МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "ЧИСЕЛЬНІ МЕТОДИ"

для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання

Харків. Вид. ХНЕУ, 2012

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 11 від 03.04.2012 р.

Укладачі: Задачин В. М.
Конюшенко І. Г.

М54 Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Чисельні методи" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання / укл. Задачин В. М., Конюшенко І. Г. – Х. : Вид. ХНЕУ, 2012. – 96 с. (Укр. мов.)

Запропоновано методичні рекомендації до виконання лабораторних робіт, метою яких є придбання студентами навичок розв'язання різноманітних математичних задач чисельними методами.

Рекомендовано для студентів напряму підготовки 6.050101 "Комп'ютерні науки".

Загальні положення

Вивчення дисципліни "Чисельні методи" дозволяє студентам оволодіти знаннями в галузі практичних методів вирішення математичних проблем, що виникають у процесі інженерної діяльності та моделювання фізичних систем, засвоїти способи розрахунків на сучасних комп'ютерах із застосуванням пакетів спеціальних прикладних програм.

Об'єктом вивчення дисципліни є типові математичні задачі, до яких зводиться вирішення практичних проблем, що виникають при розробці інформаційних систем та систем моделювання. Предметом вивчення дисципліни є чисельні методи розв'язання типових математичних задач.

Метою лабораторних робіт з навчальної дисципліни є вивчення студентами основних чисельних методів і придбання навичок застосування їх для розв'язання різноманітних математичних задач, що виникають при розробці комп'ютерних систем, а також придбання навичок застосування при цьому математичних пакетів.

При розробці комп'ютерних систем нерідко доводиться розв'язувати математичні задачі різної складності, наприклад:

розв'язання рівнянь і систем алгебраїчних рівнянь, лінійних і нелінійних;

диференціювання й інтегрування функцій;

апроксимації (наближення) даних;

розв'язання диференціальних рівнянь (звичайних та з частинними похідними);

розв'язання інтегральних рівнянь та ін.

Основним інструментом розв'язання цих задач на комп'ютері є чисельні методи.

Чисельними методами називаються методи розв'язання математичних задач, що зводяться до скінченної кількості арифметичних і деяких логічних дій над числами.

Розв'язок, отриманий за допомогою чисельного методу, зазвичай є наближеним, тобто містить деяку похибку. Джерелами похибки є:

- невідповідність математичної постановки задачі досліджуваному реальному явищу;
- похибка початкових даних;

- похибка чисельного методу розв'язання;
- помилки округлення та розрахунків на комп'ютері.

При виборі чисельного методу розв'язання поставленої математичної задачі перевага надається тому методу, який дозволяє одержати достатньо точний розв'язок з найменшими затратами (часовими, ресурсними тощо). Основними характеристиками чисельних методів є:

- трудомісткість;
- швидкість збіжності;
- стійкість до похибок обчислень;
- стійкість до похибок у початкових даних.

У світі розроблено ряд універсальних математичних пакетів, за допомогою яких можна достатньо швидко як розв'язати багато відомих математичних задач, так і провести попередні розрахунки з подальшою їх реалізацією вже в проектованій комп'ютерній системі. Найбільш відомими є такі математичні пакети: Mathematica, Maple, MATLAB, MathCad, R.

Треба зазначити, що останнім часом намітилася тенденція до зближення та інтеграції різних математичних пакетів. Так, останні версії пакетів Mathematica і Maple мають потужні засоби для візуального програмування; в пакети MATLAB та MathCad включена бібліотека аналітичних перетворень Maple; MathCad дозволяє працювати разом з MATLAB тощо. Тому, за великим рахунком, немає значення, який математичний пакет застосовується в навчальному процесі та засвоюється студентами. Головне – принципово навчитися використовувати будь-який математичний пакет для розв'язання практичних задач.

Останнім часом математичний пакет R набуває все більшої популярності у світі, тому саме пакет R використовується у цих методичних рекомендаціях для програмування чисельних методів, виконання розрахунків та наочного представлення результатів розв'язання математичних задач.

Позначення

Позна-чення	Значення
R^1	простір дійсних чисел
$x \in G$	x належить множині G (x – елемент множини G)
$i = \overline{1,n}$	i приймає значення від 1 до n (еквівалентно $i = 1,\ldots,n$)
$\forall x \in G$	для всіх x, що належать множині G
$\exists x \in G$	існує x, що належить множині G
R^n	n-вимірний дійсний простір векторів-стовпців, тобто $x = (x_1, x_2, \ldots, x_n)^T$, де $x_i \in R^1,\ i = \overline{1,n}$
$x \in R^n$	x – n-вимірний вектор-стовпець
x^T	вектор, транспонований до x (вектор-рядок)
$x^k \in R^n$	x^k – n-вимірний вектор-стовпець, k-й за номером в деякій нумерації
x_i	i-та компонента вектора x
$\{x^k\}$	послідовність x^k, тобто $x^0, x^1, x^2, \ldots$
$\{x \in G : T\}$	підмножина елементів множини G, що задовольняють умові T
$x \ge 0, x \in R^n$	усі компоненти n-вимірного вектора x невід'ємні
$A = (a_{ij})_{ij=1}^n$	матриця розмірності n на n з елементами a_{ij}, де a_{ij} – елемент i-го рядка й j-го стовпця
$R^{n \times m}$	множина матриць розмірності n на m
A^T	матриця, транспонована до A, тобто матриця з елементами a_{ji}
$f : R^1 \to R^1$	дійсна функція однієї дійсної змінної, тобто якщо $x \in R^1$ та $y = f(x)$, то $y \in R^1$
$f : R^n \to R^m$	дійсна вектор-функція n дійсних змінних, тобто якщо $x \in R^n$ й $y = f(x)$, то $y \in R^m$
$x^T y$	Скалярний добуток векторів x, y, тобто якщо $x, y \in R^n$, то $x^T y = \left[\sum_{i=1}^n x_i y_i\right]$
$\lVert x \rVert$	Евклідова норма вектора x, тобто якщо $y \in R^n$, то $\lVert x \rVert = \left[\sum_{i=1}^n x_i^2\right]^{\frac{1}{2}} = \sqrt{x^T x}$

Лабораторна робота № 1
Вступ у систему R

1.1. Мета роботи

Вивчення можливостей математичного пакета **R**, придбання навичок використання цього пакета для розв'язання математичних задач та графічного представлення результатів досліджень на комп'ютері.

1.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальні поняття лінійної алгебри та програмування; *вміти* застосовувати засоби пакета R для розв'язання простих математичних задач.

1.2.1. Числові вектори

Вектори в системі R формуються функцією **c()**. Аргументи цієї функції є компонентами вектора. Наприклад, команда

```
> a = c(1, 2, 3, 4, 5)
> a
[1] 1 2 3 4 5
```

формує вектор-рядок **(1, 2, 3, 4, 5)** і привласнює його змінній **a**.

Аргументи функції **c()** можуть бути векторами. В цьому випадку як результат маємо конкатенацію цих векторів. Скалярні значення (тобто числа) сприймаються R як вектори довжини 1. Таким чином, аргументами функції **c()** можуть бути як вектори, так і скаляри.

Наприклад:

```
> c(c(1, 2, 3, 4, 5), 6, c(7, 8))
[1] 1 2 3 4 5 6 7 8
```

Вектор, що складається з послідовних чисел, можна отримати за допомогою команди <початкове_значення>:<кінцеве_значення>. Наприклад, 1:5. На цю команду схожа функція **seq**, яка генерує відрізки арифметичної прогресії. Можна задати початкове значення, кінцеве значення і крок:

```
> seq(0, 1, by=0.1)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

В іншому варіанті задається початкове значення, кінцеве значення і кількість точок. У результаті буде згенерований вектор, що складається з заданої кількості компонент, рівномірно розподілених на заданому відрізку:

```
> seq(0, 1, len=11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

Функція **rep(v,k)** створює вектор, що складається з **k** копій вектора **v**. Наприклад:

```
> rep(c(1, 2), 3)
[1] 1 2 1 2 1 2
```

Над векторами можна виконувати арифметичні операції й елементарні функції. Елементарні математичні функції застосовуються до кожної компоненти вектора. Наприклад:

```
> x = c(0, pi/2, pi)
> sin(x)
[1] 0.000000e+00 1.000000e+00 1.224606e-16
```

Доступ до елементів вектора здійснюється оператором **[i]**. Наприклад, **u[5]** – це 5-й елемент вектора **u**. Нумерація елементів починається з 1. Вирази виду **u[i]** можуть зустрічатися і в лівій частині від знака привласнення. При цьому, якщо вектор має довжину не менше **i**, то **u[i]** просто прийме нове значення. В інакшому випадку вектор **u** збільшить свою довжину до **i**, компонента **u[i]** прийме нове значення, а останні нові компоненти приймуть значення **NA**.

```
> u = 1
> u[5] = 5
> u
[1]  1 NA NA NA  5
```

Деякі корисні функції:

crossprod(v, w) – скалярний добуток векторів **v** та **w**;

sum(v) – сума елементів вектора **v**;

prod(v) – добуток елементів вектора **v**;

max(v) – максимальний елемент;

min(v) – мінімальний елемент;

length(v) – довжина вектора;

mean(v) – середнє значення елементів вектора **v** (оцінка математичного сподівання);

var(v) – варіація елементів вектора v (оцінка дисперсії);

sort(v) – повертає вектор тієї ж довжини, що і **v**, з елементами, відсортованими в порядку збільшення.

1.2.2. Рядкові назви елементів вектора

Елементам вектора (числового, логічного, символьного) можна задавати імена. Наприклад:

```
> fruit = c(5, 10, 1, 20)
> names(fruit) = c("orange", "banana","apple", "peach")
> fruit
orange banana  apple  peach
     5     10      1     20
```

Створений вектор довжиною 4, компоненти якого мають вказані назви. Тепер звертатися до елементів вектора можна як по індексу, так і по імені:

```
> fruit[1] = 6
> fruit["peach"] = 60
> fruit
orange banana  apple  peach
     6     10      1     60
```

1.2.3. Матриці

Числову матрицю можна побудувати з числового вектора за допомогою функції **matrix**. Треба вказати кількість рядків **nrow=m** і/або кількість стовпців **ncol=m**. У результаті елементи з вектора будуть записані в матрицю вказаних розмірів. Наприклад:

```
> matrix(1:6, nrow=2, ncol=3)
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Довжина вектора має бути кратною добутку потрібної кількості рядків і потрібної кількості стовпців:

```
> matrix(1, nrow=2, ncol =2)
     [,1] [,2]
[1,]    1    1
[2,]    1    1

> matrix(1:2, nrow=3, ncol =2)
     [,1] [,2]
[1,]    1    2
[2,]    2    1
[3,]    1    2
```

Елементи записуються в матрицю по стовпцях. Щоб записати їх по рядках, треба задати значення додаткового параметра **byrow=TRUE**:

```
> matrix(1:6, nrow=2, byrow=TRUE)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

Функції **nrow(A)** і **ncol(A)** повертають відповідно кількість рядків і стовпців матриці **A**.

Доступ до елементів матриці відбувається по індексу. Так **A[i,j]** посилається на елемент **i**-го рядка і **j**-го стовпця матриці **A**.

A[i,] еквівалентне **A[i,1:ncol(A)]**, а **A[,j]** еквівалентне **A[1:nrow(A),j]**. Можливий доступ до елементів матриці з допомогою одного індексу. Наприклад, **U[5]** означає 5-й елемент матриці **U**, якщо вважати, що елементи перенумеровані по стовпцях. Якщо **i**-вектор, то **A[i]** – означає вибірку відповідних елементів і т. д.

Можна задати імена стовпцям і рядкам матриці:

```
> A = matrix(c(23, 31, 58, 16), nrow=2)
> rownames(A) = c("petal","sepal")
> colnames(A) = c("length","width")
> A
      length width
petal     23    58
sepal     31    16
```

Після цього доступ до рядків і стовпців може відбуватися за ім'ям. Наприклад:

```
> A["petal","length"]
[1] 23
```

Функція **A = cbind(A, B)** створює матрицю, приписуючи до **A** справа **B** (для цього кількість рядків у **A** і **B** має співпадати).

Функція **A = rbind(A, B)** створює матрицю, приписуючи до **A** знизу **B** (для цього кількість стовпців у **A** і **B** має співпадати). Відмітимо, що в списках аргументів функцій **cbind** и **rbind** можна вказати більше двох матриць.

Арифметичні операції над матрицями здійснюються по компонентно, тому, наприклад, щоб додати дві матриці, вони повинні мати однакові розміри.

Елементарні математичні функції також застосовуються до елементів матриці покомпонентно.

Транспонування матриці здійснює функція **t(A)**, а матричний добуток – операція **%*%**:

```
> A = matrix(1:6,nrow=2)
> A
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> B = t(A)
> B
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> A%*%B
     [,1] [,2]
[1,]   35   44
[2,]   44   56
> B%*%A
     [,1] [,2] [,3]
[1,]    5   11   17
[2,]   11   25   39
[3,]   17   39   61
```

Для розв'язання системи лінійних рівнянь **Ax = b** з квадратною невиродженою матрицею **A** є функція **solve(A,b)**:

```
> A = matrix(1:4,ncol =2)
> A
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> b = c(5,8)
> b
[1] 5 8
> solve(A,b)
[1] 2 1
```

Функція **det(A)** знаходить визначник, а **solve(A)** – обернену матрицю:

```
> det(A)
[1] -2

> solve(A)
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
```

1.2.4. Графіки функцій

Для зображення графіків функцій у R є функція **plot(x,y)**. Тут **x** – вектор значень абсцис і **y** – вектор значень ординат. Якщо вказаний тільки один аргумент – **plot(y)**, то передбачається, що **x=1:n**, де **n** – довжина вектора **y**. Наприклад, команди

```
> x = seq(-pi, pi, len =30)
> y = sin(x)
> plot(x,y)
```

рисують 30 точок, що лежать на синусоїді **y=sin(x)** (рис. 1.1). Графіки з'являються в окремому вікні (або в кількох окремих вікнах).

