

Министерство науки и высшего образования Российской Федерации
Ярославский государственный университет им. П. Г. Демидова
Кафедра математического моделирования

А. В. Секацкая

**Система «Wolfram Mathematica»
в задачах математической физики
и математическом моделировании**

Учебно-методическое пособие

Ярославль
ЯрГУ
2021

УДК 519.6(075.8)
ББК В19я73
С28

*Рекомендовано
Редакционно-издательским советом университета
в качестве учебно-методического издания. План 2021 года.*

Рецензент
кафедра математического моделирования
Ярославского государственного университета им. П. Г. Демидова

Секацкая, Алина Вадимовна.

С28 Система «Wolfram Mathematica» в задачах математической физики и математическом моделировании : учебно-методическое пособие / А. В. Секацкая; Яросл. гос. ун.-т им. П. Г. Демидова. — Ярославль : ЯрГУ, 2021. — 60 с.

Пособие содержит основные понятия практического курса «Пакеты прикладных математических программ и математическое моделирование», «Численные методы». Большое внимание уделено решению практических задач на базе применения программы «Wolfram Mathematica» с разбором теоретической части поставленных задач. Пособие включает достаточно большое число упражнений для самостоятельной работы.

Учебно-методическое пособие написано по дисциплинам «Пакеты прикладных математических программ и математическое моделирование», «Численные методы» и предназначено для студентов бакалавриата по направлению «Математика и компьютерные науки». Оно может быть использовано студентами магистратуры по указанному направлению, а также направлению «Прикладная математика и информатика».

Библиогр.: 16 назв., табл., 13 рис.

УДК 519.6(075.8)
ББК В19я73

© ЯрГУ, 2021

Оглавление

1	Математическое моделирование	4
1.1	Введение в математическое моделирование	4
1.2	Уравнение колебания струны с закрепленными концами	5
1.2.1	Метод Фурье	6
1.2.2	Разностная схема	9
1.3	Уравнение теплопроводности	11
1.3.1	Явная разностная схема	12
1.3.2	Неявная разностная схема	13
1.4	Моделирование брошенного тела под углом к горизонту	16
2	Численные методы	20
2.1	Основные определения и понятия	20
2.2	Численные методы решения линейных алгебраических систем	21
2.2.1	Метод Гаусса	21
2.2.2	Метод простой итерации	27
2.2.3	Метод Зейделя	32
2.2.4	Метод скорейшего спуска	38
2.3	Численное интегрирование	41
2.4	Численное решение ОДУ	46
2.4.1	Метод Эйлера	46
2.4.2	Метод Рунге—Кутта	49
	Литература	57

Глава 1

Математическое моделирование

1.1 Введение в математическое моделирование

Моделирование — метод познания окружающего мира, который может быть отнесен к общенаучным методам, применяемым как на эмпирическом (наблюдение, эксперимент, измерение), так и на теоретическом (идеализация, формализация) уровне познания. В процессе построения и исследования модели могут быть использованы практически все остальные методы познания.

Современную науку сложно представить без математического моделирования. Это связано с тем, что моделирование является одним из важнейших методов, которые позволяют ускорить технический прогресс, сократить сроки освоения новых видов оборудования, новых процессов.

Под **моделью** понимается такой материальный или мысленно представляемый объект, который в процессе познания (изучения) замещает объект-оригинал, сохраняя некоторые важные для данного исследования типичные его черты. Процесс построения и использования модели называется *моделированием*.

Другими словами, *модель* — это объект-заменитель объекта-оригинала, обеспечивающий изучение некоторых интересующих исследователя свойств оригинала.

Таким образом, любая модель является не тождественной объекту-оригиналу, поскольку при ее построении учитывались только важнейшие с точки зрения исследователя факторы. Поэтому любая модель является *неполной*. «Полная» модель — полностью тождественна оригиналу.

В случае когда полученные при исследовании модели результаты позволяют предугадать дальнейшее поведение или свойства исследуемого объекта, говорят, что модель *адекватна* объекту.

Математическое моделирование — это идеальное научное знаковое формальное моделирование, при котором описание объекта осуществляется на языке математики, а исследование модели проводится с использованием тех или иных математических методов.

Если модель описывается некоторыми уравнениями, то она называется *детерминированной*. Начально-краевые и краевые задачи математической физики — примеры детерминированных дифференциальных моделей.

Если модель описывается вероятностными законами, то она называется *стохастической*.

Математические модели должны обладать свойством *универсальности*. Это означает, что одна математическая модель может описывать явления и объекты, которые имеют различную физическую природу, но при этом могут быть описаны одними и теми же математическими соотношениями, логическими и структурными схемами. Такое моделирование называется **аналоговым**.

Примерами аналоговых моделей могут служить электрические и механические колебания, малые колебания при взаимодействии двух биологических популяций, колебания в простейшей модели изменения зарплаты и занятости, которые с точки зрения математики описываются уравнением колебаний:

$$u_{tt} = a^2 u_{xx},$$

но относятся к качественно отличающимся физическим процессам.

Уравнение математической физики — уравнение Лапласа

$$u_{xx} + u_{yy} + u_{zz} = 0$$

— описывает такие явления, как гравитационные, оптические, акустические, электромагнитные излучения, полет самолета, колебания упругих тел и строение атома, в том числе стационарное тепловое поле и стационарное электростатическое поле.

Обобщая, можно сказать, что модель составляется для следующих задач.

1. Понять, как устроен изучаемый объект: изучить его структуру, основные свойства, законы развития и взаимодействия с окружающим миром.
2. Научиться управлять объектом (процессом) и определять наилучшие способы управления при заданных целях и критериях.
3. Прогнозировать прямые и косвенные последствия реализации заданных способов и форм воздействия на объект.

1.2 Уравнение колебания струны с закрепленными концами

Рассмотрим простейшую задачу о поперечных колебаниях струны, закрепленной на концах в точках $x = 0$ и $x = l$ оси x -ов. График функции $f(x)$, определенной на отрезке $[0, l]$, задает профиль плоской струны в начальный момент времени $t = 0$, а функция $g(x)$ определяет начальную скорость точки струны с абсциссой x . Процесс колебаний можно описать одной функцией $u[t, x]$, $x \in [0, l]$, характеризующей вертикальное перемещение струны в момент времени $t \in [0, T]$. Таким образом, функция $u[x, t]$ дает отклонение струны от оси x и удовлетворяет дифференциальному уравнению [15]:

$$u_{tt} = a^2 u_{xx}. \quad (1.1)$$

Если концы струны закреплены $0 \leq x \leq l$, то должны выполняться «граничные условия»:

$$u(0, t) = 0, \quad u(l, t) = 0. \quad (1.2)$$

Поскольку процесс колебаний струны зависит от ее начальной формы и распределения скоростей, необходимо задать «начальные условия»:

$$u(x, t_0) = \varphi(x), \quad u_t(x, t_0) = \psi(x). \quad (1.3)$$

Таким образом, дополнительные условия состоят из граничных и начальных условий, где $\varphi(x)$ и $\psi(x)$ — заданные функции.

Данную задачу можно решить точно методом Фурье или приближенно при помощи разностного метода. Начнем с нахождения точного решения и впоследствии с этим результатом сравним приближенное решение.

1.2.1 Метод Фурье

Метод разделения переменных, или метод Фурье, является одним из наиболее распространенных методов решения уравнений с частными производными. Пусть необходимо найти решение уравнения свободных колебаний струны (1.1), удовлетворяющее однородным краевым условиям (1.2) и начальным условиям (1.3).

По методу Фурье решение приведенной задачи Коши $u[x, t]$ равно сумме бесконечного ряда Фурье [15]:

$$u(x, t) = \sum_{n=1}^{\infty} \left(A_n \cos \frac{\pi n}{l} at + B_n \sin \frac{\pi n}{l} at \right) \sin \frac{\pi n}{l} x, \quad (1.4)$$

где

$$A_n = \frac{2}{l} \int_0^l \varphi(\xi) \sin \left(\frac{\pi n}{l} x \right) dx, \quad B_n = \frac{2}{\pi n a} \int_0^l \psi(x) \sin \left(\frac{\pi n}{l} x \right) dx.$$

Рассмотрим конкретную задачу колебания струны, найдем точное решение и построим анимацию бегущих волн при следующих начальных условиях:

$$\varphi(x) = \begin{cases} \frac{x}{2} + 1, & \text{если } x \in [-2, 2], \\ 4 - x, & \text{если } x \in [2, 4], \\ 0, & \text{если } x \notin [-2, 4], \end{cases} \quad \psi(x) = 0.$$

Для начала зададим параметр a и длину струны в состоянии покоя:

$$\text{In}[1]:= a = 0.1; \quad l = 10;$$

Далее вводим исходные данные, а именно функции $\varphi(x)$ и $\psi(x)$ из начальных условий:

$$\text{In}[2]:= \phi[x_]= \text{Which}[-2 \leq x \leq 2, 1/2x + 1, 2 \leq x \leq 4, 4 - x, x < -2, 0,$$

```

x > 4, 0];
ψ[x_] = 0;

```

Чтобы наглядно посмотреть на профиль плоской струны в начальный момент времени $t = 0$, нарисуем график функции $\varphi(x)$:

```

In[3]:= Plot[φ[x], {x, -5, 5}, PlotRange → All]

```

Дополнительно можно построить график скорости точек струны при $t = 0$:

```

In[4]:= Plot[ψ[x], {x, 0, 5}]

```

Для исключения ошибки при вводе кусочной функции можно воспользоваться функцией «`PiecewiseExpand[expr]`»:

```

In[5]:= PiecewiseExpand[φ[x]]

```

$$\text{Out[5]} := \begin{cases} 4 - x & 2 < x \leq 4 \\ \frac{2 + x}{2} & -2 \leq x \leq 2 \\ 0 & \text{True.} \end{cases}$$

Затем вводим функции, которые являются коэффициентами n -го члена ряда Фурье:

$$\begin{aligned} \text{In[6]} := A[n_] &= \frac{2}{l} \int_0^l \phi[x] \sin \left[\frac{\pi n}{l} x \right] dx; \\ B[n_] &= \frac{2}{\pi n a} \int_0^l \psi[x] \sin \left[\frac{\pi n}{l} x \right] dx; \end{aligned}$$

и функцию $u(x, t)$ в виде суммы первых пяти членов ряда Фурье:

$$\text{In[7]} := u[x_, t_] = \sum_{n=1}^5 \left(A[n] \cos \left[\frac{\pi n}{l} a t \right] + B[n] \sin \left[\frac{\pi n}{l} a t \right] \right) \sin \left[\frac{\pi n}{l} x \right] // N;$$

Далее построим графики функции $u(x, t)$ при t , изменяющемся от 0 до 200 с шагом 1.

```

In[8]:= Animate[Plot[u[x, t], {x, 0, l}, PlotStyle → {Hue[0, 7]},
  Epilog → {Text[tToString[t], {10, 1.8}]}],
  PlotRange → {{0, l + 1}, {-2, 2}}, {t, 0, 200}]

```

В результате получим анимацию положений струны, состоящую из 201 кадра:

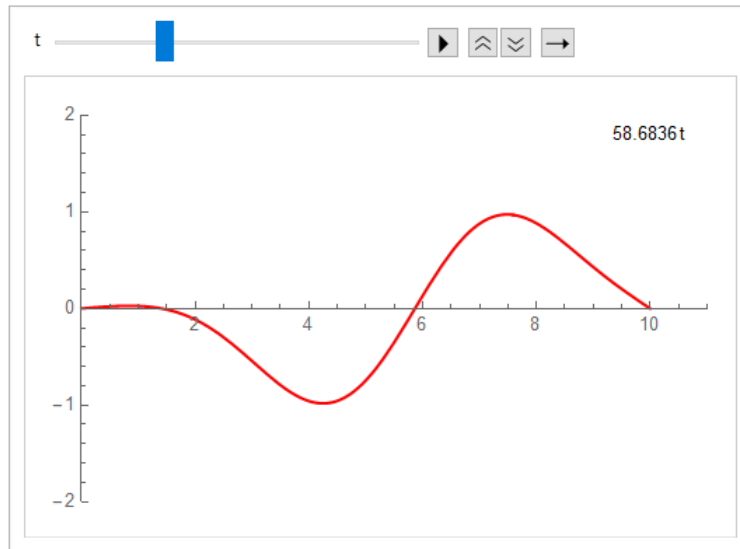


Рис. 1.1. Анимация полученного решения уравнения колебания струны. Метод Фурье

Чтобы отдельно вывести несколько положений струны в конкретные моменты времени, введем следующие команды:

```
In[9]:= gr1 = Table[Plot[u[x, t], {x, 0, l}, PlotStyle → {Hue[0, 7]},
  Epilog → {Text[tt, {10, 1.8}]}],
  PlotRange → {{0, l + 1}, {-2, 2}}, {t, 0, 100, 25}];
gr2 = Table[Plot[u[x, t], {x, 0, l}, PlotStyle → {Hue[0, 7]},
  Epilog → {Text[tt, {10, 1.8}]}],
  PlotRange → {{0, l + 1}, {-2, 2}}, {t, 100, 200, 25}];
Show[GraphicsGrid[{gr1, gr2}], ImageSize → {1000, 250}]
```

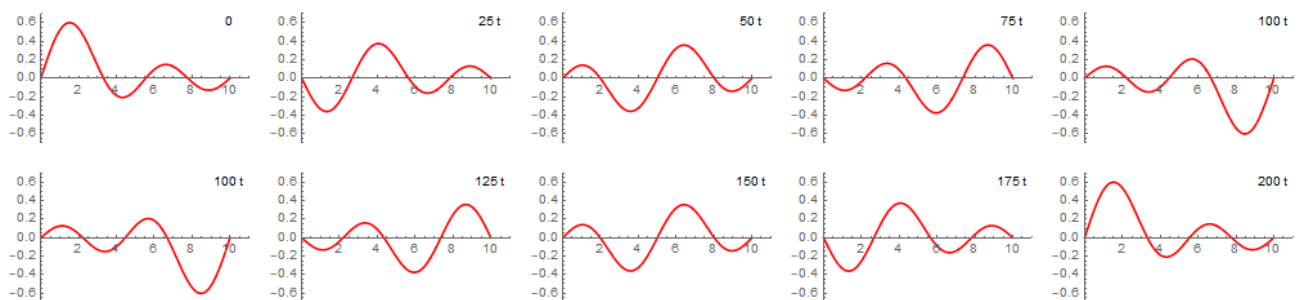


Рис. 1.2. Кадры из анимации решения уравнения колебания струны. Метод Фурье

В заключение можно отобразить все профили струны в виде трехмерного графика:

```
In[10]:= Plot3D[u[x, t], {x, 0, 5}, {t, 0, 100}]
```

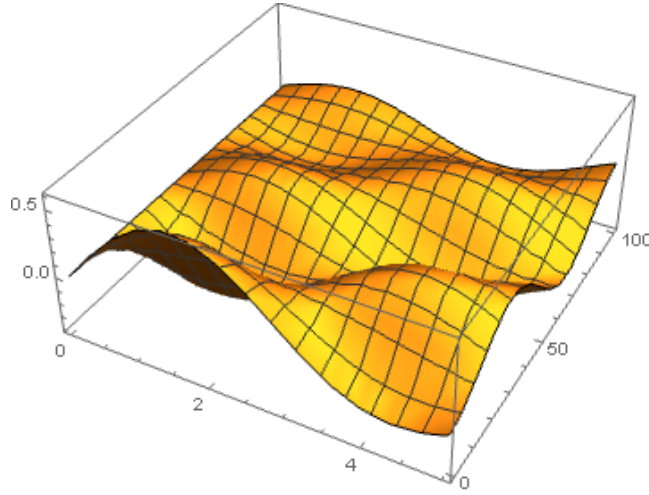



Рис. 1.3. Профили струны. Метод Фурье

1.2.2 Разностная схема

Далее найдем приближенное решение уравнения колебания струны (1.1), удовлетворяющего начально-краевым условиям (1.2), (1.3) в прямоугольнике $[0, l] \times [0, T]$ на сетке (x_i, t_k) , состоящей из $(nh + 1) \times (n\tau + 1)$ точек

$$x_i = 0 + ih, \quad i = 0, 1, \dots, nh,$$

$$t_k = 0 + \tau k, \quad k = 0, 1, \dots, n\tau,$$

где

$$h = \frac{l}{nh}, \quad \tau = \frac{T}{n\tau}.$$

Необходимо построить и решить сеточную краевую задачу, найдя точное решение в узлах сетки. Для этого будем искать сеточную функцию $\{u_{i,j}\}$, определенную на данной сетке, близкую к аналитическому решению дифференциальной задачи (1.1), (1.2), (1.3). Важно знать оценку погрешности найденного решения, то есть разностей $[u_{i,k} - u(x_i, t_k)]$.

