frac{d^2\varphi}{dx^2} + 2x \frac{d\varphi}{dx} - 2\varphi = -1 - 2/x$$

вместо первой и второй производной записываются аппроксимации:

$$[\varphi''(x)]_i \approx \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{h^2}, \quad [\varphi'(x)]_i \approx \frac{\varphi_{i+1} - \varphi_{i-1}}{2h}.$$

Разобьем отрезок $[1, 2]$ на четыре равные части. Получим 5 узлов $x_i = 1 + hi$; $i = 0, 1, \dots, 4$; $h = 0,25$. Запишем последовательно уравнения для всех узлов, начиная с x_1 (для узла $x_0 = 1$, которому соответствует индекс $i = 0$, не пишем, поскольку для него значение известно из граничного условия: $\varphi_0 = \varphi(1) = 0$).

Для $i = 1$ ($x = 1,25$) – первое уравнение:

$$1,25^2 \frac{\varphi_2 - 2\varphi_1 + \varphi_0}{0,25^2} + 2 \cdot 1,25^2 \frac{\varphi_2 - \varphi_0}{2 \cdot 0,25} - 2\varphi_1 = -1 - \frac{2}{1,25};$$

для $i = 2$ ($x = 1,5$) – второе уравнение:

$$1,5^2 \frac{\varphi_3 - 2\varphi_2 + \varphi_1}{0,25^2} + 2 \cdot 1,5 \frac{\varphi_3 - \varphi_1}{2 \cdot 0,25} - 2\varphi_2 = -1 - \frac{2}{1,5};$$

для $i = 3$ ($x = 1,75$) – третье уравнение:

$$1,75^2 \frac{\varphi_4 - 2\varphi_3 + \varphi_2}{0,25^2} + 2 \cdot 1,75 \frac{\varphi_4 - \varphi_2}{2 \cdot 0,25} - 2\varphi_3 = -1 - \frac{2}{1,75};$$

для $i = 4$ ($x = 2$) – четвертое уравнение

$$2,0^2 \frac{\varphi_5 - 2\varphi_4 + \varphi_3}{0,25^2} + 2 \cdot 2,0 \frac{\varphi_5 - \varphi_3}{2 \cdot 0,25} - 2\varphi_4 = -1 - \frac{2}{2,0}.$$

Здесь φ_5 – значение неизвестной функции в фиктивном узле $x_5 = 2,25$.

Имеем четыре уравнения при пяти неизвестных ($\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5$). Недостающее пятое уравнение получим, если аппроксимируем второе граничное условие, соответствующее узлу с номером $i = 4$:

$$\frac{\varphi_5 - \varphi_3}{2 \cdot 0,25} = 1.$$

Итак, решение полученной системы даст приближенные значения искомой функции в точках $x = 1,25$; $x = 1,5$; $x = 1,75$; $x = 2$. Найти значения в других точках отрезка $[1, 2]$ можно с помощью методов интерполяции.

Порядок решения в системе Maple

> **restart;**

1. Введем переменную **deq** для хранения дифференциального уравнения:

> **deq:=x^2*diff(y(x),x\$2)+2*x*diff(y(x),x)-2*y(x)=-1-2/x;**

$$deq := x^2 \left(\frac{d^2}{dx^2} y(x) \right) + 2x \left(\frac{d}{dx} y(x) \right) - 2y(x) = -1 - \frac{2}{x}$$

(**y(x)** здесь обозначает $\varphi(x)$).

2. Определим конечно-разностные уравнения.

Первое уравнение (переменная **eq1**) получим, осуществляя подстановку в **deq** вместо $\varphi''(x)$ и $\varphi'(x)$ аппроксимации $(\varphi_2 - 2\varphi_1 + \varphi_0)/0,25^2$ и $(\varphi_2 - \varphi_0)/(2 \cdot 0,25)$ соответственно, вместо **y(x)** узловую переменную **y1** (т.е. φ_1) и, наконец, вместо **x** значение 1,25:

> **eq1:=subs(diff(y(x),x\$2)=(y2-2*y1+y0)/0.25^2, diff(y(x),x)=(y2-y0)/(2*0.25), y(x)=y1,x=1.25,deq);**

$$eq1 := 30.00000000y2 - 52.00000000y1 + 20.00000000y0 = -2.600000000$$

Аналогично зададим следующие три уравнения для точек $x = 1,5$; $x = 1,75$; $x = 2$ – переменные **eq2**, **eq3** и **eq4** соответственно:

> **eq2:=subs(diff(y(x),x\$2)=(y3-2*y2+y1)/0.25^2, diff(y(x),x)=(y3-y1)/(2*0.25), y(x)=y2,x=1.5,deq);**
eq2 := 42.00000000y3 - 74.00000000y2 + 30.00000000y1 = -2.333333333

```

> eq3:=subs(diff(y(x),x$2)=(y4-2*y3+y2)/0.25^2,
diff(y(x),x)=(y4-y2)/(2*0.25),y(x)=y3,x=1.75,deq);
eq3:=56.00000000y4-100.00000000y3+42.00000000y2=-2.142857143
> eq4:=subs(diff(y(x),x$2)=(y5-2*y4+y3)/0.25^2,
diff(y(x),x)=(y5-y3)/(2*0.25),y(x)=y4,x=2.0,deq);
eq4:=72.00000000y5-130.00000000y4+56.00000000y3=-2.000000000

```

Пятое уравнение – это аппроксимация граничного условия $\varphi'(2)=1$:

```

> eq5:=(y5-y3)/(2*0.25)=1;
eq5:=2.000000000y5-2.000000000y3=1

```

3. Решим эту систему с помощью функции **solve**. При этом следует положить $\varphi_0=0$ (или $y_0=0$), что следует из первого граничного условия:

```

> rez:=solve({y0=0,eq1,eq2,eq3,eq4,eq5},{y0,y1,y2,
y3,y4,y5});

```

Получим результат:

```

rez:={y0=0.,y1=0.7529029084,y4=1.821710593,y2=1.218365041,
y5=2.053299821,y3=1.553299821}

```

(y_0, y_1, \dots здесь обозначают $\varphi_0, \varphi_1, \dots$). Это и есть конечно-разностное решение, которое непосредственно дает значения искомой функции лишь в отдельных точках – узлах сетки $\{x_k\}$.

4. Сравним с точным решением:

```

> st:=dsolve({deq,y(1)=0,D(y)(2)=1},y(x));
> u:=unapply(subs(st,y(x)),x);

```

$$u := x \rightarrow \frac{11}{5} \frac{1}{x^2} + \frac{7}{10} x + \frac{1}{2} \frac{2+x}{x}$$

Выведем найденные числа $\{y_i\}$ и значения функции $u(x)$ в тех узлах, для которых найдено конечно-разностное решение:

```

> X:=[1,1.25,1.5,1.75,2];
Y:=subs(rez,[y0,y1,y2,y3,y4]);
Y:=[0,0.7529029084,1.218365041,1.553299821,1.821710593]
> n:=5: # число узлов сетки
> for j from 1 to n do printf(`x=%4.2g      Y=%7.5g
u=%7.5g\n`,X[j],Y[j],evalf(u(X[j]))); od;

```

x=1.00	Y=0.00000	u=0.00000
x=1.25	Y= .75290	u= .76700
x=1.50	Y=1.21836	u=1.23888
x=1.75	Y=1.55329	u=1.57806
x=2.00	Y=1.82171	u=1.85000

Совпадение есть. Погрешность $\sim 1\%$.

5. Выполним построения графика точного решения и полученных приближенных узловых значений.

```
> i:='i';
> g3:=plot([[X[i],Y[i]]$i=1..n],x=1..2,style=POINT,
symbol=CIRCLE,color=black);
> plots[display](g3);
```

(Графическое построение не приводится).

6. Найдем приближенное решение в виде непрерывной функции с помощью интерполяционного метода.

Сначала проведем глобальную интерполяцию:

```
> Digits:=6: f:=interp(X,Y,a);
f:=-0.990296a4+7.12002a3-12.2729-19.6536a2+25.7968a
> fu:=proc(x) global a; a:=x; RETURN(f) end;
fu:=proc(x) global a; a:=x; RETURN(f) end proc
```

Построим графики интерполяционной функции $f_u(x)$ и точного решения вместе с узлами интерполяции

```
> g1:=plot(fu(x),x=1..2,
color=blue):g2:=plot(u(x),
x=1..2,color=red):
> plots[display](g1,g2,g3);
```

На практике часто (особенно при большом числе узлов n) вместо глобальной интерполяции предпочтительнее использовать локальную интерполяцию (линейную, квадратичную или сплайн-интерполяцию). Ниже приведен способ построения кусочно-линейной интерполяции.

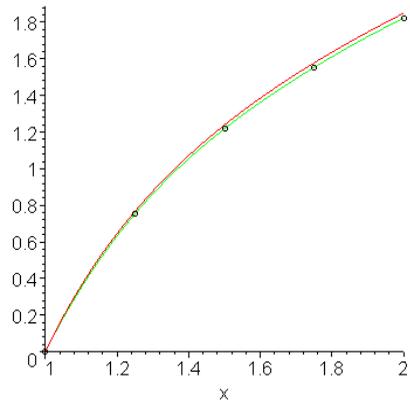


Рис. 3.1

```

> readlib(spline) ;
      proc(XX, YY, z, d) ... end proc
> f1:=spline(X,Y,x,linear) ;
      f1 := {
      -3.01161+3.01161x      x < 1.25
      -1.57444+1.86187x     x < 1.5
      -0.791200+1.33972x   x < 1.75
      -0.325560+1.07364x   otherwise
      }
> g4:=plot(f1,x=1..2,color=black,thickness=2) :
> g2:=plot(u(x),x=1..2,color=red) :
> g5:=plot([ [X[i],Y[i]]$i=1..n ],x=1..2.1,
style=POINT,symbol=CIRCLE) :
> plots[display](g2,g4,g5) ;

```

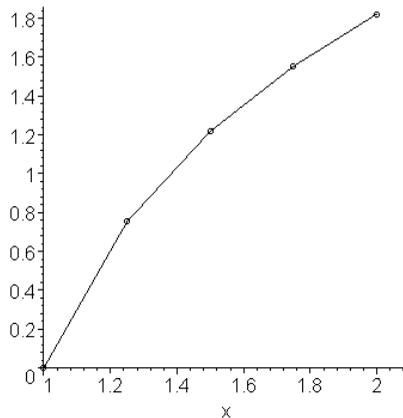


Рис. 3.2

7. Проверка сходимости метода конечных разностей.

Сходимость означает, что с уменьшением расстояния h между ближайшими узлами (а это приводит к увеличению числа узлов) точность МКР должна возрастать, т.е. должна стремиться к нулю разность $|y_i - u(x_i)|$ для любого узла с номером i при $h \rightarrow 0$. Таким образом, для проверки сходимости нужно провести серию расчетов по рассмотренной схеме МКР для разного числа узлов n .

Сделать вывод о сходимости можно визуально, отмечая число совпадающих знаков у точного и приближенного решений. Но лучше для этого использовать числовую характеристику, например

$$\delta_1 = \left\{ \frac{1}{n} \sum_{i=0}^n (y_i - y(x_i))^2 \right\}^{1/2} \quad \text{или} \quad \delta_2 = \max_i |y_i - y(x_i)|.$$

Следует, однако, заметить, что тот способ реализации МКР в системе Maple, что приведен выше, достаточно нагляден, но с ростом числа узлов становится крайне громоздким. Действительно, если это число равно n , то придется вручную вводить столько же уравнений. Попробуйте оценить, на сколько возрастет текст программы при $n = 10$, $n = 20$, $n = 100$.