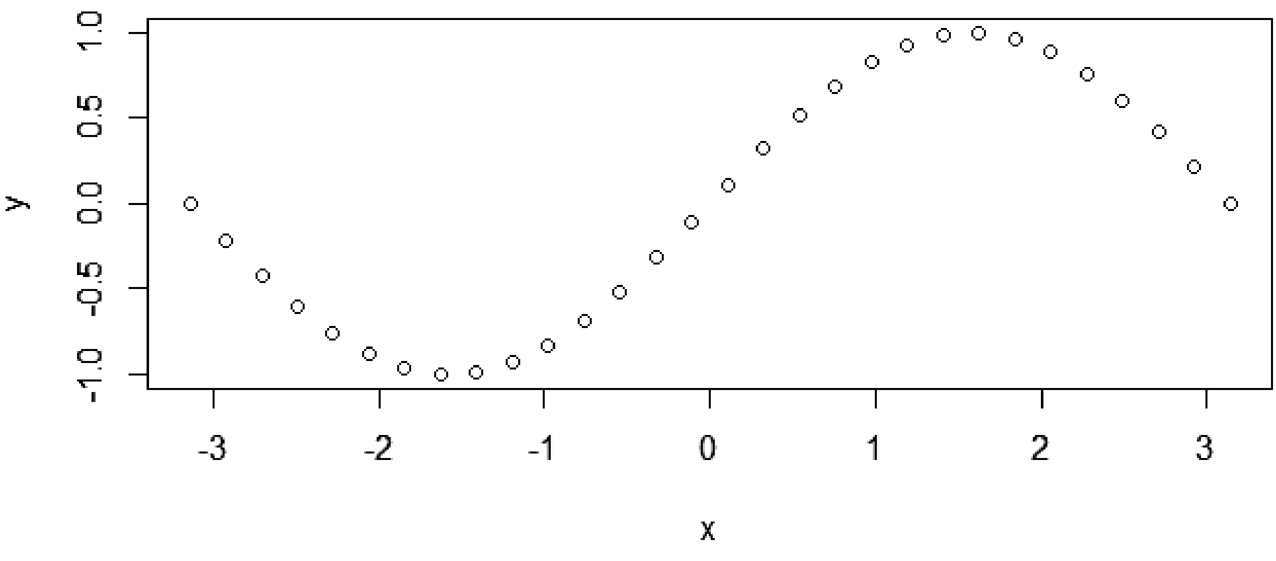


Рис. 1.1. **"Точковий" графік синуса**

Функція **plot** допускає багато додаткових параметрів. Деякі з них:

type=тип-графіка дозволяє вказати тип графіка, що виводиться; деякі значення параметра – "p" (точки, за замовчуванням), "l" (лінії), "b" (лінії і точки), "o" (лінії і точки перекриваються), "n" (нічого не рисується);

xlab=назва-осі-абсцис і **ylab=назва-осі-ординат** дозволяє вказати підписи до осі абсцис і ординат відповідно;

main=основний-надпис і **sub=додатковий-надпис** створюють надписи зверху і знизу графіка;

col=колір задає колір графіка; можливі значення "blue", "red", "green", "cyan", "magenta", "yellow", "black" та багато інших. Кольори можна задати rgb-вектор-функцією **rgb(r, g, b)**. Параметри цієї функції – значення від 0 до 1.

lty=стиль-лінії визначає стиль лінії; можливі значення "blank" (немає лінії), "solid", "dashed", "dotted", "dotdash", "longdash", "twodash".

Розглянемо наступний приклад:

```
> plot(x, y, type="l", col="blue", main="Sine of x")
```

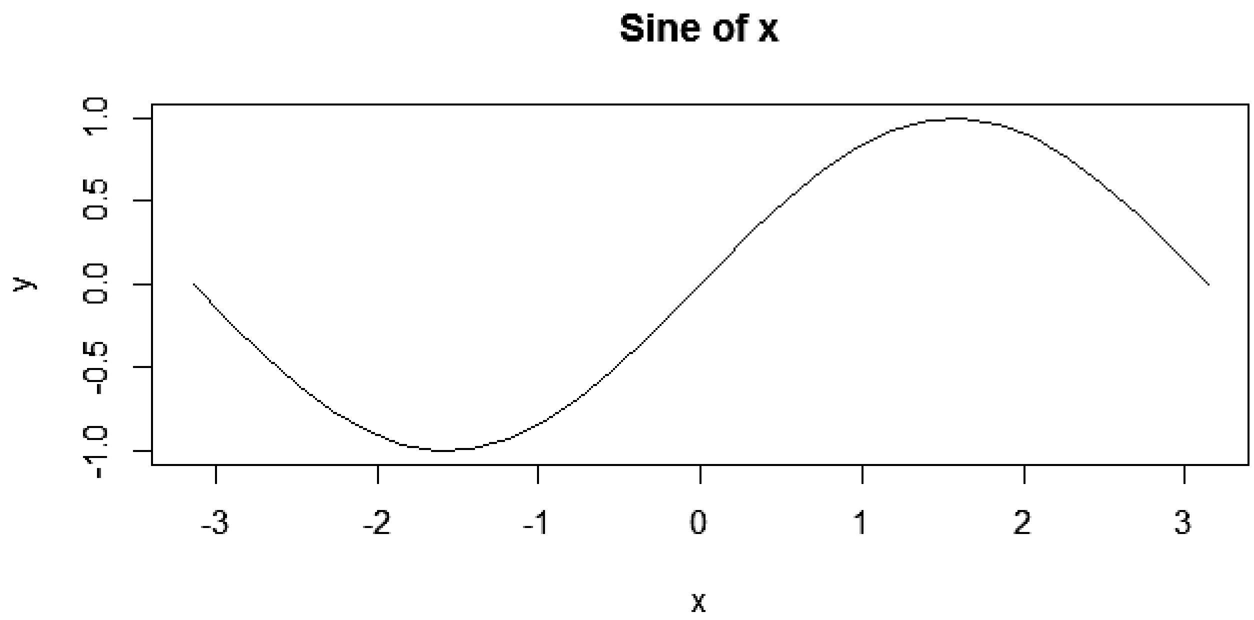


Рис. 1.2. **Графік синуса**

Наступні функції (що називаються низькорівневими) ніколи не стирають зображення, яке вже є. Їм можуть бути передані також додаткові параметри, аналогічні відповідним параметрам функції **plot**:

points(x,y) рисує додаткові точки, **lines(x,y)** рисує ломану лінію, яка з'єднує вказані точки. Цим функціям можуть бути передані додаткові аргументи, наприклад, **type=тип-графика** (за замовчуванням він дорівнює "p" для points і "l" для lines).

text(x,y,текст) додає текст. За замовчуванням він центрується навколо точки **(x,y)**. Вказані параметри можуть бути векторами. В цьому випадку буде розміщено кілька надписів. Можна вказати шрифт **font=шрифт** та ін.

abline(k,b) рисує пряму y=kx+b. Можна вказати колір і стиль лінії тощо.

abline(h=y), **abline(v=x)** – для рисування горизонтальних і вертикальних прямих. Вказуються відповідно координата **y** і **x**.

polygon(x,y) рисує багатокутник з вершинами **(x,y)**. Можна вказати колір заповнення **col=колір** і колір границі **border=колір**.

legend(x,y,легенда) додає легенду у вказану точку.

Розглянемо приклад:

```
> x = seq(-10, 10, by=0.1)
> y1 = x^2+2*x-5
> y2 = -x^2+60
> plot(x, y1, type="n",main="Графіки функцій")
> colors = c("blue", "red", "green")
> lines(x,y1, col=colors[1])
> lines(x,y2, col=colors[2], lty="dashed")
> legends = paste("y",1:2, sep="")
> legend(3, 110, legends, lty="solid",col=colors)
> abline(h=0) #Горизонтальна вісь
> abline(v=0) #Вертикальна вісь
```

Результат наведений на рис. 1.3

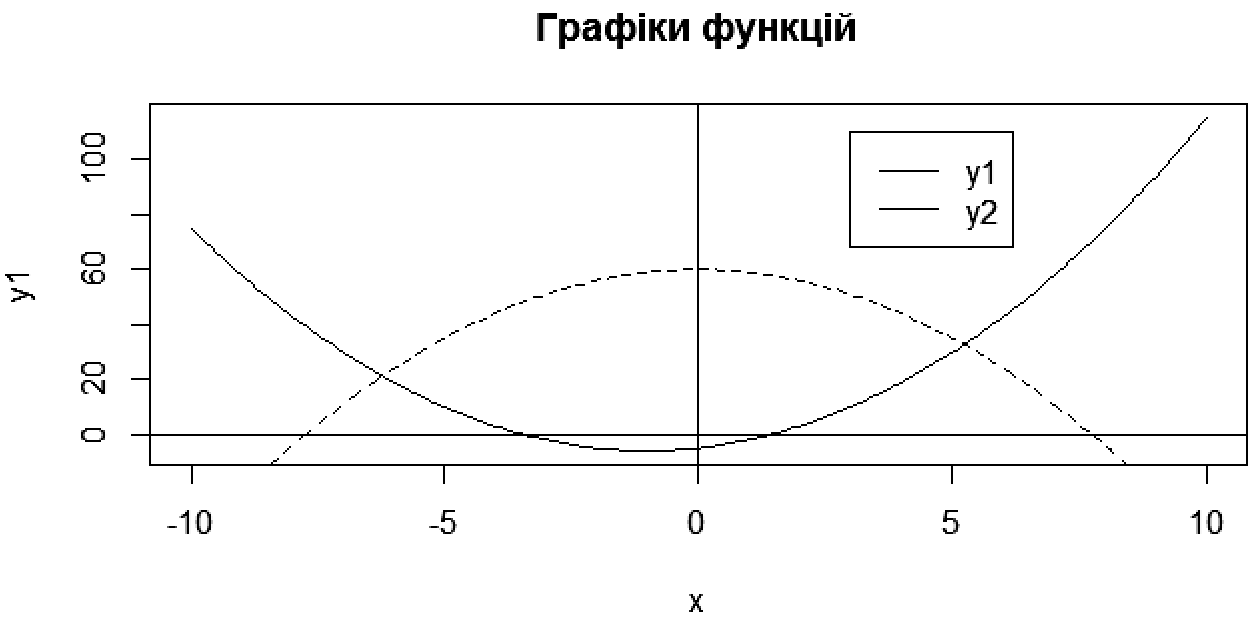


Рис. 1.3. **Два графіка в одному графічному вікні**

1.2.5. Управляючі конструкції

Кожна команда чи вираз у мові R повертає деякий результат. Навіть привласнення – це вираз, результат якого – привласнюване значення. Вираз може бути частиною іншого виразу. Зокрема, можливі множинні привласнювання, наприклад:

```
> x = y = z = 0
```

Перевірка умов здійснюється так, як і в мові С:

```
> if(умова) {вираз1} else {вираз2}
```

де *умова* – це логічний вираз, результатом якого є скалярне логічне значення.

Цикли з управляючою змінною реалізуються з допомогою конструкції:

```
> for (змінна in вираз1) {вираз2}
```

Результатом *вираз1* має бути вектор, при цьому змінна на кожній ітерації циклу приймає значення чергового елемента цього вектора. Кількість ітерацій дорівнює кількості елементів у векторі.

```
> s=0
> for(i in 1:20)  s = s+i
> s
[1] 210
```

Відмітимо, що той же результат швидше можна отримати за допомогою команди. **sum(1:20)**

Цикли з передумовою реалізуються за допомогою конструкції:

```
> while (умова) {вираз}
```

Передчасний вихід з циклу здійснюється командою **break**. Для переривання поточної ітерації і переходу до наступної служить команда **next**. Ці команди можна також використовувати і з конструкцією **for**.

Безкінечний цикл реалізує команда **repeat**:

```
> repeat {вираз}
```

Всередині цієї конструкції можна використовувати команди **break** і **next**.

1.2.6. Функції користувача

Формат визначення функції такий:

```
> ім'я = function(arg1,arg2, ...) {вираз}
```

Тут список формальних аргументів (формальних параметрів) **арг1, арг2,...** може мати довільну довжину (в тому числі, бути пустим). Як правило, вираз (тіло функції) є блоком. Значення, яке повертає функція, це значення цього виразу. Дострокове повернення з функції здійснює команда **return(вираз1)**. У цьому випадку функція повертає значення вказаного виразу.

Звернення до функції має вид **ім'я(вираз1, вираз2, ...)**. Тут значення виразів **вираз1, вираз2,...** є фактичними аргументами, які підставляються замість відповідних формальних аргументів.

При зміні значень формальних аргументів всередині функції, не змінюються відповідні значення фактичних аргументів. Єдиний спосіб зробити це – використовувати в тілі функції <сильне> привласнювання: **arg1 <-- вираз**.

Якщо фактичні аргументи задані в <іменованому> вигляді **ім'я = вираз**, то вони можуть бути пераховані в довільному порядку. Більш того, список фактичних аргументів може починатися з аргументів, представлених у звичайній позиційній формі, після чого можуть іти аргументи в іменованій формі.

Наприклад, є функція **fun**, задана таким чином:

```
> fun = function(arg1,arg2,arg3,arg4)
+   {
+     #тіло функції опущене
+   }
```

Тоді **fun** може бути викликана численними еквівалентними способами, наприклад:

```
> ans = fun(d,f, 20, TRUE)
> ans = fun(d,f, arg4=TRUE, arg3=20)
> ans = fun(arg1=d, arg3=20, arg2=f, arg4=TRUE)
```

У багатьох випадках при визначенні функції деяким аргументам можуть бути привласнені значення за замовчуванням. Тоді при виклику функції ці аргументи можуть бути опущені. Припустимо, що функція **fun** визначена як

```
> fun = function(arg1, arg2, arg3=20, arg4=TRUE)
+ {
+   #тіло функції опущене
+ }
```

Тоді звертання до цієї функції виду **fun(d,f)** еквівалентне трьом, приведеним вище. Наступне звертання

```
> fun(d,f, arg4=FALSE)
```

змінює одне із значень за замовчуванням.

Відзначимо, що в значеннях за замовчуванням можна використовувати будь-які вирази, у тому числі такі, що містять інші аргументи функції.

Усі символи, що зустрічаються в тілі функції, діляться на три групи: це формальні аргументи, локальні змінні й вільні змінні.

Формальні аргументи (формальні параметри) – це аргументи, перераховані в заголовку функції (в круглих дужках після ключового слова **function**).

Локальні змінні – це змінні, які не є формальними аргументами, значення яких визначаються під час виконання функції.

Змінні, які не є формальними аргументами і локальними змінними, є вільними змінними.

Наприклад, у функції

```
> f = function(x)
+ {
+   y = 2*x
+   print(x)
+   print(y)
+   print(z)
+ }
```

x – формальний аргумент, **y** – локальна змінна, **z** – вільна змінна.

Область видимості формальних аргументів і локальних змінних – лише сама ця функція. Це означає, що зміни цих змінних всередині функції ніяк не відображаються на змінних з такими ж іменами у зовнішній функції. Область видимості вільних змінних розповсюджується до зовнішньої функції, в якій вони були визначені. Зміна таких змінних у тілі функції впливає також на відповідні змінні в цій зовнішній функції.

1.3. Порядок виконання роботи і варіанти завдань

1.3.1. Зміст звіту

У практичній частині роботи необхідно:

- виконати всі приклади наведені в розділі 1.2;

- виконати завдання наведене далі;
- привести текст складеної програми та результати її виконання.

1.3.2. Завдання

Запишіть функцію, яка приймає на вхід числовий вектор **x** і число розбиття інтервала **k** (за замовчуванням дорівнює кількості елементів вектора, поділеному на 10) і виконує таке: знаходить мінімальне і максимальне значення елементів вектора **xmin** й **xmax**, ділить отриманий відрізок **[xmin; xmax]** на **k** рівних інтервалів і підраховує кількість елементів вектора, що належать кожному інтервалу. Далі має будуватися графік, де по осі абсцис – середини інтервалів, по осі ординат – кількість елементів вектора, що належать інтервалу, поділене на загальну кількість точок. Проведіть експеримент на даній функції, де x – вектор довжиною 5 000, згенерований із нормально розподіленої випадкової величини з математичним сподіванням $a = 5.5$ і середньоквадратичним відхиленням – $\sigma = 1.5$ (функція **rnorm**).

1.4. Контрольні запитання

1. Які є засоби в системі R для створення та роботи з векторами та матрицями?
2. Які є засоби в системі R для побудови графіків?
3. Які є засоби в системі R управляючі конструкції при програмуванні?
4. Як описати функцію користувача в системі R?

Лабораторна робота № 2
Чисельні методи розв'язання систем лінійних алгебраїчних рівнянь

2.1. Мета роботи

Вивчення методу Гауса із частковим вибором головного елемента та ітераційних методів для практичного розв'язання систем лінійних алгебраїчних рівнянь, придбання навичок використання цих методів для розв'язання систем лінійних алгебраїчних рівнянь із застосуванням комп'ютера.