Для построения сеточной краевой задачи аппроксимируем производные их разностными аналогами. Заменим значения вторых производных вторыми разностными производными:

$$u_{tt} \approx \frac{u_{i,k-1} - 2u_{i,k} + u_{i,k+1}}{\tau^2},$$

$$u_{xx} \approx \frac{u_{i-1,k} - 2u_{i,k} + u_{i+1,k}}{h^2}.$$

В итоге получаем разностное уравнение:

$$\frac{u_{i,k-1} - 2u_{i,k} + u_{i,k+1}}{\tau^2} = a^2 \frac{u_{i-1,k} - 2u_{i,k} + u_{i+1,k}}{h^2}. \quad (1.5)$$

Обозначим через

$$\beta = \frac{a^2 \tau^2}{h^2} \quad (1.6)$$

и выразим из уравнения (1.5) $u_{i,k+1}$:

$$u_{i,k+1} = -u_{i,k-1} + \beta \cdot u_{i-1,k} + (2 - 2\beta)u_{i,k} + \beta \cdot u_{i+1,k}. \quad (1.7)$$

Начальное условие (1.3) аппроксимируем следующим образом:

$$u_t(x_i, 0) \approx \frac{u_{i,1} - u_{i,0}}{\tau}. \quad (1.8)$$

Отсюда и из (1.3) находим $u_{i,1}$, и начальные условия (1.3) можно переписать так:

$$u_{i,0} = f(x_i), \quad u_{i,1} = u_{i,0} + \tau g(x_i). \quad (1.9)$$

Начальные условия (1.9) и формула (1.7) позволяет найти все значения табличной функции — численного решения данной дифференциальной задачи $u_{i,k}$.

Разностная схема устойчива, если выполняется соотношение шагов вида:

$$\tau \leq \frac{h}{a}. \quad (1.10)$$

При выполнении этого условия разностное решение сходится к аналитическому решению, то есть

$$|u_{i,k} - u(x_i, u_k)| \rightarrow 0$$

при стремлении шагов к нулю $h \rightarrow 0, \tau \rightarrow 0$ с соблюдением условия (1.10).

Приведем численное решение задачи Коши для уравнения колебания струны. Введем параметры задачи (параметр струны, длина струны, шаг для разбиения по переменным x и t):

$$\begin{aligned} \text{In[11]} := & \mathbf{a = 0.1; l = 5; T = 100; h = 0.1;} \\ & \mathbf{\tau = 1; n\tau = 100/\tau; nh = l/h; \beta = a^2\tau^2/h^2;} \end{aligned}$$

При этом шаги должны быть выбраны так, чтобы выполнялось условие устойчивости разностной схемы.

Далее необходимо вычислить все координаты узлов сетки, а именно x_i и t_k . Для этого создадим таблицу с вычисленными значениями:

$$\begin{aligned} \text{In[12]} := & \mathbf{Table[x_i = 0 + hi, \{i, 0, nh\}] // N;} \\ & \mathbf{Table[t_i = 0 + \tau i, \{i, 0, n\tau\}] // N;} \end{aligned}$$

Аналогично создадим таблицу с начальными и краевыми условиями:

$$\begin{aligned} \text{In[13]} := & \mathbf{Table[u_{0,k} = u_{nh,k} = 0, \{k, 0, n\tau\}] // N;} \\ & \mathbf{Table[u_{i,0} = f[x_i] // N, \{i, 0, nh\}];} \\ & \mathbf{Table[u_{i,1} = u_{i,0} + \tau g[x_i] // N, \{i, 1, nh - 1\}];} \end{aligned}$$

И последнее, что осталось выполнить — это найти решения построенной разностной схемы:

```
In[14]:= Table[ui,k+1 = -ui,k-1 + βui-1,k + (2 - 2β)ui,k +  
βui+1,k, {k, 1, nτ - 1}, {i, 1, nh - 1}];
```

Ответом в данной задаче будет таблица со следующими значениями:

```
In[15]:= gr = Table[{xi, ui,k}, {k, 0, nτ}, {i, 0, nh}];
```

Построим анимацию найденных профилей струны:

```
In[18]:= Animate[ListPlot[gr[[k]], Joined → True,  
PlotRange → {{0, 6}, {-15, 15}}], {k, 1, nτ, 1}]
```

По желанию можно отдельно вывести профили струны в конкретный момент времени аналогично тому, как это делалось при решении задачи на базе применения метода Фурье.

1.3 Уравнение теплопроводности

Уравнения с частными производными 2-го порядка параболического типа наиболее часто встречаются при изучении процессов теплопроводности и диффузии.

Рассмотрим процесс распространения тепла в пластине. Этот процесс может быть описан функцией $u(x, y, t)$, представляющей собой температуру в точке (x, y) плоской изотропной квадратной пластины $\Sigma = [0, 1] \times [0, 1]$ в момент времени $t \in [0, T]$.

Пусть начальное распределение температуры:

$$u(x, y, 0) = \varphi(x, y), \quad (1.11)$$

$(x, y) \in \Sigma$, и для всех значений температуры $t \in [0, T]$, для всех граничных точек пластины $(x, y) \in \partial\Sigma$ известно, что

$$u(x, y, t) = \psi(x, y, t). \quad (1.12)$$

При выполнении этих условий функция $u(x, y, t)$ удовлетворяет уравнению вида

$$u_{tt} = a^2 (u_{xx} + u_{yy}) + f(x, y, t), \quad (1.13)$$

где a — коэффициент температуропроводности [15], а выделение тепла может быть охарактеризовано объемной плотностью тепловых источников $f(x, y, t)$ в точке (x, y) в момент времени t . Если источники отсутствуют, то есть $f(x, y, t) = 0$, то уравнение теплопроводности принимает простой вид:

$$u_t = a^2 u_{xx}.$$

Требуется найти решение $u(x, y, t)$ дифференциального уравнения (1.13), удовлетворяющее начально-краевым условиям (1.11) и (1.12).

Решим задачу разностным методом.

1.3.1 Явная разностная схема

Зададим все необходимые параметры:

$$\begin{aligned} \text{In[16]} &:= f[x_ , y_ , t_] = x^2 + yt; \phi[x_ , y_] = 1; \\ \psi[x_ , y_ , t_] &= \text{Exp}[-xyt]; \\ a &= 1/10; T = 1; n = 10; m = 10; h = 1/n; \\ t &= T/m; \gamma = a^2 2t/h^2; \end{aligned}$$

Введем координаты узлов:

$$\begin{aligned} \text{In[17]} &:= \text{Table}[\{x_i = ih, y_i = ih\}, \{i, 0, n\}]; \\ &\text{Table}[t_i = it, \{i, 0, m\}]; \end{aligned}$$

Пусть сеточная функция $u_{i,j,k} \approx u(x_i, y_j, t_k)$. Тогда начально-краевые условия (1.11) и (1.12) с помощью такой сеточной функции можно записать в следующем виде:

$$\begin{aligned} \text{In[18]} &:= \text{Table}[u_{i,j,0} = \phi[x_i, y_j], \{i, 0, n\}, \{j, 0, n\}]; \\ &\text{Table}[u_{0,i,k} = \psi[0, y_i, t_k], \{i, 0, n\}, \{k, 1, m\}]; \\ &\text{Table}[u_{n,i,k} = \psi[1, y_i, t_k], \{i, 0, n\}, \{k, 1, m\}]; \\ &\text{Table}[u_{i,0,k} = \psi[x_i, 0, t_k], \{i, 1, n-1\}, \{k, 1, m\}]; \\ &\text{Table}[u_{i,n,k} = \psi[x_i, 1, t_k], \{i, 1, n-1\}, \{k, 1, m\}]; \end{aligned}$$

Запишем дифференциальное уравнение (1.13) в виде разностных уравнений. Для этого заменим производные в (1.13) разностными производными.

Разностные производные в точке с координатами (x_i, y_j, z_k) определяются по следующим формулам:

$$\begin{aligned} u_t &\approx \frac{u_{i,j,k+1} - u_{i,j,k}}{t}, \\ u_{xx} &\approx \frac{u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}}{h^2}, \quad u_{yy} \approx \frac{u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k}}{h^2}. \end{aligned}$$

Заменим производные в (1.13) соответствующими разностными производными и после преобразований получим разностные уравнения:

$$u_{i,j,k+1} = u_{i,j,k} + \gamma(u_{i-1,j,k} + u_{i+1,j,k} + u_{i,j-1,k} + u_{i,j+1,k} - 4u_{i,j,k}) + tf[x_i, y_j, t_k],$$

где

$$\gamma = a^2 \frac{t}{h^2}.$$

Разностные уравнения позволяют найти значения сеточной функции $u_{i,j,k+1}$ на $k+1$ слое, если известны все значения функции $u_{i,j,k}$ на нижележащем k слое.

Так как на границе значения функции $u_{i,j,k}$ заданы начально-краевыми условиями, то значения функции в остальных точках сетки можно получить в результате выполнения следующего цикла:

```
In[19]:=Do[ui,j,k+1 = ui,j,k + γ(ui-1,j,k + ui+1,j,k + ui,j-1,k + ui,j+1,k - 4ui,j,k) +  
tf[xi, yj, tk], {k, 0, m - 1}, {i, 1, n - 1}, {j, 1, n - 1}];
```

Ответ задачи запишем в виде:

```
In[20]:=Table[{xi, yj, tk, ui,j,k}, {i, 0, n}, {j, 0, n}, {k, 0, m}];
```

Для визуального восприятия решения введем список значений температур на k -м слое:

```
In[21]:=g[k_]:=Table[ui,j,k, {j, 0, 10}, {i, 0, 10}]
```

и построим приближенно графики функций $u = u(x, y, t)$ для значений времени $t = 0, 1, \dots, m$, идущего с шагом 3:

```
In[22]:=p = Table[ListPlot3D[g[k]], {k, 0, m, 3}]
```

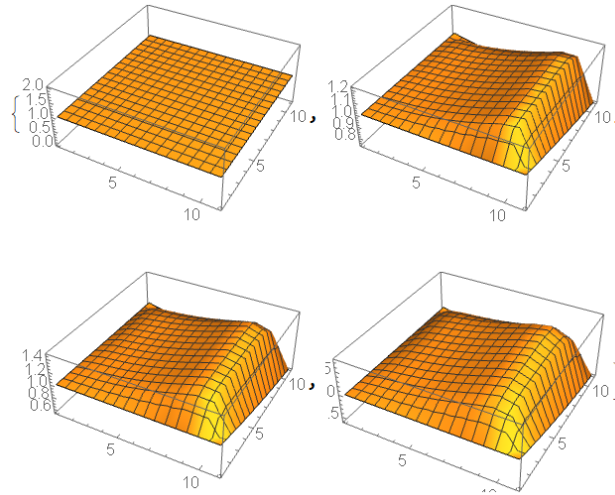


Рис. 1.4. Графики решения уравнения теплопроводности. Явная разностная схема

Рассмотренные разностные уравнения вместе с начальными условиями есть явная разностная схема. Она аппроксимирует дифференциальную задачу с точностью $O(h^2 + t)$. Разностная схема устойчива при

$$t < \frac{h^2}{4a^2}. \quad (1.14)$$

Это неравенство накладывает существенные ограничения на шаг по времени, что может привести к большим вычислительным затратам при больших значениях T .

1.3.2 Неявная разностная схема

Условие (1.14) можно снять. Для этого надо от уравнения (1.13) перейти к разностным уравнениям, заменяя первую частную производную в (1.13) на левую разностную

производную по формуле:

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j,k} - u_{i,j,k-1}}{t},$$

а вторые частные производные заменяются так же, как и в предыдущем случае. После подстановки разностных производных в (1.13) и преобразования получим следующее разностное уравнение (неявную разностную схему):

$$\gamma u_{i-1,j,k} + \gamma u_{i+1,j,k} + \gamma u_{i,j-1,k} + \gamma u_{i,j+1,k} - (1 + 4\gamma)u_{i,j,k} = -u_{i,j,k-1} - tf[x_i, y_j, t_k].$$

Очистим все переменные командой `ClearAll["Global`*"]`:

```
In[1]:=ClearAll["Global`*"]
```

Зададим начальные условия, параметры и шаг:

```
In[2]:= f[x_, y_, t_] = x^2 + yt;
phi[x_, y_] = 1;
psi[x_, y_, t_] = Exp[-xyt];
a = 1/10;
T = 1;
n = 10;
m = 10;
h = 1/n;
t = T/m;
gamma = a^2 t / h^2;
```

Отдельно вычислим все узлы сеточной функции:

```
In[3]:= Table[{x_i = ih, y_i = ih}, {i, 0, n}];
Table[t_i = it, {i, 0, m}];
```

Получим разностное уравнение:

```
In[4]:= eq =
Flatten[Table[gamma u_{i-1,j,k} + gamma u_{i+1,j,k} + gamma u_{i,j-1,k} + gamma u_{i,j+1,k} -
(1 + 4gamma)u_{i,j,k} + u_{i,j,k-1} ==
- (tf[x_i, y_j, t_k] // N), {k, 1, m}, {j, 1, n - 1}, {i, 1, n - 1}]];
```

Начально-краевые условия запишем в виде уравнений:

```

In[5]:= eq1 = Flatten [Table [ $u_{i,j,0} = \phi[x_i, y_j]$ , { $i, 0, n$ }, { $j, 0, n$ }]];
eq2 = Flatten [Table [ $u_{0,i,k} = \psi[0, y_i, t_k]$ , { $i, 0, n$ }, { $k, 1, m$ }]];
eq3 = Flatten [Table [ $u_{n,i,k} = \psi[1, y_i, t_k]$ , { $i, 0, n$ }, { $k, 1, m$ }]];
eq4 = Flatten [Table [ $u_{i,0,k} = \psi[x_i, 0, t_k]$ , { $i, 1, n - 1$ }, { $k, 1, m$ }]];
eq5 = Flatten [Table [ $u_{i,n,k} = \psi[x_i, 1, t_k]$ , { $i, 1, n - 1$ }, { $k, 1, m$ }]];

```

Объединим все уравнения в одну систему.

```

In[6]:= eq = Join[eq, eq1, eq2, eq3, eq4, eq5];

```

Введем список переменных системы:

```

In[7]:= per = Flatten [Table [ $u_{i,j,k}$ , { $k, 0, m$ }, { $j, 0, n$ }, { $i, 0, n$ }]];

```

Решаем систему встроенной функцией:

```

In[8]:= sw = Solve[eq, per][[1]];

```

Находим значения сеточной функции, то есть ответ поставленной задачи:

```

In[9]:= Do [ $u_{i,j,k} = u_{i,j,k}/sw$ , { $k, 0, m$ }, { $j, 0, n$ }, { $i, 0, n$ }];

```

Для наглядности введем список значений температур на k -м слое:

```

In[10]:= g[k_] := Table [ $u_{i,j,k}$ , { $j, 0, 10$ }, { $i, 0, 10$ }]

```

и построим приближенно графики функций $u = u(x, y, t)$ для значений времени $t = 0, 1, \dots, m$, идущего с шагом 3:

```

In[11]:= p = Table[ListPlot3D[g[k]], { $k, 0, m, 3$ }]

```

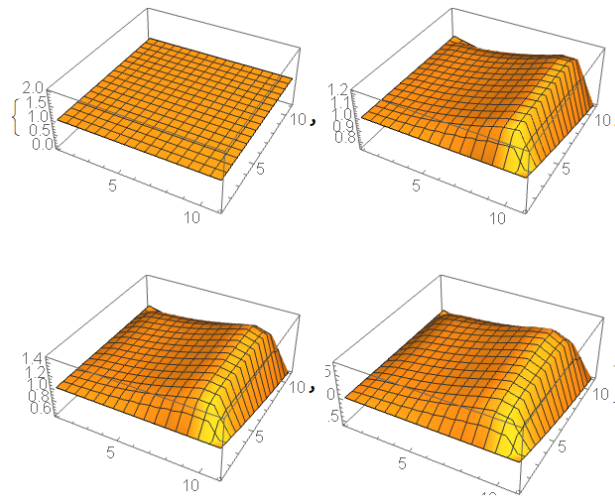


Рис. 1.5. Графики решения уравнения теплопроводности. Неявная разностная схема

К недостаткам данного подхода надо отнести большой объем вычислений при решении системы при малых шагах и большом интервале времени.