Поэтому приведем другую, более универсальную, Maple-программу, дающую решение данной краевой задачи методом конечных разностей при любом n (т.е. n выступает как параметр):

```
> restart;
> Digits:=15;
> a:=1; b:=2; n:=10; h:=(b-a)/n;
```

Граничные условия зададим в общем виде (10), устанавливая для коэффициентов соответствующие значения (**A0** соответствует α_0 , **B0** соответствует β_0 , **C0** соответствует γ_0 и т.д.):

```
> A0:=0; B0:=4; C0:=0; A1:=1.; B1:=0; C1:=1;
> diffeqn:=x^2*diff(y(x),x$2)+2*x*diff(y(x),x)-
2*y(x)=-1-2/x;
> Eqns_bas:={seq(subs(diff(y(x),x$2)=
(g[k+1]-2*g[k]+g[k-1])/h^2,diff(y(x),x)=(g[k+1]-
g[k-1])/(2.*h),y(x)=g[k],
x=a+h*k,diffeqn),k=0..n)};
> eq1:={A0*(g[1]-g[-1])/2./h+B0*g[0]=C0}; # гранич-
ное условие в точке x = a
> eq2:={A1*(g[n+1]-g[n-1])/2./h+B1*g[n]=C1}; # гра-
ничное условие в точке x = b
> rez:=solve(Eqns_bas union eq1 union eq2,
{g[i]$i=-1..n+1}):
```

```

> Y:=evalf([seq(subs(rez,g[k]),k=0..n)]):
> X:=evalf[5]([seq(a+k*h,k=0..n)]):
> gr1:=plot([X[i],Y[i]]$i=1..n+1,x=a..b,
style=POINT,symbol=CIRCLE,color=black):
> readlib(spline); # – получение непрерывного решения с
помощью сплайн-интерполяции
> Digits:=10: f1:=spline(X,Y,x,linear):
> gr2:=plot(f1,x=a..b,color=blue):
> x:='x': y:='y':
> p:=dsolve({diffeqn,A0*D(y)(a)+B0*y(a)=C0,
A1*D(y)(b)+B1*y(b)=C1},y(x));
> u:=unapply(subs(p,y(x)),x); # u(x) – точное решение
> gr3:=plot(u(x),x=a..b,color=red):
> plots[display]([gr1,gr2,gr3])

```

Вычисление величин δ_1 и δ_2 – норм ошибки решения:

```

> delta1:=(1/(n+1)*sum( (Y['i]-u(X[i])) ^2',
'i'=1..n+1))^(0.5);
> delta2:=max( seq(abs(Y[i]-u(X[i])),i=1..n+1) );

```

Следующие строки используются для табулирования:

```

> np:=trunc(0.1/h); if(np=0) then np:=1; fi;
> for j from 1 by np to n+1 do printf(`x=%4.2g
Y=%8.5g    u=%8.5g\n`,
X[j],Y[j],evalf(u(X[j]))); od;

```

Результаты вычисления величин δ_1 и δ_2 в зависимости от n приведены в таблице.

	$n = 4$	$n = 8$	$n = 16$	$n = 50$	$n = 100$
δ_1	0,04509	0,01527	0,005263	$9,3449 \cdot 10^{-4}$	$3,2883 \cdot 10^{-4}$
δ_2	0,02829	0,007121	0,001784	$1,1828 \cdot 10^{-4}$	$4,5699 \cdot 10^{-5}$

Видно, что сходимостью данный метод конечных разностей обладает.

Упражнения

1. Решить методом конечных разностей одну из следующих краевых задач. Использовать не менее 5 узлов. Сравнить с точным решением. Построить графики. Провести аппроксимацию полученного решения по методу кусочно-линейной или кусочно-квадратичной интерполяции. Показать сходимость, увеличивая каждый раз число узлов в 2 раза.

Вариант 1. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2, \quad \varphi(1) = 0, \varphi(2) = 0.$

Вариант 2. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2, \quad \varphi'(1) = 0, \varphi'(2) = 0.$

Вариант 3. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2, \quad \varphi'(1) = 0, \varphi'(4) - \varphi(4) = -2.$

Вариант 4. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - 2 \ln x,$
 $\varphi(1) + \varphi'(1) = 0, \varphi(2) - \varphi'(2) = 0.$

Вариант 5. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - 2 \ln x,$
 $\varphi(1) - \varphi(2) = 2, \varphi'(2) + \varphi(1) = 1.$

Вариант 6. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 0, \quad \varphi'(1) = -1, \varphi'(2) = 1.$

Вариант 7. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 1 - \frac{1}{x}, \quad \varphi(1) = 0, \varphi'(2) = 1.$

Вариант 8. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 2 - \ln x, \quad \varphi(1) - 2\varphi'(1) = 0, \varphi'(2) = 1.$

Вариант 9. $\frac{d}{dx} \left((x+1)^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1, \quad \varphi(0) = 0, \varphi'(1) = 1.$

Вариант 10. $\frac{d}{dx} \left((x+1)^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln(x+1), \quad \varphi(0) = 0, \varphi'(2) = 1.$

2. Решить краевую задачу методом конечных разностей. Сравнить с точным решением. Показать сходимость. Построить график непрерывного решения на основе сплайн-интерполяции.

Вариант 1. $\frac{d}{dx} \left((x^2 + x + 1) \frac{d\varphi}{dx} \right) = x^2 - 1$; а) $\varphi(0) = 0$, $\frac{d\varphi}{dx}(1) = \frac{1}{3}$;

б) $\varphi(0) = 0$, $\varphi(1) = \frac{1}{3}$; в) $\varphi(0) + \frac{d\varphi}{dx}(0) = 0$, $\varphi(1) - \frac{d\varphi}{dx}(1) = -1$.

Вариант 2. $\frac{d}{dx} \left(e^x \frac{d\varphi}{dx} \right) = -2x$; а) $\frac{d\varphi}{dx}(0) = 0$, $\varphi(4) = 1$;

б) $\varphi(0) = 0$, $\varphi(4) = 1$; в) $2\varphi(0) - \frac{d\varphi}{dx}(0) = 0$, $\frac{d\varphi}{dx}(4) + \varphi(4) = -1$.

Вариант 3. $\frac{d}{dx} \left(\frac{1}{1+x^2} \frac{d\varphi}{dx} \right) = -2x$; а) $\varphi(0) = 0$, $\frac{d\varphi}{dx}(1) = -1$;

б) $\varphi(0) = 0$, $\varphi(1) = -1$; в) $\frac{d\varphi}{dx}(0) = -1$, $\frac{d\varphi}{dx}(1) + 4\varphi(1) = 2$.

Вариант 4. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) + 2\varphi = 2 - x$; а) $\varphi(1) = 1$, $\frac{d\varphi}{dx}(2) = -1$;

б) $\varphi(1) = 1$, $\varphi(2) = -1$; в) $\varphi(1) - \frac{d\varphi}{dx}(1) = 5$, $\frac{d\varphi}{dx}(2) + 2\varphi(2) = 0$.

Вариант 5. $\frac{d}{dx} \left(e^{-x} \frac{d\varphi}{dx} \right) + e^x \varphi = 0$; а) $\varphi(0) = 1$, $\frac{d\varphi}{dx}(1) = \frac{1}{2}$;

б) $\varphi(0) = 1$, $\varphi(1) = \frac{1}{2}$; в) $\varphi(0) - 5 \frac{d\varphi}{dx}(0) = -2$, $\frac{d\varphi}{dx}(1) + \varphi(1) = 2$.

Вариант 6. $\frac{d}{dx} \left((x+2) \frac{d\varphi}{dx} \right) - \frac{1}{x+2} \varphi = x^2 - \frac{1}{2}$, а) $\varphi(0) = -1$, $\frac{d\varphi}{dx}(2) = \frac{1}{2}$;

б) $\varphi(0) = 1$, $\varphi(2) = \frac{1}{2}$; в) $\frac{d\varphi}{dx}(0) - 2\varphi(0) = 1$, $\frac{d\varphi}{dx}(2) = \frac{1}{2}$.

Вариант 7. $\frac{d}{dx}\left(e^{-x} \frac{d\varphi}{dx}\right) - e^x \varphi = \frac{e^{2x}}{2}$; а) $\varphi(0) = 1$, $\frac{d\varphi}{dx}(1) = 3$;

б) $\varphi(0) = 1$, $\varphi(1) = 3$; в) $2\varphi(0) - 3\frac{d\varphi}{dx}(0) = 8$, $\frac{d\varphi}{dx}(1) + 2\varphi(1) = 2$.

Вариант 8. $\frac{d^2\varphi}{dx^2} = -\varphi + 1 - x + x^2$; а) $\frac{d\varphi}{dx}(1) = -1$, $\varphi(3) = 1$;

б) $\varphi(1) = -1$, $\varphi(3) = 1$; в) $\varphi(1) - \frac{d\varphi}{dx}(1) = -1$, $\frac{d\varphi}{dx}(2,5) + 2\varphi(2,5) = 1$.

Вариант 9. $\frac{d}{dx}\left(x^4 \frac{d\varphi}{dx}\right) - 4x^2\varphi = \frac{1}{x}$; а) $\frac{d\varphi}{dx}(1) = -3$, $\varphi(2) = 3$;

б) $\varphi(1) = -3$, $\varphi(1,9) = 3$; в) $\frac{d\varphi}{dx}(1) - 2\varphi(1) = -10$, $20\frac{d\varphi}{dx}(2) - \varphi(2) = 10$.

Вариант 10. $\frac{d}{dx}\left((e^x + 2) \frac{d\varphi}{dx}\right) = e^{x+2}$; а) $\frac{d\varphi}{dx}(0) = -10$, $\varphi(2) = \frac{1}{2}$;

б) $\varphi(0) = 10$, $\varphi(4) = \frac{1}{2}$; в) $\frac{d\varphi}{dx}(0) + 2\varphi(0) = \frac{15}{2}$, $20\frac{d\varphi}{dx}(2) + \varphi(2) = 10$.

3. С помощью МКР найти решение краевой задачи, не имеющей аналитического решения. Показать сходимость.

Вариант 1. $y''(x) + (x+1)y'(x) - 2y(x) = 2(2x-1)$,
 $4y(0,5) - y'(0,5) = 2$, $y(1) + y'(1) = 0$.

Вариант 2. $y''(x) + (x-1)y'(x) - 2y(x) = -4x$, $4y(0,5) - y'(0,5) = 2$,
 $y(1) = 3$.

Вариант 3. $y''(x) + \frac{2}{x+1}y'(x) - (x+1)y(x) = -1$, $y(0,5) = 2/3$,
 $y'(1) + 2y(1) = 0,75$.

Вариант 4. $y''(x) - (x+1)y'(x) - y(x) = \frac{2}{(x+1)^3}$,
 $y(0,5) - y'(0,5) = 2$, $y(1) = 0,5$.

Вариант 5. $y''(x) + \frac{4x}{x^2 + 1} y'(x) - \frac{1}{x^2 + 1} y(x) = -\frac{3}{(x^2 + 1)^2}$,
 $y'(0) = 0$, $y(1) = 0,5$.

Вариант 6. $y''(x) + \frac{2}{x-2} y'(x) + (x-2)y(x) = 1$, $y(0) = -0,5$,
 $y(1) = -1$.

Вариант 7. $x^2 \ln x \cdot y''(x) - xy'(x) + y(x) = 0$, $y(1) = 0$,
 $y(e) = e - 2$.

Вариант 8. $y''(x) - \ln x \cdot y'(x) - 2y(x) = 1$, $y(0,5) + y'(0,5) = 1$,
 $y(1,5) - y'(1,5) = 1$.

Вариант 9. $y''(x) + y'(x) - \frac{2}{\cos^2 x} y(x) = \frac{2}{\cos^2 x}$, $y(0) = 0$,
 $y'(0,5) = 1/\cos^2 0,5$.

Вариант 10. $y''(x) - \frac{2}{x} y'(x) - \frac{4}{x^2 + 2} y(x) = 8$, $y'(0,5) = 0,5$,
 $y(1) + y'(1) = 1$.

4*. Решить нелинейное уравнение $e^{-y} y'' = 1$ с краевыми условиями $y(0) = y(1) = 0$, используя сетку с шагом не более $h = 0,25$. (Это уравнение типично для задач о распространении тепла в химически реагирующих материалах, для которых плотность источника тепла пропорциональна e^y , где y – температура).

Указание. Для решения подобных задач используются стандартные итерационные (с последовательным уточнением) процедуры, сводящиеся к многократному решению систем линейных алгебраических уравнений. Наиболее известные из таких процедур – метод простой итерации и метод Ньютона.

4. МЕТОД РИТЦА

4.1. О прямых методах вариационного исчисления

Дифференциальные уравнения (Эйлера) вариационных задач интегрируются в конечном виде лишь в исключительных случаях. В связи с этим естественно возникает потребность в иных методах решения этих задач. Основная идея так называемых *прямых методов* заключается в том, что вариационная задача рассматривается как предельная для некоторой задачи на экстремум функции конечного числа переменных. Эта задача на экстремум функции конечного числа переменных решается обычными методами математического анализа и алгебры.

Функционал $F[y(x)]$ можно рассматривать как функцию бесконечного множества переменных. Это утверждение становится совершенно очевидным, если предположить, что допустимые функции могут быть разложены в степенные ряды

$$y(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n + \dots,$$

или в ряды Фурье

$$y(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx),$$

или вообще в какие-нибудь ряды вида

$$y(x) = \sum_{n=0}^{\infty} a_n \varphi_n(x), \quad (4.1)$$

где $\varphi_n(x)$ – заданные функции. Для задания функции $y(x)$, представимой в виде ряда (4.1), достаточно задать значения всех коэффициентов a_n , и, следовательно, значение функционала $F[y(x)]$ в этом случае определяется заданием бесконечной последовательности чисел $a_0, a_1, a_2, \dots, a_n, \dots$, т. е. функционал является функцией бесконечного множества переменных:

$$F[y(x)] = \Phi(a_0, a_1, \dots, a_n, \dots).$$

Следовательно, различие между вариационными задачами и задачами на экстремум функций конечного числа переменных состоит в том, что в вариационном случае приходится исследовать на экстремум функции бесконечного множества переменных.