2.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі розв'язання системи лінійних алгебраїчних рівнянь; *вміти* розв'язувати системи лінійних алгебраїчних рівнянь методом виключення Гауса [1 – 6] і методом ітерацій; *вміти* застосовувати процедури пакета R для розв'язання систем лінійних алгебраїчних рівнянь.

Задача розв'язання системи алгебраїчних лінійних рівнянь у загальному вигляді формулюється в такий спосіб: необхідно знайти n невідомих $x_i \in R^1$, $i = \overline{1,n}$, що задовольняють системі рівнянь:

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

де $a_{ij} \in R^1$, $i = \overline{1,n}$, $j = \overline{1,n}$ – задані коефіцієнти при невідомих, $b_i \in R^1$, $i = \overline{1,n}$ – задані праві частини (вільні члени).

Якщо ввести позначення $A = (a_{ij})_{ij=1}^{n}$ – матриця розмірності $n \times n$ з коефіцієнтами a_{ij}, $b = (b_i)_{i=1}^{n}$ – вектор-стовпець розмірності n з елементами b_i, $x = (x_i)_{i=1}^{n}$ – вектор-стовпець розмірності n з елементами x_i, то систему (1.1) можна записати у матричному вигляді:

$$A\,x = b. \qquad (2.2)$$

Для того, щоб система (2.2) мала єдиний розв'язок необхідно й достатньо, щоб $\det A \neq 0$.

Найбільш відомим з методів розв'язання системи (2.1) є **метод виключення Гауса**, ідея якого полягає в послідовному виключенні невідомих з рівнянь.

Розрахункові формули метода Гауса:

Позначимо $a_{ij}^{(0)} = a_{ij}$, $i = \overline{1,n}$, $j = \overline{1,n}$; $a_{i,n+1}^{(0)} = b_i$, $i = \overline{1,n}$.

Прямий хід, k-й крок ($k=\overline{1,n}$): $a_{kk}^{(k)}=1$, $a_{kj}^{(k)}=\dfrac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$, $j=\overline{k+1,n+1}$;

$a_{ij}^{(k)}=a_{ij}^{(k-1)}-a_{ik}^{(k-1)}\times a_{kj}^{(k)}$, $i=\overline{k+1,n}$, $j=\overline{k+1,n+1}$.

Зворотний хід: $x_n=a_{n,n+1}^{(n)}$, $x_k=a_{k,n+1}^{(k)}-\sum\limits_{j=k+1}^{n}a_{kj}^{(k)}x_j$, $k=\overline{n-1,1}$.

При ручному рахунку застосовують схему (табл. 2.1 для $n=4$).

Таблиця 2.1

Схема ручного розрахунку (метод Гауса)

Коефіцієнти при невідомих				Вільні члени b
x_1	x_2	x_3	x_4	
a_{11}	a_{12}	a_{13}	a_{14}	$a_{15}=b_1$
a_{21}	a_{22}	a_{23}	a_{24}	$a_{25}=b_2$
a_{31}	a_{32}	a_{33}	a_{34}	$a_{35}=b_3$
a_{41}	a_{42}	a_{43}	a_{44}	$a_{45}=b_4$
1	$a_{12}^{(1)}$	$a_{13}^{(1)}$	$a_{14}^{(1)}$	$a_{15}^{(1)}$
0	$a_{22}^{(1)}$	$a_{23}^{(1)}$	$a_{24}^{(1)}$	$a_{25}^{(1)}$
0	$a_{32}^{(1)}$	$a_{33}^{(1)}$	$a_{34}^{(1)}$	$a_{35}^{(1)}$
0	$a_{42}^{(1)}$	$a_{43}^{(1)}$	$a_{44}^{(1)}$	$a_{45}^{(1)}$
	1	$a_{23}^{(2)}$	$a_{24}^{(2)}$	$a_{25}^{(2)}$
	0	$a_{33}^{(2)}$	$a_{34}^{(2)}$	$a_{35}^{(2)}$
	0	$a_{43}^{(2)}$	$a_{44}^{(2)}$	$a_{45}^{(2)}$
		1	$a_{34}^{(3)}$	$a_{35}^{(3)}$
		0	$a_{44}^{(3)}$	$a_{45}^{(3)}$
			1	$a_{45}^{(4)}$

Рядки, що містять одиницю, називаються виділеними рядками. Діагональний елемент $a_{kk}^{(k-1)}$, на який проводиться ділення, називається **головним** (ведучим) **елементом**. Якщо головний елемент близький до нуля за абсолютною величиною, то необхідно знайти у відповідному (k-му) стовпці максимальний за модулем елемент і переставити рядки місцями так, щоб цей елемент став головним.

Процес одержання виділених рядків (приведення системи до трикутного вигляду) називається **прямим ходом**, а процес знаходження невідомих шляхом використання виділених рядків – **зворотним ходом** методу Гауса.

Метод ітерацій є найбільш типовим прикладом ітераційного (наближеного) методу розв'язання системи лінійних рівнянь (2.1).

Для застосування методу ітерацій систему (2.1) необхідно представити у вигляді:

$$x = Cx + d, \tag{2.3}$$

де $C \in R^{n \times n}$, $d \in R^n$.

Алгоритм методу ітерацій.

1. Задається $\varepsilon > 0$ – точність розв'язання задачі і початкове наближення розв'язку $x^0 \in R^n$ (наприклад $x^0 = d$).

2. На k-й ітерації методу ($k = 0, 1, 2, \ldots$) обчислюється наступне наближення x^{k+1} за формулою:

$$x^{k+1} = Cx^k + d.$$

3. Перевіряється критерій "останову" $\frac{q}{1-q}\left\|x^{k+1} - x^k\right\| \le \varepsilon$, де $0 < q < 1$ (визначається з умови збіжності).

Метод простої ітерації збігається тільки при виконанні умови:

$$\|C\| \le q, \tag{2.4}$$

де $0 < q < 1$ (умова збіжності) [1 – 3; 5; 6]. При цьому як норму матриці C можна розглядати величину

$$\|C\| \equiv \underbrace{\max}_{j} \sum_{i=1}^{n} \left|c_{ij}\right| \text{ або } \|C\| \equiv \underbrace{\max}_{i} \sum_{j=1}^{n} \left|c_{ij}\right|. \tag{2.5}$$

Приведення системи (2.1) до виду (2.3) можна здійснити різними способами, важливо тільки, щоб виконувалася умова збіжності (2.4). Розглянемо один з них.

Якщо діагональні елементи матриці відмінні від нуля, тобто $a_{ii} \neq 0$, $i = \overline{1, n}$, то систему (2.1) можна записати у вигляді

$$\begin{cases} x_1 = \dfrac{1}{a_{11}}(-a_{12}x_2 - a_{13}x_3 - \ldots - a_{1n}x_n + b_1), \\ x_2 = \dfrac{1}{a_{22}}(-a_{21}x_1 - a_{23}x_3 - \ldots - a_{2n}x_n + b_2), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \dfrac{1}{a_{nn}}(-a_{n1}x_1 - a_{n3}x_3 - \ldots - a_{n,n-1}x_{n-1} + b_n). \end{cases} \quad . \qquad (2.6)$$

Застосування методу ітерацій до системи (2.6) ще називають **методом Якобі.** З (2.4) – (2.5) слідує, що для збіжності методу Якобі повинна виконуватись умова $|a_{ii}| > \sum\limits_{j=1}^{n} |a_{ij}| \quad \forall\ i = \overline{1, n}$.

2.3. Контрольні приклади

Приклад 1. Розв'язати методом Гауса систему рівнянь:

$$\begin{cases} 0{,}14x_1 + 0{,}24x_2 - 0{,}84x_3 = 1{,}11 \\ 1{,}07x_1 - 0{,}83x_2 + 0{,}56x_3 = 0{,}48 \\ 0{,}64x_1 + 0{,}43x_2 - 0{,}38x_3 = -0{,}83 \end{cases}$$

Ручний рахунок:

x_1	x_2	x_3	b
0,14	0,24	–0,84	1,11
1,07	–0,83	0,56	0,48
0,64	0,43	–0,38	–0,83
1	1,7143	–6	7,926
0	–2,6643	6,98	–8,0036
0	0,6672	–3,46	–5,9043
	1	–2,6198	3,004
	0	1,7121	–3,9
		1	–2,2779
	1	0	–2,9636
1			–0,6583

Розв'язок: $x_1 = -0{,}6583$, $x_2 = -2{,}9636$, $x_3 = -2{,}2779$.

Використання процедур пакета R.

У пакеті **R** є процедура *solve*, призначена для розв'язання системи лінійних рівнянь у матричному вигляді (2.2). Для її виклику необхідно визначити й задати матрицю A й вектор b, тоді розв'язком є $x = solve(A,b)$.

Задається матриця коефіцієнтів A і вектор вільних членів b:

```
> A = matrix(nrow=3,ncol=3,c(0.14,1.07,0.64,0.24,-0.83,
+                            0.43,-0.84,0.56,-0.38))
> A
     [,1]  [,2]  [,3]
[1,] 0.14  0.24 -0.84
[2,] 1.07 -0.83  0.56
[3,] 0.64  0.43 -0.38
> b = matrix(c(1.11,0.48,-0.83))
> b
      [,1]
[1,]  1.11
[2,]  0.48
[3,] -0.83
```

Розв'язання з використанням процедури *solve*:

```
> x = solve(A,b)
> x
          [,1]
[1,] -0.658156
[2,] -2.963664
[3,] -2.277882
```

Перевірка розв'язку:

```
> A%*%x
      [,1]
[1,]  1.11
[2,]  0.48
[3,] -0.83
```

Приклад 2. Необхідно побудувати площину у тривимірному просторі, що проходить через три точки з координатами $(10,5,3)$, $(1,7,5)$, $(-2,6,1)$.

Розв'язання. В загальному вигляді рівняння площини у тривимірному просторі має вид: $z = a\,x + b\,y + c$, де (x, y, z) – тримірні координати. Для побудови площини достатньо знайти параметри рівняння a, b, c. Оскільки, площина має проходити через задані точки, то параметри рівняння a, b, c повинні задовольняти системі рівнянь:

$$\begin{cases} 10\,a + 5\,b + c = 3 \\ 1a + 7\,b + c = 5 \\ -2\,a + 6\,b + c = 1 \end{cases}.$$

Реалізація в пакеті R:

```
> A = matrix(nrow=3,ncol=3,c(10, 1, -2, 5, 7, 6, 1, 1, 1))
> A
     [,1] [,2] [,3]
[1,]   10    5    1
[2,]    1    7    1
[3,]   -2    6    1
> b = c(3,5,1)
> b
[1] 3 5 1
> x = solve(A,b)
> x
[1]   0.4   2.8 -15.0
```

Розв'язок: $a = 0.4$, $b = 2.8$, $c = -15$, тобто рівняння площини має вигляд: $z = 0.4\,x + 2.8\,y - 15$.

Приклад 3. Розв'язати методом ітерацій систему рівнянь:

$$\begin{cases} 5x_1 + x_2 + 2x_3 = 10, \\ 6x_1 + 18x_2 + 6x_3 = 54, \\ 10x_1 + 20x_2 + 40x_3 = 160. \end{cases}$$

Розв'язання. Перепишемо систему у вигляді

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

Реалізація методу ітерацій у пакеті R:

```
> MetIterSLE = function(C,d,n,eps)
+ {
+  x0 = d
+  repeat
+   {
+   x = C%*%x0 + d
+   if(norm(x-x0) <= eps)
+    break
+   x0 = x
+   }
+  return(x)
+ }
```

Введення початкових даних і виклик запрограмованої процедури для отримання розв'язку:

```
> C = rbind(c(0,-1/5,-2/5), c(-1/3,0,-1/3), c(-1/4,-1/2,0))
> C
           [,1] [,2]       [,3]
[1,]  0.0000000 -0.2 -0.4000000
[2,] -0.3333333  0.0 -0.3333333
[3,] -0.2500000 -0.5  0.0000000
> d = c(2,3,4)
> d
[1] 2 3 4
> x = MetIterSLE(C,d,3, 0.0001)
> x
          [,1]
[1,] 0.4444534
[2,] 1.8666762
[3,] 2.9555659
```

Перевірка розв'язку:

```
> y = C%*%x + d
> y
          [,1]
[1,] 0.4444384
[2,] 1.8666602
[3,] 2.9555486
```

2.4. Порядок виконання роботи і варіанти завдань

2.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання систем лінійних алгебраїчних рівнянь;
- чисельні методи розв'язання систем лінійних алгебраїчних рівнянь.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремого модуля метод Гауса;
- запрограмувати у вигляді окремого модуля метод Якобі;
- привести тексти складених програм;
- розв'язати задачу з індивідуального завдання з застосуванням складених програм та засобів математичного пакету R.

2.4.2. Варіанти індивідуальних завдань

Варіант 1.
$$\begin{cases} 5,23x_1 + 0,57x_2 + 0,61x_3 + 0,48x_4 = 6,72 \\ 0,66x_1 + 7,04x_2 + 0,77x_3 + 0,36x_4 = -6,96 \\ 0,34x_1 + 0,82x_2 + 6,81x_3 + 0,18x_4 = 80,33 \\ 0,11x_1 + 0,16x_2 + 0,90x_3 + 8,33x_4 = 26,05 \end{cases}$$

Варіант 2.
$$\begin{cases} 0,70x_1 + 5,09x_2 + 0,89x_3 + 0,17x_4 = -4,43 \\ 0,32x_1 + 0,15x_2 + 9,11x_3 + 0,87x_4 = 8,10 \\ 0,85x_1 + 0,26x_2 + 0,19x_3 + 7,33x_4 = 12,73 \\ 0,21x_1 + 0,26x_2 + 0,95x_3 + 5,13x_4 = 16,06 \end{cases}$$

Варіант 3.
$$\begin{cases} 4,97x_1 + 0,07x_2 + 0,36x_3 + 0,81x_4 = -15,43 \\ 0,66x_1 + 7,12x_2 + 0,32x_3 + 0,77x_4 = 8,28 \\ 0,59x_1 + 0,16x_2 + 9,43x_3 + 0,36x_4 = 17,00 \\ 0,87x_1 + 0,91x_2 + 0,42x_3 + 8,15x_4 = 8,22 \end{cases}$$

Варіант 4.
$$\begin{cases} 7,70x_1 + 0,38x_2 + 0,43x_3 + 0,83x_4 = -13,79 \\ 0,97x_1 + 6,49x_2 + 0,14x_3 + 0,06x_4 = 17,35 \\ 0,81x_1 + 0,53x_2 + 9,13x_3 + 0,05x_4 = 12,41 \\ 0,19x_1 + 0,27x_2 + 0,38x_3 + 8,47x_4 = 14,58 \end{cases}$$

Варіант 5.
$$\begin{cases} 5,19x_1 + 0,17x_2 + 0,32x_3 + 0,57x_4 = -10,23 \\ 0,86x_1 + 7,49x_2 + 0,75x_3 + 0,96x_4 = 11,25 \\ 0,63x_1 + 0,42x_2 + 8,13x_3 + 0,52x_4 = 11,33 \\ 0,27x_1 + 0,65x_2 + 0,28x_3 + 6,99x_4 = 9,56 \end{cases}$$

Варіант 6.
$$\begin{cases} 6,17x_1 + 0,63x_2 + 0,46x_3 + 0,24x_4 = 4,12 \\ 0,43x_1 + 7,08x_2 + 0,83x_3 + 0,66x_4 = 9,72 \\ 0,34x_1 + 0,52x_2 + 5,44x_3 + 0,17x_4 = -5,88 \\ 0,71x_1 + 0,93x_2 + 0,65x_3 + 8,83x_4 = 14,25 \end{cases}$$