1.4 Моделирование брошенного тела под углом к горизонту

Решим задачу Коши, описывающую движение тела, брошенного с начальной скоростью v_0 под углом α к горизонтальной плоскости, рассматривая его как материальную точку с массой m и считая, что сопротивление воздуха пропорционально квадрату скорости [13]. Полагаем, что дальность полета и высота траектории малы по сравнению с радиусом Земли. На тело действует только сила тяжести \mathbf{P} , проекции которой на координатные оси равны:

$$P_x = 0, \quad P_y = -P = -mg, \quad P_z = 0.$$

Подставляя эти величины в дифференциальные уравнения движения, а именно

$$m\ddot{x} = \sum F_{kx}, \quad m\ddot{y} = \sum F_{ky}, \quad m\ddot{z} = \sum F_{kz}, \quad (1.15)$$

F_1, \dots, F_k — действующие на материальную точку силы, получаем:

$$\dot{v}_x = 0, \quad \dot{v}_y = -g, \quad \dot{v}_z = 0.$$

Умножая обе части этих уравнения на dt и интегрируя, находим:

$$v_x = C_1, \quad v_y = -gt + C_2, \quad v_z = C_3.$$

Запишем начальные условия для данной задачи:

$$\text{при } t = 0 : \quad x = 0, \quad y = 0, \quad z = 0;$$

$$v_x = v_0 \cos \alpha, \quad v_y = v_0 \sin \alpha, \quad v_z = 0.$$

Используя заданные начальные условия, находим, что:

$$C_1 = v_0 \cos \alpha, \quad C_2 = v_0 \sin \alpha, \quad C_3 = 0.$$

Подставим полученные значения C_1, C_2, C_3 в найденные решения:

$$\dot{x} = v_0 \cos \alpha, \quad \dot{y} = v_0 \sin \alpha - gt, \quad \dot{z} = 0.$$

Осталось проинтегрировать данные уравнения. Получим:

$$x = v_0 t \cos \alpha + C_4, \quad y = v_0 t \sin \alpha - \frac{gt^2}{2} + C_5, \quad z = C_6.$$

Начальные условия дают:

$$C_4 = C_5 = C_6 = 0.$$

Таким образом, окончательные уравнения движения материальной точки имеют вид:

$$x = v_0 t \cos \alpha, \quad y = v_0 t \sin \alpha - \frac{gt^2}{2}, \quad z = 0.$$

Из последнего уравнения можно сделать вывод, что движение происходит в плоскости Oxy .

Для моделирования траектории движения материального тела, брошенного под углом к горизонту, составим разностное уравнение. Из дифференциальных уравнений движения (1.15) получаем систему:

$$\begin{cases} m\ddot{x} = 0, \\ m\ddot{y} = -mg, \end{cases} \Rightarrow \begin{cases} \ddot{x} = 0, \\ \ddot{y} = -g. \end{cases} \quad (1.16)$$

Заменяем вторые производные по переменной t разностными аналогами:

$$x_i''(t) = \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2},$$

$$y_i''(t) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Подставляя разностные производные в систему (1.16), получим систему, состоящую из двух разностных уравнений:

$$\begin{cases} \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} = 0, \\ \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = -g, \end{cases} \Rightarrow \begin{cases} x_{i+1} = 2x_i - x_{i-1}, \\ y_{i+1} = -gh^2 + 2y_i - y_{i-1}. \end{cases}$$

Далее, используя полученные разностные уравнения движения материальной точки, решим задачу в «Wolfram Mathematica» при начальной скорости $v_0 = 0.2$, угле между вектором начальной скорости и горизонтом $\alpha = \pi/6$, шагом $h = 0.01$, начальными условиями:

$$x(0) = 0, \quad y(0) = 0,$$

$$\dot{x}(0) = v_0 \cos \alpha, \quad \dot{y}(0) = v_0 \sin \alpha.$$

Очистим все переменные и правила:

```
In[1]:= ClearAll["Global*"];
```

Введем все исходные данные задачи:

```
In[2]:= v0 = 0.2; g = 9.8; α = π/6; h = 0.01; x0 = 0; y0 = 0;
```

```
x1 = v0Cos[α]; y1 = v0Sin[α];
```

Значения, вычисленные по построенным разностным уравнениям, будут записываться в таблицу *lst*, в начало которой добавлены координаты исходного положения точки:

```
In[3]:= lst = Table[{x_{i+1} = 2x_i - x_{i-1}, y_{i+1} = -gh^2 + 2y_i - y_{i-1}},
{ i, 1, 1000 };
lst = Prepend[lst, {x0, y0}];
```

Выведем полученную таблицу *lst*. Здесь «headings» — заголовки столбцов:

```
In[4]:= headings = {{}, {x, y}};
```

```
TableForm[lst, TableHeadings → headings]
```

Out[4]:=

x	y
0	0
0.34641	0.19902
0.519615	0.29706
0.69282	0.39412
0.866025	0.4902
...	...

По найденным точкам из таблицы *lst* построим график траектории падения тела, брошенного под углом $\pi/6$ к горизонту:

```
In[5]:= p = ListPlot[lst, PlotRange → {{0, 40}, {0, 6}}];
```

```
Show[p]
```

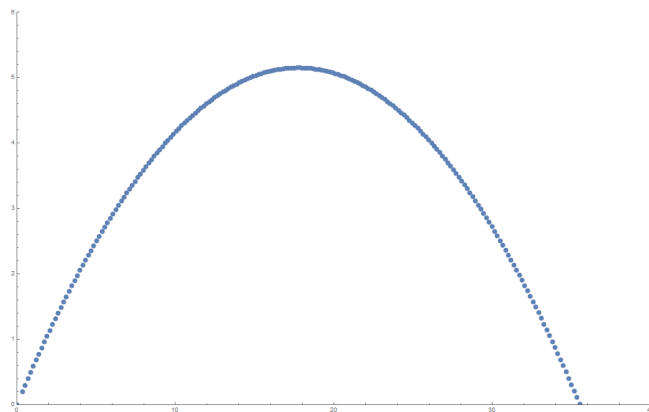


Рис. 1.6. Траектория падения тела, брошенного под углом к горизонту

Далее рассмотрим задачу о влиянии коэффициента сопротивления среды на траекторию падения тела, брошенного под углом к горизонту. Построим графики для параметра $b = 0$ (предыдущая задача), $b = 0.4$ и $b = 3$:

```
In[6]:= v0 = 0.2; g = 9.8; α = π/6; h = 0.01; m = 3;
```

```
x0 = 0; y0 = 0; x1 = v0 Cos[α]; y1 = v0 Sin[α];
```

```
lst[b_]:=Table [ { x_{i+1} = -\frac{b}{m} h (x_i - x_{i-1}) + 2x_i - x_{i-1},
```

```


$$y_{i+1} = -\frac{b}{m}h(y_i - y_{i-1}) - gh^2 - y_{i-1} + 2y_i \Big\} , \{i, 300\} \Big];$$

lst1 = lst[0]; lst1 = Prepend[lst1, {x0, y0}];
lst2 = lst[0.4]; lst2 = Prepend[lst2, {x0, y0}];
lst3 = lst[3]; lst3 = Prepend[lst3, {x0, y0}];
p1 = ListPlot[lst1, PlotRange → {{0, 40}, {0, 6}}];
p2 = ListPlot[lst2, PlotRange → {{0, 40}, {0, 6}}];
p3 = ListPlot[lst3, PlotRange → {{0, 40}, {0, 6}}];
Show[{p1, p2, p3}]

```

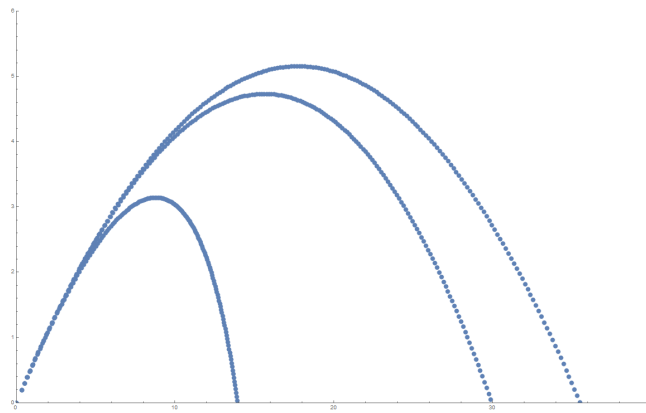


Рис. 1.7. Траектория тела, брошенного под углом к горизонту при различных значениях параметра, b — сопротивление среды

Глава 2

Численные методы

2.1 Основные определения и понятия

Погрешность

Пусть x — точное значение некоторого числа, а x^* — приближенное.

Абсолютная погрешность приближенного числа равна модулю разности между его точным и приближенным значениями:

$$\Delta x = |x - x^*|.$$

Относительная погрешность — это отношение абсолютной погрешности к приближенному значению числа:

$$\delta x = \frac{\Delta x}{|x^*|}.$$

Норма матрицы. Норма вектора

Напомним следующие стандартные обозначения: \mathbb{R} — поле вещественных чисел, \mathbb{C} — комплексных чисел. \mathbb{R}^n — вещественное n -мерное пространство, \mathbb{C}^n — комплексное n -мерное пространство.

Каждому элементу $x \in \mathbb{R}^n$ ($x \in \mathbb{C}^n$) можно поставить в соответствие вещественное число $\|x\|$, которое называется нормой, если:

1. $\|x\| \geq 0$, $\|x\| = 0 \Leftrightarrow x = 0$;
2. $\|\lambda x\| = |\lambda| \cdot \|x\|$, $\lambda \in \mathbb{R}$ ($\lambda \in \mathbb{C}$);
3. $\|x + y\| \leq \|x\| + \|y\|$.

Для вектора $x = (x_1, \dots, x_n)$, $x_j \in \mathbb{R}$ ($x_j \in \mathbb{C}$), $j = 1, \dots, n$, как правило, вводят следующие нормы:

1. $\|x\| = \left(\sum_{j=1}^n |x_j|^2 \right)^{1/2}$ — евклидова норма.
2. $\|x\|_1 = \max_{1 \leq j \leq n} |x_j|$;

$$3. \|x\|_2 = \sum_{j=1}^n |x_j|.$$

2.2 Численные методы решения линейных алгебраических систем

2.2.1 Метод Гаусса

Рассмотрим систему линейных алгебраических уравнений (СЛАУ), состоящую из n уравнений относительно n неизвестных:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n, \end{cases} \quad (2.1)$$

где x_1, x_2, \dots, x_n — неизвестные переменные, a_{ij} , $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$ — действительные числа, b_1, b_2, \dots, b_n — свободные члены.

Если $b_1 = b_2 = \dots = b_n = 0$, то СЛАУ называется *однородной*, в противном случае — *неоднородной*.

Совокупность значения неизвестных переменных $x_1 = \alpha_1$, $x_2 = \alpha_2$, \dots , $x_n = \alpha_n$, при которых все уравнения системы обращаются в тождества, называется *решением СЛАУ*.

Если существует хотя бы одно решение системы алгебраических уравнений, то она называется *совместной*, в противном случае — *несовместной*.

Если СЛАУ имеет единственное решение, то она называется *определенной*. Если решений больше одного, то система называется *неопределенной*.

Запишем систему (2.1) в матричной форме:

$$Ax = b, \quad (2.2)$$

где

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad x = (x_1, x_2, \dots, x_n)^T, \quad b = (b_1, b_2, \dots, b_n)^T.$$

Здесь A — квадратная матрица порядка n системы линейных алгебраических уравнений, x — вектор-столбец неизвестных переменных, а b — вектор-столбец свободных коэффициентов СЛАУ.

Численные методы решения СЛАУ можно разделить на два типа: *прямые (точные)* методы и *итерационные (приближенные)*. Прямые методы позволяют найти решение за конечное число вычислений. Если при использовании прямых методов не округлять полученные значения, то решение поставленной задачи будет точным. Для приближенного нахождения решения используются итерационные методы. В результате их применения

приближенное решение находится в результате построения достаточного числа последовательности, сходящейся к точному решению.

Если к матрице A добавить в качестве $n + 1$ -го столбца вектор-столбец свободных коэффициентов, то получим так называемую *расширенную матрицу* системы линейных уравнений:

$$T = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_n & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array} \right).$$

Квадратная матрица A называется *вырожденной*, если ее определитель равен нулю. Если $|A| \neq 0$, то матрица A называется *невырожденной*.

Далее перейдем непосредственно к методу Гаусса. Данный метод состоит из:

- прямого хода метода Гаусса;
- обратного хода метода Гаусса.

Прямой ход

Пусть определитель матрицы A отличен от нуля и $a_{11} \neq 0$. В противном случае перенумеруем переменные так, чтобы a_{11} не равнялось нулю. Исключим неизвестную переменную x_1 из всех уравнений системы, начиная со второго. Для этого умножим первое уравнение на

$$\frac{a_{21}}{a_{11}}$$

и вычтем из второго уравнения. Продолжим процесс, умножая первое уравнение на

$$\frac{a_{31}}{a_{11}}$$

и вычитая из третьего уравнения, и так далее. В заключение из n -го уравнения вычтем первое, умноженное на $\frac{a_{n1}}{a_{11}}$.

В результате получим систему, в которой, кроме первого уравнения, отсутствует переменная x_1 :

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + \dots + a_{3n}^{(1)}x_n = b_3^{(1)}, \\ \dots \\ a_{n2}^{(1)}x_2 + a_{n3}^{(1)}x_3 + \dots + a_{nn}^{(1)}x_n = b_n^{(1)}, \end{array} \right.$$

где

$$a_{ij}^{(1)} = a_{ij} - a_{1j} \cdot \left(\frac{a_{i1}}{a_{11}} \right), \quad b_i^{(1)} = b_i - b_1 \cdot \left(\frac{a_{i1}}{a_{11}} \right), \quad i, j = 2, 3, \dots, n.$$

Далее исключаем неизвестную переменную x_2 из всех уравнений, начиная с третьего. Для этого из третьего уравнения вычтем второе, умноженное на $\frac{a_{32}^{(1)}}{a_{22}^{(1)}}$, а из четвертого

уравнения вычтем второе, умноженное на $\frac{a_{42}^{(1)}}{a_{22}^{(1)}}$, и так далее. Из n -го уравнения вычтем второе, умноженное на $\frac{a_{n2}^{(1)}}{a_{22}^{(1)}}$. Система уравнений после таких преобразований примет вид:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(2)}x_n = b_2^{(2)}, \\ a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n = b_3^{(2)}, \\ \dots \\ a_{n3}^{(2)}x_3 + \dots + a_{nn}^{(2)}x_n = b_n^{(2)}, \end{cases}$$

где

$$a_{ij}^{(2)} = a_{ij}^{(1)} - a_{2j}^{(1)} \cdot \left(\frac{a_{i2}^{(1)}}{a_{22}^{(1)}} \right), \quad b_i^{(2)} = b_i^{(1)} - b_2^{(1)} \cdot \left(\frac{a_{i2}^{(1)}}{a_{22}^{(1)}} \right), \quad i, j = 3, 4, \dots, n.$$

Таким образом, переменная x_2 исключена из всех уравнений, начиная с третьего.