Идея *метода Ритца* заключается в том, что значения некоторого функционала $F[y(x)]$ рассматриваются не на произвольных допустимых кривых данной вариационной задачи, а лишь на всевозможных линейных комбинациях вида

$$y_m = \sum_{i=1}^m \alpha_i N_i(x)$$

с постоянными коэффициентами, составленных из m первых функций некоторой выбранной последовательности функций, называемых координатными или *базисными*,

$$N_1(x), N_2(x), \dots, N_m(x), \dots$$

Функции $y_m = \sum_{i=1}^m \alpha_i N_i(x)$ должны быть допустимыми в рассматриваемой задаче, что налагает некоторые ограничения на выбор последовательности функций $N_i(x)$. На таких линейных комбинациях функционал $F[y(x)]$ превращается в функцию $\Phi(\alpha_1, \alpha_2, \dots, \alpha_m)$ коэффициентов $\alpha_1, \alpha_2, \dots, \alpha_m$. Эти коэффициенты выбираются так, чтобы функция $\Phi(\alpha_1, \alpha_2, \dots, \alpha_m)$ достигала экстремума; следовательно, $\alpha_1, \alpha_2, \dots, \alpha_m$ должны быть определены из системы уравнений

$$\partial\Phi/\partial\alpha_i = 0, \quad i = 1, 2, \dots, m.$$

Рассмотрим конкретные подходы к решению некоторых задач методом Ритца.

4.2. Простейшая задача вариационного исчисления.

Краевая задача 1-го рода

В этой задаче требуется найти функцию $y(x)$, доставляющую экстремум функционалу

$$F[y(x)] = \int_a^b f(x, y, y') dx$$

при условиях $y(a) = y_0$, $y(b) = y_1$. Одновременно эта функция будет являться решением краевой задачи

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \frac{\partial f}{\partial y'} = 0; \quad y(a) = y_0, \quad y(b) = y_1.$$

Граничные условия 1-го рода обычно играют роль главных условий. В этом случае множество допустимых функций, на которых ищется экстремум, приходится ограничивать только теми функциями, которые априори удовлетворяют этим условиям. Поэтому, если в исходной краевой или вариационной задаче имеются граничные условия 1-го рода

$$y(a) = y_0 \quad \text{и/или} \quad y(b) = y_1,$$

то непосредственный учет этих условий следует производить путем ограничения множества пробных функций вариационной задачи. Применительно к методу Ритца это означает выбор такой аппроксимации $y_m = \sum_{i=1}^m \alpha_i N_i(x)$, которая заведомо удовлетворяет указанным условиям вне зависимости от того, какие значения принимают параметры $\{\alpha_i\}$.

Если граничные условия однородны, т.е. $y(a) = y(b) = 0$, то проще всего в качестве базисных функций выбрать функции, удовлетворяющие этим граничным условиям: $N_i(a) = N_i(b) = 0$, $i = 1, 2, \dots, m$. Например,

$$N_i(x) = (x - a)(x - b)\varphi_i(x),$$

где $\varphi_i(x)$ – какие-нибудь непрерывные функции, или

$$N_k(x) = \sin \frac{k\pi(x - a)}{b - a} \quad (k = 1, 2, \dots).$$

Очевидно, что при этом и аппроксимация $y_m = \sum_{i=1}^m \alpha_i N_i(x)$

при любых α_i будет удовлетворять тем же однородным граничным условиям.

Если условия неоднородны, т.е. $y(a) = y_0$, $y(b) = y_1$, где хотя бы одно из чисел y_0 или y_1 отлично от нуля, то решение вариационной задачи можно искать в виде

$$y_m = \psi(x) + \sum_{i=1}^m \alpha_i N_i(x), \quad (4.2)$$

где $\psi(x)$ удовлетворяет заданным граничным условиям: $\psi(a) = y_0$, $\psi(b) = y_1$, а функции $N_i(x)$ удовлетворяют соответствующим однородным граничным условиям, т.е. в рассматриваемом случае $N_i(a) = N_i(b) = 0$. Очевидно, что при таком выборе при любых α_i функции $y_m(x)$ удовлетворяют заданным граничным условиям. В качестве функций $\psi(x)$ можно выбрать, например, линейную функцию

$$\psi(x) = \frac{y_1 - y_0}{b - a}(x - a) + y_0,$$

а в качестве базисных – функции

$$N_i(x) = (x - a)(x - b)x^{i-1}.$$

(Убедитесь, что оба условия $y(a) = y_0$ и $y(b) = y_1$ выполняется здесь априори).

На аппроксимации вида (4.1) или (4.2) функционал $F[y(x)]$ превращается в функцию $\Phi(\alpha_1, \alpha_2, \dots, \alpha_m)$ коэффициентов $\alpha_1, \alpha_2, \dots, \alpha_m$. Эти коэффициенты выбираются так, чтобы функция $\Phi(\alpha_1, \alpha_2, \dots, \alpha_m)$ достигала экстремума; следовательно, $\alpha_1, \alpha_2, \dots, \alpha_m$ должны быть определены из системы уравнений

$$\partial\Phi/\partial\alpha_i = 0, \quad i = 1, 2, \dots, m.$$

Совершая предельный переход при $m \rightarrow \infty$, получим в случае существования предела функцию $y = \sum_{i=1}^{\infty} \alpha_i N_i(x)$, являющуюся точным решением рассматриваемой вариационной задачи. Если не совершать предельного перехода, а ограничиться лишь m первыми членами, то получим приближенное решение вариационной задачи.

4.3. Вариационная задача Больца. Краевая задача 2-го и 3-го рода

Задача Больца состоит в определении функции $y(x) \in C_1[a, b]$, доставляющей экстремум функционалу

$$F[y(x)] = \int_a^b f(x, y, y') dx + g(y(a), y(b)).$$

Задача Больца – это особый тип задачи с подвижными границами, в которой граничные точки кривой могут перемещаться вдоль соответствующих вертикальных прямых $x = a$ и $x = b$. Экстремум здесь достигается только на кривых, удовлетворяющих уравнению Эйлера

$$\frac{\partial f}{\partial y} - \frac{d}{dx} \frac{\partial f}{\partial y'} = 0$$

и условиям трансверсальности

$$\left[\frac{\partial f}{\partial y'} - \frac{\partial g}{\partial y(a)} \right]_{x=a} = 0, \quad \left[\frac{\partial f}{\partial y'} + \frac{\partial g}{\partial y(b)} \right]_{x=b} = 0.$$

Надлежащим выбором функций f и g в функционале можно добиться эквивалентной вариационной формулировки дифференциальных краевых задач 2 и 3 рода. Например, функционал

$$F[y(x)] = \frac{1}{2} \int_a^b (py'^2 + qy^2 - 2fy) dx +$$

$$+\frac{\beta_2}{2} p(b)y(b)^2 - \gamma_2 p(b)y(b) - \frac{\beta_1}{2} p(a)y(a)^2 + \gamma_1 p(a)y(a) \quad (4.3)$$

достигает экстремума на функции $y(x)$, которая одновременно удовлетворяет дифференциальному уравнению 2-го порядка (уравнению Эйлера)

$$\frac{d}{dx} \left(p(x) \frac{d}{dx} \right) - q(x)y = -f(x) \quad (4.4)$$

и граничным условиям 3-го рода (условиям трансверсальности)

$$y'(a) + \beta_1 y(a) = \gamma_1, \quad y'(b) + \beta_2 y(b) = \gamma_2. \quad (4.5)$$

Отсюда следует, что если уравнение для некоторой краевой задачи имеет вид (4.4), а граничные условия – (4.5), то эквивалентная вариационная формулировка этой задачи дается функционалом (4.3). При этом заметим, что граничные условия (4.5) могут быть не только 3-го, но и 2-го рода; для этого достаточно положить $\beta_1 = 0$ и/или $\beta_2 = 0$. Указанные граничные условия являются естественными. Это означает, что функция, доставляющая экстремум функционалу на множестве всех допустимых функций, будет автоматически удовлетворять как дифференциальному уравнению, так и естественному граничному условию.

В силу естественности граничных условий аппроксимация неизвестной функции может выбираться без ограничений на границе. В методе Ритца обычно $y(x)$ записывают в виде разложения по системе линейно независимых функций $\{1, x, x^2, x^3, \dots\}$, т.е. в виде полного многочлена с неопределенными коэффициентами:

$$y(x) \approx y_m(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \dots + \alpha_m x^{m-1}.$$

Разумеется, в смешанной задаче, когда присутствуют одновременно и главное, и естественное условия, аппроксимация должна обеспечивать выполнение граничного условия 1-го рода, записанного для соответствующей точки (как в простейшей задаче). Например, если в точке $x=a$ выполняется условие

$y(a) = y_0$, а в точке $x = b$ – условие типа $y'(b) + \beta y(b) = \gamma$, то аппроксимация может быть взята в виде

$$y_m(x) = y_0 + (\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \dots + \alpha_m x^{m-1})(x - a),$$

а функционал (4.3) не будет содержать граничных слагаемых, вычисляемых в точке $x = a$:

$$F = \frac{1}{2} \int_a^b [p(x)y'^2 - q(x)y^2 + 2f(x)y] dx + \frac{1}{2} \beta p(b)y^2(b) - \gamma p(b)y(b).$$

Пример 1. Решить краевую задачу методом Рунге:

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \frac{2}{x}; \quad \varphi(1) = 0, \quad \varphi'(2) = 1.$$

Решение. Сначала найдем эквивалентную вариационную формулировку данной задачи. Сравнивая уравнение (4.4) с нашим уравнением, получаем, что $p(x) = x^2$, $q(x) = 2$, $f(x) = 1 + 2/x$. Аналогично из вида граничного условия $y'(2) + \beta_2 y(2) = \gamma_2$ находим $\beta_2 = 0$, $\gamma_2 = 1$. Граничное условие при $x = 1$ (1-го рода) является главным. Таким образом, в данной задаче используем функционал

$$F[\varphi(x)] = \frac{1}{2} \int_1^2 (x^2 \varphi'^2 + 2\varphi^2 - 2(1 + 2/x)\varphi) dx - 4\varphi(2).$$

Аппроксимирующие функции, при любых значениях параметров $\{\alpha_i\}$ удовлетворяющие главному условию $\varphi(1) = 0$, можно выбрать в виде

$$\varphi_m(x) = (\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \dots + \alpha_m x^{m-1})(x - 1). \quad (4.6)$$

Порядок решения в системе Maple

1. Задание аппроксимации, заведомо удовлетворяющей условию $\varphi(1) = 0$.

> restart; # очистка оперативной памяти

Возьмем четыре параметра аппроксимации, т. е. в (4.6) положим $m = 4$.

> **z := x -> (a1 + a2*x + a3*x^2 + a4*x^3) * (1 - x) ;**
 $z := x \rightarrow (a_1 + a_2 x + a_3 x^2 + a_4 x^3) (1 - x)$

Тем самым, аппроксимирующая функция задана как **z (x)** .

2. Вычисление функционала от этой функции

> **F := z -> (1/2.) * int (x^2 * diff (z (x) , x) ^2 + 2 * z (x) ^2 -**
2 * (1 + 2 * 1/x) * z (x) , x = 1 .. 2) - 4 * z (2) ;

$$F := z \rightarrow \frac{1}{2} \int_1^2 x^2 \left(\frac{d}{dx} z(x) \right)^2 + 2 z(x)^2 - 2 \left(1 + \frac{2}{x} \right) z(x) dx - 4 z(2)$$

> **P := F (z) ;**

$$F := \frac{154 a_2 a_3}{5} + \frac{271 a_1 a_4}{10} + \frac{79 a_1 a_3}{6} + \frac{13 a_1}{2} + \frac{59 a_2}{6} + \frac{71 a_2^2}{10} + \frac{19 a_2 a_1}{3} + \frac{3 a_1^2}{2} + \frac{229 a_3}{12} - 2 a_1 \ln(2) + \frac{2237 a_4}{60} + \frac{4607 a_4^2}{28} + \frac{458 a_2 a_4}{7} + \frac{597 a_3 a_4}{4} + \frac{2407 a_3^2}{70}$$

Функционал теперь зависит от параметров a_1, a_2, a_3, a_4 , т.е. он превратился в функцию многих переменных (обозначена через **P**).

Замечания. 1) В ряде случаев Maple может некорректно выводить функционал, представляя его в виде дробно-рациональной функции относительно параметров $\{a_i\}$, в то время как он должен быть квадратичным по ним. Подобная ситуация делает применение метода Ритца невозможным. Как правило, действие Maple-функции **expand** на второе слагаемое под знаком интеграла в функционале, т.е. qy^2 , помогает устранить проблему. В частности, в рассматриваемом примере в функционале **F** вместо $2 * z (x) ^2$ следует использовать **expand (2 * z (x) ^2)** .