Варіант 7. $\begin{cases} 7{,}01x_1 + 0{,}93x_2 + 0{,}75x_3 + 0{,}53x_4 = -13 \\ 0{,}22x_1 + 3{,}67x_2 + 0{,}31x_3 + 0{,}57x_4 = 6 \\ 0{,}46x_1 + 0{,}67x_2 + 9{,}16x_3 + 0{,}69x_4 = 12 \\ 0{,}82x_1 + 0{,}37x_2 + 0{,}35x_3 + 7{,}24x_4 = 2 \end{cases}$

Варіант 8. $\begin{cases} 3{,}47x_1 + 0{,}37x_2 + 0{,}56x_3 + 0{,}72x_4 = -6{,}57 \\ 0{,}52x_1 + 5{,}88x_2 + 0{,}23x_3 + 0{,}40x_4 = 7{,}27 \\ 0{,}72x_1 + 0{,}63x_2 + 9{,}65x_3 + 0{,}75x_4 = 9{,}77 \\ 0{,}17x_1 + 0{,}97x_2 + 0{,}47x_3 + 8{,}12x_4 = 11{,}72 \end{cases}$

Варіант 9. $\begin{cases} 3{,}97x_1 + 0{,}17x_2 + 0{,}53x_3 + 0{,}68x_4 = -10{,}71 \\ 0{,}60x_1 + 8{,}71x_2 + 0{,}47x_3 + 0{,}93x_4 = 11{,}69 \\ 0{,}27x_1 + 0{,}59x_2 + 6{,}43x_3 + 0{,}63x_4 = 14{,}99 \\ 0{,}32x_1 + 0{,}51x_2 + 0{,}84x_3 + 8{,}77x_4 = 18{,}49 \end{cases}$

Варіант 10. $\begin{cases} 4{,}47x_1 + 0{,}93x_2 + 0{,}76x_3 + 0{,}52x_4 = -14{,}66 \\ 0{,}38x_1 + 6{,}63x_2 + 0{,}43x_3 + 0{,}61x_4 = 21{,}65 \\ 0{,}53x_1 + 0{,}76x_2 + 8{,}11x_3 + 0{,}27x_4 = 14{,}25 \\ 0{,}44x_1 + 0{,}68x_2 + 0{,}83x_3 + 7{,}09x_4 = 16{,}86 \end{cases}$

Варіант 11. $\begin{cases} 5{,}12x_1 + 0{,}82x_2 + 0{,}47x_3 + 0{,}68x_4 = -11{,}44 \\ 0{,}26x_1 + 6{,}49x_2 + 0{,}55x_3 + 0{,}37x_4 = 12{,}03 \\ 0{,}49x_1 + 0{,}42x_2 + 7{,}95x_3 + 0{,}95x_4 = 16{,}22 \\ 0{,}31x_1 + 0{,}59x_2 + 0{,}91x_3 + 8{,}55x_4 = 13{,}59 \end{cases}$

Варіант 12. $\begin{cases} 6{,}01x_1 + 0{,}22x_2 + 0{,}33x_3 + 0{,}44x_4 = 7{,}62 \\ 0{,}11x_1 + 5{,}42x_2 + 0{,}55x_3 + 0{,}66x_4 = -6{,}84 \\ 0{,}63x_1 + 0{,}43x_2 + 6{,}88x_3 + 0{,}77x_4 = 10{,}43 \\ 0{,}99x_1 + 0{,}34x_2 + 0{,}57x_3 + 8{,}13x_4 = 18{,}44 \end{cases}$

Варіант 13. $\begin{cases} 5{,}49x_1 + 0{,}35x_2 + 0{,}57x_3 + 0{,}79x_4 = -9{,}56 \\ 0{,}51x_1 + 7{,}82x_2 + 0{,}86x_3 + 0{,}64x_4 = 13{,}42 \\ 0{,}37x_1 + 0{,}38x_2 + 7{,}93x_3 + 0{,}42x_4 = 12{,}84 \\ 0{,}91x_1 + 0{,}53x_2 + 0{,}88x_3 + 0{,}02x_4 = 16{,}02 \end{cases}$

Варіант 14. $\begin{cases} 4,75x_1 + 0,49x_2 + 0,07x_3 + 0,33x_4 = -14,30 \\ 0,24x_1 + 5,68x_2 + 0,63x_3 + 0,74x_4 = 10,07 \\ 0,46x_1 + 0,98x_2 + 7,08x_3 + 0,92x_4 = 8,53 \\ 0,53x_1 + 0,55x_2 + 0,76x_3 + 9,13x_4 = 9,99 \end{cases}$

Варіант 15. $\begin{cases} 10,9x_1 + 1,2x_2 + 2,1x_3 + 0,9x_4 = -7 \\ 1,2x_1 + 11,2x_2 + 1,5x_3 + 2,5x_4 = 5,3 \\ 2,1x_1 + 1,5x_2 + 9,8x_3 + 1,3x_4 = 10,3 \\ 0,9x_1 + 2,5x_2 + 1,3x_3 + 12,1x_4 = 24,6 \end{cases}$

1.5. Контрольні запитання

1. Які є типи методів розв'язання систем лінійних алгебраїчних рівнянь?
2. У чому полягає суть методу Гауса з вибором головного елемента?
3. Які проблеми виникають при розв'язанні систем лінійних алгебраїчних рівнянь великої розмірності і як вони вирішуються?
4. У чому полягає ідея методу ітерації?
5. Чим метод Якобі відрізняється від методу ітерацій?
6. При яких умовах метод ітерації збігається? Якою при цьому є оцінка швидкості збіжності?
7. Які проблеми виникають при розв'язанні систем лінійних алгебраїчних рівнянь великої розмірності і як вони вирішуються?

Лабораторна робота № 3
Чисельні методи розв'язання нелінійних рівнянь з одним невідомим

3.1. Мета роботи

Вивчення чисельних методів розв'язання нелінійних рівнянь, придбання навичок використання цих методів для розв'язання задач пошуку коренів нелінійних рівнянь із застосуванням комп'ютера.

3.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі пошуку коренів нелінійного рівняння; *вміти* розв'язувати цю задачу з використанням чисельних методів [1 – 6].

Розглянемо рівняння виду:

$$f(x) = 0, \tag{3.1}$$

де $f(x)$ – нелінійна неперервна функція однієї змінної. Розв'язати рівняння – означає знайти таке x^*, при якому рівняння перетворюється в тотожність (x^* ще називають коренем рівняння). У загальному випадку рівняння може мати багато коренів. Розглянуті нижче чисельні методи розв'язання нелінійних рівнянь дозволяють знаходити один корінь на заданому відрізку $[a,b]$. При цьому на відрізку повинен існувати тільки один корінь.

Розглянемо кілька методів розв'язання нелінійного рівняння (3.1) на відрізку $[a,b]$.

Метод половинного ділення (метод дихотомії). При розв'язанні нелінійного рівняння методом половинного ділення задаються відрізок $[a,b]$, на якому існує тільки один розв'язок, і бажана точність $\varepsilon > 0$. На 0-й ітерації методу $[a_0,b_0] = [a,b]$, на k-й ітерації методу маємо поточний відрізок $[a_k,b_k]$. Далі визначається середина відрізка $x_k = \dfrac{a_k + b_k}{2}$ й перевіряється умова $f(a_k) \times f(x_k) < 0$. Якщо зазначена умова виконується, то $b_{k+1} = x_k$, $a_{k+1} = a_k$. Якщо умова не виконується, то $a_{k+1} = x_k$, $b_{k+1} = b_k$. Ділення відрізка навпіл припиняється при досягненні заданої точності, тобто $|b_k - a_k| \le \varepsilon$. Тут x_k є наближенням розв'язку на k-й ітерації методу. Очевидно, що

$$\left|x_{k+1} - x^*\right| \le \frac{1}{2}\left|x_k - x^*\right| \le \cdots \le \left(\frac{1}{2}\right)^k |b - a|,$$

тобто метод збігається з лінійною швидкістю з коефіцієнтом $q = 1/2$.

Метод хорд. При розв'язанні нелінійного рівняння методом хорд задаються відрізок $[a,b]$, на якому існує тільки один розв'язок, і точність ε.

На 0-й ітерації методу $[a_0, b_0] = [a, b]$, на k-й ітерації методу маємо поточний відрізок $[a_k, b_k]$. Потім через дві точки з координатами $(a_k, f(a_k))$ й $(b_k, f(b_k))$ проводимо відрізок прямої лінії (хорду) і визначаємо точку перетину цієї лінії з віссю абсцис (точка x_k). Якщо при цьому $f(a_k) \times f(x_k) < 0$, то праву межу інтервалу переносимо в точку x_k (тобто $b_{k+1} = x_k$, $a_{k+1} = a_k$). Якщо зазначена умова не виконується, то в точку x_k переноситься ліва межа інтервалу ($a_{k+1} = x_k$, $b_{k+1} = b_k$). Пошук розв'язку припиняється при досягненні заданої точності, тобто $|f(x_k)| < \varepsilon$. Точка перетину хорди з віссю абсцис визначається за формулою:

$$x_{k+1} = a_k + \left| \frac{f(a_k)}{f(b_k) - f(a_k)} \right| \times (b_k - a_k). \tag{3.2}$$

Метод дотичних (метод Ньютона). При розв'язанні нелінійного рівняння методом дотичних задаються початкове наближення x_0 і точність ε. Потім у точці $(x_0, f(x_0))$ проводиться дотична до графіка $f(x)$ й визначається точка x_1 перетину дотичної з віссю абсцис. У точці $(x_1, f(x_1))$ знову будується дотична, обчислюється наступне наближення шуканого розв'язку x_2 і т. д. Зазначена процедура повторюється доти, поки $|f(x_i)| > \varepsilon$. Точка перетину $(k+1)$-ої дотичної з віссю абсцис визначається за формулою:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{3.3}$$

Умова збіжності методу дотичних: $f(x_0) \times f''(x_0) > 0$. При цьому швидкість збіжності буде квадратичною: $\left| x_{k+1} - x^* \right| \le M \left| x_k - x^* \right|^2$ для всіх $k > k_0$, $k_0 > 0$, $M > 0$.

Метод простої ітерації. При розв'язанні нелінійного рівняння (3.1) методом ітерацій його потрібно записати у вигляді:

$$x = \phi(x). \tag{3.4}$$

Задаються початкове наближення x_0 й точність ε. Перше наближення розв'язку x_1 знаходиться з виразу $x_1 = \phi(x_0)$, друге – $x_2 = \phi(x_1)$ і т. д. У загальному випадку $k+1$ наближення обчислюється за формулою $x_{k+1} = \phi(x_k)$. Зазначена процедура припиняється при досягненні заданої точності, тобто $|f(x_k)| = |x_k - \phi(x_k)| \le \varepsilon$. Умова збіжності методу ітерацій: $|\phi'(x)| \le q < 1$ для всіх $x \in [a,b]$. При цьому швидкість збіжності буде лінійною: $|x_k - x^*| \le Mq^k$ для всіх $k > 0$, де $M > 0$.

3.3. Контрольні приклади

Приклад 1. Розв'язати методом простої ітерації рівняння $x^3 - 5x = 0$ з точністю $\varepsilon = 10^{-4}$.

Розв'язання.

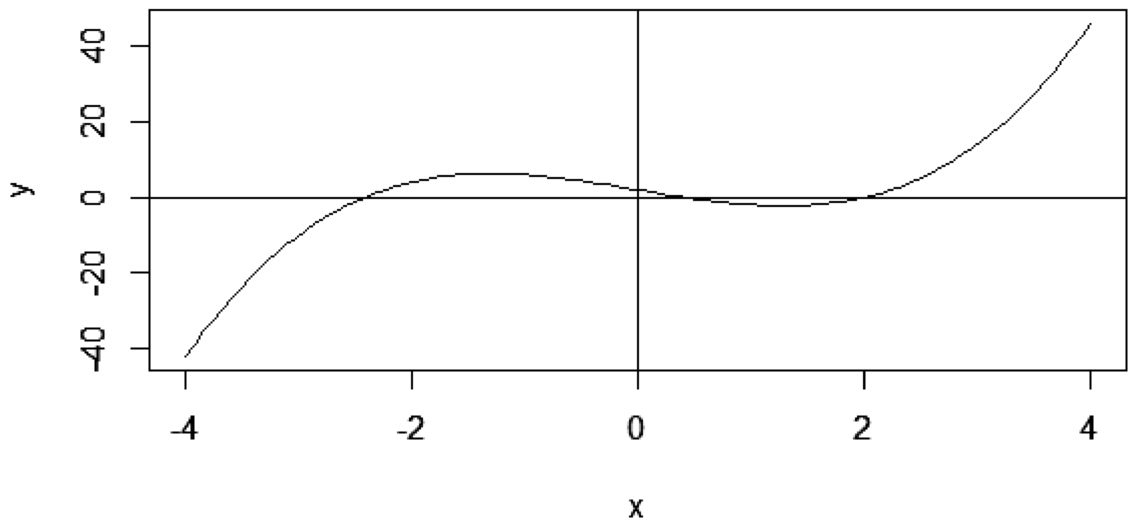
Спочатку визначимо відрізок, що містить розв'язок. Зробимо це графічно.

Вводимо функцію лівої частини рівняння:

```
> f = function(x)
+ {
+   return (x^3-5*x+2)
+ }
```

Будуємо графік заданої функції:

```
> x = seq(-4, 4, len=100)
> y = f(x)
> plot(x, y, type="l")
> abline(h=0)
> abline(v=0)
```



Із графіка видно, що коренів три. Знайдемо корінь на відрізку $[2, 4]$. Спробуємо застосувати метод простої ітерації:

а) розглянемо рівняння $x = \frac{x^3}{5}$, тобто $\phi(x) = \frac{x^3}{5}$. Тоді $\phi'(x) = \frac{3x^2}{5}$.

Але $\left|\frac{3x^2}{5}\right| \geq \frac{12}{5} = 2.4$ для всіх $x \in [2, 4]$, тобто $|\phi'(x)| \geq 2.4$, тобто умова збіжності методу простої ітерації не виконується;

б) розглянемо рівняння $x = \sqrt[3]{5x}$, тобто $\phi(x) = \sqrt[3]{5x}$. Тоді $\phi'(x) = \frac{5 \times (5x)^{-\frac{2}{3}}}{3} = \frac{5}{3(5x)^{\frac{2}{3}}}$, тому для всіх $x \in [2, 4]$ $|\phi'(x)| \leq \frac{5}{3 \times 10^{\frac{2}{3}}} < 0.36$, тобто умова збіжності методу простої ітерації виконується $(q = 0.36)$.

Візьмемо $x_0 = 4$, далі $x_1 = \sqrt[3]{5x_0}$, $x_2 = \sqrt[3]{5x_1}$ і т. д. до досягнення точності $|x_k - x_{k-1}| < \varepsilon$.

Вводимо функцію правої частини рівняння (3.4):

```
> g = function(x)
+ {
+   return ((5*x)^(1/3))
+ }
```

Запрограмуємо в пакеті R метод ітерацій для розв'язання нелінійних рівнянь. Процедура має такий вигляд:

```
> MetIter = function(x0,eps)
+ {
+  repeat
+   {
+   x = g(x0)
+   if(abs(x-x0) <= eps ) break
+   x0 = x
+   }
+  return(x)
+ }
```

Виклик процедури для розв'язання нелінійного рівняння:

```
> x = MetIter(4, 0.0001)
> x
[1] 2.23609
```

Перевірка розв'язку:

```
> y = g(x)-x
> y
[1] -1.468216e-05
```

Приклад 2. Траєкторія польоту снаряда в безповітряному просторі під дією тільки однієї сили тяжіння описується рівнянням (параболою) $y = xtg\theta_0 - \dfrac{gx^2}{2V_0^2\cos^2\theta_0}$, де V_0 – початкова швидкість у момент пострілу, θ_0 – кут відносно горизонту в момент пострілу (рад), *x, y* – відповідно абсциса і ордината в площині польоту снаряда, *g* – прискорення вільного падіння (9.81м/сек^2). Необхідно знайти, під яким кутом був зроблений постріл, якщо снаряд пролетів 3 600 м, а початкова швидкість снаряда дорівнювала 800 км/год.