Далее приступаем к исключению неизвестной x_3 и т. д. Так продолжаем прямой ход Гаусса, пока система не примет треугольный вид:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1, \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)}, \\ a_{33}^{(2)}x_3 + \dots + a_{3n}^{(2)}x_n = b_3^{(2)}, \\ \dots \\ a_{nn}^{(n-1)}x_n = b_n^{(n-1)}. \end{cases}$$

С этого момента начинаем обратный ход метода Гаусса.

Обратный ход метода Гаусса

Вычисляем x_n из предпоследнего уравнения как

$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}.$$

С помощью полученного значения x_n находим x_{n-1} из предпоследнего уравнения, а x_1 — из первого уравнения. Определяя из последнего уравнения x_n , затем из предпоследнего x_{n-1} и так далее, обратным ходом определим все x_i . Процесс остановится, если на k -м шаге не удастся найти ненулевой элемент в k -й строке даже в результате перенумерации переменных от x_k до x_n . Это будет означать, что все коэффициенты этой строки равны нулю. Если b_k не равно нулю, то система не имеет решения, если же и $b_k = 0$, то решение не единственно.

Невязка решения

Неизбежные округления при использовании метода Гаусса дают определенную погрешность, величину которой характеризует *вектор невязки*, который вычисляется по следу-

ющей формуле:

$$\xi = b - Ax,$$

где A — матрица исходной системы, x — найденный вектор решения. Любая норма вектора невязки называется *невязкой* решения.

Разберем алгоритм на примере.

Задание

Найти решение системы алгебраических уравнений

$$\begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ x_1 + 2x_2 + 3x_3 - x_4 = 3, \\ 2x_1 - 3x_2 + 2x_3 - 2x_4 = 2, \\ 3x_1 + 3x_2 - 2x_3 + 4x_4 = 5 \end{cases}$$

методом Гаусса.

Решение

Коэффициент a_{11} отличен от нуля, так что приступим к прямому ходу метода Гаусса, то есть к исключению неизвестной переменной x_1 из всех уравнений системы, кроме первого. Для этого из левой и правой частей второго, третьего и четвертого уравнения вычтем левую и правую части первого уравнения, умноженные соответственно на

$$\frac{a_{21}}{a_{11}} = \frac{1}{2}, \quad \frac{a_{31}}{a_{11}} = 1, \quad \frac{a_{41}}{a_{11}} = \frac{3}{2} :$$

$$\begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ x_1 + 2x_2 + 3x_3 - x_4 - \frac{1}{2} \cdot (2x_1 - x_2 + x_3 + 2x_4) = 3 - \frac{1}{2} \cdot 5, \\ 2x_1 - 3x_2 + 2x_3 - 2x_4 - 2x_1 + x_2 - x_3 - 2x_4 = 2 - 5, \\ 3x_1 + 3x_2 - 2x_3 + 4x_4 - \frac{3}{2} \cdot (2x_1 - x_2 + x_3 + 2x_4) = 5 - \frac{3}{2} \cdot 5; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ \frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4 = \frac{1}{2}, \\ -2x_2 + x_3 - 4x_4 = -3, \\ \frac{9}{2}x_2 - \frac{7}{2}x_3 + x_4 = -\frac{5}{2}. \end{cases}$$

Неизвестную переменную x_1 исключили, переходим к исключению x_2 . Из левой и правой частей третьего и четвертого уравнений системы вычитаем левую и правую части второго уравнения, умноженные соответственно на

$$\frac{a_{32}^{(1)}}{a_{22}^{(1)}} = -\frac{2}{\frac{5}{2}} = -\frac{4}{5}, \quad \frac{a_{42}^{(1)}}{a_{22}^{(1)}} = \frac{\frac{9}{2}}{\frac{5}{2}} = \frac{9}{5} :$$

$$\begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ \frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4 = \frac{1}{2}, \\ -2x_2 + x_3 - 4x_4 - \left(-\frac{4}{5}\right) \cdot \left(\frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4\right) = -3 - \left(-\frac{4}{5}\right) \cdot \frac{1}{2}, \\ \frac{9}{2}x_2 - \frac{7}{2}x_3 + x_4 - \frac{9}{5} \cdot \left(\frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4\right) = -\frac{5}{2} - \frac{9}{5} \cdot \frac{1}{2}; \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ \frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4 = \frac{1}{2}, \\ 3x_3 - \frac{28}{5}x_4 = -\frac{13}{5}, \\ -8x_3 + \frac{23}{5}x_4 = -\frac{17}{5}. \end{cases}$$

Для завершения прямого хода Гаусса осталось исключить неизвестную переменную x_3 из последнего уравнения системы. Вычтем из левой и правой частей четвертого уравнения соответственно левую и правую части третьего уравнения, умноженные на:

$$\frac{a_{43}^{(2)}}{a_{33}^{(2)}} = -\frac{8}{3} :$$

$$\begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5 \\ \frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4 = \frac{1}{2} \\ 3x_3 - \frac{28}{5}x_4 = -\frac{13}{5} \\ -8x_3 + \frac{23}{5}x_4 - \left(-\frac{8}{3}\right) \cdot \left(3x_3 - \frac{28}{5}x_4\right) = -\frac{17}{5} - \left(-\frac{8}{3}\right) \cdot \left(-\frac{13}{5}\right) \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5, \\ \frac{5}{2}x_2 + \frac{5}{2}x_3 - 2x_4 = \frac{1}{2}, \\ 3x_3 - \frac{28}{5}x_4 = -\frac{13}{5}, \\ -\frac{31}{3}x_4 = -\frac{31}{3}. \end{cases}$$

Можно начинать обратный ход метода Гаусса.

Из последнего уравнения имеем:

$$x_4 = 1,$$

из третьего уравнения получаем :

$$x_3 = \frac{1}{3} \left(-\frac{13}{5} + \frac{28}{5}x_4 \right) = \frac{1}{3} \cdot \frac{15}{5} = 1,$$

из второго :

$$x_2 = \frac{2}{5} \left(\frac{1}{2} - \frac{5}{2}x_3 + 2x_4 \right) = \frac{2}{5} \left(\frac{1}{2} - \frac{5}{2} + 2 \right) = 0,$$

из первого уравнения:

$$x_1 = \frac{1}{2} (5 + x_2 - x_3 - 2x_4) = \frac{1}{2} (5 - 1 - 2) = 1.$$

Для проверки можно подставить полученные значения неизвестных переменных в исходную систему уравнений. Все уравнения обращаются в тождества, что говорит о том, что решение по методу Гаусса найдено верно.

Ответ: $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$.

Решим данное задание в «Wolfram Mathematica». Пусть дана расширенная матрица :

`In[1]:= a={{2,-1,1,2,5},{1,2,3,-1,3},{2,-3,2,-2,2},{3,3,-2,4,5}};`

системы алгебраических уравнений:

$$\begin{cases} 2x_1 - x_2 + x_3 + 2x_4 = 5 \\ x_1 + 2x_2 + 3x_3 - x_4 = 3 \\ 2x_1 - 3x_2 + 2x_3 - 2x_4 = 2 \\ 3x_1 + 3x_2 - 2x_3 + 4x_4 = 5. \end{cases}$$

Решим данную систему методом Гаусса.

Проверим правильность введения матрицы

`In[2]:= a//MatrixForm`

`Out[2]:=`
$$\begin{pmatrix} 2 & -1 & 1 & 2 & 5 \\ 1 & 2 & 3 & -1 & 3 \\ 2 & -3 & 2 & -2 & 2 \\ 3 & 3 & -2 & 4 & 5 \end{pmatrix}$$

Зададим количество уравнений в системе:

`In[3]:= n = 4;`

Обозначим через

`In[4]:= b=a[[All, n+1]]`

`Out[4]:= {5, 3, 2, 5}`

столбец свободных членов системы уравнений и введем матрицу системы, удаляя из матрицы a последний столбец:

`In[5]:= A=a[[1;;,1;n]]`

`Out[5]:=`
$$\begin{pmatrix} 2 & -1 & 1 & 2 \\ 1 & 2 & 3 & -1 \\ 2 & -3 & 2 & -2 \\ 3 & 3 & -2 & 4 \end{pmatrix}.$$

Перед каждым шагом исключения нужно проверять, не является ли элемент главной диагонали ведущей строки нулевым, чтобы не получить деления на ноль.

`In[6]:= For[i = 1, i ≤ n - 1, i++,`

`If[a[[i, i]] == 0,`

```

    a[[i]] = a[[i]] + a[[i + 1]]//N;];
If[a[[i, i]] ≠ 0,
    a[[i]] = a[[i]]/a[[i, i]]//N;
    For[j = i + 1, j ≤ n, j++,
        a[[j]] = a[[j]] - a[[i]]a[[j, i]]/a[[i, i]]//N;];];
If[a[[n, n]] ≠ 0,
    a[[n]] = a[[n]]/a[[n, n]]//N;
    For[i = n, i ≥ 1, i--,
        For[j = i - 1, j ≥ 1, j--,
            a[[j]] = a[[j]] - a[[i]] a[[j, i]]//N;];];
x = a[[All, n + 1]]
ξ = b - A.x

```

Out[7]:= Решение $x = \{1., 0., 1., 1.\}$ с вектором невязки $r = \{0., 0., 0., 0.\}$.

Так как невязка равна:

In[8]:= Norm[ξ]

Out[8]:= 0

то все команды, приводящие к решению, введены правильно и решение найдено верно.

Ответ: $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 1$.

Варианты заданий

$$\begin{array}{ll}
 1. \begin{cases} 2x_1 - x_2 - 2x_3 = -1, \\ 2x_2 + x_3 = 3, \\ -2x_1 - 2x_2 + 2x_3 = -2; \end{cases} & 2. \begin{cases} 2x_1 + x_2 - 3x_3 = 7, \\ 2x_1 + 3x_2 + x_3 = 1, \\ 3x_1 + 2x_2 + x_3 = 6; \end{cases} \\
 3. \begin{cases} 3x_1 + x_2 - 3x_3 = 8, \\ 3x_2 - x_3 = 7, \\ 3x_1 - x_2 - 3x_3 = 4; \end{cases} & 4. \begin{cases} 4x_1 + x_2 + 4x_3 = -3, \\ x_1 + x_2 + 2x_3 = -4, \\ 2x_1 - x_2 + 2x_3 = 3; \end{cases}
 \end{array}$$

2.2.2 Метод простой итерации

Метод простой итерации применяется тогда, когда размерность системы велика. В этом случае применение точных методов, например метода Гаусса, приводит к большому объему вычислений и, как следствие, накоплению вычислительной погрешности,

что зачастую существенно искажает результат. Итерационные методы более устойчивы к вычислительной погрешности, однако в отличие от точных методов не являются универсальными. Обычно для того, чтобы запустить итерационный процесс, требуется преобразовать систему так, чтобы итерационный процесс сходил. Одним из таких методов является метод итерации.

Если для получения приближения решения системы алгебраических уравнений (2.2) на очередной итерации используется лишь предыдущее значение x , то такой итерационный метод называется *одношаговым*.

Приведем систему (2.2) некоторыми равносильными преобразованиями к следующему виду:

$$x = Bx + G. \quad (2.3)$$

Это можно сделать, записав систему (2.2) в виде

$$x = x - Ax + b$$

и обозначив через B матрицу $E - A$ или, предварительно умножив (2.2) на некую неособенную матрицу H , проделать то же самое. Можно предложить и другие способы перехода от системы (2.2) к системе (2.3).

Построим последовательность векторов $\{x^{(k)}\}_{k=0,1,2,\dots}$, начиная с некоторого начального приближения $x^{(0)}$, по формуле:

$$x^{(k+1)} = Bx^{(k)} + G. \quad (2.4)$$

Если к системе (2.3) мы перешли с помощью некой матрицы H и эта матрица не зависит от номера итераций, то итерационный процесс называют *стационарным*.

Выбор матрицы H может быть осуществлен с использованием особенностей данной системы. Если матрица A положительно определена, то система $Ax = b$ всегда может быть приведена к виду, в котором метод последовательных приближений будет сходиться. В самом деле, вычислим, например, первую норму μ матрицы A . Мы получим, что собственные значения матрицы A лежат в интервале $(0, \mu)$. Положим

$$H = \frac{2}{\mu}E,$$

где E — единичная матрица. Умножим систему (2.2) на матрицу H и преобразуем к виду:

$$x = x - \frac{2}{\mu}Ax + \frac{2}{\mu}Ax + \frac{2}{\mu}b = Bx + G.$$

Здесь

$$B = E - \frac{2}{\mu}A,$$

а

$$G = \frac{2}{\mu}b.$$

Необходимые и достаточные признаки сходимости итерационной последовательности

1. Последовательность $\{x^{(k)}\}$ сходится к решению ξ системы уравнений (2.2), если норма матрицы B системы (2.4) удовлетворяет условию

$$\|B\| \leq q < 1 \quad (2.5)$$

при некотором q . При этом погрешность приближенного решения $x^{(k)}$ допускает следующую оценку:

$$\|\xi - x^{(k)}\| < \frac{q}{1-q} \|x^{(k+1)} - x^{(k)}\|.$$

2. Последовательность $\{x^{(k)}\}$ сходится к решению ξ системы уравнений (2.2), если:

- собственные значения матрицы B по модулю меньше единицы;
- матрица A системы (2.2) есть матрица с диагональным преобладанием:

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}|, \quad i = 1, \dots, n,$$

$$\sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| < |a_{jj}|, \quad j = 1, \dots, n$$

$$\text{или} \quad 2|a_{ii}| > \sum_{j=1}^n |a_{ij}|, \quad i = 1, \dots, n.$$

Построение члена такой последовательности, являющегося приближенным значением вектора решения данной системы с заданной точностью $\varepsilon > 0$, составляет суть метода итераций.

Признаком окончания итерационного процесса является условие

$$\frac{\|x^{(k+1)} - x^{(k)}\|_1}{\|x^{(k)}\|_1} < \varepsilon.$$

Кроме решения системы, следует найти вектор невязки

$$R = Ax^{(k+1)} - b.$$

Задание

Решить систему алгебраических уравнений, используя метод простой итерации:

$$\begin{cases} 10x_1 + x_2 - x_3 = 11 \\ x_1 + 10x_2 - x_3 = 10 \\ -x_1 + x_2 + 10x_3 = 10 \end{cases}$$

с заданной точностью $\varepsilon = 0,001$.

Для начала приведем систему алгебраических уравнений к удобному виду для итерации:

$$\begin{cases} x_1 = -0.1x_2 + 0.1x_3 + 1.1 \\ x_2 = -0.1x_1 + 0.1x_3 + 1 \\ x_3 = 0.1x_1 - 0.1x_2 + 1. \end{cases}$$

Выбираем начальное приближение, например:

$$x^{(0)} = \begin{pmatrix} 1.1 \\ 1 \\ 1 \end{pmatrix}$$

— вектор правой части.

Для начала введем команду, проверяющую, что собственные значения матрицы B по модулю меньше единицы.

```
In[1]:= us11[m_]:=Block[{n = Length[m]}, If[Abs[Eigenvalues[m]] < 1, False,  
Print["Условие не выполняется"]]
```

Зададим необходимую точность приближенного решения.

```
In[2]:=  $\epsilon = 0.0001$ ;
```

Введем матрицу B

```
In[3]:=  $B = \{\{0, -0.1, 0.1\}, \{-0.1, 0, 0.1\}, \{0.1, -0.1, 0\}\}$ ;
```

и столбец свободных коэффициентов:

```
In[4]:=  $G = \{1.1, 1, 1\}$ ;
```

Проверим условия сходимости итерационного процесса для заданной матрицы B :

```
In[5]:= us11[B]
```

```
Out[5]:= Собственные значения меньше 1
```

```
In[6]:= Norm[B] < 1
```

```
Out[6]:= True
```

Так как собственные значения матрицы B по модулю меньше единицы, а также норма матрицы B меньше единицы, то итерационный процесс сходится.