2) Как известно, Maple, в первую очередь, стремится проводить символьные вычисления, поэтому и функционал здесь выведен в символьном виде, что не всегда бывает целесообразным. Добиться числового формата можно, как всегда, с помощью **evalf**. Например, вместо строки **P := F (z) ;** записать **P := evalf (F (z)) ;**

3. Поиск минимума функционала как функции многих переменных.

Применяя необходимое условие экстремума $\partial F/\partial a_i = 0$, $i = 1, 2, 3, 4$, получим систему алгебраических уравнений:

> eqns := {diff (P, a1)=0, diff (P, a2)=0, diff (P, a3)=0, diff (P, a4)=0};

$$eqns := \left\{ \frac{271 a1}{10} + \frac{2237}{60} + \frac{4607 a4}{14} + \frac{458 a2}{7} + \frac{597 a3}{4} = 0, \right.$$

$$\left. \frac{271 a4}{10} + \frac{79 a3}{6} + \frac{13}{2} + \frac{19 a2}{3} + 3 a1 - 2 \ln(2) = 0, \right.$$

$$\left. \frac{154 a3}{5} + \frac{59}{6} + \frac{71 a2}{5} + \frac{19 a1}{3} + \frac{458 a4}{7} = 0, \right.$$

$$\left. \frac{154 a2}{5} + \frac{79 a1}{6} + \frac{229}{12} + \frac{597 a4}{4} + \frac{2407 a3}{35} = 0 \right\}$$

Для решения используем функцию **solve**:

> r := solve (eqns, {a1, a2, a3, a4});

$$r := \left\{ a4 = \frac{232015}{429} - \frac{1336405}{1716} \ln(2), a2 = \frac{1377873}{286} - \frac{23785055}{3432} \ln(2), \right.$$

$$\left. a3 = -\frac{401681}{143} + \frac{6938435}{1716} \ln(2), a1 = -\frac{390421}{143} + \frac{4485175}{1144} \ln(2) \right\}$$

Maple вывел точный результат, но он неудобен для восприятия. Представим его в числовом формате в виде десятичных дробей:

> r1 := evalf (r);

$$r1 := \{ a1 = -12.658736, a4 = 1.0108637, a2 = 13.966255, a3 = -6.302647 \}$$

Это и есть оптимальные параметры, обеспечивающие наилучшее приближение в классе функций вида (4.6).

Решение данной задачи – это функция **z (x)**, заданная в п. 1, с только что найденными числовыми параметрами **a1**, **a2**, **a3** и **a4**, которые хранятся в переменной **r1** (или **r**). Результат подстановки этих параметров в **z (x)** оформим в виде новой функции **phi (x)**:

> phi := unapply (subs (r1, z (x)), x);

$$\phi := x \rightarrow (-12.658736 + 13.966255 x - 6.302647 x^2 + 1.0108637 x^3) (1 - x)$$

Следует помнить, что здесь коэффициенты записаны с ограниченным числом значащих цифр. В некоторых случаях это может привести к дополнительной ошибке в решении. Уменьшить такую ошибку можно заданием фиксированного числа разрядов в переменной **Digits**. Другой способ – использование переменной **r** вместо **r1** в определении функции **phi**.

4. Представление полученной функции в виде таблицы и графика.

Обращаться к функции-решению теперь можно обычным способом, например, вычислить значение в некоторой точке:

```
> phi(1.25) ;  
0.7686150060
```

Можно также протабулировать эту функцию, т.е. вывести ее значения на определенном интервале изменения аргумента в определенных (чаще равноотстоящих) точках. К примеру, изменяя аргумент от 1 до 2 с шагом 0,25, получим

```
> for t from 1 by 0.25 to 2 do printf('x=%g  
z=%g\n', t, phi(t)); od;  
x=1 z=0  
x=1.25 z=.768615  
x=1.50 z=1.239322  
x=1.75 z=1.576536  
x=2.00 z=1.849904
```

Построим график решения (рис. 4.1):

```
> plot(phi(x), x=1..2) ;
```

5. Получение точного решения.

Данная краевая задача может быть решена точно аналитическими методами. Точное решение позволит оценить погрешность приближенного решения, полученного по методу Рунге.

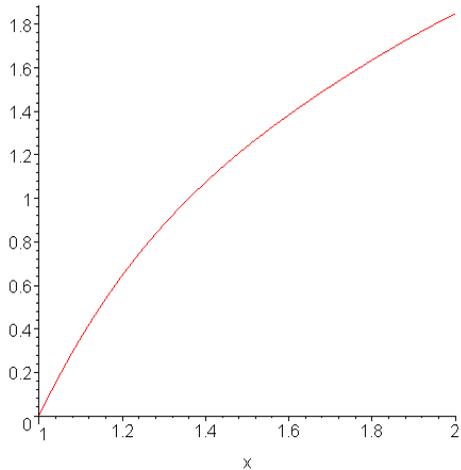


Рис. 4.1

```
> st:=dsolve({diff(x^2*diff(y(x),x),x)-2*y(x)=
-1-2*1/x,y(1)=0,D(y)(2)=1),y(x)});
```

$$st := y(x) = -\frac{11}{5x^2} + \frac{7x}{10} + \frac{2+x}{2x}$$

```
> u:=unapply(subs(st,y(x)),x); # теперь точное решение
есть функция u(x)
```

6. Сравнение точного и приближенного решений.

Сначала сравним результаты в отдельных равноотстоящих точках:

```
> for t from 1 by 0.25 to 2 do printf(`x=%g z=%g
u=%g\n`,t,phi(t),u(t)); od;
```

x=1	z=0	u=0
x=1.25	z=.768615	u=.767
x=1.50	z=1.239322	u=1.238889
x=1.75	z=1.576536	u=1.578061
x=2.00	z=1.849904	u=1.85

Видим хорошее соответствие. Расхождение – менее 1%.

Для визуального сравнения построим графики:

```
> plot([phi(x),u(x)],x=1..2,color=[red,blue],
thickness=[2,2]);
```

Они практически совпадают.

7. Проверка сходимости метода.

Для получения адекватного результата численный или приближенный метод должен обладать свойством сходимости. Это означает, что с увеличением сила параметров аппроксимации приближенное решение должно стремиться к точному решению.

Конечно, в определенной степени дать оценку совпадения результатов можно, анализируя табличные данные или графики функции, как это делалось в предыдущем пункте. Но такой подход математически нестрог и годится только для качественного анализа. Для количественной характеристики процесса сходимости используется особая величина – норма ошибки решения, определяемая по формуле

$$\delta = \left(\int_a^b [\varphi_m(x) - \varphi(x)]^2 dx \right)^{1/2}$$

(здесь $\varphi(x)$ – точное решение). Если метод сходится, то δ приближается к нулю с увеличением m . То, насколько быстро это происходит, характеризует скорость сходимости. Вычисление нормы ошибки можно организовать следующим образом:

```
> evalf(int(phi(x)-u(x))^2, x=1..2)^(1/2);
0.001770384139
```

Итак, для проверки сходимости нужно провести ряд вычислений по предложенной схеме для разного числа параметров аппроксимации. В приведенном алгоритме реализации это потребует изменения текста в п. 1, а также при необходимости в п. 3. Например, при $m=3$ поменять надо только выражение для аппроксимации:

```
> z:=x->((a1+a2*x+a3*x^2)*(1-x));
```

а все остальные команды просто перезапустить на выполнение. Если же взять больше четырех параметров, к примеру, $m=5$, то помимо аппроксимации

```
> z:=x->((a1+a2*x+a3*x^2+a4*x^3+a5*x^4)*(1-x));
```

из-за появления нового параметра $a5$ также должны быть модифицированы команды пункта 3:

```
> eqns:={diff(P,a1)=0,diff(P,a2)=0,diff(P,a3)=0,
diff(P,a4)=0,diff(P,a5)=0};
```

```
> r:=solve(eqns,{a1,a2,a3,a4,a5});
```

Помимо оценки сходимости решения интерес представляет сходимость производной $\varphi'_m(2)$ к точному значению 1 (естественное граничное условие) и сходимость функционала. Вывод этих величин осуществляется очень простыми командами:

```
> D(phi)(2);
```

```
> F(phi);
```

Точное значение функционала определяется так

> evalf(F(u));

-4.993147181

Результаты вычислений представлены в таблице.

	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$
δ	0.0513	0.00913	$1.77 \cdot 10^{-3}$	$3.57 \cdot 10^{-4}$	$6.81 \cdot 10^{-5}$
$\varphi'_m(2)$	0.673	1.118	0.964	1.010	0.997
$F[\varphi_m]$	-4.916386	-4.986417	-4.992677	-4.993119	-4.993146

Данные из таблицы свидетельствуют о высокой скорости сходимости метода Рунге.

Упражнения

1. Решить краевую задачу методом Рунге. Сравнить с точным решением. Построить графики. Показать сходимость для $m = 2 \div 6$.

Вариант 1. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2$, $\varphi(1) = 0$, $\varphi(2) = 0$.

Вариант 2. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2$, $\varphi'(1) = 0$, $\varphi'(2) = 0$.

Вариант 3. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2$, $\varphi'(1) = 0$, $\varphi'(4) - \varphi(4) = -2$.

Вариант 4. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - 2 \ln x$,
 $\varphi(1) + \varphi'(1) = 0$, $\varphi(2) - \varphi'(2) = 0$.

Вариант 5. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - 2 \ln x$,
 $\varphi(1) - \varphi(2) = 2$, $\varphi'(2) + \varphi(1) = 1$.

Вариант 6. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 0$, $\varphi'(1) = -1$, $\varphi'(2) = 1$.

Вариант 7. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 1 - \frac{1}{x}$, $\varphi(1) = 0$, $\varphi'(2) = 1$.

Вариант 8. $\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 2 - \ln x$, $\varphi(1) - 2\varphi'(1) = 0$, $\varphi'(2) = 1$.

Вариант 9. $\frac{d}{dx} \left((x+1)^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1$, $\varphi(0) = 0$, $\varphi'(1) = 1$.

Вариант 10. $\frac{d}{dx} \left((x+1)^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln(x+1)$,
 $\varphi(0) = 0$, $\varphi'(2) = 1$.

2. Решить уравнение $d^2u/dx^2 + \omega^2 u = x$, $0 \leq x \leq 1$ с краевыми условиями:

- 1) $u = 0$ при $x = 0$ и $u = 0$ при $x = 1$;
- 2) $u = 1$ при $x = 0$ и $du/dx = 0$ при $x = 1$;
- 3) $u = 0$ при $x = 0$ и $du/dx + u = 0$ при $x = 1$.

Сравнить с точным решением.

3. Решить уравнение $d^2u/dx^2 - \omega^2 u = 0$, $0 \leq x \leq 1$ с краевыми условиями $u = 0$ при $x = 0$ и $du/dx + 10u = 20$ при $x = 1$. Сравнить с точным решением.

4. Решить краевую задачу методом Рунца для разных типов граничных условий. Сравнить с точным решением. Показать сходимость.

Вариант 1. $\frac{d}{dx} \left((x^2 + 1) \frac{d\varphi}{dx} \right) - 2\varphi = 1 - x - x^2$;

а) $\varphi(1) = -\frac{2}{3}$, $\frac{d\varphi}{dx}(3) = -\frac{1}{2}$; б) $\varphi(1) = 2$, $\varphi(4) = -\frac{1}{2}$;

в) $\varphi(1) - 2\frac{d\varphi}{dx}(1) = -2$, $\varphi(3) + 3\frac{d\varphi}{dx}(3) = -2$.

Вариант 2. $\frac{d}{dx}\left(e^{-x} \frac{d\varphi}{dx}\right) - e^{-x}\varphi = -2\cos(x+1)$;

а) $\frac{d\varphi}{dx}(0) = -3$, $\varphi(2) = 5$; б) $\varphi(0) = 1$, $\varphi(2) = 0$;

в) $\frac{d\varphi}{dx}(0) = -1$, $\frac{d\varphi}{dx}(3) + 2\varphi(3) = -6$.

Вариант 3. $\frac{d}{dx}\left(\frac{1}{1+x^2} \frac{d\varphi}{dx}\right) - \varphi = -2$; а) $\frac{d\varphi}{dx}(-1) = \frac{5}{2}$, $\varphi(1) = 3$;

б) $\varphi(-1) = 2$, $\varphi(1.5) = 4$; в) $\frac{d\varphi}{dx}(-1) + 2\varphi(-1) = 0$, $\frac{d\varphi}{dx}(1) + \varphi(1) = -4$.

Вариант 4. $\frac{d}{dx}\left(x^2 \frac{d\varphi}{dx}\right) - \varphi = 5\ln x$; а) $\frac{d\varphi}{dx}(1) = -\frac{3}{2}$, $\varphi(2.4) = 2$;

б) $\varphi(1) = 1$, $\varphi(2) = 0$; в) $\varphi(1) + \frac{d\varphi}{dx}(1) = 4$, $\frac{d\varphi}{dx}(2) - 2\varphi(2) = -3$.