Розв'язання.

Місце падіння снаряду має координати $(x_F, y_F) = (3600;0)$ тому для знаходження кута θ_0, під яким був зроблений постріл треба розв'язати нелінійне рівняння $y_F = x_F tg\theta_0 - \dfrac{gx_F^2}{2V_0^2\cos^2\theta_0}$. Перепишемо його у вигляді (3.1), тобто $f(\theta_0) = x_F tg\theta_0 - \dfrac{gx_F^2}{2V_0^2\cos^2\theta_0} - y_F = 0$, і розв'яжемо за допомогою процедури *uniroot()* пакета R:

```
> V0 = 800000/3600; xF = 3600; yF = 0; g = 9.81
> fun = function(Teta)
+    {xF*tan(Teta)-g*xF*xF/(2*V0*V0*cos(Teta)*cos(Teta))-yF}
> fun(0)
[1] -1287.268
> fun(pi/3)
[1] 1086.31
> rc = uniroot(fun, lower=0, upper=pi/3, tol=1e-9)
> rc$root
[1] 0.3984186
> fun(rc$root)
[1] 3.751666e-11
```

Відповідь. Постріл було зроблено під кутом 0.3984186 радіан або близько 23 градусів.

3.4. Порядок виконання роботи і варіанти завдань

3.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання нелінійного рівняння з однією змінною;
- чисельні методи розв'язання нелінійного рівняння з однією змінною.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів методи половинного ділення, хорд, дотичних і простої ітерації;
- привести тексти складених програм;
- розв'язати задачу з індивідуального завдання з застосуванням складених програм та засобів математичного пакета R;
- порівняти трудомісткість і швидкість збіжності методів.

3.4.2. Варіанти індивідуальних завдань

Варіант	**Рівняння**
1	$x^2 = \exp(-x^2) - 1$
2	$x = \cos(x)$
3	$x = x^2 - 1$
4	$x = 2\exp(-x)$
5	$x = 3\cos(x)$
6	$x = 2\exp(-x)$
7	$x = tg(2x) - 1$
8	$x = \ln(x) + 2$
9	$x = \cos(2x)$
10	$x = \exp(-3x)$
11	$x = \exp(-3x) + 1$
12	$x = \exp(-x^2)$
13	$x = \exp(-3x^2)$
14	$x^2 = \exp(-x^2)$
15	$x = tg(x) - 2$

3.5. Контрольні запитання

1. Сформулюйте постановку задачі розв'язання рівнянь з одним невідомим.
2. У чому полягає суть методу дихотомії? Яка у нього швидкість збіжності?
3. У чому полягає суть методу хорд? Яка у нього швидкість збіжності?
4. У чому полягає суть методу Ньютона? Яка у нього швидкість збіжності?
5. У чому полягає суть методу простої ітерації? Яка у нього швидкість збіжності?

Лабораторна робота № 4
Чисельні методи розв'язання систем нелінійних рівнянь

4.1. Мета роботи

Вивчення чисельних методів розв'язання систем нелінійних рівнянь, придбання навичок використання цих методів для пошуку розв'язків систем нелінійних рівнянь із застосуванням комп'ютера.

4.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі розв'язання системи нелінійних рівнянь; *вміти* розв'язувати цю задачу з використанням чисельних методів [1 – 6].

Розглянемо систему нелінійних рівнянь виду:

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 \\ f_2(x_1, x_2, ..., x_n) = 0 \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ f_n(x_1, x_2, ..., x_n) = 0 \end{cases}, \tag{4.1}$$

де $f_i(x_1, x_2, ..., x_n)$, $i = \overline{1,n}$ – деякі нелінійні функції n змінних. Якщо ввести позначення $x = (x_i)_{i=1}^{n}$ – вектор-стовпець розмірності n з елементами x_i, $F(x) = \begin{pmatrix} f_1(x) \\ \dots \\ f_n(x) \end{pmatrix}$ – векторна функція розмірності n, елементами якої є функції $f_i(x) = f_i(x_1, x_2, ..., x_n)$, $i = \overline{1,n}$, то систему (4.1) можна записати у векторному вигляді:

$$F(x) = 0, \tag{4.2}$$

Розв'язати систему (4.2) – означає знайти таке $x^* \in R^n$, для якого $F(x^*) \equiv 0$, тобто $f_i(x_1^*, x_2^*, ..., x_n^*) \equiv 0 \;\; \forall i = \overline{1,n}$.

Розглянемо три основні методи розв'язання систем нелінійних рівнянь виду (4.2).

Метод Ньютона. Метод будує ітераційну послідовність $\{x^k\}$, $k = 0, 1, 2, \dots$ $(x^k \in R^n)$ наближень розв'язків x^* (початкове наближення x^0 задається) за такою ітераційною формулою:

$$x^{k+1} = x^k - \left[F'(x^k)\right]^{-1} F(x^k). \tag{4.3}$$

Тут $F'(x) = \begin{pmatrix} \frac{\partial f_1(x)}{\partial x_1} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n(x)}{\partial x_1} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{pmatrix}$, тобто $F'(x^k)$ – матриця розмірності $n \times n$. Ітераційний процес (4.3) триває доти, поки не виконається умова $\left\|F(x^k)\right\| \le \varepsilon$, де ε – задана точність розв'язання задачі (4.2).

Якщо послідовність $\{x^k\}$, $k = 0, 1, 2, \dots$ збігається, то швидкість її збіжності квадратична, тобто $\left\|x^{k+1} - x^*\right\| \le M \left\|x^k - x^*\right\|^2$ починаючи з якогось k. Тут M – деяка додатна константа [1; 2; 6].

Основним недоліком методу Ньютона є те, що він збігається тільки для достатньо близьких до розв'язку початкових наближень x^0.

Метод ітерацій. При розв'язанні системи нелінійних рівнянь (4.2) методом ітерацій її потрібно записати у вигляді $x = \Phi(x)$. Тут $\Phi(x)$ – векторна функція розмірності n від $x \in R^n$. Задаються початкове наближення x^0 й точність ε. Метод будує ітераційну послідовність $\{x^k\}$, $k = 0, 1, 2, \ldots$, наближень розв'язків x^* за такою ітераційною формулою:

$$x^{k+1} = \Phi(x^k). \tag{4.4}$$

Метод ітерацій збігається, якщо $\|\Phi'(x)\| \le q < 1$ для всіх x приналежних деякому околі $V(x^*)$ розв'язку x^* й $x^0 \in V(x^*)$. При цьому швидкість збіжності буде лінійною, тобто $\|x^{k+1} - x^*\| \le q\|x^k - x^*\|$ починаючи з якогось k.

Метод найменших квадратів. Для розв'язання системи нелінійних рівнянь (4.2) методом найменших квадратів треба ввести функцію виду

$$\Phi(x) \equiv \|F(x)\|^2 = \sum_{i=1}^{n} f_i^2(x). \tag{4.5}$$

Тоді розв'язком задачі (4.2) буде точка, в якій функція $\Phi(x)$ досягає мінімального значення.

4.3. Контрольний приклад

Приклад 1. Розв'язати методом Ньютона систему нелінійних рівнянь $\begin{cases} 2x_1^2 + x_2^2 - 1 = 0 \\ x_1^3 + 6x_1^2 x_2 - 1 = 0 \end{cases}$ з точністю $\varepsilon = 10^{-4}$ (початкове наближення $x_0 = (0.65, 0.35)^T$).

Розв'язання.

Для розв'язання поставленої задачі реалізуємо метод Ньютона у пакеті R:

```
> Newton = function(Fx, FFx, x, eps)
+ {
+   y = Fx(x)
+   while ((sqrt(t(y)%*%y)) > eps)
+   {
+     yy = FFx(x)
+     x = x - solve(yy)%*%y
+     y = Fx(x)
+   }
+   return(x)
+ }
```

Введення початкових даних:

```
> # Вводимо вектор початкового наближення розв'язку
> x = c(0.65, 0.35)

> # Вводимо векторну функцію F
> Fx = function(x)
+ {
+   f = c(1:2)
+   f[1] = 2*(x[1])^2+(x[2])^2-1
+   f[2] = (x[1])^3+(6*(x[1])^2)*x[2]-1
+   return(f)
+ }

> # Вводимо матрицю частинних похідних
> FFx = function (x)
+ {
+   ff = matrix(nrow=2, ncol=2)
+   ff[1,1] = 4*x[1]
+   ff[1,2] = 2*x[2]
+   ff[2,1] = 3*(x[1]^2)+12*x[1]*x[2]
+   ff[2,2] = 6*(x[1]^2)
+   return(ff)
+ }
```

Виклик запрограмованої процедури для отримання розв'язку і перевірка правильності розв'язку:

```
> X = Newton(Fx, FFx, x, 0.00001)
> X
          [,1]
[1,] 0.6865944
[2,] 0.2391155
> Fx(X)
[1]  1.176896e-10 -2.472814e-10
```

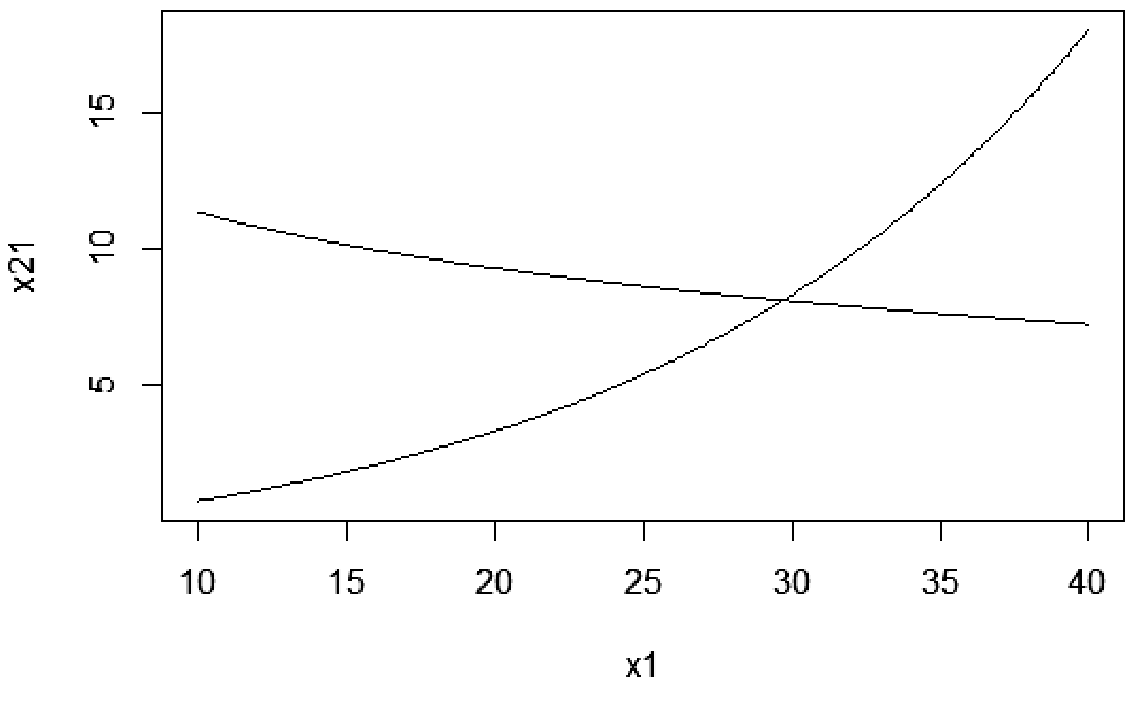
Приклад 2. Пропозиція на деякий товар описується функцією $S = \exp\left(\frac{P+5}{15}\right) - 2$, а попит на цей товар описується функцією $D = -3\ln\left(\frac{P}{3}\right) + 15$, де S – величина пропозиції товару, D – величина попиту товару, P – ціна товару. Знайдіть рівноважну ціну та кількість товару, при якій встановиться рівноважна ціна. (Рівноважна ціна – це ціна при якій попит дорівнює пропозиції).

Позначимо через x_1 ціну, а через x_2 кількість товару. Тоді систему перепишемо так:

$$\begin{cases} \exp\left(\frac{x_1+5}{15}\right) - 2 - x_2 = 0, \\ -3\ln\left(\frac{x_1}{3}\right) + 15 - x_2 = 0 \end{cases}.$$

Для вибору початкового наближення побудуємо графік:

```
> x = seq(10, 40, 0.1)
> y1 = exp((x+5)/15)-2
> y2 = -3*log(x/3)+15
> plot(x, y1, type="l")
> lines(x, y2, type="l")
```



Задамо початкове наближення таким: $x_0 = (28, 10)^T$.

Введемо початкові дані задачі:

```
> # Вводимо вектор початкового наближення розв'язку
> x = c(28, 10)

> # Вводимо векторну функцію F
> Fx = function(x)
+ {
+   f = c(1:2)
+   f[1] = exp((x[1]+5)/15)-2-x[2]
+   f[2] = -3*log(x[1]/3)+15-x[2]
+   return(f)
+ }
> # Вводимо матрицю частинних похідних
> FFx = function (x)
+ {
+   ff = matrix(nrow=2, ncol=2)
+   ff[1,1] = exp((x[1]+5)/15)*(1/15)
+   ff[1,2] = -1
+   ff[2,1] = -9/x[1]
+   ff[2,2] = -1
+   return(ff)
+ }
```

Скористаємося записаною в прикладі 1 процедурою і перевіримо отриманий результат:

```
> X = Newton(Fx, FFx, x, 0.00001)
> X
          [,1]
[1,] 29.718483
[2,]  8.120525
>
> Fx(X)
[1] 9.489298e-12 4.145522e-06
```

Таким чином, при ціні товару 8,12 грошових одиниць попит і пропозиція будуть майже однакові і становитимуть 29 або 30 одиниць товару.

4.4. Порядок виконання роботи і варіанти завдань

4.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання системи нелінійних рівнянь;
- чисельні методи розв'язання системи нелінійних рівнянь.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів методи Ньютона, ітерацій та найменших квадратів;
- привести тексти складених програм;
- розв'язати задачу з індивідуального завдання з застосуванням складених програм та засобів математичного пакету R;
- порівняти трудомісткість і швидкість збіжності методів.