Далее сделаем необходимые преобразования, сделав замену $A = E - B$, $b = G$, где матрица E — единичная матрица.

```
In[7]:= A = IdentityMatrix[3] - B;
```

```
b = G;
```

Затем запускаем итерационный процесс по формулам (2.4), а также выводим вектор невязки и невязку:

```
In[8]:= xn+1 = G;
```

```
Do [xn = xn+1; xn+1 = B.xn + G;
```

```
If [Norm[xn+1 - xn, 1] / Norm[xn, 1] < ε, Print["Решение x = ",  
xn+1 // MatrixForm,
```

```
" получено на ", i, " шаге"]; Break[], {i, 1, 100}]
```

```
R = A.xn+1 - G
```

```
Norm[R]
```

```
Out[8]:= Решение x =  $\begin{pmatrix} 1.102 \\ 0.9909 \\ 1.0111 \end{pmatrix}$  получено на 3 шаге
```

```
{-0.00002, -0.00001, -0.00001}
```

```
0.000064433
```

Варианты заданий

1. $\begin{cases} x_1 = -0.3x_2 + 0.4x_3 - 1.9, \\ x_2 = -0.1x_1 - 0.1x_3 + 0.5, \\ x_3 = 0.5x_1 - 0.2x_2 + 2.2. \end{cases}$
2. $\begin{cases} x_1 = 0.2x_2 - 0.1x_3 + 1.2, \\ x_2 = 0.2x_1 - 0.4x_3 - 1.8, \\ x_3 = -0.2x_1 - 0.6x_2 - 0.1. \end{cases}$
3. $\begin{cases} x_1 = 0.1x_2 - 0.8x_3 + 1, \\ x_2 = 0.5x_1 + 0.3x_3 + 10, \\ x_3 = 0.7x_1 - 0.25x_2 + 15. \end{cases}$
4. $\begin{cases} x_1 = 0.75 + 0.25x_2 - 0.25x_3, \\ x_2 = 0.4 - 0.2x_1 + 0.2x_3, \\ x_3 = -1.2 + 0.2x_1 - 0.2x_2. \end{cases}$
5. $\begin{cases} x_1 = 0.1x_2 - 0.1x_4, \\ x_2 = -2.3 - 0.1x_3 + 0.1x_4, \\ x_3 = 1.9 - 0.1x_1 + 0.1x_2, \\ x_4 = 1.8 - 0.1x_2. \end{cases}$
6. $\begin{cases} x_1 = -0.1x_2 - 0.5x_3 + 0.6, \\ x_2 = -0.1x_1 - 0.2x_2 - 0.6, \\ x_3 = -0.1x_1 + 0.1x_2 + 4.2. \end{cases}$

$$7. \begin{cases} x_1 = -0.3x_2 + 0.1x_3 - 0.2x_4 - 26.8, \\ x_2 = -0.1x_1 + 0.1x_3 - 0.1x_4 - 17.3, \\ x_3 = -0.2x_1 + 0.3x_2 - 0.1x_4 + 72.9, \\ x_4 = -0.1x_1 + 0.3x_2 + 0.1x_3 + 50.1. \end{cases} \quad 8. \begin{cases} x_1 = 1.2 - 0.1x_2 - 0.1x_3, \\ x_2 = 1.3 - 0.2x_1 - 0.1x_3, \\ x_3 = 1.4 - 0.2x_1 - 0.2x_2. \end{cases}$$

2.2.3 Метод Зейделя

Метод Зейделя характеризуется тем, что при вычислении $(k+1)$ -е приближения к решению системы вектор $x^{(k+1)}$ определяется по следующему алгоритму: j -я компонента вектора $x^{(k+1)}$ определяется из j -го уравнения системы, в котором используются компоненты вектора $x^{(k)} : x_{j+1}^{(k)}, \dots, x_n^{(k)}$, т. е. вектора, полученного на предыдущей итерации и компоненты $x_1^{(k+1)}, \dots, x_{j-1}^{(k+1)}$ вектора $x^{(k+1)}$, определенные из предыдущих $j-1$ уравнений. Вычисления членов последовательности $\{x^{(k)}\}$ прекращаются как только

$$\frac{\|x^{(k+1)} - x^{(k)}\|_1}{\|x^{(k)}\|_1} < \varepsilon,$$

а $x^{(k+1)}$ будет решением системы (2.2) с точностью ε .

Запишем систему (2.2)

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n = b_3 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases}$$

Зададим произвольный начальный вектор

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})^T$$

и запустим итерационный процесс по формулам:

$$\begin{cases} a_{11}x_1^{(k+1)} + a_{12}x_2^{(k)} + \dots + a_{1n}x_n^{(k)} = b_1 \\ a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} + \dots + a_{2n}x_n^{(k)} = b_2 \\ a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k+1)} + \dots + a_{3n}x_n^{(k)} = b_3 \\ \dots \\ a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \dots + a_{nn}x_n^{(k+1)} = b_n. \end{cases}$$

Этот алгоритм, записанный в виде

$$x_j^{(k+1)} = - \sum_{i=1}^{j-1} \frac{a_{ij}}{a_{jj}} x_i^{(k+1)} - \sum_{i=j+1}^n \frac{a_{ij}}{a_{jj}} x_i^{(k)} + b_j, \quad j = 1, 2, \dots, n,$$

называют иногда методом Некрасова.

Необходимые и достаточные условия сходимости метода Зейделя

1. Для того чтобы метод Зейделя сходился с любого начального приближения, необходимо и достаточно, чтобы корни уравнения

$$\det A = \begin{vmatrix} a_{11}\lambda & a_{12} & \cdots & a_{1n} \\ a_{21}\lambda & a_{22}\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}\lambda & a_{n2}\lambda & \cdots & a_{nn}\lambda \end{vmatrix} = 0 \quad (2.6)$$

по модулю были меньше 1.

2. Достаточные признаки сходимости:

— матрица A системы алгебраических уравнений с диагональным преобладанием:

$$\sum_{j=1, j \neq i}^n |a_{ij}| < |a_{jj}|,$$

$$\sum_{i=1, i \neq j}^n |a_{ij}| < |a_{jj}|,$$

или

$$2|a_{ii}| > \sum_{j=1}^n |a_{ij}|, \quad i = 1, \dots, n;$$

— матрица системы алгебраических уравнений симметричная и положительно определенная.

Замечание. Зачастую приведения матрицы к диагональному преобладанию удастся добиться простой перестановкой уравнений или перенумерацией переменных. Если домножить обе части системы алгебраических уравнений $Ax = b$ на транспонированную матрицу A^T , то получим равносильную систему, матрица которой симметрична и положительно определена (теорема Коши-Бине).

Задание

Решить систему алгебраических уравнений:

$$\begin{cases} 2.7x_1 + 3.3x_2 + 3.1x_3 = 2.7, \\ 3.5x_1 - 1.7x_2 + 2.8x_3 = 1.7, \\ 4.1x_1 + 5.8x_2 - 1.7x_3 = 0.8. \end{cases}$$

Зададим матрицу A системы и вектор свободных коэффициентов b :

`In[1]: A = {{2.7, 3.3, 3.1}, {3.5, -1.7, 2.8}, {4.1, 5.8, -1.7}};`

`b = {2.7, 1.7, 0.8};`

Для начала проверим выполнение необходимого и достаточного условия, а именно найдем корни уравнения (2.6). Для этого введем команду, проверяющую, что найденные корни по модулю меньше единицы.

```
In[2]:= provkorni[m_]:=Block[ {l = Solve[
  Det[{ {m[[1, 1]]λ, m[[1, 2]], m[[1, 3]]}, {m[[2, 1]]λ, m[[2, 2]]λ, m[[2, 3]]},
  {m[[3, 1]]λ, m[[3, 2]]λ, m[[3, 3]]λ} } ] == 0, λ},
  If[AllTrue[{Abs[l[[1, 1, 2]]], Abs[l[[2, 1, 2]]], Abs[l[[3, 1, 2]]]}, # < 1 &],
  Print["Корни уравнения по модулю меньше 1"],
  Print["Условие на корни не выполняется"]]]
```

Выполним проверку необходимого и достаточного условия:

```
In[3]:= provkorni[A]
```

```
Out[3]:= Условие на корни не выполняется
```

Так как условие сходимости не выполняется, а также видно, что матрица не является симметричной, то далее приведем матрицу A системы алгебраических уравнений к симметричному виду, домножив обе части системы на транспонированную матрицу:

```
In[4]:= b = Transpose[A].b;
A = Transpose[A].A;
```

После таких преобразований попробуем снова проверить необходимое и достаточное условие сходимости метода Зейделя:

```
In[5]:= provkorni[A]
```

```
Out[5]:= Корни уравнения по модулю меньше 1
```

Так как необходимое и достаточное условие сходимости метода Зейделя выполняется, перейдем к проверке достаточных условий. Для этого создадим блок, проверяющий матрицу на диагональное преобладание ее элементов, а также на симметричность.

```
In[6]:= provdiag[m_]:=
  Block[{n = Length[m]},
    If[MemberQ[Table[Abs[2m[[i, i]]] >
      Sum^n Abs[m[[i, j]]], {i, 1, n}], False],
    Print["Условие не выполняется"],
    Print["Матрица с преобладанием диагональных элементов"]]]
```

```

In[7]:= provsim[m_]:=
      Block[{n = Length[m]},
        If[SymmetricMatrixQ[m],
          Print["Матрица является симметричной"],
          Print["Матрица не является симметричной"]]]

```

И выполним проверку данных условий:

```

In[8]:= provdiag[A]

```

```

Out[8]:= Условие не выполняется

```

```

In[9]:= provsim[A]

```

```

Out[9]:= Матрица является симметричной

```

Введем команду, которая в дальнейшем будет проверять на симметричность матрицу A , используя встроенную функцию «SymmetricMatrixQ».

В случае если матрица окажется не симметричной, то левую и правую части исходной системы алгебраических уравнений домножим на транспонированную матрицу:

```

In[10]:= a = If[SymmetricMatrixQ[a], a, a.Transpose[a]];
        b = If[SymmetricMatrixQ[a], b, b.Transpose[a]];

```

Осталось проверить положительную определенность матрицы A . Это можно сделать двумя способами:

- посмотреть на угловые миноры матрицы A . Они все должны быть > 0 ;
- посчитать собственные значения матрицы A . Если они все будут > 0 , то матрица является положительно определенной.

Создадим блок, который будет проверять на положительность собственные значения матрицы A :

```

In[11]:= provpol[m_]:=
      Block[{n = Length[m]}, If[AllTrue[Eigenvalues[m], # > 0 &],
        Print["Собственные значения положительны"],
        Print["Условие не выполняется"]]]

```

И выполним проверку достаточного условия сходимости:

In[12]:= **provpol**[A]

Out[12]:= Собственные значения положительны

Так как необходимые и достаточные условия сходимости метода Зейделя при решении систем алгебраических уравнений выполняются, а именно корни уравнения (2.6) по модулю меньше единицы, матрица A исходной системы является симметричной и положительно определенной (угловые миноры > 0 , либо собственные значения матрицы все > 0), то запустим итерационный процесс.

Зададим необходимую точность приближенного решения:

In[13]:= $\epsilon = 0.0001$;

и начальное приближение:

In[14]:= $\mathbf{x1} = \{1, 0, 0\}$;

In[15]:= **Do**{ $\mathbf{x0} = \mathbf{x1}$, $k = \text{Length}[A]$ };

For[$j = 1, j \leq k, j++$,
 $\mathbf{x1}[[j]] = -\sum_{i=1}^{j-1} \frac{A[[j]][[i]]}{A[[j]][[j]]} \mathbf{x1}[[i]] - \sum_{i=j+1}^3 \frac{A[[j]][[i]]}{A[[j]][[j]]} \mathbf{x0}[[i]] +$
 $\frac{b[[j]]}{A[[j]][[j]]}$];

If [$\frac{\text{Norm}[\mathbf{x1} - \mathbf{x0}, 1]}{\text{Norm}[\mathbf{x0}, 1]} < \epsilon$, **Print**["Решение $\mathbf{x} = "$,
 $\mathbf{x1} // \text{MatrixForm}$, " получено на ", n , " шаге"];

Break[], { $n, 1, 100$ }

$R = A.\mathbf{x1} - b$;

Print["Вектор невязки $R = "$, $R // \text{MatrixForm}$]

Norm[R]; **Print**["Невязка = ", **Norm**[R]]

Out[15]:= Решение $x = \begin{pmatrix} 0.147856 \\ 0.191278 \\ 0.538531 \end{pmatrix}$ получено на 24 шаге

Вектор невязки $R = \begin{pmatrix} 0.000870389 \\ -0.000104569 \\ 0. \end{pmatrix}$

Невязка = 0.000876648

Ответ: $x_1 = 0.147856$, $x_2 = 0.191278$, $x_3 = 0.538531$.

Далее решим эту же систему, но предварительно приведя матрицу к диагональному преобладанию ее элементов. Для этого из первой строки вычтем третью, из второй строки вычтем третью, а к третьей прибавим вторую:

$$\left(\begin{array}{ccc|c} 2.7 & 3.3 & 3.1 & 2.7 \\ 3.5 & -1.7 & 2.8 & 1.7 \\ 4.1 & 5.8 & -1.7 & 0.8 \end{array} \right) \Rightarrow \left(\begin{array}{ccc|c} -1.4 & -2.5 & 4.8 & 1.9 \\ -0.6 & -7.5 & 4.5 & 0.9 \\ 7.6 & 4.1 & 1.1 & 2.5 \end{array} \right).$$

Затем поменяем местами первую и третью строки, перейдя таким образом к матрице с диагональным преобладанием:

$$\left(\begin{array}{ccc|c} 7.6 & 4.1 & 1.1 & 2.5 \\ -0.6 & -7.5 & 4.5 & 0.9 \\ -1.4 & -2.5 & 4.8 & 1.9 \end{array} \right).$$

Выполним все сделанные выше вычисления в «Wolfram Mathematica» для той же системы, но уже с матрицей, приведенной к матрице с диагональным преобладанием.

```
In[16]:= A = {{7.6, 4.1, 1.1}, {-0.6, -7.5, 4.5}, {-1.4, -2.5, 4.8}};
```

```
b = {2.5, 0.9, 1.9};
```

```
provkorni[A]
```

```
provdiag[A]
```

```
provsim[A]
```

Корни уравнения по модулю меньше 1

Матрица с преобладанием диагональных элементов

Матрица не является симметричной

```
ϵ = 0.0001;
```

```
x1 = {1, 0, 0};
```

```
Do[{x0 = x1, k = Length[A]};
```

```
For[j = 1, j ≤ k, j++,
```

```
  x1[[j]] = - Sum[ $\frac{A[[j]][[i]]}{A[[j]][[j]]}$  x1[[i]] - Sum[ $\frac{A[[j]][[i]]}{A[[j]][[j]]}$  x0[[i]] +  

 $\frac{b[[j]]}{A[[j]][[j]]}$ ], {i, 1, j-1}];
```

```
If [ $\frac{\text{Norm}[x1 - x0, 1]}{\text{Norm}[x0, 1]}$  < ϵ, Print["Решение x = ",
```

```
x1//MatrixForm, " получено на ",
```

```
n, " шаг"]];
```

```
Break[], {n, 1, 100}]
```

```
R = A.x1 - b;
```

```
Print["Вектор невязки R = ", R//MatrixForm]
```

```
Norm[R];
```

Print["Невязка = ", Norm[R]]

Решение $x = \begin{pmatrix} 0.147776 \\ 0.191337 \\ 0.538589 \end{pmatrix}$ получено на 10 шаге

Вектор невязки $R = \begin{pmatrix} 0.0000262011 \\ -0.0000387703 \\ 0. \end{pmatrix}$

Невязка = 0.0000467935

Ответ: $x_1 = 0.147776$, $x_2 = 0.191337$, $x_3 = 0.538589$.