Вариант 5. $\frac{d}{dx}\left(x^2 \frac{d\varphi}{dx}\right) + 2\varphi = 2 - x$; а) $\varphi(1) = 1$, $\frac{d\varphi}{dx}(2) = -1$;

б) $\varphi(1) = 1$, $\varphi(2) = -1$; в) $\varphi(1) - \frac{d\varphi}{dx}(1) = 5$, $\frac{d\varphi}{dx}(2) + 2\varphi(2) = 0$.

Вариант 6. $\frac{d}{dx}\left(e^{-x} \frac{d\varphi}{dx}\right) + e^{-x}\varphi = -4$; а) $\varphi(0) = 0$, $\frac{d\varphi}{dx}(1) = -2$;

б) $\varphi(0) = 1$, $\varphi(1) = \frac{1}{2}$; в) $\varphi(0) - 5\frac{d\varphi}{dx}(0) = -2$, $\frac{d\varphi}{dx}(1) + \varphi(1) = 2$.

Вариант 7. $\frac{d}{dx}\left((x+2) \frac{d\varphi}{dx}\right) - \frac{1}{x+2}\varphi = x^2 - \frac{1}{2}$; а) $\varphi(0) = -1$, $\frac{d\varphi}{dx}(2) = \frac{1}{2}$;

б) $\varphi(0) = 1$, $\varphi(2) = \frac{1}{2}$; в) $\frac{d\varphi}{dx}(0) - 2\varphi(0) = 1$, $\frac{d\varphi}{dx}(2) = \frac{1}{2}$.

Вариант 8. $\frac{d^2\varphi}{dx^2} = -\varphi + 1 - x + x^2$; а) $\frac{d\varphi}{dx}(1) = -1$, $\varphi(3) = 1$;

б) $\varphi(1) = -1$, $\varphi(3) = 1$; в) $\varphi(1) - \frac{d\varphi}{dx}(1) = -1$, $\frac{d\varphi}{dx}(2,5) + 2\varphi(2,5) = 1$.

Вариант 9. $\frac{d}{dx}\left(x^4 \frac{d\varphi}{dx}\right) - 4x^2\varphi = \frac{1}{x}$; а) $\frac{d\varphi}{dx}(1) = -3$, $\varphi(2) = 3$;

б) $\varphi(1) = -3$, $\varphi(1,9) = 3$; в) $\frac{d\varphi}{dx}(1) - 2\varphi(1) = -10$, $20\frac{d\varphi}{dx}(2) - \varphi(2) = 10$.

Вариант 10. $\frac{d}{dx}\left((e^x + 2)\frac{d\varphi}{dx}\right) = e^{x+2}$; а) $\frac{d\varphi}{dx}(0) = -10$, $\varphi(2) = \frac{1}{2}$;

б) $\varphi(0) = 10$, $\varphi(4) = \frac{1}{2}$; в) $\frac{d\varphi}{dx}(0) + 2\varphi(0) = \frac{15}{2}$, $20\frac{d\varphi}{dx}(2) + \varphi(2) = 10$.

5. Решить краевую задачу методом Рунге. Сравнить с точным решением. Показать сходимость.

1. $\frac{d^2\varphi}{dx^2} + 2\frac{d\varphi}{dx} + 1 = 0$; $\varphi(0) = 1$, $\frac{d\varphi}{dx}(1) = -1$; [$\varphi(0) = 1$, $\varphi(1) = -1$].

2. $x^2\frac{d^2\varphi}{dx^2} + 2x\frac{d\varphi}{dx} - 2\varphi = 0$; $\varphi(1) = 1$, $\frac{d\varphi}{dx}(2) = -1$; [$\varphi(1) = 1$, $\varphi(2) = -1$].

3. $\frac{1}{4}\frac{d^2\varphi}{dx^2} + \varphi = 1$; $\varphi(0) + 2\frac{d\varphi}{dx}(0) = 0$, $\varphi(1) = 1$; [$\varphi(0) = 0$, $\varphi(1) = 1$].

4. $\frac{d^2\varphi}{dx^2} - \frac{d\varphi}{dx} = e^x$; $\varphi(0) = 0$, $\frac{d\varphi}{dx}(4) = -10$; [$\varphi(0) = 0$, $\varphi(4) = -10$].

5. $\frac{d}{dx}\left(e^{-x}\frac{d\varphi}{dx}\right) + e^x\varphi = 0$; $\varphi(0) = 0$, $\frac{d\varphi}{dx}(1) = 1$; [$\varphi(0) = 0$, $\varphi(1) = 1$].

6. $\frac{d^2\varphi}{dx^2} + \varphi = \frac{1}{\cos x}$; $\varphi(0) = 1$, $\frac{d\varphi}{dx}(1) = 1$; [$\varphi(0) = 1$, $\varphi(1) = 1$].

$$7. \frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = 0; \quad \varphi(1) = 0, \quad \frac{d\varphi}{dx}(2) = 1; \quad [\varphi(1) = 0, \quad \varphi(2) = 1].$$

$$8. \frac{d}{dx} \left(e^{-x/2} \frac{d\varphi}{dx} \right) = 1 - x; \quad \varphi(0) = 0, \quad \frac{d\varphi}{dx}(4) = 2; \quad [\varphi(0) = 0, \quad \varphi(4) = 2].$$

$$9. \frac{d}{dx} \left(\frac{1}{1+x} \frac{d\varphi}{dx} \right) = -x; \quad \varphi(0) + \frac{d\varphi}{dx}(0) = 1, \quad \varphi(1) = 0; \quad [\varphi(0) = 1, \quad \varphi(1) = 0].$$

$$10. \frac{d}{dx} \left(\frac{1}{x} \frac{d\varphi}{dx} \right) = \frac{1}{1+x^2}; \quad \varphi(1) + \frac{d\varphi}{dx}(1) = 0, \quad \varphi(2) = 0; \quad [\varphi(1) = 0, \quad \varphi(2) = 0].$$

5. МЕТОД ГАЛЁРКИНА

Основная идея проекционных методов основана на свойстве гильбертовых функциональных пространств, согласно которому, если проекции некоторой функции f на систему линейно независимых функций $\{W_i\}$ равны нулю, то такая функция $f \equiv 0$. За такую функцию, которая должна обращаться в ноль, принимают невязку – разность между левой и правой частью дифференциального уравнения и граничного условия (соответственно R_Ω и R_Γ). Поскольку в функциональном анализе проекция f на функцию g определяется как скалярное произведение (f, g) , то система уравнений имеет вид

$$(R_\Omega, W_i) = 0, \quad i = 1, \dots, m$$

или

$$(R_\Omega, W_i) + (R_\Gamma, \bar{W}_i) \Big|_\Gamma = 0$$

в зависимости от того учитывает ли аппроксимация неизвестной функции граничные условия или не учитывает.

Рассмотрим основные моменты реализации метода взвешенных невязок для уравнения

$$\frac{d^2 y}{dx^2} + s(x) \frac{dy}{dx} + t(x)y = r(x)$$

и граничных условий

$$\mu_1 y'(a) + \beta_1 y(a) = \gamma_1, \quad \mu_2 y'(b) + \beta_2 y(b) = \gamma_2.$$

Вначале строится аппроксимация с помощью базисных функций, как обычно, в виде суммы

$$y_m(x) = \sum_{i=1}^m \alpha_i N_i(x) \quad (5.1)$$

Для этой аппроксимации записываются невязки по отрезку $[a, b]$ и граничным точкам

$$\begin{aligned} R_\Omega &= \frac{d^2 y_m}{dx^2} + s(x) \frac{dy_m}{dx} + t(x) y_m - r(x), \\ R_a &= \mu_1 y'_m(a) + \beta_1 y_m(a) - \gamma_1, \\ R_b &= \mu_2 y'_m(b) + \beta_2 y_m(b) - \gamma_2. \end{aligned}$$

Выбираются системы линейно независимых функций – весовых функций W_i , $W_i^{(1)}$ и $W_i^{(2)}$. Эти системы произвольны, а, значит, в частности, могут быть каким-либо образом связаны между собой. Важный частный случай, когда $W_i = N_i$, $W_i^{(1)}(a) = \pm N_i(a)$, $W_i^{(2)}(b) = \pm N_i(b)$, т.е. весовые функции совпадают с аппроксимирующими, носит название метода Галеркина. Далее составляется система алгебраических уравнений

$$(R_\Omega, W_i) + [R_a W_i^{(1)}]_{x=a} + [R_b W_i^{(2)}]_{x=b} = 0, \quad i = 1, \dots, m, \quad (5.2)$$

или в развернутом виде

$$\begin{aligned} &\int_a^b \left[\frac{d^2 y_m}{dx^2} + s(x) \frac{dy_m}{dx} + t(x) y_m - r(x) \right] W_i dx + \\ &+ (\mu_1 y'_m(a) + \beta_1 y_m(a) - \gamma_1) W_i^{(1)}(a) + \\ &+ (\mu_2 y'_m(b) + \beta_2 y_m(b) - \gamma_2) W_i^{(2)}(b) = 0. \end{aligned} \quad (5.2^*)$$

Эта система решается относительно параметров $\{\alpha_i\}$, которые при подстановке в (5.1) дают приближенное решение краевой задачи.

Возможен вариант, когда аппроксимация выбирается так, что изначально удовлетворяет одному или двум граничным условиям вне зависимости от того, какие значения принимают параметры $\{\alpha_i\}$. Эта аппроксимация, как правило, имеет вид

$$y_m(x) = \psi(x) + \sum_{i=1}^m \alpha_i N_i(x),$$

где $\psi(x)$ удовлетворяет граничным условиям задачи при $x=a$ и/или $x=b$, а функции $\{N_i\}$ – соответствующим однородным условиям. Тогда в уравнениях (5.2) одно или оба граничных слагаемых исчезают. Действительно, поскольку соответствующие граничные условия будут выполняться точно для функции y_m , то граничная невязка будет равна нулю.

Замечание. Невязку R_Ω (так же, как и R_Γ) можно рассматривать как функционал, поскольку она определяется выбором функции $y_m(x)$. Фактически метод Галеркина минимизирует $|R_\Omega|$ и его можно считать вариационным.

Пример. Решить краевую задачу методом Галеркина:

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \frac{2}{x}; \quad \varphi(1) = 0, \quad \varphi'(2) = 1.$$

Решение. Включая в систему уравнений взвешенных невязок дополнительно невязки для левой и правой граничных точек, в соответствии с (9.2) получим

$$\left(\frac{d}{dx} \left(x^2 \varphi' \right) - 2\varphi + 1 + \frac{2}{x}, W_i \right) + (\varphi - 0, W_i^{(1)}) \Big|_{x=1} + (\varphi' - 1, W_i^{(2)}) \Big|_{x=2} = 0,$$

или

$$\int_1^2 \left[\frac{d}{dx} \left(x^2 \frac{d\varphi_m}{dx} \right) - 2\varphi_m + 1 + \frac{2}{x} \right] W_i dx + \varphi_m(1) W_i^{(1)}(1) + \varphi_m'(2) W_i^{(2)}(2) = 0.$$

В качестве аппроксимации возьмем линейную комбинацию степенных функций:

$$\varphi_5(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 + \alpha_5 x^4, \quad (5.3)$$

и составим уравнения взвешенных невязок, используя одинаковые (с точностью до знака) весовые функции W_i , $W_i^{(1)}$ и $W_i^{(2)}$: $1, x, x^2, x^3, x^4$. Как видно, такой выбор означает галеркинскую аппроксимацию.

Порядок решения в системе Maple

1. Зададим аппроксимацию и три невязки

```
> restart;
> z:=x->(a1+a2*x+a3*x^2+a4*x^3+a5*x^4);      #[1]
          z:=x → a1 + a2 x + a3 x2 + a4 x3 + a5 x4
> R1:=diff(x^2*diff(z(x),x),x)-2*z(x)+1+2/x;
R2:=z(1);   R3:=D(z)(2)-1;
```

2. Определим пять весовых функций – по числу неизвестных параметров аппроксимации – соответственно для всех трех введенных невязок:

```
> W:=[1,x,x^2,x^3,x^4,x^5]; WL:=subs(x=1,W);
WR:=subs(x=2,W);   #[2]
```

3. Построим пять уравнений как сумму взвешенных невязок:

```
> g1:=int(R1*W[1],x=1..2)+R2*WL[1]+R3*WR[1]=0;
> g2:=int(R1*W[2],x=1..2)+R2*WL[2]+R3*WR[2]=0;
> g3:=int(R1*W[3],x=1..2)+R2*WL[3]+R3*WR[3]=0;
> g4:=int(R1*W[4],x=1..2)+R2*WL[4]+R3*WR[4]=0;
> g5:=int(R1*W[5],x=1..2)+R2*WL[5]+R3*WR[5]=0;
```

4. Решим полученную систему линейных алгебраических уравнений:

```
> r:=solve({g1,g2,g3,g4,g5},{a1,a2,a3,a4,a5});
> r1:=evalf(r);
{a3=-23.167272 a4=8.348938 a5=-1.1531431 a2=30.546853 a1=-14.707363}
```

Это и есть оптимальные параметры, обеспечивающие наилучшее приближение в классе функций вида (5.3).