4.4.2. Варіанти індивідуальних завдань

Варіант	Система рівнянь	Початкове наближення
1	2	3
1	$\begin{cases} \ln\left(1+\dfrac{x_1+x_2}{5}\right)-\sin\left(\dfrac{x_2}{3}\right)-x_1+1.1=0, \\ \cos\left(\dfrac{x_1x_2}{6}\right)-x_2+0.5=0. \end{cases}$	$x^0=(1;1)$
2	$\begin{cases} \lg\left(\dfrac{x_2}{x_3}\right)-x_1+1=0, \\ 2x_1^2+x_2-x_3-0.4=0, \\ \dfrac{x_1x_2}{20}-x_3+2=0. \end{cases}$	$x^0=(1;2.2;2)$
3	$\begin{cases} x_1+x_1^2-2x_2x_3-0.1=0, \\ x_2-x_2^2+3x_1x_3+0.2=0, \\ x_3+x_3^2+2x_1x_2-0.3=0. \end{cases}$	$x^0=(0;0;0)$
4	$\begin{cases} x_1^3+x_2^3-6x_1+3=0, \\ x_1^3-x_2^3-6x_2+2=0. \end{cases}$	$x^0=(0.5;0.5)$
5	$\begin{cases} x_1+3\lg x_1-x_2^2=0, \\ 2x_1^2-x_1x_2-5x_1+1=0. \end{cases}$	$x^0=(3.4;2.2)$
6	$\begin{cases} 5x_1-6x_2+20\lg x_1+16=0, \\ 2x_1+x_2-10\lg x_2-4=0. \end{cases}$	$x^0=(0;0)$

1	2	3
7	$\begin{cases} 0.5\sin\left(\dfrac{x_2}{3}\right) - x_1 + 1 = 0, \\ 0.3\cos x_1 - x_2 = 0. \end{cases}$	$x^0 = (1; 0)$
8	$\begin{cases} x_1^2 - x_2^2 - 1 = 0, \\ x_2(x_2 - 1) - 1 = 0. \end{cases}$	$x^0 = (2; 10)$
9	$\begin{cases} 2x_1 - x_2 - 6\lg x_1 - 3 = 0, \\ 15x_1 - 10x_2 - 60\lg x_2 - 6 = 0. \end{cases}$	$x^0 = (0; 0)$
10	$\begin{cases} x_1^2 - x_2^2 = 0, \\ x_1 x_2 - 4 = 0. \end{cases}$	$x^0 = (0; 0)$
11	$\begin{cases} x_1 - x_2 - 6\lg x_1 - 1 = 0, \\ x_1 - 3x_2 - 6\lg x_2 - 2 = 0. \end{cases}$	$x^0 = (0.5; 0.2)$
12	$\begin{cases} 2x_1^3 - x_2^2 - 1 = 0, \\ x_1 x_2^3 - x_2 - 4 = 0. \end{cases}$	$x^0 = (0; 0)$
13	$\begin{cases} x_1 + x_2 + x_3 - 13 = 0, \\ x_1^2 + x_2^2 + x_3^2 - 91 = 0, \\ x_2^2 - x_1 x_3 = 0. \end{cases}$	$x^0 = (0; 0)$
14	$\begin{cases} \sin x_1 + 2x_2 - 2 = 0, \\ \cos x_1 + x_2 - 1.5 = 0. \end{cases}$	$x^0 = (0; 0)$
15	$\begin{cases} \cos(x_1 + 0.5) - x_2 - 2 = 0, \\ \sin x_2 - 2x_1 - 1 = 0. \end{cases}$	$x^0 = (0; 0)$

4.5. Контрольні запитання

1. Сформулюйте постановку задачі розв'язання системи нелінійних рівнянь.
2. У чому полягає суть методу Ньютона? Яка у нього швидкість збіжності?
3. У чому полягає суть методу ітерацій? Яка у нього швидкість збіжності?
4. У чому полягає суть методу найменших квадратів?

Лабораторна робота № 5
Чисельні методи знаходження власних значень і власних векторів матриці

5.1. Мета роботи

Вивчити методи знаходження власних чисел матриці. Навчитися застосовувати ці методи для розв'язання різних математичних задач. Засвоїти програмні засоби пакета R, призначені для знаходження власних чисел і власних векторів квадратної матриці.

5.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* поняття власних чисел (значень) і власних векторів квадратної матриці; *вміти* розв'язувати задачу знаходження найбільшого власного значення матриці методом ітерацій [1 – 6].

Метод ітерацій. Метод ітерацій застосовується для пошуку найбільшого за модулем власного значення квадратної матриці.

Нехай $A \in R^{n \times n}$. Тоді алгоритм методу ітерацій такий:

1) задається початковий вектор $z^0 \in R^n$ (початкове наближення для власного вектора матриці A, що відповідає найбільшому за модулем власному значенню матриці A);

2) k-та ітерація ($k = 0, 1, 2, \ldots$). На ній маємо поточний вектор $z^k \in R^n$. Далі проводиться нормування вектора z^k, а саме обчислюється число $\lambda_{\max}^k$ таке, що $\left|\lambda_{\max}^k\right| = \max\limits_{i=\overline{1,n}} \left|z_i^k\right|$, і вектор $y^k = \dfrac{z^k}{\lambda_{\max}^k}$, і після чого визначається наступний вектор $z^{k+1} = Ay^k$;

3) обчислення припиняються, коли виконується умова $\dfrac{\left|\lambda_{\max}^{k+1} - \lambda_{\max}^k\right|}{\left|\lambda_{\max}^{k+1}\right|} \le \varepsilon$, де ε – задана відносна точність обчислень.

У пакеті R є процедура *eigen*, призначена для знаходження власних чисел та власних векторів матриці. Для її виклику необхідно визначити й задати матрицю A і виконати команду $eigen\,(A)$.

5.3. Контрольний приклад

Приклад. Досліджується тривісний напружений стан елементів тіла, представлений на рис. 5.1.

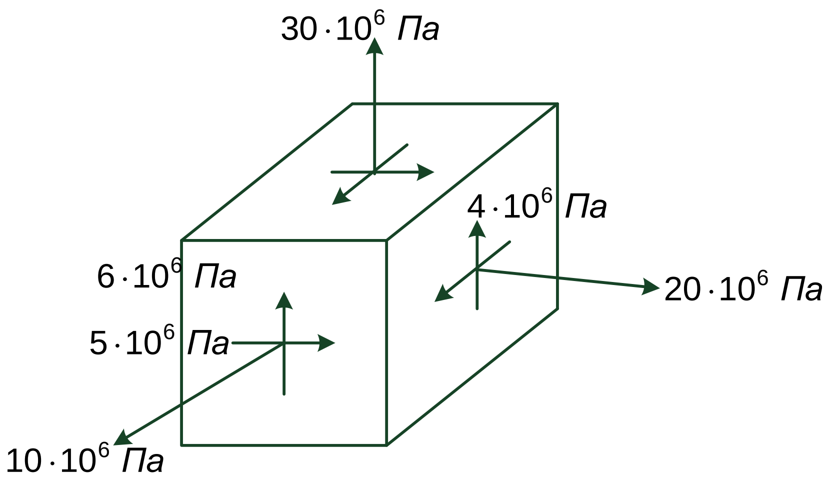


Рис. 5.1. **Тривісний напружений стан елементів тіла**

Матриця напруги для нього має вигляд:

$$\begin{bmatrix} 10 & 5 & 6 \\ 5 & 20 & 4 \\ 6 & 4 & 30 \end{bmatrix} \times 10^6 \; \frac{H}{м^2}.$$

Якщо виходити з того, що руйнування відбудеться при максимальній напрузі, то необхідно знати величину найбільшої головної напруги, яка відповідає найбільшому власному значенню матриці напруги.

Для знаходження величини цієї напруги можна скористатися методом ітерацій. Ітераційну процедуру здійснювати до тих пір, поки різниця між власними значеннями, обчисленими в послідовних ітераціях, не стане менше 0,01 %.

Відповідь: $\lambda_{\max} = 0.33712 \times 10^8$, $z^T = (0.34091, 0.41636, 1.0)$.

5.4. Порядок виконання роботи і варіанти завдань

5.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі пошуку власних значень і відповідних власних векторів квадратної матриці;
- чисельні методи пошуку власних значень і відповідних власних векторів квадратної матриці.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремого модуля метод ітерацій для визначення найбільшого (за модулем) власного значення і відповідного власного вектора квадратної матриці;
- привести текст складеної програми;
- розв'язати задачу з прикладу у розділі 5.3 з використанням складеної програми;
- знайти найбільше і найменше (за модулем) власні значення квадратної матриці з індивідуального завдання з застосуванням складеної програми та засобів математичного пакета R.

5.4.2. Варіанти індивідуальних завдань

Матрицю взяти з системи лінійних алгебраїчних рівнянь з індивідуального завдання до лабораторної роботи 1.

5.5. Контрольні запитання

1. Що називається власними значеннями і власними векторами матриці?
2. Для яких матриць вводяться поняття власних значень і власних векторів?
3. Які ви знаєте чисельні методи визначення власних значень і власних векторів матриць?
4. Наведіть приклади практичних задач, що зводяться до визначення найбільшого власного значення і відповідного власного вектора квадратної матриці.

Лабораторна робота № 6
Чисельні методи наближення функцій. Апроксимація та інтерполяція функцій

6.1. Мета роботи

Вивчення методу найменших квадратів (МНК) для практичного розв'язання задач апроксимації функцій. Вивчення методів інтерполяції для практичного розв'язання задачі наближення функцій. Придбання навичок використання вказаних методів для розв'язання задачі наближення функцій із застосуванням комп'ютера.

6.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі наближення (апроксимації) функцій й, зокрема, задачі інтерполяції; *вміти* розв'язувати задачу апроксимації функцій методом найменших квадратів [1 – 6]; *вміти* розв'язувати задачу наближення функцій за допомогою кусково-лінійної й кусково-квадратичної інтерполяції й полінома Лагранжа [1 – 6]*; вміти* застосовувати процедури пакета R для розв'язання задачі наближення функцій.

6.2.1. Апроксимація функцій.

Задача апроксимації функцій у загальному вигляді формулюється наступним чином.

Нехай є деяка функція $f(x)$, $f: R^1 \to R^1$, про яку відомо, що в n точках $x_1, x_2, \ldots, x_n$ вона приймає, відповідно, значення $y_1, y_2, \ldots, y_n$, тобто $y_i = f(x_i)$, $i = \overline{1,n}$. Потрібно відновити її значення при інших значеннях $x \in [x_1, x_n]$.

Функція $y = f(x)$ може бути як невідомою, тобто її значення тільки вимірюються, а може бути й просто дуже складною для обчислень. У цих випадках функцію $f(x)$ намагаються замінити більш простою функцією $g(x,a)$, близькою до $f(x)$, тобто

$$f(x) \approx g(x,a), \tag{6.1}$$

де $a \in R^k$, $a = (a_1, a_2, \ldots, a_k)^T$ – деякі параметри функції g, вид якої відомий. Процес наближення однієї функції іншою називають **апроксимацією**, а функцію g при цьому називають **апроксимуючою** для f.

Найбільш відомим й ефективним з методів розв'язання задачі апроксимації функцій є **метод найменших квадратів**, суть якого полягає в наступному. Вводиться функція (від a) виду

$$\Phi(a) \equiv \sum_{i=1}^{n} \left[y_i - g(x_i, a)\right]^2, \tag{6.2}$$

яку можна розглядати як міру відхилення функції $g(x,a)$ (по аргументу x) від функції $f(x)$ по сукупності точок $x_1, x_2, \ldots, x_n$. Тоді параметри a функції $g(x,a)$ можна визначити з умови найменшого відхилення $g(x,a)$ від $f(x)$, тобто параметри a визначаються як точка, у якій функція $\Phi(a)$ досягає по $a \in R^k$ мінімального значення (точка мінімуму). Нехай a^* – шукані параметри. При цьому величини $r_i \equiv y_i - g(x_i, a^*)$, $i = \overline{1,n}$, називаються **залишками** (**залишковими відхиленнями**) й використовуються для аналізу отриманого розв'язку задачі апроксимації.

У випадках, коли функція $g(x,a)$ є лінійною за параметрами a точку мінімуму функції $\Phi(a)$ можна знайти аналітично з необхідної умови мінімуму 1-го порядку $\Phi'(a) = 0$, тобто:

$$\frac{\partial \Phi}{\partial a_j} = 0, \forall j = \overline{1,k}. \tag{6.3}$$

Розглянемо загальний випадок, коли функція $g(x,a)$ лінійна за параметрами a, тобто має вигляд:

$$g(x,a) = \sum_{m=1}^{k} a_m \psi_m(x), \tag{6.4}$$

де $\psi_m(x)$, $m=1,\ldots,k$ – деякі відомі функції. Тоді з (6.2) – (6.4) одержуємо

$$\forall j=\overline{1,k}:\ \frac{\partial\Phi}{\partial a_j}=-2\sum_{i=1}^{n}\left[y_i-g(x_i,a)\right]\psi_j(x_i)=-2\sum_{i=1}^{n}\left[y_i-\sum_{m=1}^{k}a_m\psi_m(x_i)\right]\psi_j(x_i)=0\,.$$

Звідки для $\forall j=\overline{1,k}$:

$$\sum_{m=1}^{k}a_m\left[\sum_{i=1}^{n}\psi_m(x_i)\psi_j(x_i)\right]=\sum_{i=1}^{n}y_i\,\psi_j(x_i)\,. \qquad (6.5)$$

Таким чином, параметри a можуть бути знайдені як розв'язок лінійної системи алгебраїчних рівнянь (6.5), яку можна записати в матричному вигляді:

$$Ca=b\,, \qquad (6.6)$$

де $c_{jm}=\sum_{i=1}^{n}\psi_m(x_i)\psi_j(x_i)$, , $j=\overline{1,k}$, $m=\overline{1,k}$, $b_j=\sum_{i=1}^{n}y_i\,\psi_j(x_i)$, $j=\overline{1,k}$.

Розглянемо ще випадок, коли апроксимуюча функція є поліномом заданої $(k-1)$-ї степені. Тоді вона, у загальному випадку, може бути записана у вигляді $g(x,a)=b_0+b_1x+b_2x^2+\cdots+b_{k-1}x^{k-1}$, де $a\in R^k$, $a_1=b_0$, $a_2=b_1,\ldots$, $a_k=b_{k-1}$, b_j – коефіцієнти полінома. При цьому функція $g(x,a)$ є лінійною за параметрами a і має вигляд (6.4), де $\psi_1(x)=1$, $\psi_2(x)=x,\ldots$, $\psi_k(x)=x^{k-1}$, тобто $\psi_m(x)=x^{m-1}$.

Тоді параметри a можна знайти, розв'язавши лінійну систему алгебраїчних рівнянь (6.6), з елементами:

$$c_{jm}=\sum_{i=1}^{n}x_i^{\,m-1}x_i^{\,j-1}=\sum_{i=1}^{n}x_i^{\,m+j-2},\ j=\overline{1,k},\ m=\overline{1,k}, \qquad (6.7)$$

$$b_j=\sum_{i=1}^{n}y_i x_i^{\,j-1},\ j=\overline{1,k}\,. \qquad (6.8)$$

У пакеті R є процедура *nlm*, призначена для знаходження точки мінімуму довільної функції $\varphi(z)$, $z\in R^k$. Для її виклику необхідно визначити функцію $\varphi(z)$ й задати початкове наближення $z^0\in R^k$, тоді розв'язком є $z^*=nlm(\varphi,z^0)$.

У випадку, коли апроксимуюча функція лінійна за параметрами, система (6.5) може бути розв'язана за допомогою процедури *solve*, описаної у рекомендаціях до лабораторної роботи № 1.

6.2.2. Інтерполяція функцій.