Как видно из полученного ответа, после приведения матрицы к матрице с диагональным преобладанием, решение системы алгебраических уравнений получено не на 24 шаге, как это было найдено для симметричной матрицы без диагонального преобладания элементов, а на 10 шаге.

Варианты заданий

$$1. \begin{cases} 3.2x_1 - 11.5x_2 + 3.8x_3 = 2.8, \\ 0.8x_1 + 1.3x_2 - 6.4x_3 = -6.5, \\ 2.4x_1 + 7.2x_2 - 1.2x_3 = 4.5. \end{cases} \quad 2. \begin{cases} 4x_1 + x_2 + 9x_3 = -7, \\ 3x_1 + 8x_2 - 7x_3 = -6, \\ x_1 + x_2 - 8x_3 = 7. \end{cases}$$

$$3. \begin{cases} 3.6x_1 + 1.8x_2 - 4.7x_3 = 3.8, \\ 2.7x_1 - 3.6x_2 + 1.9x_3 = 0.4, \\ 1.5x_1 + 4.5x_2 + 3.3x_3 = -1.6. \end{cases} \quad 4. \begin{cases} 10x_1 - x_2 + x_4 = 0, \\ 12x_2 + x_3 - x_4 = -28, \\ x_1 - 2x_2 + 16x_3 = 31, \\ 2x_2 + 16x_4 = 29. \end{cases}$$

$$5. \begin{cases} 3.1x_2 + 2.8x_3 + 1.9x_4 = 0.2, \\ 1.9x_1 + 3.1x_2 + 2.1x_3 = 2.1, \\ 7.5x_1 + 3.8x_2 + 4.8x_3 = 5.6. \end{cases} \quad 6. \begin{cases} 10x_1 + x_2 + x_3 = 12, \\ 2x_1 + 10x_2 + x_3 = 13, \\ 2x_1 + 2x_2 + 10x_3 = 14. \end{cases}$$

$$7. \begin{cases} 10x_2 + 30x_3 + 40x_4 = -50, \\ 10x_1 + 20x_3 + 30x_4 = -40, \\ 30x_1 + 20x_2 - 50x_4 = 120, \\ 40x_1 + 30x_2 + 50x_3 = 50. \end{cases} \quad 8. \begin{cases} 0.3x_2 - 0.1x_3 + 0.2x_4 = -1, \\ -0.2x_1 - 0.21x_3 + 0.2x_4 = -4, \\ 0.3x_1 - 0.1x_2 + 0.3x_4 = 2, \\ 0.1x_1 - 0.1x_2 - 0.2x_3 = 0.1. \end{cases}$$

2.2.4 Метод скорейшего спуска

Данный метод применяется для решения систем вида (2.2) с положительно определенной симметричной матрицей.

Метод скорейшего спуска является итерационным методом и заключается в построении последовательности векторов $\{x^{(k)}\}$, $k = 0, 1, 2, \dots$, где x_0 — любое начальное приближе-

ние, сходящейся к вектору ξ — решению системы (2.2). Для построения такой последовательности рассматривается вспомогательная функция

$$F(x) = (Ax, x) - 2(b, x).$$

Задача нахождения решения системы (2.2) подменяется экстремальной задачей — задачей отыскания точки минимума функции $F(x)$. Эта функция является квадратичной функцией относительно координат вектора x .

Градиентный метод минимизации этого функционала приводит к итерационным формулам отыскания решения системы (2.2)

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \lambda_k R^{(k)}, \\ R^{(k)} &= b - Ax^{(k)}, \quad \lambda_k = \frac{(R^{(k)}, R^{(k)})}{(AR^{(k)}, R^{(k)})}. \end{aligned} \quad (2.7)$$

Вектор R — вектор невязки. Следует отметить, что покоординатная минимизация функционала $F(x)$ приводит к методу Зейделя.

Задание

Решить систему алгебраических уравнений:

$$\begin{cases} 2x_1 + x_2 = 3, \\ x_1 - 2x_2 = 1. \end{cases}$$

В качестве начального приближения выберем вектор

$$x^{(0)} = \begin{pmatrix} 1.5 \\ -0.5 \end{pmatrix}.$$

Введем матрицу системы и столбец свободных коэффициентов:

`In[1]: A = {{2, 1}, {1, -2}}; b = {3., 1};`

Зададим необходимую точность:

`In[2]: ε = 0.0001;`

Прежде чем перейти к нахождению приближенного решения, необходимо проверить, что матрица системы алгебраических уравнений является симметричной и положительно определенной. Напомним, что матрица является положительно определенной, если все ее собственные значения являются положительными либо если можно воспользоваться критерием Сильвестра. Для этого необходимо проверить на положительности все угловые миноры матрицы A .

Для начала проверим матрицу A на симметричность. В случае если матрица не является симметричной, то домножим обе части системы (2.2) на транспонированную матрицу A^T .

```
In[3]:= B = If[SymmetricMatrixQ[A], B, Transpose[A].B];
A = If[SymmetricMatrixQ[A], A, Transpose[A].A];
```

Затем выполним проверку, что матрица A является положительно определенной. Для этого можно воспользоваться одним из критериев:

```
In[4]:= B = If[A[[1]][[1]] > 0, B, Transpose[A].B];
A = If[A[[1]][[1]] > 0, A, Transpose[A].A];
B = If[Minors[A][[1]][[1]] > 0, B, Transpose[A].B];
A = If[Minors[A][[1]][[1]] > 0, A, Transpose[A].A];
B = If[Det[A] > 0, B, Transpose[A].B];
A = If[Det[A] > 0, A, Transpose[A].A];
```

После проверки всех условий сходимости запустим итерационный процесс. Сначала зададим начальный вектор, с которого и начнем нахождение приближенного решения заданной системы:

```
In[5]:= X = {0, 0};
R = B - A.X;
While[Total[Abs[R]] > ε,
  R = B - A.X; λ = (R.R)/(A.R.R)//N;
  X = X + λR;]
Print["Решение системы X = ", X//MatrixForm]
Print["Вектор невязки R = ", R//MatrixForm]
```

Решение системы $X = \begin{pmatrix} 1.4 \\ 0.2 \end{pmatrix}$

Вектор невязки $R = \begin{pmatrix} -8.881784197001252 \times 10^{-16} \\ 0. \end{pmatrix}$

Ответ: $x_1 = 1.4, x_2 = 0.2$.

Варианты заданий

$$\begin{array}{ll} 1. \begin{cases} 1.4x_1 + 0.4x_2 + 0.5x_3 + 0.6x_4 = 0.7, \\ 0.4x_1 + 1.4x_2 + 0.3x_3 + 0.4x_4 = 0.5, \\ 0.5x_1 + 0.3x_2 + 1.4x_3 + 0.2x_4 = 1.3, \\ 0.6x_1 + 0.4x_2 + 0.2x_3 + 1.4x_4 = 1.2. \end{cases} & 2. \begin{cases} x_1 + 2.1x_2 = 0, \\ 2.1x_1 + 2x_2 + 4x_3 = 4, \\ 4x_2 + 5x_3 = 5. \end{cases} \\ \\ 3. \begin{cases} x_1 + 0.42x_2 + 0.54x_3 + 0.66x_4 = 0.3, \\ 0.42x_1 + x_2 + 0.32x_3 + 0.44x_4 = 0.5, \\ 0.54x_1 + 0.32x_2 + x_3 + 0.22x_4 = 0.7, \\ 0.66x_1 + 0.44x_2 + 0.22x_3 + x_4 = 0.9. \end{cases} & 4. \begin{cases} 2x_2 + 2x_3 = 1.8, \\ 2x_1 - 2x_3 = 2.2, \\ 2x_1 - 2x_2 = 0.77. \end{cases} \\ \\ 5. \begin{cases} 0.5x_1 + 1.6x_2 + 0.6x_3 = 2.1, \\ 1.6x_1 + 0.6x_2 - 2x_3 = -0.8, \\ 1.6x_1 - 2x_2 + 0.8x_3 = 7.8. \end{cases} & 6. \begin{cases} 2x_1 + 4x_2 - 4x_3 = 2.1, \\ 4x_1 + 5x_2 - 8x_3 = 0.98, \\ -4x_1 - 8x_2 + 5x_3 = -7.2. \end{cases} \end{array}$$

2.3 Численное интегрирование

Рассмотрим задачу о вычислении значения функции

$$F(t) = \int_a^b f(t, x) dx \quad (2.8)$$

в точках $t_j = c + j\tau$, $\tau = \frac{d-c}{m}$ методом удвоения числа шагов для достижения заданной точности, используя стандартные квадратурные формулы.

Обозначив через $g_i(x)f(t_j, x)$, мы тем самым сводим задачу в цикле по j к вычислению интеграла от функции $g_j(x)$ по заданным квадратурным формулам.

Все квадратурные формулы имеют вид:

$$I(g) \equiv \int_a^b g(x) dx \approx \sum_{i=1}^n c_i g(x_i) \equiv S_n(g). \quad (2.9)$$

Метод удвоения числа шагов состоит в следующем. Пусть

$$\int_a^b g(u) du \approx \sum_{i=1}^n d_i g(u_i) = S_n.$$

Разобьем отрезок интегрирования точками u_1, \dots, u_N и составим сумму S_N . Удвоим N и составим сумму S_{2N} . Если величина

$$|S_{2N} - S_N| < \varepsilon,$$

то считается, что заданная точность достигнута. В противном случае процесс продолжается.

Обозначим через

$$h = \frac{b-a}{N}, \quad g_i = g(x_i), \quad g_{i+\frac{1}{2}} = g\left(x_i + \frac{h}{2}\right), \quad g_0 = g(a), \quad g_N = g(b).$$

Начальное разбиение N выбирается произвольно, но при использовании формул Симпсона начальное N четно.

Квадратурные формулы прямоугольников

$$I_R = h \cdot \sum_{i=0}^{n-1} f\left(a + \frac{h}{2}(2i+1)\right), \quad i = 0, \dots, n-1.$$

Задание

Вычислить значение функции

$$F(t) = \int_a^b \sin\left(\frac{t}{1+x^2} + 0.001x\right) dx, \quad 0 \leq x \leq 1,$$

с точностью $\varepsilon = 0.001$.

In[1]:= ClearAll["Global*"]

Зададим необходимую точность вычислений

In[2]:= $\epsilon = 0.001$;

Введем функцию, которая стоит под интегралом:

In[3]:= $f[t_ , x_] = \text{Sin}\left[\frac{t}{1+x^2} + 0.001x\right]$;

Зададим отрезки, в пределах которых изменяются значения переменных $x \in [a, b]$, $t \in [c, d]$:

In[4]:= $\{a, b\} = \{0.0, 1.0\}$; $\{c, d\} = \{0.0, 1.0\}$;

Шаг разбиения отрезка $[a, b]$: $h = (b-a)/n$ и отрезка $[c, d]$: $\tau = (d-c)/m$:

In[5]:= $m = 4$; $n = 200$;

$$\tau = \frac{d-c}{m}; \quad h = \frac{b-a}{n};$$

Запустим вычисления с использованием квадратурных формул прямоугольников. Процесс остановится, когда будет выполнено условие метода удвоения шагов:

$$|S_{2N} - S_N| < \varepsilon.$$

На каждом шаге значение определенного интеграла сравнивается со значением, которое получено при помощи встроенной функции «NIntegrate».

```
In[6]:= iR = 0;
For [j = 0, j <= m, j++, tj = c + jτ;
CellPrint["NIntegrate"];
CellPrint [NIntegrate [Sin [  $\frac{t_j}{1+x^2} + 0.001x$  ], {x, a, b}]]];
iR = hSum [f [tj, a +  $\frac{h}{2}(2k+1)$ ], {k, 0, n-1}]; CellPrint [iR];
If[Abs [hSum [f [tj, a +  $\frac{h}{2}(2k+1)$ ], {k, 0, 2n-1}] -
hSum [f [tj, a +  $\frac{h}{2}(2k+1)$ ], {k, 0, n-1}] < ε], Break]]];
```

```
Out[6]:= NIntegrate
0.0005
0.0005
NIntegrate
0.195427
0.195427
NIntegrate
0.381934
0.381934
...
```

Квадратурные формулы трапеций

$$I_T = h \left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(a + ih) \right), \quad i = 1, \dots, n-1.$$

Решим ту же самую задачу, но используя квадратурные формулы трапеций.

```
In[7]:= iT = 0;
For [j = 0, j <= m, j++, tj = c + jτ;
CellPrint["NIntegrate"];
```

```

CellPrint [NIntegrate [Sin [  $\frac{t_j}{1+x^2} + 0.001x$  ], {x, a, b}]]];
i_T = h (  $\frac{f[t_j, a] + f[t_j, b]}{2} + \text{Sum}[f[t_j, a + kh], \{k, 1, n-1\}]$  );
CellPrint [i_T];
If[Abs [  $h ( \frac{f[t_j, a] + f[t_j, b]}{2} + \text{Sum}[f[t_j, a + kh], \{k, 1, 2n-1\}]$  ) -
h (  $\frac{f[t_j, a] + f[t_j, b]}{2} + \text{Sum}[f[t_j, a + kh], \{k, 1, n-1\}]$  ) ] <  $\epsilon$  ],
Break]];

```

```

Out[7]:= NIntegrate
0.0005
0.0005
NIntegrate
0.195427
0.195427
NIntegrate
0.381934
0.381933
...

```

Квадратурные формулы Симпсона

$$I_S = \frac{2h}{3} \left(\frac{f(a) + f(b)}{2} + 2 \sum_{i=1}^m f(a + (2i-1)h) + \sum_{i=1}^{m-1} f(a + 2kh) \right), \quad N - \text{четно}.$$

```

In[8]:= {c, d} = {0.0, 1.0};
v = 4; m = n/2; n = 50;
 $\tau = \frac{d-c}{v}; h = \frac{b-a}{n};$ 
i_S = 0;
For[j = 0, j <= m, j++, t_j = c + j  $\tau$ ;
CellPrint["NIntegrate"];
CellPrint [NIntegrate [Sin [  $\frac{t_j}{1+x^2} + 0.001x$  ], {x, a, b}]]];
i_S =
 $\frac{2h}{3} ( \frac{f[t_j, a] + f[t_j, b]}{2} + 2\text{Sum}[f[t_j, a + (2k-1)h], \{k, 1, m\}] +$ 

```

```

Sum [f [tj, a + 2kh], {k, 1, m - 1}]]);
CellPrint [iS];
If[
  Abs[
     $\frac{2h}{3} \left( \frac{f[t_j, a] + f[t_j, b]}{2} + 2\text{Sum} [f [t_j, a + (2k - 1)h], \{k, 1, 2m\}] + \right.$ 
    Sum [f [tj, a + 2kh], {k, 1, 2m - 1}]] -
     $\left( \frac{2h}{3} \left( \frac{f[t_j, a] + f[t_j, b]}{2} + \right.$ 
    2Sum [f [tj, a + (2k - 1)h], {k, 1, m}] +
    Sum [f [tj, a + 2kh], {k, 1, m - 1}])) < ϵ], Break]]);

```

```

Out[8]:= NIntegrate
0.0005
0.0005
NIntegrate
0.195427
0.195427
NIntegrate
0.381934
0.381934

```

Варианты заданий

Вычислить значение функции

$$F(t) = \int_a^b f(t, x) dx$$

в точках $t_j = c + j\tau$, $\tau = \frac{d-c}{m}$ методом удвоения шагов для достижения заданной точности. Положить $\varepsilon = 0.001$.

1. $f(x, t) = \text{Exp} \left[\frac{\sqrt{t+1}}{1+x^2} \right] \sin 2x.$
2. $f(x, t) = \cos \left(\frac{t}{1+x^2} + 0.001x \right).$
3. $f(x, t) = \frac{\sin(t+x^2+0.5)}{2+\cos(x^2+1)}.$
4. $f(x, t) = \ln \left(\frac{1+t}{1+x^2} + 0.001x \right).$

В функции `NIntegrate` заложено много различных алгоритмов численного интегрирования. Обычно при ее использовании функция сама выбирает подходящий метод, но также присутствует возможность выбрать необходимый метод самостоятельно. Для этого используется свойство «`Method`».