Решение данной задачи – это функция $z(x)$, заданная в п. 1, с только что найденными числовыми параметрами a_1, a_2, a_3, a_4 и a_5 , которые хранятся в переменной r_1 (или r). Ре-

зультат подстановки этих параметров в $\mathbf{z}(\mathbf{x})$ оформим в виде новой функции $\mathbf{phi}(\mathbf{x})$:

```
> phi:=unapply(subs(r1,z(x)),x);
```

$$\phi := x \rightarrow -14.707363 + 30.546853x - 23.167272x^2 + 8.348938x^3 - 1.1531431x^4$$

5. Получим точное решение задачи.

Данная краевая задача может быть решена точно аналитическими методами. Точное решение позволит оценить погрешность приближенного решения, полученного по методу Галеркина.

```
> st:=dsolve({diff(x^2*diff(y(x),x),x)-2*y(x)=-1-2*1/x,y(1)=0,D(y)(2)=1},y(x));
```

$$st := y(x) = -\frac{11}{5x^2} + \frac{7x}{10} + \frac{2+x}{2x}$$

```
> u:=unapply(subs(st,y(x)),x); # теперь точное решение есть функция u(x)
```

6. Сравним точное и приближенное решения.

Сначала выведем результаты в отдельных равноотстоящих точках:

```
> for t from 1 by 0.25 to 2 do printf(`x=%g
z=%8.6g      u=%g\n`,t,phi(t),evalf(u(t))); od;
```

x=1	z=-.131987	u=0
x=1.25	z= .76857	u=.767
x=1.50	z=1.326433	u=1.238889
x=1.75	z=1.729728	u=1.578061
x=2.00	z=2.058469	u=1.85

Полученные по методу Галеркина значения не вполне хорошо согласуются с точным решением, что наглядно видно на графике:

```
> plot([phi(x),u(x)],x=1..2, color=[red,blue],
thickness=[2,2]);
```

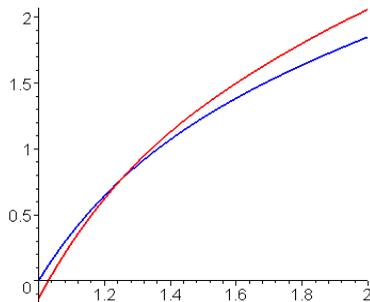


Рис. 5.1

7. Проверим сходимость.

Чтобы проверить адекватность решения, следует доказать сходимость метода. Для этого нужно выполнить все шаги метода с увеличенным на единицу числом параметров аппроксимации:

$$\varphi_6(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 + \alpha_5 x^4 + \alpha_6 x^5.$$

Для этого строку [1] следует поменять на

```
> z:=x-(a1+a2*x+a3*x^2+a4*x^3+a5*x^4+a6*x^5);
```

Список весовых функций W можно оставить как есть, так как он уже содержит требуемую шестую функцию x^5 :

```
> W:=[1, x, x^2, x^3, x^4, x^5];
```

Далее в пункте 3 следует проделать вычисления для g_1 , g_2 , g_3 , g_4 , g_5 и добавить еще g_6 :

```
> g6:=int(R1*W[6], x=1..2)+R2*WL[6]+R3*WR[6]=0;
```

Решим полученную систему из шести уравнений

```
> r:=solve({g1, g2, g3, g4, g5, g6}, {a1, a2, a3, a4, a5, a6});
```

Решение будет иметь вид

$$\phi := x \rightarrow -18.43098 + 47.0351x - 48.6239x^2 + 26.5966x^3 - 7.45583x^4 + 0.848104x^5$$

Построим, как и раньше, графики. Видно, что новое решение более точное, чем предыдущее. Этот же вывод следует и из поточечного сравнения решений.

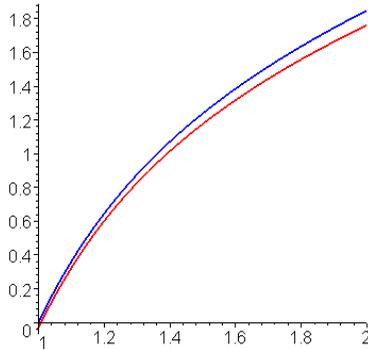


Рис. 5.2

Если взять семь параметров, т.е. аппроксимацию

$$\varphi_7(x) = \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 + \alpha_5 x^4 + \alpha_6 x^5 + \alpha_7 x^6,$$

то обнаружим на порядок еще более точное решение. Итак, данный метод сходится. Для тщательного анализа сходимости можно вычислить норму ошибки

> `evalf(int((phi(x)-u(x))^2, x=1..2))^(1/2);`

и осуществить проверку граничных условий, поскольку они выполняются приближенно:

> `D(phi)(2); #` вычисление значения производной $\varphi'_m(2)$

> `phi(1); #` вычисление значения функции $\varphi_m(1)$

Таблица 5.1

	$m=5$	$m=6$	$m=7$
δ	0.1188	0.06313	0.00744
$\varphi'_m(2)$	1.164	0.961	1.010
$\varphi_m(1)$	-0.132	-0.0309	-0.0078

Данные из таблицы свидетельствуют о сходимости метода.

Замечание. По умолчанию Maple использует в вычислениях 10 значащих цифр. С увеличением порядка аппроксимации может потребоваться более высокая точность. Поэтому рекомендуется проверять расчеты с измененным параметром `Digits`. Например,

> **Digits:=25;** # устанавливает 25 значащих цифр
(вводится сразу за командой **restart**).

Другой подход. Рассмотрим вариант метода Галеркина, когда одно или два граничных условия удовлетворяются точно путем специального выбора аппроксимации.

Выберем аппроксимацию

$$\varphi_m(x) = (\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3 + \dots + \alpha_m x^{m-1})(x-1), \quad (5.4)$$

автоматически удовлетворяющую граничному условию 1-го рода $\varphi(1) = 0$. Тогда невязка **R2** обнулится и формально ее присутствие в уравнениях **g1**, **g2**, ... ни на что не влияет. Таким образом, переопределив только функцию **z(x)** в соответствии с (5.4) в самом начале листинга, все остальные команды можно просто перезапустить. В общем случае такой подход, когда одно или два граничных условия учитываются точно через аппроксимацию, дает более высокую сходимость, чем в случае, когда граничные условия удовлетворяются приближенно путем введения соответствующих невязок. В табл. 5.2 показана сходимость нормы ошибки и производной функции $\varphi(x)$ в точке $x = 2$.

Таблица 5.2

	$m = 4$	$m = 5$	$m = 6$	$m = 7$
δ	0.56170	0.11291	0.03008	0.007037
$\varphi'_m(2)$	1.504	0.899	1.027	0.994

Сходимость, пусть не намного, но выше, чем в предыдущем подходе.

Для удобства проведения анализа сходимости предлагается следующий листинг, в котором при изменении числа параметров аппроксимации достаточно только поменять значение параметра m :

```
> restart; Digits:=25;
> m:=5; a:=1: b:=2: # a и b – границы интервала изменения x
> a:=vector(m); g:=vector(m);
> z:=x->(sum(alpha[j]*x^(j-1), j=1..m)*(x-1)); # Для
# хранения параметров аппроксимации используем массив alpha
```

```

> R1:=diff(x^2*diff(z(x),x),x)-2*z(x)+1+2/x;
> R2:=z(a); R3:=D(z)(b)-1;
> W:=[1,x,x^2,x^3,x^4,x^5,x^6]; WL:=subs(x=a,W);
WR:=subs(x=b,W);
> for i from 1 to m do
g[i]:=int(R1*W[i],x=a..b)+R2*WL[i]+R3*WR[i];od;
> r:=solve({g[k]$k=1..m},{alpha[k]$k=1..m});
# Дальнейший текст почти полностью повторяет
# соответствующий фрагмент предыдущего варианта
> r1:=evalf(r);
> phi:=unapply(subs(r1,z(x)),x);
> st:=dsolve({diff(x^2*diff(y(x),x),x)-2*y(x)=-1-2*1/x,
y(1)=0,D(y)(2)=1},y(x));
> u:=unapply(subs(st,y(x)),x);
> for t from a by 0.25 to b do
printf(`x=%g z=%8.6g u=%g\n`,t,phi(t),u(t)); od;
> plot([phi(x),u(x)],x=a..b,color=[red,blue],
thickness=[2,2]);
> evalf(int((phi(x)-u(x))^2,x=a..b)^(1/2));
> D(phi)(b);

```

Упражнения

1. Произвести вычисления по той же схеме, что в примере, но с другими весовыми функциями W_i :

а) $W_i = 1, -x, x^2, -x^3, \dots$ б) $W_i = 1, x^2, x^4, x^6, \dots$

в) $W_i = 1, \operatorname{ch} x, \operatorname{ch} 2x, \operatorname{ch} 3x, \dots$ г) $W_i = e^x, x-1, e^{2x}, x^2-x, e^{3x}, x^3-x^2, \dots$

д) $W_i = 1, \sin \pi x, \sin 2\pi x, \sin 3\pi x, \dots$

е) $W_i = \delta(x - x_i)$, где x_i – узлы отрезка $[1, 2]$.

2. Выполнить расчет по приведенной схеме различных вариантов метода взвешенных невязок, отличающихся только знаками у весовых функций $W_i^{(1)}$ и $W_i^{(2)}$:

а) $W_i = N_i, W_i^{(1)} = -N_i, W_i^{(2)} = N_i$;

б) $W_i = N_i, W_i^{(1)} = N_i, W_i^{(2)} = -N_i$;

в) $W_i = N_i, W_i^{(1)} = -N_i, W_i^{(2)} = -N_i$.

В каком случае сходимость аппроксимации выше?

3. Предложить вариант решения задачи из примера с использованием аппроксимирующих базисных функций

$$N_1 = \sin \frac{\pi(x-1)}{2}, N_2 = \sin \frac{3\pi(x-1)}{2}, \dots, N_k = \sin \frac{(2k-1)\pi(x-1)}{2}$$

и любых подходящих весовых функций. Выполнить анализ сходимости.

4. Методом взвешенных невязок решить уравнение на указанном отрезке с указанными граничными условиями

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \frac{2}{x}; \quad \text{а) } x \in [0,5; 2,5]; \quad \varphi(0,5) = 0, \quad \varphi'(2,5) = 1;$$

$$\text{б) } x \in [1; 2,5]; \quad \varphi'(1) = 2, \quad \varphi(2,5) = 0.$$

5. Методом Галеркина решить краевые задачи из упражнения **4** раздела 5 («Метод Ритца»):

а) включая в уравнения невязки в обеих граничных точках;

б) удовлетворяя краевому условию 1-го рода выбором аппроксимации, а условиям 2-3-рода – вводом соответствующей невязки.

Сравнить с точным решением. Провести анализ сходимости аппроксимаций.

6*. С помощью метода Галёркина найти решение краевой задачи, не имеющей аналитического решения. Условия задач взять из упражнения **6** раздела 4. Показать сходимость.

6. МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ

Метод конечных элементов (МКЭ) на сегодняшний день является одним из наиболее эффективных методов решения краевых задач математической физики. Он обычно базируется на вариационной или проекционной формулировке и тем самым несет в себе преимущества методов Ритца или Галеркина. С другой стороны, подобно методу конечных разностей, решение здесь строится в виде каркаса значений в узлах сетки, а значит, он хорошо приспособлен для компьютерных вычислений.

Основное отличие метода конечных элементов от методов Рунге или Галеркина – в построении базисных функций. Теперь они имеют малый носитель, т.е. отличны от нуля только в окрестности некоторой геометрически выделенной точки – узла. В классическом методе конечных элементов область задачи разделяется на подобласти (конечные элементы). На каждой такой подобласти локально строится аппроксимация в виде интерполяционного полинома. Коэффициенты каждого полинома находятся из решения глобальной системы алгебраических уравнений, полученной путем применения процедуры Рунге или Галеркина и требования непрерывности решения.

Пример. Рассмотрим ту же задачу, что была решена ранее другими методами:

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \frac{2}{x}, \quad \varphi(1) = 0, \quad \varphi'(2) = 1.$$

Решение.