Загальна задача апроксимації функцій була сформульована в розділі 6.2.1. Якщо при цьому параметри $a_1, a_2, \dots, a_k$ апроксимуючої функції $g(x,a)$ визначаються з умови збігу вихідної функції $f(x)$ й функції $g(x,a)$ в точках $x_1, x_2, \dots, x_n$, тобто $g(x_i, a) = f(x_i)$, $\forall\ i = \overline{1,n}$, то такий спосіб апроксимації (наближення) називається **інтерполяцією.**

Кусково-лінійна інтерполяція. Такий вид інтерполяції полягає в тому, що на кожному окремому інтервалі (x_i, x_{i+1}) функція $f(x)$ наближається лінійною функцією $g_i(x)$, що проходить через дві точки (x_i, y_i), (x_{i+1}, y_{i+1}). Неважко перевірити, що така лінійна функція може бути також записана у вигляді:

$$g_i(x) = \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \times y_i + \frac{(x - x_i)}{(x_{i+1} - x_i)} \times y_{i+1}. \tag{6.9}$$

Таким чином, при використанні кусково-лінійної інтерполяції для таблично заданої функції, при обчисленні наближеного значення функції $f(x)$ у деякій точці $\hat{x} \in (x_1, x_n)$ дотримуються наступного алгоритму. Спочатку знаходять інтервал (x_i, x_{i+1}), якому точка $\hat{x}$ належить, а потім обчислюють наближене значення функції $f(x)$ в точці $\hat{x}$ за формулою $f(\hat{x}) \approx g_i(\hat{x})$.

Якщо $f(x)$ двічі неперервно-диференційовна на відрізку $[x_1, x_n]$, то оцінка похибки при кусково-лінійній інтерполяції має вигляд [1 – 3, 5, 6]:

$$|f(x) - g(x)| \le \frac{M_2 h^2}{2},\ \forall x \in (x_1, x_n),$$

де $M_2 = \max\limits_{\xi \in (x_1, x_n)} |f''(\xi)|$, $h = \max\limits_{i = \overline{1,n}} |x_{i+1} - x_i|$.

Кусково-квадратична інтерполяція. Подібно кусково-лінійній інтерполяції можна використати для наближення $f(x)$ на кожному окремому інтервалі (x_i, x_{i+1}) й поліноми більш високого порядку, наприклад, квадратичну функцію $g_i(x)$, яка проходить вже через три точки (x_i, y_i), (x_{i+1}, y_{i+1}), (x_{i+2}, y_{i+2}). Неважко перевірити, що така квадратична функція може бути записана у вигляді:

$$g_i(x) = \frac{(x - x_{i+1})(x - x_{i+2})}{(x_i - x_{i+1})(x_i - x_{i+2})} \times y_i + \frac{(x - x_i)(x - x_{i+2})}{(x_{i+1} - x_i)(x_{i+1} - x_{i+2})} \times y_{i+1} + \\ + \frac{(x - x_i)(x - x_{i+1})}{(x_{i+2} - x_i)(x_{i+2} - x_{i+1})} \times y_{i+2} \,. \qquad (6.10)$$

Якщо $f(x)$ тричі неперервно-диференційовна на відрізку $[x_1, x_n]$, то оцінка похибки при кусково-квадратичній інтерполяції має вигляд [1 – 3; 5; 6]:

$$|f(x) - g(x)| \le \frac{M_3 h^3}{6}, \ \forall x \in (x_1, x_n),$$

де $M_3 = \max\limits_{\xi \in (x_1, x_n)} \left| f^{(3)}(\xi) \right|$, $h = \max\limits_{i = \overline{1,n}} |x_{i+1} - x_i|$.

Інтерполяційний многочлен Лагранжа. Якщо відомі значення функції $f(x)$ в n точках, то поліном мінімальної степені, що інтерполює $f(x)$, буде мати степінь $n-1$. Він називається **інтерполяційним многочленом Лагранжа** й має вигляд:

$$L_{n-1}(x) = \sum_{i=1}^{n} \left[\prod_{\substack{j=1 \\ j \ne i}}^{n} \left(\frac{x - x_j}{x_i - x_j} \right) \right] y_i \,. \qquad (6.11)$$

Неважко перевірити, що $L_{n-1}(x_i) = y_i$ для всіх $i = \overline{1,n}$.

Похибка наближення інтерпольованої функції многочленом Лагранжа сильно залежить від розкиду точок x_i, тобто чим менший інтервал (x_1, x_n), тим точніше $L_{n-1}(x)$ наближає значення функції $f(x)$ в точках $x \in (x_1, x_n)$.

Якщо функція $f(x)$ n разів неперервно-диференційовна на відрізку (x_1, x_n). Тоді для всіх $x \in (x_1, x_n)$:

$$|f(x) - L_{n-1}(x)| \le \frac{M_n h^n}{n!},$$

де $M_n = \max\limits_{\xi \in (x_1, x_n)} \left| f^{(n)}(\xi) \right|$, $h = \max\limits_{i = \overline{1,n}} |x_{i+1} - x_i|$ [1 – 3; 5; 6].

6.3. Контрольний приклад

Приклад 1. Залежність (HQ-характеристика насоса) створюваного насосом напору (H) від витрати води (Q) описується функцією $H = H_F - S_F Q^2$, де H_F, S_F – параметри залежності. У результаті проведення натурних експериментів з насосом отримані такі дані:

Q, м^3/год.	5	20	30	35
H, м вод. ст.	209	194	165	142

Треба побудувати графік HQ-характеристики насоса та оцінити значення напору, який буде створювати насос при витраті води $Q = 32$ м^3/год.

Розв'язання 1. HQ-характеристика насоса задана таблицею даних, при цьому функція $H(Q) = H_F - S_F Q^2$ є апроксимуючою для неї. Використаємо метод найменших квадратів для оцінки параметрів H_F, S_F апроксимуючої функції.

Використання процедури *optim* **пакета R:**

Вводимо початкові дані:

```
> n = 4
> x = c(5, 20, 30, 35)
> y = c(209, 194, 165, 142)
>
> # Задаємо вид апроксимуючої функції
> FunModel = function(x,a)
+ {
+  return( a[1]-a[2]*x^2 )
+ }
```

Визначаємо цільову функцію методу МНК:

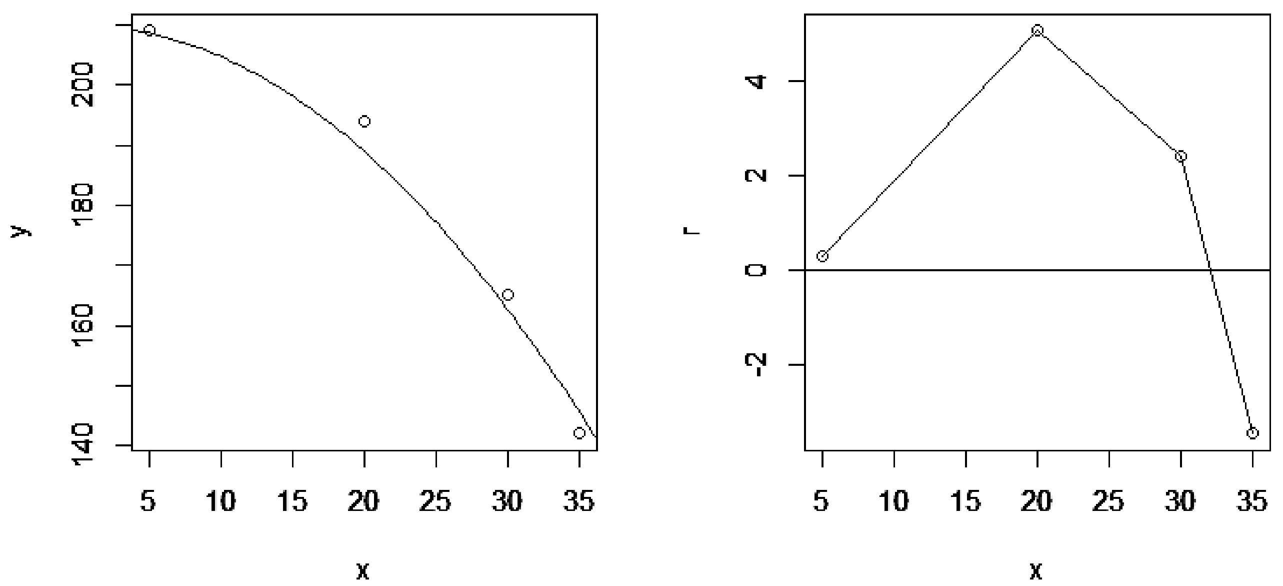
```
> FunMNK = function(a)
+ {
+  S = 0
+  for(i in 1:n)
+   S = S + (y[i]-FunModel(x[i],a))^2
+  return( S )
+ }
```

Задаємо початкове значення для параметрів **a** апроксимуючої функції й оцінюємо параметри:

```
> a0 = c(210,1)
> res = optim(fn=FunMNK, par=a0)
> a1 = res$par
> a1
[1] 213.22313362   0.05605678
> FunMNK(a1)
[1] 29.68398
```

Для перевірки правильності отриманого розв'язку, а також наочної інтерпретації результатів, побудуємо графіки:

```
> # Обчислюємо значення апроксимуючої функції в точках x
> ym = FunModel(x,a1)
>
> # Обчислюємо залишки
> r = y - ym
>
> a = 0; b = 40; n = 100
>
> # Обчислюємо n точок на відрізку [a,b]
> h = (b-a)/(n-1)
> x1 = c(1:n)
> for(i in 1:n){ x1[i] = a + h*(i-1) }
>
> # Обчислюємо значення апроксимуючої функції в точках x
> y1m = FunModel(x1,a1)
>
> par(mfrow=c(1, 2))              # задаємо дві графічні підобласті
> plot(x, y, type="p", col="red") # рисуємо перший графік у вигляді точок
> lines(x1, y1m, col="blue")     # додаємо в 1-у графічну підобласть ще лінію
> plot(x, r, type="p")           # рисуємо перший графік у вигляді точок
> lines(x, r, col="blue")        # додаємо в 2-у графічну підобласть ще лінію
> abline(h=0)
> abline(v=0)
```



Розв'язання 2. Використання процедури solve пакета R

Апроксимуюча функція $H(Q) = H_F - S_F Q^2$ є поліномом 2-ї степені й може бути записана у вигляді $g(x,a) = a_1 + a_2 x^2$, де $a \in R^2$ – коефіцієнти полінома $(a_1 = H_F, a_2 = S_F)$. При цьому функція $g(x,a)$ є лінійною за параметрами a і має вигляд (6.4), де $\psi_1(x) = 1$, $\psi_2(x) = -x^2$. Таким чином, у даному прикладі $n = 4$, $k = 2$. Тоді параметри a можна знайти, розв'язавши лінійну систему рівнянь (6.6).

Для оцінки параметрів апроксимуючої функції виду (6.4) в загальному вигляді запишемо процедуру *MNKSolve*:

```
> MNKSolve = function(x, y, n, Fi, k)
+ {
+   # Формуємо матрицю значень векторної функції Fi в точках x[i]
по рядкам
+   FiX = matrix(nrow=n, ncol=k)
+   for(i in 1:n)
+   {
+     FiX[i,] = Fi(x[i])
+   }

+   # Формуємо матрицю C
+   C = matrix(nrow=k, ncol=k)
+  for(j in 1:k)
+  {
+    for(m in 1:k)
+    {
+      S = 0
+      for(i in 1:n)
+      {
+       S = S + FiX[i,j]*FiX[i,m]
+       }
+      C[j,m] = S
+    }
+  }

+   # Формуємо вектор d
+   d = c(1:k)
+   for(j in 1:k)
+  {
+    S = 0
+    for(i in 1:n)
+    {
+      S = S + y[i]*FiX[i,j]
+    }
+     d[j] = S
+  }
```

```
+   # Розв'язуємо систему
+   a = solve(C, d)
+  return(a)
+ }
```

Застосуємо цю процедуру для розв'язання поставленої задачі:

```
> # Вводимо початкові дані
> n = 4
> x = c(5, 20, 30, 35)
> y = c(209, 194, 165, 142)
> # Вводимо векторну функцію Fi
> Fi = function(x)
+ {
+   z = c(1:2)
+   z[1] = 1
+   z[2] = -x*x
+   return( z )
+ }
> # Знаходимо параметри апроксимуючої функції
> k=2
> a1 = MNKSolve(x, y, n, Fi, k)
> a1
[1] 213.19623060   0.05599409
```

Як бачимо, значення параметрів апроксимуючої функції такі ж самі.

Приклад 2. Використовуючи кусково-лінійну й кусково-квадратичну інтерполяцію, а також поліном Лагранжа, обчислити в точці $\hat{x}=3$ наближене значення функції $f(x)$, заданої таблицею:

x	0,5	2,5	4,5	6,5	8,5
y	4,0	205,	1240	3840	8750

Розв'язання.

Кусково-лінійна інтерполяція

Оскільки точка $\hat{x}=3$ належить інтервалу $(2.5; 4.5)$, то наближене значення $f(3)$ можна обчислити за формулою (6.9), тобто

$$f(3)\approx g_2(3)=\frac{(3-4.5)}{(2.5-4.5)}\times 205+\frac{(3-2.5)}{(4.5-2.5)}\times 1240 = 463.75\,.$$

Кусково-квадратична інтерполяція

Оскільки точка $\hat{x}=3$ належить інтервалу $(2.5; 4.5)$, то наближене значення $f(3)$ можна обчислити за формулою (6.10), тобто

$$f(3) \approx g_2(3) = \frac{(3-4.5)(3-6.5)}{(2.5-4.5)(2.5-6.5)} \times 205 + \frac{(3-2.5)(3-6.5)}{(4.5-2.5)(4.5-6.5)} \times 1240 +$$
$$+ \frac{(3-2.5)(3-4.5)}{(6.5-2.5)(6.5-4.5)} \times 3840 = 497 .$$

Інтерполяційний многочлен Лагранжа

Проводимо обчислення за формулою (6.11):

$$f(3) \approx L_4(3) = \frac{(3-2.5)(3-4.5)(3-6.5)(3-8.5)}{(0.5-2.5)(0.5-4.5)(0.5-6.5)(0.5-8.5)} \times 4 +$$
$$+ \frac{(3-0.5)(3-4.5)(3-6.5)(3-8.5)}{(2.5-0.5)(2.5-4.5)(2.5-6.5)(2.5-8.5)} \times 205 +$$
$$+ \frac{(3-0.5)(3-2.5)(3-6.5)(3-8.5)}{(4.5-0.5)(4.5-2.5)(4.5-6.5)(4.5-8.5)} \times 1240 +$$
$$+ \frac{(3-0.5)(3-2.5)(3-4.5)(3-8.5)}{(6.5-0.5)(6.5-2.5)(6.5-4.5)(6.5-8.5)} \times 3840 +$$
$$+ \frac{(3-0.5)(3-2.5)(3-4.5)(3-6.5)}{(8.5-0.5)(8.5-2.5)(8.5-4.5)(8.5-6.5)} \times 8750 = 357.247 .$$

6.4. Порядок виконання роботи і варіанти завдань

6.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі наближення функцій;
- метод найменших квадратів для апроксимації функцій;
- методи інтерполяції функцій.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремого модуля метод найменших квадратів для оцінки параметрів будь-якої апроксимуючої функції;
- запрограмувати у вигляді окремого модуля метод найменших квадратів для оцінки параметрів апроксимуючого полінома деякої степені;
- привести тексти складених програм;
- підібрати з застосуванням складених програм найкращу апроксимацію для функції, заданої таблично, серед функцій виду:

1) $y = b_0 + b_1 x$;

2) $y = b_0 + b_1 x + b_2 x^2$;

3) $y = b_0 + b_1 x + b_2 x^2 + b_3 x^3$;

4) $y = b_0 + b_1 \sin(b_2 x) + b_3 \cos(b_2 x)$;

5) $y = b_0 \exp(b_1 x)$.

Побудувати графіки апроксимуючих функцій і відповідних залишків.

- запрограмувати у вигляді окремих модулів методи кусково-лінійної та кусково-квадратичної інтерполяції та інтерполяційний поліном Лагранжа;
- привести тексти складених програм;
- обчислити в точці $\hat{x} = \dfrac{(x_1 + x_2)}{2}$ наближене значення функції $f(x)$, заданої таблично в індивідуальному завданні;
- побудувати графіки всіх інтерполюючих функцій.