Также у функции «`NIntegrate`» есть свойства, с помощью которых можно управлять точностью вычислений.

«`AccuracyGoal`» — абсолютная погрешность вычислений. «`PrecisionGoal`» — относительная погрешность вычислений.

2.4 Численное решение ОДУ

Пусть необходимо решить задачу Коши для дифференциального уравнения первого порядка:

$$y'(x) = f(x, y(x)), \quad y(x_0) \equiv y_0 \quad (2.10)$$

на отрезке $[a, b]$.

Разобьем интервал точками

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

расположенными с шагом

$$h = \frac{b - a}{n}.$$

Численным решением задачи Коши (2.10) является табличная функция $y = (x_i, y_i)$, для которой известна оценка погрешности $[y_i - y(x_i)]$, где $y(x)$ — аналитическое решение задачи (2.10). Оценка погрешности позволяет судить, насколько найденное численное решение удовлетворяет требованиям практической задачи.

2.4.1 Метод Эйлера

Применяя ряд Тейлора функции $y(x)$ в точке x_i , можно записать

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + y'(x_i)h + ch^2,$$

где ch^2 — остаточный член в форме Лагранжа. Перепишем это равенство с учетом (2.10):

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) + ch^2. \quad (2.11)$$

При $i = 0$ из (2.11) и начального условия (2.10) получим

$$y(x_1) = y(x_0) + hf(x_0, y(x_0)) + ch^2 = y_0 + hf(x_0, y_0) + ch^2. \quad (2.12)$$

Обозначим через

$$y_1 = y_0 + hf(x_0, y_0) \quad (2.13)$$

и объявим y_1 вторым значением табличной функции — численным решением задачи Коши, первое значение — y_0 . Из (2.12) и (2.13) получаем погрешность приближенного значения y_1 :

$$y(x_1) - y_1 = ch^2. \quad (2.14)$$

При $i = 1$ из (2.11) и (2.14) получим

$$y(x_2) = y(x_1) + hf(x_1, y(x_1)) + c_1 h^2 = y_1 + ch^2 + hf(x_1, y_1 + ch^2) + c_1(h^2).$$

Применяя формулу Тейлора, запишем:

$$y(x_2) = y_1 + ch^2 + h[f(x_1, y_1) + f'_y(x_1, y_1)ch^2 + \tilde{c}h^4] + c_1 h^2$$

или, пренебрегая достаточно малыми слагаемыми с h^3 и h^5 , получим

$$y(x_2) = y_1 + hf(x_1, y_1) + ch^2 + c_1 h^2 = y_1 + hf(x_1, y_1) + 2c_2 h^2, \quad (2.15)$$

где $c_2 = \frac{c + c_1}{2}$. Обозначим через

$$y_2 = y_1 + hf(x_1, y_1) \quad (2.16)$$

и объявим y_2 третьим значением табличной функции — численным решением задачи Коши. Из (2.15) и (2.16) получаем погрешность приближенного значения y_2 :

$$y(x_2) - y_2 = 2c_2 h^2. \quad (2.17)$$

И так далее, другими словами, все численные решения задачи Коши будем вычислять по формуле:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad (2.18)$$

при $i = 0, 1, \dots, n-1$. При этом глобальная погрешность найденного численного решения

$$y(x_n) - y_n = nc_n h^2 = \frac{b-a}{h} c_n h^2 = (b-a)c_n h.$$

Таким образом, метод Эйлера имеет глобальную погрешность порядка h и поэтому практически не применяется. Можно увеличить число членов ряда Тейлора в рассуждениях выше, тем самым повысить точность метода. Но в этом случае возникнет необходимость вычислять производные функции $f(x, y)$. Следующий метод позволяет избежать таких трудностей и имеет достаточно высокую точность.

Отметим, что данный метод является k -шаговым, если при определении приближенного значения y_{i+1} требуются k предыдущих значений $y_i, y_{i-1}, \dots, y_{i-k+1}$. В этом смысле метод Эйлера является одношаговым методом, из формулы (2.18) видно, что для определения приближенного значения y_{i+1} требуется одно предыдущее значение.

Задание

Решить систему дифференциальных уравнений

$$\begin{cases} \dot{x} = xy - 4, \\ \dot{y} = (x - 4)(y - x). \end{cases}$$

методом Эйлера.

Зададим систему дифференциальных уравнений, отрезок, начальные условия и шаг:

```

In[1]:= f[t_, x_, y_] := xy - 4;
        g[t_, x_, y_] := (x - 4)(y - x);
        {a, b} = {0, 100};
        {t0, x0, y0} = {0.0, -2.1, -2.1};
        h = 0.02;

```

Будем записывать в *xylst* все значения координат x, y , вычисленных методом Эйлера. В самое начало этого списка добавим начальные координаты:

```

In[2]:= {t, x, y} = {t0, x0, y0};
        xylst = List[{x0, y0}];
        m = Floor[(b - a)/h];

```

Запустим метод Эйлера и по вычисленным значениям построим фазовую траекторию системы с заданными начальными условиями:

```

In[3]:= For[k = 1, k ≤ m, k++, {t, x, y} =
        {t + h, x + hf[t, x, y], y + hg[t, x, y]};
        xylst = Append[xylst, {x, y}]];
        lst1 = {AbsoluteThickness[3], RGBColor[0, 1, 0]};
        gr1 = ListPlot[xylst, Frame → True];
        Show[gr1, PlotLabel → "Фазовая траектория"]

```

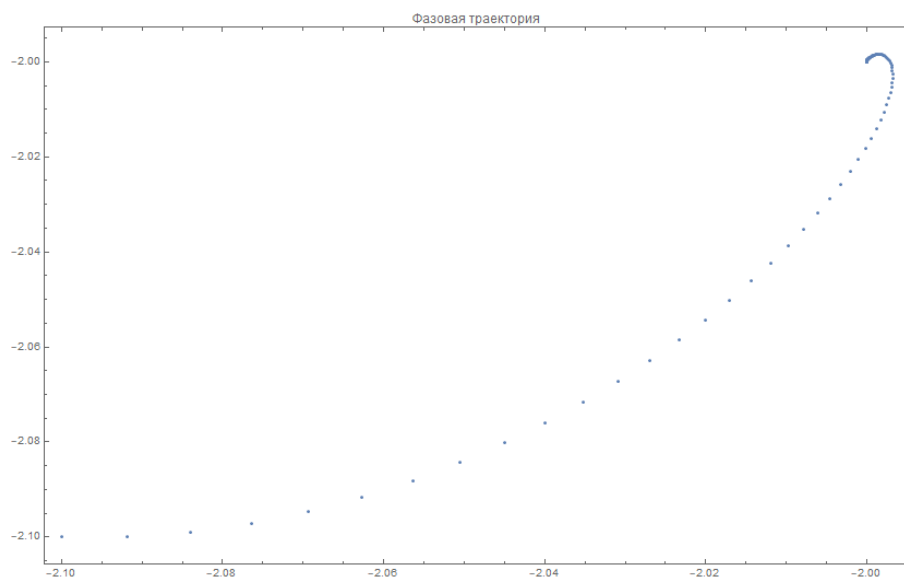


Рис. 2.1. Фазовая траектория системы, построенная с помощью метода Эйлера

2.4.2 Метод Рунге—Кутта

Пусть необходимо исследовать поведение решений системы дифференциальных уравнений в окрестности особых точек и сделать вывод о характере особой точки.

Пусть задана система

$$\begin{cases} \frac{dx}{dt} = P(x, y), \\ \frac{dy}{dt} = Q(x, y). \end{cases} \quad (2.19)$$

Напомним, что особой точкой (x_0, y_0) называется такая точка, в которой

$$P(x_0, y_0), \quad Q(x_0, y_0) = 0.$$

Необходимо найти все особые точки системы, численно исследовать поведение траекторий в окрестности особой точки и сделать вывод о том, является ли данная точка устойчивым положением равновесия и к какому типу (узел, седло, фокус, центр) относится эта особая точка. Результат численного исследования должен быть согласован с выводами, полученными аналитическим путем. Напомним, что аналитическое исследование проводится по следующей схеме.

1. Система линеаризуется в окрестности исследуемого положения равновесия.

Это означает, что в системе делается замена

$$u = x - x_0, \quad v = y - y_0$$

и в полученной системе отбрасываются все нелинейные члены.

2. В полученной системе вида

$$\begin{cases} \frac{du}{dt} = au + bv, \\ \frac{dv}{dt} = cu + dv \end{cases}$$

находятся собственные значения матрицы

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

3. Если собственные значения вещественные, различные и одного знака, то особая точка — узел, устойчивый или неустойчивый, в зависимости от того, отрицательны или положительны оба собственных значения. Если собственные значения вещественны и разных знаков, то особая точка — седло. Оно всегда неустойчиво. Если же собственные значения чисто мнимые, то особая точка будет центром, если система линейная. Для нелинейной системы эта точка может быть как центром, так и фокусом. В случае когда собственные значения равны, особая точка — дикретический узел.

Численное исследование поведения интегральных кривых в фазовой плоскости Oxy можно проводить по следующей схеме. Пусть найдено положение равновесия (a, b) .

Используя предложенный вариант метода Рунге—Кутта, найдем несколько приближенных решений заданной системы, запуская метод Рунге—Кутта с разными начальными условиями. Так, например, можно численно просчитать траекторию, задав начальные условия

$$x(0) = a + \varepsilon, \quad y(0) = b,$$

от 0 до некоторого t_0 , затем проделать то же самое, но уже с начальными условиями $x(0) = a - \varepsilon, y(0) = b$, затем с начальными условиями

$$x(0) = a, \quad y(0) = b + \varepsilon,$$

затем $x(0) = a, y(0) = b - \varepsilon$ и так далее. В вариантах задания задается шаг по t , а величина t_0 и ε выбирается произвольно.

Методы Рунге—Кутта

Пусть задано уравнение

$$y' = f(x, y)$$

и начальное условие

$$y(x_0) = y_0.$$

Требуется найти решение на некотором интервале (x_0, b) . Разобьем отрезок (x_0, b) на n частей с шагом h . Значения в точке $x_{i+1} = x_i + h$ находятся по формуле:

$$y_{i+1} = y_i + h\varphi(x_i, y_i, h), \quad (2.20)$$

где правая часть в (2.20) является приближением отрезка ряда Тейлора функции $y(x)$ в точке x_i до p -го порядка и не содержит производных. Рассмотрим наиболее простые случаи.

1. При $p = 1$ формула (2.18) представляет собой метод Рунге—Кутта первого порядка. При этом правая часть (2.18) не содержит производных и приближает отрезок ряда Тейлора функции $y(x)$ в точке x_i до первой производной. В этом случае метод Рунге—Кутта первого порядка совпадает с методом Эйлера.
2. При $p = 2$ формулы метода Рунге—Кутта второго порядка имеют вид:

$$y_{i+1} = y_i + \frac{k_1 + k_2}{2},$$

где

$$k_1 = hf(x_i, y_i),$$

$$k_2 = hf(x_i + h, y_i + k_1).$$

3. При $p = 3$ получаем метод Рунге—Кутта третьего порядка:

$$y_{i+1} = y_i + \frac{k_1 + 4k_2 + k_3}{6},$$

где

$$k_1 = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right),$$

$$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}),$$

$$k_3 = hf(x_i + h, y_i - k_1 + 2k_2).$$

4. Отметим, что на практике обычно применяется метод Рунге—Кутты четвертого порядка. По методу Рунге—Кутты четвертого порядка приближенные значения неизвестной функции y_i в точках x_i определяются последовательно из следующих равенств:

$$y_{i+1} = y_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4),$$

где

$$k_1 = hf(x_i, y_i),$$

$$k_2 = hf(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}),$$

$$k_3 = hf(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}),$$

$$k_4 = hf(x_i + h, y_i + k_3).$$

Далее перейдем к решению задачи.

Задание

Построить фазовые траектории в окрестности состояний равновесия следующей системы дифференциальных уравнений:

$$\begin{cases} \dot{x} = xy - 4, \\ \dot{y} = (x - 4)(y - x). \end{cases}$$

Первое, что необходимо сделать, — найти состояния равновесия данной системы. Для этого приравняем правые части дифференциальных уравнений и решим систему:

$$\begin{cases} xy - 4 = 0, \\ (x - 4)(y - x) = 0. \end{cases}$$

Из второго уравнения находим, что $x = 4$ и $x = y$. Подставляя эти значения в первое уравнение, получаем состояния равновесия: $(4, 1)$, $(2, 2)$ и $(-2, -2)$.

Далее зададим начальные условия для фазовых траекторий:

```
In[1]:=Points:={{2., 2.}, {-2., -2.}, {4., 1.}};
InArea[x_, y_] := (Abs[x] ≤ 10) && (Abs[y] ≤ 10);
nP = 3; ε = 0.2;
ε1[x_, y_, ε_] := {x + ε, y}; ε2[x_, y_, ε_] := {x - ε, y};
ε3[x_, y_, ε_] := {x, y + ε}; ε4[x_, y_, ε_] := {x, y - ε};
```

```

 $\epsilon 5[x\_ , y\_ , \epsilon\_ ] := \{x + \epsilon, y + \epsilon\}; \epsilon 6[x\_ , y\_ , \epsilon\_ ] := \{x + \epsilon, y - \epsilon\};$ 
 $\epsilon 7[x\_ , y\_ , \epsilon\_ ] := \{x - \epsilon, y + \epsilon\}; \epsilon 8[x\_ , y\_ , \epsilon\_ ] := \{x - \epsilon, y - \epsilon\};$ 
 $\epsilon s = \{\epsilon 1, \epsilon 2, \epsilon 3, \epsilon 4, \epsilon 5, \epsilon 6, \epsilon 7, \epsilon 8\};$ 

```

Зададим отрезок изменения переменной t , количество уравнений в системе и шаг:

```

In[2]:={t0, tn} = {0.0, 20.0};

n = 2;

h = 0.02;

grS = {};

```

Запустим метод Рунге—Кутты. Здесь u — массив, в котором хранятся текущие значения решений, v — массив для хранения значений, полученных на предыдущем шаге, а p — массив со значениями правых частей. В конце выведем на экран фазовые траектории отдельно для каждого состояния равновесия, а также отобразим все фазовые траектории на одном графике:

```

In[3]:= For[pointIndex = 1, pointIndex ≤ nP, pointIndex++,

{a, b} = Points[[pointIndex]];

grs = {};

For[ $\epsilon$ Index = 1,  $\epsilon$ Index ≤ Length[ $\epsilon$ s],  $\epsilon$ Index++,

t = t0; w0 = {x0, y0} =  $\epsilon$ s[[ $\epsilon$ Index]][a, b,  $\epsilon$ ];

Array[u, n, 1]; Array[v, n, 1];

Array[p, n, 1]; Array[q, n, 1];

xlst = List[{t0, x0}]; ylst = List[{t0, y0}];

xylst = List[{x0, y0}];

For[i = 1, i ≤ n, i++, u[i] = w0[[i]]; v[i] = u[i]];

fp[t_, h_, u_][p_] :=

Block[{x = u[1], y = u[2]}, p[1] = h * (xy - 4);

p[2] = h * ((x - 4)(y - x)); m = (tn - t0)/h + 1;

{x, y} = {x0, y0};

For[i = 1, InArea[x, y]&&(i ≤ m), i++, fp[t, h, u][p];

For[k = 1, k ≤ n, k++, q[k] = p[k];

u[k] = v[k] + p[k]/2; t = t + h/2;

fp[t, h, u][p];