Разобьем отрезок $1 \leq x \leq 2$ на n равные части: $[x_k, x_{k+1}]$, $k = 1, 2, \dots, n$; $x_k \equiv 1 + (k-1)/n$. (Точки x_k называются узлами.) Каждую такую часть будем называть конечным элементом. На k -м элементе – отрезке $[x_k, x_{k+1}]$ – неизвестную функцию представим в виде линейной комбинации

$$\varphi^{(k)}(x) = \varphi(x_k) N_{1k}(x) + \varphi(x_{k+1}) N_{2k}(x), \quad x_k \leq x \leq x_{k+1}, \quad (6.1)$$

что соответствует случаю линейной интерполяции. Интерполирующие функции $N_{1k}(x)$ и $N_{2k}(x)$ (их называют *базисными функциями* или *функциями формы*) равны

$$N_{1k}(x) = \frac{x_{k+1} - x}{x_{k+1} - x_k}, \quad N_{2k}(x) = \frac{x - x_k}{x_{k+1} - x_k}. \quad (6.2)$$

Учитывая, что длины всех элементов одинаковы и равны $x_{k+1} - x_k = 1/n$, а также $x_k = 1 + (k-1)/n$, можно записать

$$N_{1k}(x) = n + k - nx, \quad N_{2k}(x) = nx - n - k + 1.$$

Итак, на всей области задачи $1 \leq x \leq 2$ неизвестная функция строится кусочным образом:

$$\hat{\varphi}(x) = \begin{cases} \varphi^{(1)}(x), & \text{если } x_1 \leq x \leq x_2, \\ \varphi^{(2)}(x), & \text{если } x_2 \leq x \leq x_3, \\ \dots\dots\dots \\ \varphi^{(n)}(x), & \text{если } x_n \leq x \leq x_{n+1}. \end{cases} \quad (6.3)$$

и рассматривается как пробная функция в методах Ритца или Галеркина. Проверьте, что эта функция принадлежит множеству S_0 , т.е. является непрерывной, но ее производная терпит разрыв 1-го рода в точках $x = x_k$, $k = 2, 3, \dots, n - 1$. Важно отметить, что функция $\hat{\varphi}(x)$ однозначно определяется своими узловыми значениями $\varphi_k \equiv \varphi(x_k)$ по формулам (6.1)–(6.3), и поэтому метод конечных элементов сводится к нахождению этих значений.

Для данного дифференциального уравнения существует вариационная формулировка. Поэтому для решения воспользуемся методом Ритца. Подставляя в функционал задачи

$$F[\varphi(x)] = \frac{1}{2} \int_1^2 (x^2 \varphi'^2 + 2\varphi^2 - 2(1 + 2/x)\varphi) dx - 4\varphi(2)$$

кусочную аппроксимацию (10.3), получим

$$\begin{aligned} F[\hat{\varphi}(x)] = & \frac{1}{2} \int_{x_1}^{x_2} [x^2 (\varphi^{(1)})'^2 + 2(\varphi^{(1)})^2 - 2(1 + 2/x)(\varphi^{(1)})] dx + \\ & + \int_{x_2}^{x_3} [x^2 (\varphi^{(2)})'^2 + 2(\varphi^{(2)})^2 - 2(1 + 2/x)(\varphi^{(2)})] dx + \\ & + \dots + \frac{1}{2} \int_{x_n}^{x_{n+1}} [x^2 (\varphi^{(n)})'^2 + 2(\varphi^{(n)})^2 - 2(1 + 2/x)(\varphi^{(n)})] dx - 4\varphi^{(n)}, \end{aligned} \quad (6.4)$$

т.е. сумму функционалов отдельных конечных элементов.

Учитывая конкретный вид функции $\varphi^{(k)}(x)$ из (6.1) с базисными функциями (6.2), в этом выражении можно выпол-

нить интегрирование по конечным элементам (при этом считая $\varphi_k \equiv \varphi(x_k)$ постоянными числами). В итоге функционал превратится в функцию многих переменных $\Phi(\varphi_2, \dots, \varphi_{n+1})$. Числа φ_k теперь выступают как параметры. Обратите внимание, что в списке аргументов Φ отсутствует параметр $\varphi_1 = \varphi(1)$, поскольку он известен из соответствующего граничного условия Дирихле при $x = 1$ и должен быть зафиксирован. Далее, как обычно необходимое условие экстремума $\Phi(\varphi_2, \dots, \varphi_{n+1})$ приводит к системе алгебраических уравнений

$$\frac{\partial \Phi}{\partial \varphi_k} = 0, \quad k = 2, \dots, n.$$

Решая эту систему, найдем требуемые значения параметров $\{\varphi_2^*, \varphi_3^*, \dots, \varphi_{n+1}^*\}$ и тем самым приближенное решение краевой задачи. В отличие от метода конечных разностей, МКЭ позволяет непосредственно найти значения функции во всех точках отрезка $1 \leq x \leq 2$ (а не только в узловых) согласно представлению (6.3).

Порядок решения в системе Maple

1. В соответствии с МКЭ, разбиваем отрезок на 4 конечных элемента, и на каждом из них строим аппроксимацию:

```
> restart;
> N1 := [5-4.*x, 6-4*x, 7-4*x, 8-4*x];
   N2 := [4.*x-4., 4*x-5, 4*x-6, 4*x-7];
> z := unapply([y1*N1[1]+y2*N2[1], y2*N1[2]+y3*N2[2],
y3*N1[3]+y4*N2[3], y4*N1[4]+y5*N2[4]], x);
   z := x -> [y1(5-4x) + y2(4x-4), y2(6-4x) + y3(4x-5),
             y3(7-4x) + y4(4x-6), y4(8-4x) + y5(4x-7)]
```

Таким образом, ввели функцию $z(x)$ в виде вектора, т.е. вектор-функцию, i -й элемент которой есть аппроксимация на i -м элементе. Обращение к ней дополнительно содержит номер конечного элемента в квадратных скобках: например, $\mathbf{z}(\mathbf{x})[2]$

означает аппроксимацию второго элемента и, следовательно, будет замещаться выражением $y_2 (6 - 4x) + y_3 (4x - 5)$.

2. Для каждого конечного элемента составляем функционалы (вид функционалов – как в методе Ритца). Для 1-го элемента

```
> f1 := (1/2) * int (x^2 * diff (z (x) [1] , x) ^2 +
2 * expand (z (x) [1] ^2) - 2 * (1 + 2 ./ x) * z (x) [1] , x = 1 .. 1.25) ;
```

Для второго элемента

```
> f2 := (1/2) * int (x^2 * diff (z (x) [2] , x) ^2 +
2 * expand (z (x) [2] ^2) - 2 * (1 + 2 ./ x) * z (x) [2] , x = 1.25 .. 1.5) ;
```

Для третьего элемента

```
> f3 := (1/2) * int (x^2 * diff (z (x) [3] , x) ^2 +
2 * expand (z (x) [3] ^2) - 2 * (1 + 2 ./ x) * z (x) [3] , x = 1.5 .. 1.75) ;
```

Для 4-го элемента в функционале присутствует граничная добавка $-4 * z(2) [4]$, поскольку этот элемент примыкает к границе $x = 2$:

```
> f4 := (1/2) * int (x^2 * diff (z (x) [4] , x) ^2 +
2 * expand ((z (x) [4] ) ^2) - 2 * (1 + 2 ./ x) * z (x) [4] ,
x = 1.75 .. 2) - 4 * z (2) [4] ;
```

3. Из условия, что суммарный функционал должен достигать экстремума, строим систему уравнений. При этом, заметьте, что первое уравнение, соответствующее узлу с номером 1, заменено на $y_1 = 0$, поскольку в левой граничной точке задано условие 1-го рода $\varphi(1) = 0$.

```
> F := f1 + f2 + f3 + f4 ;
> eqns := {y1 = 0, diff (F, y2) , diff (F, y3) , diff (F, y4) ,
diff (F, y5) } ;
eqns := { -14.00000000 y4 + 14.25000000 y5 - 4.255560503,
-5.000000000 y1 + 13.00000000 y2 - 0.6527102710 - 7.500000000 y3,
-7.500000000 y2 + 18.50000000 y3 - 0.5848939496 - 10.50000000 y4,
-10.50000000 y3 + 25.00000000 y4 - 0.5366941241 - 14.00000000 y5, y1 = 0 }
```

4. Находим ее решение:

```
> r := solve (eqns, {y1, y2, y3, y4, y5}) ;
> r1 := evalf (r) ;
```

```
r1 := {y1 = 0, y5 = 1.8374183725364, y2 = 0.75877819934262, y4 = 1.5662608075276
      y3 = 1.2281875093920}
```

Эти числа и есть значения искомой функции в узловых точках. Чтобы сравнить их с точными значениями, воспользуемся набором команд:

```
> X := [1, 1.25, 1.5, 1.75, 2] :
   Y := subs (r1, [y1, y2, y3, y4, y5]) ;
> st := dsolve ({diff (x^2*diff (y (x), x), x) - 2*y (x) =
- (1+2.*1/x), y (1)=0, D (y) (2)=1}, y (x)) ;
> u := unapply (subs (st, y (x)), x) ;
> for i from 1 to 5 do printf (`x=%6.3g   z=%6.4g
u=%g\n`, X[i], Y[i], u (X[i])) ; od;
x= 1.000   z=0.0000   u=0
x= 1.250   z= .7587   u=.767
x= 1.500   z=1.2281   u=1.238888
x= 1.750   z=1.5662   u=1.578061
x= 2.000   z=1.8374   u=1.85
```

5. Решение конечно-элементной задачи как непрерывная функция однозначно определяется своими узловыми значениями. Она строится кусочным образом по всем конечным элементам, причем для задания функции внутри конечного элемента берутся узловые значения только этого элемента и используется формула $y^e = \sum_i y_i N_i^e(x)$.

С учетом сказанного определим искомую функцию **phi**, являющуюся решением нашей задачи. Это можно сделать с помощью встроенной функции **piecewise**:

```
> phi := x -> piecewise (x < X[2], subs (r1, z (x) [1]),
x < X[3], subs (r1, z (x) [2]), x < X[4], subs (r1, z (x) [3]),
subs (r1, z (x) [4])) : evalf (phi (x), 4) ;
```

Понимается эта запись так: если значение **x** попадает в интервал до **X[2]**, т.е. принадлежит первому конечному элементу, то функции **phi(x)** присваивается **z(x)[1]** – первая функция вектора **z(x)**, если **x** принадлежит отрезку **[X[2], X[3]]** (второй конечный элемент), присваивается **z(x)[2]** – вторая функция списка **z[x]** и т.д. Теперь к этой функции можно об-

ращаться как к обычной функции системы Maple. Например, построить график:

```
> g4:=plot(phi(x),x=X[1]..X[5],color=magenta,
thickness=2);
> g2:=plot([[X[i],Y[i]]$i=1..5],x=X[1]..X[5],
style=POINT,symbol=CIRCLE,color=black):
plots[display](g2,g4);
```

Чтобы этот график сравнить с точным решением, следует задать:

```
> plot([phi(x),u(x)],x=X[1]..X[5],color=[red,blue],
thickness=[2,2]);
```

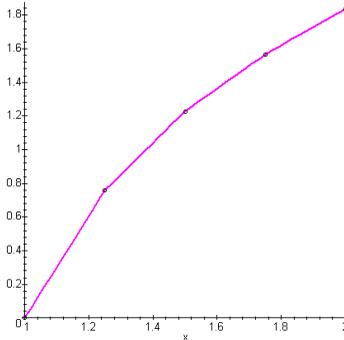
Найти норму погрешности аппроксимации можно так:

```
> int((phi(x)-u(x))^2,x=1..2)^(1/2);
0.02838184159
```

6. Изобразим на графике в виде точек полученные конечно-элементные значения, для чего на координатной плоскости отдельно нанесем точки, а затем совместим два построения:

```
> g2:=plot([[X[i],Y[i]]$i=1..5],x=X[1]..X[5],
style=POINT,symbol=CIRCLE,color=black):
plots[display](g2,g4);
```

Вид этого рисунка наводит на мысль, что искомую функцию можно получить посредством обычной локальной интерполяции. В данном случае имеем кусочно-линейную интерполяцию (используются линейные конечные элементы). А это значит, что две соседние точки соединяются прямой линией.