```

```

For[k = 1, k ≤ n, k++, q[k] = q[k] + 2p[k];
  u[k] = v[k] + p[k]/2];
fp[t, h, u][p];
For[k = 1, k ≤ n, k++, q[k] = q[k] + 2p[k];
  u[k] = v[k] + p[k]; t = t + h/2;
fp[t, h, u][p];
For[k = 1, k ≤ n, k++, u[k] = v[k] + (q[k] + p[k])/6;
  v[k] = u[k]];
{x, y} = {v[1], v[2]};
xlst = Append[xlst, {t, x}];
ylst = Append[ylst, {t, y}]; xylst = Append[xylst, {x, y}];
];

lst10 = {AbsoluteThickness[2], RGBColor[0, 1, 0]};
lst11 = {AbsoluteThickness[2], AbsoluteDashing[{4, 4}]}];
lst12 = {AbsoluteThickness[2], RGBColor[1, 0, 0]};

tab = Table[Point[xylst[[i]]], {i, 1, Length[xylst]}];
gr1 = Graphics[tab];
grS = Append[grS, gr1];
grs = Append[grs, gr1];
];
Print[Show[grs, Frame → True]];
];
Show[grS, Frame → True]

```

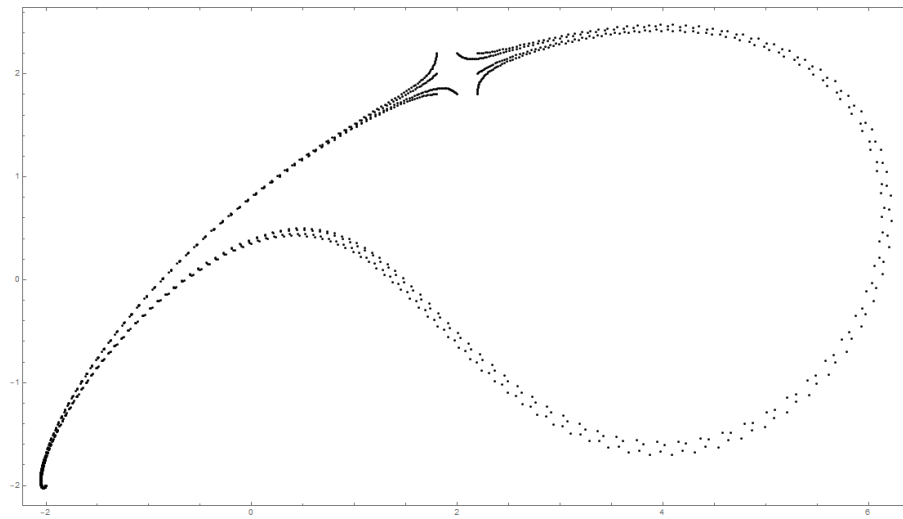


Рис. 2.2. Фазовые траектории в окрестности состояния равновесия $(2, 2)$

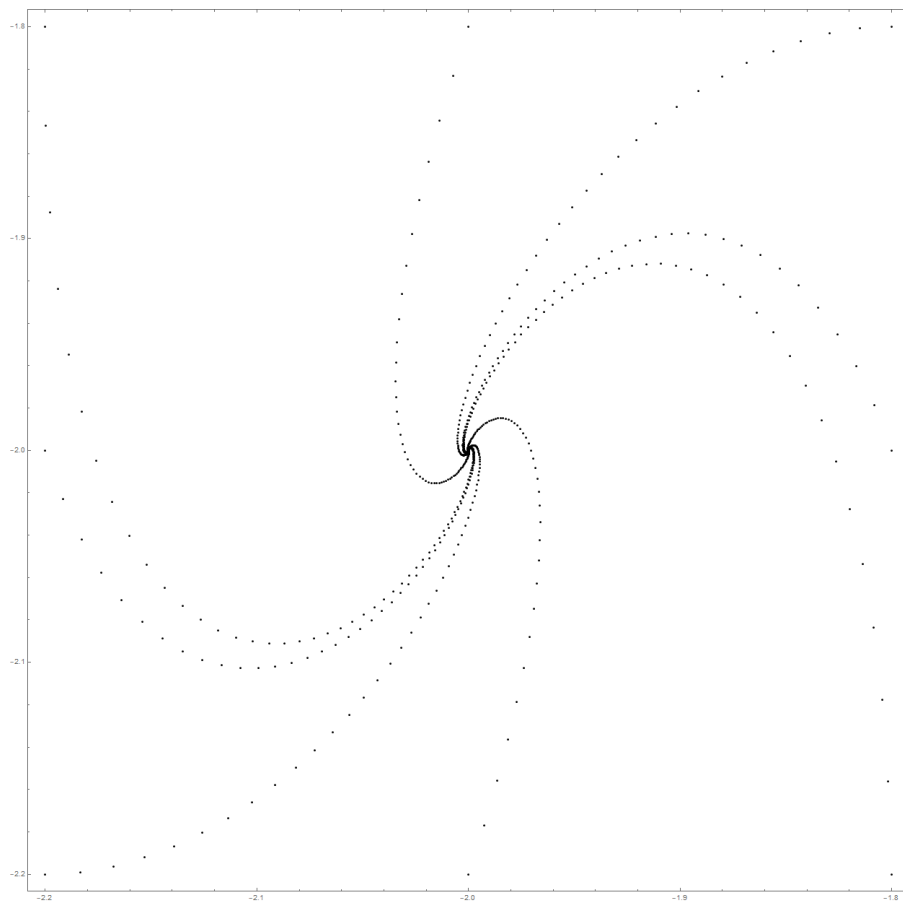


Рис. 2.3. Фазовые траектории в окрестности состояния равновесия $(-2, -2)$

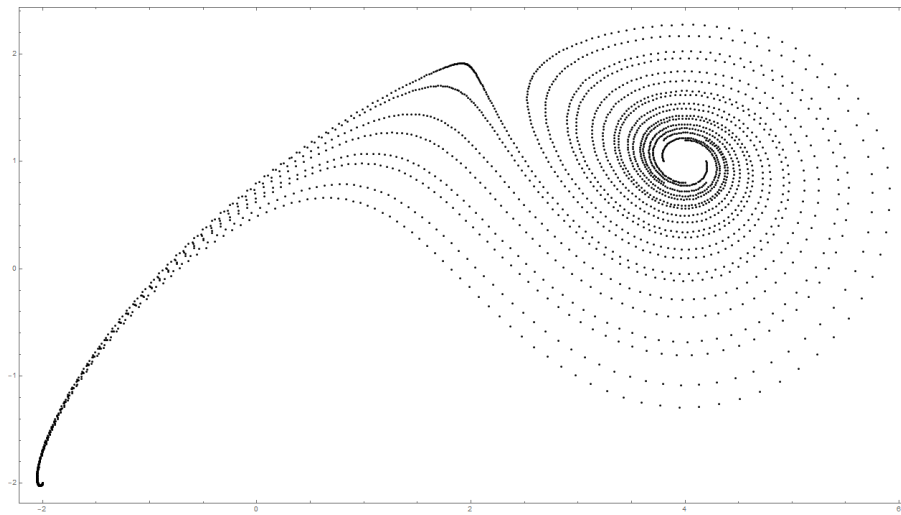


Рис. 2.4. Фазовые траектории в окрестности состояния равновесия $(4, 1)$

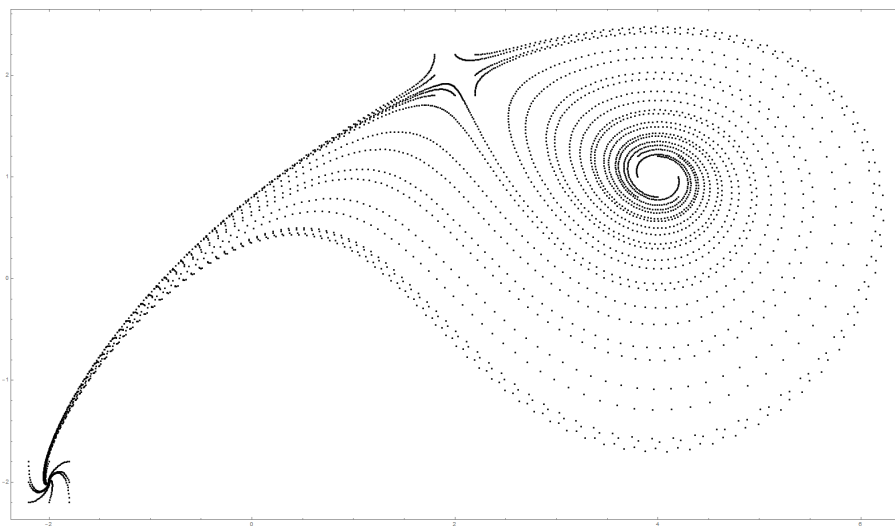


Рис. 2.5. Фазовые траектории в окрестности состояний равновесия $(2, 2)$, $(-2, -2)$, $(4, 1)$

Чтобы убедиться, что все найдено верно, можно воспользоваться встроенной функцией «StreamPlot», которая нарисует диаграмму потоков заданной системы дифференциальных уравнений:

```
In[4]:=StreamPlot[{xy - 4, (x - 4)(y - x)}, {x, -10, 10}, {y, -10, 10}]
```

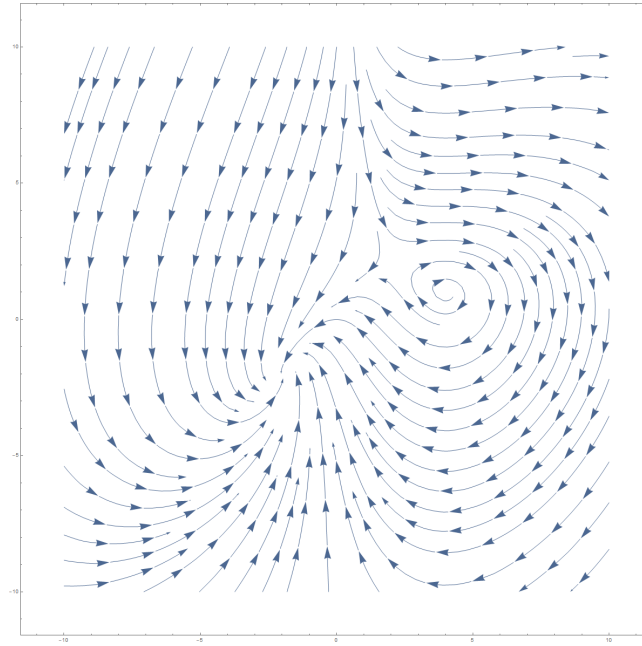


Рис. 2.6. Фазовые траектории системы, построенные с помощью встроенной функции «StreamPlot»

Варианты заданий

Используя различные варианты метода Рунге—Кутты, исследовать поведение траекторий в окрестности каждой особой точки системы дифференциальных уравнений. Если их бесконечно много, то в окрестности двух любых точек. Аналитически обосновать поведение решений в окрестности каждой особой точки.

$$1. \begin{cases} \dot{x} = x - y, \\ \dot{y} = x^2 + y^2 - 2. \end{cases} \quad 2. \begin{cases} \dot{x} = (2x - y)(x - 2), \\ \dot{y} = xy - 2. \end{cases}$$

$$3. \begin{cases} \dot{x} = x^2 - y, \\ \dot{y} = x^2 - (y - 2)^2. \end{cases} \quad 4. \begin{cases} \dot{x} = 2x + y^2 - 1, \\ \dot{y} = 6x - y^2 + 1. \end{cases}$$

$$5. \begin{cases} \dot{x} = y^2 - 4x, \\ \dot{y} = 4y - 8. \end{cases} \quad 6. \begin{cases} \dot{x} = 4 - 4x - 2y, \\ \dot{y} = xy. \end{cases}$$

$$7. \begin{cases} \dot{x} = 1 - x^2 - y^2, \\ \dot{y} = 2x. \end{cases} \quad 8. \begin{cases} \dot{x} = 2 + y - x^2, \\ \dot{y} = 2x(x - y). \end{cases}$$

$$9. \begin{cases} \dot{x} = xy - 4, \\ \dot{y} = (x - 4)(y - x). \end{cases} \quad 10. \begin{cases} \dot{x} = 1 - x^2 - y^2, \\ \dot{y} = 2xy. \end{cases}$$

$$11. \begin{cases} \dot{x} = 2(x - 1)(y - 2), \\ \dot{y} = y^2 - x^2. \end{cases} \quad 12. \begin{cases} \dot{x} = (x + y)^2 - 1, \\ \dot{y} = -y^2 - x + 1. \end{cases}$$

Литература

- [1] Бахвалов, Н. С. Численные методы / Н. С. Бахвалов. — 2-е изд. — М. : Наука, 1975. — 632 с.
- [2] Введение в математическое моделирование : учебное пособие / под ред. П. В. Трусова. — М. : Университетская книга, Логос, 2007. — 440 с.
- [3] Демидович, Б. П. Основы вычислительной математики / Б. П. Демидович, И. А. Марон. — 4-е изд. — М. : Наука, 1970. — 664 с.
- [4] Доля, П. Г. Mathematica для математиков. [Электронный ресурс] — Харьковский Национальный Университет, 2015. — Режим доступа : http://geometry.karazin.ua/resources/documents/20150217125800_2fe9e7bfd.pdf.
- [5] Дьяконов, В. П. Mathematica 5/6/7. Полное руководство / В. П. Дьяконов. — М. : ДМК Пресс, 2010. — 624 с.
- [6] Звонарев, С. В. Основы математического моделирования : учебное пособие / С. В. Звонарев. — Екатеринбург : Изд-во Урал. ун-та, 2019. — 112 с.
- [7] Матвеев, В. Н. Лабораторные работы по курсу «Методы вычислений» : учебное пособие / В. Н. Матвеев; Яросл. гос. ун-т. — Ярославль, ЯрГУ, 2004. — 84 с.
- [8] Матвеев, В. Н. Методы вычислений : учебное пособие / В. Н. Матвеев; Яросл. гос. ун-т. — Ярославль : ЯрГУ, 2007. — 156 с.
- [9] Мостовской, А. П. Численные методы и система Mathematica : учебное пособие / А. П. Мостовской. — Мурманск, 2009. — 249 с.
- [10] Мышкис, А. Д. Элементы теории математических моделей : учебное пособие / А. Д. Мышкис. — М. : КомКнига, 2007. — 192 с.
- [11] Самарский, А. А. Введение в численные методы / А. А. Самарский. — 2-е изд. — М. : Наука, 1987. — 288 с.
- [12] Самарский, А. А. Математическое моделирование : Идеи. Методы. Примеры. 2-е изд., испр. / А. А. Самарский, А. П. Михайлов. — М. : Физматлит, 2001. — 320 с.
- [13] Тарг, С. М. Краткий курс теоретической механики : учебник для втузов. — 10-е изд., перераб. и доп. / С. М. Тарг. — М. : Высш. шк., 1986. — 416 с.
- [14] Тихонов, Н. А. Основы математического моделирования : учебное пособие / Н. А. Тихонов, М. Г. Токмачев. — М. : Физический факультет МГУ, 2013. — 175 с.

- [15] Тихонов, А. Н. Уравнения математической физики : учебное пособие. — 6-е изд., испр. и доп. / А. Н. Тихонов, А. А. Самарский. — М. : Изд-во МГУ, 1999. — 799 с.
- [16] Wolfram, S. Mathematica : A system for Doing Mathematics by Computer. — Second Edition, Addison-Wesley / S. Wolfram. — New York, 1991. — 961 p.

Учебное издание

Алина Вадимовна Секацкая

Система «Wolfram Mathematica» в задачах математической физики и математическом моделировании

Учебно-методическое пособие

Редактор, корректор Л. Н. Селиванова
Компьютерная верстка А.В. Секацкая

Подписано в печать 24.09.2021. Формат 60 x 84 1/16.

Усл. печ. л. 3,49. Уч.-изд. л. 3,0.

Тираж 3 экз. Заказ 047/021.

Оригинал-макет подготовлен
в редакционно-издательском отделе
Ярославского государственного университета

Адрес типографии:
Ярославский государственный университет.
150003, Ярославль, ул. Советская, 14.