7. Чтобы доказать сходимость метода, надо провести несколько вычислений с большим числом конечных элементов. Это удобно сделать с помощью следующего фрагмента

```

> restart;
> a:=1; b:=2; n:=8; h:=(b-a)/n; m:=n; #[1]
> N1:=x->seq(a/h-x/h+i,i=1..n);
  N2:=x->seq(x/h-a/h-i+1,i=1..n); #[2]
> z:=x->seq(y[i-1]*N1(x)[i]+y[i]*N2(x)[i],i=1..n); #[3]
> F:=0; for i from 1 to n do
F:=F+(1/2)*int(x^2*diff(z(x)[i],x)^2+
2*expand((z(x)[i]^2))-2*(1+2./x)*z(x)[i],
x=a+(i-1)*h..a+i*h);od; #[4]
F:=F-4*z(2)[n]; #[5]
> eqts:=[seq(diff(F,y[i])=0,i=0..m)];
eqts[1]:=y[0]=0; #[6]
> rez:=solve({seq(eqts[i],i=1..m+1)},
{seq(y[i],i=0..m)}); #[7]
> X:=[seq(a+i*h,i=0..m)];
  Y:=subs(rez,[seq(y[i],i=0..m)]); #[8]
> readlib(spline);
> f1:=spline(X,Y,x,linear);
> g4:=plot(f1,x=a..b,color=magenta,thickness=2):
> g2:=plot([[X[j],Y[j]]$j=1..m+1],x=X[1]..X[m],
style=POINT,symbol=CIRCLE,color=black):
> st:=dsolve({diff(x^2*diff(y(x),x),x)-2*y(x)=
-(1+2.*1/x),y(1)=0,D(y)(2)=1},y(x));
> u:=unapply(subs(st,y(x)),x);
> g5:=plot(u(x),x=a..b,color=blue,thickness=2):
> plots[display](g2,g4,g5);
> delta:=evalf(int((f1-u(x))^2,x=a..b)^(1/2)); #[9]

```

Комментарий. Строка 1: задание переменных **a**, **b** (начало и конец интервала), **n** (число элементов), **h** (длина конечного элемента). Строка 2: ввод базисных функций для линейных элементов; **N1** и **N2** – последовательности из **n** базисных функ-

ций, соответствующих левым и правым узлам конечных элементов. Строка 3: определение последовательности $\mathbf{z}[\mathbf{x}]$, включающего аппроксимирующие выражения для всех конечных элементов. Строки 4: вычисление функционала через цикл по всем элементам. Строка 5: учет граничного условия 2-го рода в виде добавки к функционалу, соответствующей правой граничной точке. Строка 6: формирование системы уравнений; их количество равно $(n+1)$ – по числу узлов. Здесь же учет граничного условия 1-го рода в левой граничной точке. Строка 7: решение полученной системы; результат записывается в **rez**. Строка 8 и далее: построение графиков, сравнение с точным решением. Строка 9: вычисление нормы ошибки решения.

Для проверки сходимости нужно менять всего один параметр в приведенном фрагменте – число n . В результате получим таблицу:

	$n = 4$	$n = 8$	$n = 16$
δ	0.02838184159	0.007361378471	0.001858766257

8. Точность МКЭ повышается не только с ростом числа элементов (или узлов), но и с увеличением порядка интерполяции на каждом конечном элементе. Найдем решение задачи, используя не линейные элементы, как раньше, а квадратичные. Это означает, что неизвестная функция внутри конечного элемента строится как квадратичный полином

$$\varphi^e = C_1^e x^2 + C_2^e x + C_3^e. \quad (6.5)$$

Самый простой способ – за такой полином взять многочлен Лагранжа

$$\varphi = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \varphi_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \varphi_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \varphi_2,$$

где x_0, x_1 и x_2 – координаты трех узлов элемента – двух граничных (x_0 и x_2) и одной внутренней (x_1) точек; $\varphi_0, \varphi_1, \varphi_2$ – значения функции в соответствующих узлах. Очевидно, многочлен

Лагранжа обеспечивает непрерывность функции при переходе от элемента к элементу.

Сравнивая последнее выражение с общим видом аппроксимации $\varphi^e = \sum_i \varphi_i N_i^e(x)$, получаем на конечном элементе e три базисные функции

$$N_1^e(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}, \quad N_2^e(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)},$$

$$N_3^e(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)},$$

каждая из которых приписывается одному из узлов x_0, x_1 и x_2 элемента по правилу $N_i^e(x_j) = \delta_{ij}$ (т.е. для «своего» узла эта функция равна единице, а для остальных – нулю). Эти базисные функции меняются от элемента к элементу. Поэтому при программировании придется вводить $3n$ таких функций. Это легко реализуется, если все элементы имеют одинаковую длину. К примеру,

```
> restart;
> a:=1;b:=2;n:=4;h:=(b-a)/n; m:=2*n;
> N1:=x->seq(subs(x0=a+(i-1)*h,x1=a+(i-1)*h+h/2,
x2=a+i*h,(x-x1)*(x-x2)/((x0-x1)*(x0-x2))),i=1..n):
```

Аналогично определяются две другие базисные функции:

```
> N2:=x->seq(subs(x0=a+(i-1)*h,x1=a+(i-1)*h+h/2,
x2=a+i*h,(x-x0)*(x-x2)/((x1-x0)*(x1-x2))),i=1..n):
> N3:=x->seq(subs(x0=a+(i-1)*h,x1=a+(i-1)*h+h/2,
x2=a+i*h,(x-x0)*(x-x1)/((x2-x0)*(x2-x1))),i=1..n);
```

Есть другой, более универсальный, способ задания базисных функций. Он основан на вычислении коэффициентов $\{C_i^e\}$ в выражении типа (6.5) из каких-либо условий, например, что функция в узлах $\{x_i\}$ должна принимать значения $\{\varphi_i\}$. Кстати, эти условия могут быть наложены и на производные, и здесь мы приходим к эрмитовой интерполяции, сплайн-интерполяции и

др. Разберите следующий листинг программы, реализующий ввод этим универсальным способом базисных функций интерполяции Лагранжа 2-го порядка. (Кстати, этот способ лучше приспособлен для случая неравноотстоящих узлов.)

```
> restart;
> a:=1;b:=2;n:=4;h:=(b-a)/n; m:=2*n;
> wt:=x->c1*x^2+c2*x+c3;
> wr:=solve({wt(xi)=w1,wt(xi+h/2)=w2,wt(xi+h)=w3},
{c1,c2,c3});
> uf:=collect(subs(wr,wt(x)),{w1,w2,w3});
uv:=[diff(uf,w1),diff(uf,w2),diff(uf,w3)];
> N1:=x->seq(subs(xi=a+(i-1)*h,uv[1]),i=1..n):
N2:=x->seq(subs(xi=a+(i-1)*h,uv[2]),i=1..n):
N3:=x->seq(subs(xi=a+(i-1)*h,uv[3]),i=1..n):
```

Следующий шаг – задание аппроксимации неизвестной функции. Задавая узловые величины через массив **y**, имеем

```
> z:=x->seq(y[2*i-2]*N1(x)[i]+y[2*i-1]*N2(x)[i]+
y[2*i]*N3(x)[i],i=1..n);
```

Дальнейшие шаги почти полностью повторяют строки [4]–[8] листинга из пункта 7. Следует только учесть, что переменная **m** – число узлов минус один – здесь определяется иначе.

```
> F:=0; for i from 1 to n do
F:=F+(1/2)*int(x^2*diff(z(x)[i],x)^2+
2*expand((z(x)[i]^2))-2*(1+2./x)*z(x)[i],
x=a+(i-1)*h..a+i*h);od: # [4]
> F:=F-4*z(2)[n]: # [5]
> eqts:=[seq(diff(F,y[i])=0,i=0..m)]:
eqts[1]:=y[0]=0: # [6]
> rez:=solve({seq(eqts[i],i=1..m+1)},
{seq(y[i],i=0..m)}); # [7]
> X:=[seq(a+i*h/2,i=0..m)];
Y:=subs(rez,[seq(y[i],i=0..m)]); # [8]
```

Изображение вычисленных значений в узловых точках:

```
> g2:=plot([X[j],Y[j]]$j=1..m+1,x=X[1]..X[m],
style=POINT,symbol=CIRCLE,color=black):
```

Нахождение точного решения:

```
> st:=dsolve({diff(x^2*diff(y(x),x),x)-2*y(x)=
-(1+2.*1/x),y(1)=0,D(y)(2)=1},y(x));
> u:=unapply(subs(st,y(x)),x);
> g5:=plot(u(x),x=a..b,color=blue,thickness=2):
```

Представление конечно-элементного решения в виде кусочной функции:

```
> v1:=x<=X[1],0;
for k from 1 to n do
v1:=v1,x<=X[2*k+1],subs(rez,z(x)[k]); od:
> z2:=piecewise(v1,0):
> g4:=plot(z2,x=a..b):
```

Совмещение всех построений:

```
> plots[display](g2,g4,g5);
```

Вычисление нормы ошибки решения:

```
> Digits:=20;
> evalf(int((z2-u(x))^2,x=a..b)^(1./2); # [9]
```

Заметим, данный листинг позволяет проанализировать сходимость МКЭ в рамках квадратичной аппроксимации при увеличении числа элементов (нужно менять только **n**).

Упражнения

1. Решить задачу из примера

$$(x^2\varphi)' - 2\varphi = -1 - 2/x, \quad \varphi(1) = 0, \quad \varphi'(2) = 1$$

для разных чисел конечных элементов ($n = 2 \div 6$ и более). Показать сходимость решения, исследуя поведение функционала, аппроксимации и граничного значения $\varphi'(2)$.

2. Решить ту же задачу, используя квадратичные элементы Лагранжа. Показать сходимость решения, исследуя поведение функционала, аппроксимации и граничного значения $\varphi'(2)$ для разного количества конечных элементов (начать с $n = 2$).

Провести также анализ сходимости решения в аспекте сравнения задач с одинаковым полным числом узлов, но с разными типами конечных элементов.

3. Решить методом конечных элементов уравнение

$$\frac{d}{dx} \left(x^2 \frac{d\varphi}{dx} \right) - 2\varphi = -1 - \ln x^2$$

- а) на отрезке $1 \leq x \leq 2$ с краевыми условиями $\varphi(1) = 2, \varphi'(2) = -1$;
- б) на отрезке $1 \leq x \leq 2$ с краевыми условиями $\varphi'(1) = 0, \varphi(2) = 1$;
- в) на отрезке $1 \leq x \leq 2$ с краевыми условиями $\varphi'(1) = 1, \varphi'(2) = 0$;
- г) на отрезке $0,5 \leq x \leq 2,5$ с краевыми условиями $\varphi(0,5) = 0, \varphi'(2,5) = 1$;
- д) на отрезке $1 \leq x \leq 2,5$ с краевыми условиями $\varphi'(1) = 2, \varphi(2,5) = 0$;
- е) на отрезке $0,6 \leq x \leq 3,2$ с краевыми условиями $\varphi(0,6) = 2, \varphi(3,2) = 0$.

4. Решить методом конечных элементов краевые задачи из упражнения **4** раздела 4 «Метод Рунге». Использовать линейные элементы Лагранжа. Сравнить с точным решением. Построить графики. Провести анализ сходимости аппроксимации.

5. Решить ту же задачу, но с использованием квадратичных и кубических элементов.

6. Сравнить вычисления по МКЭ для случаев одинаковых и неодинаковых размеров конечных элементов. В последнем случае длины элементов выбираются в зависимости от характера поведения решения.

7. С помощью МКЭ найти решение краевой задачи, не имеющей аналитического решения. Условия задач взять из упражнения **6** раздела 4. Показать сходимость аппроксимаций.

10. Решить задачу одномерной теплопроводности

$$\frac{d}{dx} \left(k \frac{d\varphi}{dx} \right) = -Q; \quad 0 \leq x \leq 1, \quad \varphi(0) = 0, \quad k \frac{d\varphi}{dx}(1) = 1,$$

где

а) $k = 1$, $Q = 1$ при $0 \leq x \leq 1/2$ и $Q = 0$ при $1/2 \leq x \leq 1$;

б) $Q = 1$, $k = 1$ при $0 \leq x \leq 1/2$ и $k = 3$ при $1/2 \leq x \leq 1$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кострюков С.А., Пешков В.В., Шунин Г.Е., Шулгина В.А. Компьютерный практикум по методам вычислений: учеб. пособие. Воронеж: ВГТУ, 2018. – 179 с.; электронный ресурс кафедры ВМФММ.

2. Кострюков С.А., Пешков В.В., Шунин Г.Е. Основы вариационного исчисления [Текст]: учеб. пособие / С.А. Кострюков, В.В. Пешков, Г.Е. Шунин. – Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2011. – 158 с.

3. Алексеев, Е.Р. Решение задач вычислительной математики в пакетах Mathcad 12, MATLAB 7, Maple 9. (Самоучитель). – М.: ИТ Пресс, 2006. – 496 с.

ОГЛАВЛЕНИЕ

СОДЕРЖАНИЕ РАЗДЕЛА РАБОЧЕЙ ПРОГРАММЫ	3
1. РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ	3
1.1. Основные понятия	3
1.2. Численное решение задачи Коши	6
Упражнения	11
2. ПРИБЛИЖЕНИЕ (АППРОКСИМАЦИЯ) ФУНКЦИЙ	15
2.1. Введение	15
2.2. Интерполирование	17
2.3. Локальная интерполяция	22
2.4. Интерполирование сплайнами	25
2.5. Среднеквадратическое приближение	29
Упражнения	35
3. МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ	39
Упражнения	49
4. МЕТОД РИТЦА	53
4.1. О прямых методах вариационного исчисления.	53
4.2. Простейшая задача вариационного исчисления. Краевая задача 1-го рода	54
4.3. Вариационная задача Больца. Краевая задача 2-го и 3-го рода	57
Упражнения	65
5. МЕТОД ГАЛЁРКИНА	69
Упражнения	77
6. МЕТОД КОНЕЧНЫХ ЭЛЕМЕНТОВ	78
Упражнения	89
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	91