6.4.2. Варіанти індивідуальних завдань

Варіант 1

x	–5	–4	–3	–2	–1	0	1
y	54,9	37,6	23,7	14,4	8,3	6,6	8,9

Варіант 2

x	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,0
y	0,4	0,3	1,0	1,7	2,1	3,4	4,1	5,8

Варіант 3

x	4,5	5,0	5,5	6,0	6,5	7,0	7,5	8,0
y	7,7	9,4	11,4	13,6	15,6	18,6	21,2	24,1

Варіант 4

x	0,4	0,8	1,2	1,6	2,0	2,4	2,8	3,2	3,6
y	0,43	0,94	1,91	3,01	4,0	4,56	6,45	8,59	11,15

Варіант 5

x	4,0	4,4	4,8	5,2	5,6	6,0	6,4	6,8
y	13,88	16,93	20,47	24,15	28,29	32,61	37,41	42,39

Варіант 6

x	1	2	3	4	5	6
y	2,11	2,45	2,61	2,73	2,75	2,81

Варіант 7

x	0,3	0,6	0,9	1,2	1,5	1,8
y	4,39	4,75	4,98	5,11	5,12	5,18

Варіант 8

x	1	2	3	4	5	6
y	0,1	0,21	0,43	0,51	0,62	0,81

Варіант 9

x	1,0	1,5	2,0	2,5	3,0	3,5	4,0
y	4,11	4,16	4,23	4,29	4,36	4,42	4,53

Варіант 10

x	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,0
y	2,47	2,86	3,01	2,91	2,55	2,11	2,61	1,25

Варіант 11

x	–1	0	1	2	3	4	5
y	2,4	4,3	4,3	0,1	–6,1	–15,5	–26,5

Варіант 12

x	0,16	0,17	0,18	0,19	0,20	0,21
y	6,61	6,5	6,1	5,80	4,3	3,81

Варіант 13

x	–5	–4	–3	–2	–1	0	1
y	45,0	32,5	20,7	13,9	10,4	6,5	11,1

Варіант 14

x	–5	–4	–3	–2	–1	0	1
y	37,0	29,6	18,5	16,2	9,2	9,5	9,1

Варіант 15

x	–1	0	1	2	3	4	5
y	–0,6	3,4	3,4	–2,2	–12,6	–28,8	–48,9

6.5. Контрольні запитання

1. Сформулюйте постановку задачі наближення функцій.
2. У чому полягає метод найменших квадратів для апроксимації функцій?
3. Яким чином проводиться аналіз побудованої апроксимуючої функції?
4. У чому полягає кусково-лінійна інтерполяція функцій? Коли її слід застосовувати?
5. У чому полягає кусково-квадратична інтерполяція функцій? Коли її слід застосовувати?
6. Що називається інтерполяційним поліномом Лагранжа? Коли його слід застосовувати?
7. Що називають сплайном? Чим він відрізняється від інших методів інтерполяції?

Лабораторна робота № 7
Чисельне диференціювання функцій

7.1. Мета роботи

Вивчення формул чисельного диференціювання для практичного розв'язання задачі чисельного диференціювання функцій, придбання навичок використання цих формул для розв'язання задачі чисельного диференціювання із застосуванням комп'ютера.

7.2. Методичні рекомендації по організації самостійної роботи

По темі лабораторної роботи студент повинен: *знати* загальне формулювання задачі чисельного диференціювання функцій; *вміти* розв'язувати задачу чисельного диференціювання функцій з використанням формул чисельного диференціювання [1 – 6].

Задача чисельного диференціювання полягає в знаходженні значень похідних функції $y = f(x)$ у заданих точках у випадках, коли аналітичний вид функції $f(x)$ невідомий, дуже складний або функція $f(x)$ задана таблично. Привабливість чисельного підходу пояснюється наявністю простих формул, за допомогою яких похідні в заданій точці можна приблизно обчислити по кількох значеннях функції $f(x)$ в цій і близьких до неї точках.

Формули чисельного диференціювання застосовуються в тих випадках, коли функція $y = f(x)$ може бути задана таблицею своїх значень $y_i = f(x_i)$ у рівновіддалених вузлах $x_i = x_0 + i \times h$ $i = 0, \pm 1, \pm 2, \ldots$, де x_0 – точка, в якій обчислюється похідна. Вибравши яку-небудь множину $(n+1)$-го вузлів, функцію $y = f(x)$ наближають інтерполяційним многочленом $P_n(x)$. Тоді похідна від цього многочлена $P_n'(x)$ застосовується для наближеного представлення шуканої похідної $y' = f(x)$ в точці x_0: $f'(x_0) \approx P_n'(x_0)$.

Найбільш зручним інтерполяційним многочленом для чисельного диференціювання є поліном Ньютона. На його основі отримані формули різного порядку точності.

Наприклад, для похідних 1-го порядку:

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h} = \frac{y_1 - y_0}{h} \tag{7.1}$$

(формула диференціювання вперед),

$$f'(x_0) \approx \frac{f(x_0-h) - f(x_0)}{h} = \frac{y_0 - y_{-1}}{h} \tag{7.2}$$

(формула диференціювання назад),

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h} = \frac{y_1 - y_{-1}}{2h} \tag{7.3}$$

(симетрична формула диференціювання),

$$f'(x_0) \approx \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h} \tag{7.4}$$

(симетрична формула диференціювання).

Наприклад, для похідних 2-го порядку (симетричні формули диференціювання):

$$f''(x_0) \approx \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2} = \frac{y_1 - 2y_0 + y_{-1}}{h^2}, \tag{7.5}$$

$$f''(x_0) \approx \frac{-y_2 + 16y_1 - 30y_0 + 16y_{-1} - y_{-2}}{12h^2}. \tag{7.6}$$

Похибка формул (7.1), (7.2), (7.5) порядку $O(h)$, формули (7.3) – $O(h^2)$, формули (7.6) – $O(h^3)$, формули (7.4) – $O(h^4)$.

При цьому h називають кроком чисельного диференціювання.

7.3. Контрольні приклади

Приклад 1. Використовуючи формули чисельного диференціювання знайти чисельно 1-шу й 2-гу похідні функції $f(x) = 5\exp(1.5x)$ в точці $x_0 = 0.5$ з кроком $h = 0.01$.

Порівняти отримані значення з істинними значеннями похідних у тій же точці.

Розв'язання. Для чисельного обчислення 1-ої похідної скористаємося спочатку симетричною формулою диференціювання (7.3). Тоді:

$$f'(0.5) \approx \frac{f(0.5+0.01) - f(0.5-0.01)}{2 \times 0.01} = 15.8781.$$

Тепер обчислимо 1-шу похідну з використанням симетричної формули диференціювання (7.4). Тоді:

$$f'(0.5) \approx \frac{-y_2 + 8y_1 - 8y_{-1} + y_{-2}}{12h} =$$

$$= \frac{-f(0.5+2\times 0.01) + 8f(0.5+0.01) - 8f(0.5-0.01) + f(0.5-2\times 0.01)}{12 \times 0.01} = 15.8775.$$

Знайдемо значення похідної аналітично: $f'(x) = 7.5\exp(1.5x)$. Тоді $f'(0.5) = 15.8775$. Як видно з отриманих результатів, формула (7.4) дає меншу похибку порівняно з формулою (7.3), однак використовує вдвічі більше обчислень значень функції $f(x)$.

Для чисельного обчислення 2-ої похідної скористаємося спочатку симетричною формулою диференціювання (7.5). Тоді:

$$f''(0.5) \approx \frac{f(0.5+0.01) - 2f(0.5) + f(0.5-0.01)}{2 \times 0.01} = 23.8167.$$

Тепер обчислимо 2-гу похідну з використанням симетричної формули диференціювання (7.6). Тоді:

$$f''(0.5) \approx \frac{-y_2 + 16y_1 - 30y_0 + 16y_{-1} - y_{-2}}{12h^2} =$$

$$= \frac{-f(0.5 + 2\times 0.01) + 16f(0.5 + 0.01) - 30f(0.5) + 16f(0.5 - 0.01) - f(0.5 - 2\times 0.01)}{12\times (0.01)^2} =$$

$$= 23.81625 \; .$$

Знайдемо значення 2-ої похідної аналітично: $f''(x) = 11.25\exp(1.5x)$. Тоді $f''(0.5) = 23.81625$. Таким чином, формула (7.6) дає меншу похибку порівняно з формулою (7.5), однак використовує більше обчислень значення функції $f(x)$.

Приклад 2. Назустріч автомобілю, що наближається, посилається 3 радіосигнали з інтервалом 0,5 сек. По віддзеркалених сигналах визначається відстань до автомобіля у відповідні моменти часу: 301 м, 283 м, 269 м. Треба оцінити швидкість та прискорення автомобіля (км/год.) у момент часу, що відповідає 0.5 сек. після початку спостереження за автомобілем.

Розв'язання. Відстань від автомобіля до місця спостереження за ним y будемо розглядати, як функцію від часу t, тобто $y = f(t)$. Фізичний зміст значення 1-ої похідної від y по t у деякий момент часу – це і є швидкість автомобіля, а фізичний зміст значення 2-ої похідної – прискорення автомобіля у цей момент часу.

Функція $y = f(t)$ задана таблицею

t, сек.	0	0.5	1
y, м	301	283	269

відстань між вузлами t_i однакова ($h = 0.5$) й момент часу $t = 0.5$ є внутрішнім вузлом, то для чисельного обчислення 1-ої похідної скористаємося симетричною формулою (7.3), а для чисельного обчислення 2-ої похідної – симетричною формулою (7.5). При цьому точку $t = 0.5$ розглядаємо як центральну, тобто t_0, а інші точки нумеруються відносно неї. Тоді:

$$f'(0.5) \approx \frac{y_1 - y_{-1}}{2h} = \frac{269 - 301}{2\times 0.5} = -32,$$

$$f''(0.5) \approx \frac{y_1 - 2y_0 + y_{-1}}{h^2} = \frac{269 - 2\times 283 + 301}{(0.5)^2} = 16\,.$$

Тому швидкість автомобіля в момент часу, що відповідає 0.5 сек. після початку спостереження за ним буде –32 м/сек. або –115.2 км/год., а прискорення автомобіля – 16 м/сек^2. Тут швидкість автомобіля від'ємна, оскільки автомобіль наближається до місця спостереження за ним.

7.4. Порядок виконання роботи і варіанти завдань

7.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі чисельного диференціювання функцій;
- формули чисельного диференціювання функцій.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів формули чисельного диференціювання для аналітично заданої функції та функції, заданою таблично;
- привести тексти складених програм;
- знайти чисельно 1-шу й 2-гу похідні функції, заданої таблично, в третій за порядком точці;
- використовуючи формули чисельного диференціювання знайти чисельно 1-шу й 2-гу похідні функції, яка була отримана як найкраща апроксимуюча в завданні з лабораторної роботи 6, у 3-й точці з кроком $h = 0.01$. Порівняти отримані значення з істинними значеннями похідних цієї функції у тій же точці.

7.4.2. Варіанти індивідуальних завдань

Варіанти індивідуальних завдань взяти з завдань до лабораторної роботи 6.

7.5. Контрольні запитання

1. Сформулюйте постановку задачі диференціювання функцій.
2. У яких випадках виникає задача чисельного диференціювання?
3. Що називається інтерполяційним багаточленом Ньютона? Як його застосовують при виведенні формул чисельного диференціювання функцій?

4. На якому принципі заснований чисельний розрахунок похідних функції в точці?

5. У яких випадках краще застосовувати формули диференціювання вперед, назад та симетричні формули?

6. Від чого залежить точність чисельного обчислення похідних?

Лабораторна робота № 8
Чисельне інтегрування функцій

8.1. Мета роботи

Вивчення формул чисельного інтегрування для практичного розв'язання задачі чисельного інтегрування функцій, придбання навичок використання цих формул для розв'язання задачі чисельного інтегрування із застосуванням комп'ютера.

8.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі чисельного інтегрування функцій; *вміти* розв'язувати задачу чисельного інтегрування функцій з використанням формул чисельного інтегрування [1 – 6].

Задача чисельного інтегрування полягає в обчисленні визначеного інтеграла

$$\int_a^b f(x)dx \tag{8.1}$$

у випадках, коли аналітичне обчислення неможливе або дуже складне.

Методи чисельного обчислення інтеграла засновані на тому, що в якості наближеного значення інтеграла (8.1) береться значення інтеграла від інтерполюючої для $f(x)$ функції, побудованої по точках розбиття відрізка $[a,b]$.

Найбільш відомими й ефективними методами розв'язання задачі чисельного інтегрування функцій є методи трапецій і Сімпсона.

Формула трапецій (кусково-лінійна інтерполяція функції $f(x)$):

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} S_i = h\sum_{i=0}^{n-1}\frac{f(x_i)+f(x_{i+1})}{2} = h\left[\frac{f(x_0)+f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i)\right].$$

Формула Сімпсона (кусково-квадратична інтерполяція функції $f(x)$):

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(x_0)+f(x_n)+2\left[f(x_2)+f(x_4)+\cdots+f(x_{n-2})\right]+4\left[f(x_1)+f(x_3)+\cdots+f(x_{n-1})\right]\right].$$

При цьому n має бути обов'язково парним (число ж точок – непарним).

8.3. Контрольні приклади

Приклад 1. Обчислити значення визначеного інтеграла $\int_0^5 x^2 dx$ чисельно методом трапецій для числа розбиття інтервалу інтегрування $n = 10$ та $n = 50$.

Розв'язання. Для розв'язання поставленої задачі запишемо процедуру для методу трапецій:

```
> trapec = function(a, b, n)
+ {
+   h = (b-a)/n
+   s = (f(a)+f(b))/2
+   x = c(1:(n+1))
+   for (i in 2:n)
+   {
+     x[i] = a + (i-1)*h
+     s = s + f(x[i])
+   }
+   return (s*h)
+ }
```

Скористаємося цією процедурою для обчислення значення визначеного інтеграла:

```
> # підінтегральна функція
> f = function(x)
+ {
+   return (x^2)
+ }
> # початкові дані
> a = 0
> b = 5
> S = trapec(a, b, 10) # виклик функції користувача
> S
[1] 41.875
> S = trapec(a, b, 50) # виклик функції користувача
> S
[1] 41.675
```

Приклад 2. Після аварії в Чорному морі на поверхні води утворилась нафтова пляма. Для оцінки ступеня небезпеки екологічна станція має визначити площу забруднення води нафтою. З цією метою були визначені відстані від берега до країв плями (координати **у**) через кожні 10 м. Результати замірів (у метрах) приведені в таблиці.

Номер заміру	1	2	3	4	5	6	7	8	9	10	11
Ближній край, м	2	3,5	5	2,5	1,2	1,8	1,7	2,2	2,7	3,5	3,4
Дальній край, м	18	21	19	20	20	18	15	12	16,5	18	12

Треба оцінити площу нафтової плями, вважаючи, берегову лінію прямою.

Розв'язання. Для розв'язання поставленої задачі скористаємося засобами пакета R.

За умовою задачі маємо 11 точок спостережень, відстань між якими становить 10 метрів. Заміри до країв плями представимо векторами. Маємо дві таблично задані функції.

```
> # початкові дані
> n = 11
> h = 10
> y_bl = c(2, 3.5, 5, 2.5, 1.2, 1.8, 1.7, 2.2, 2.7, 3.5, 3.4)
> y_dal = c(18, 21, 19, 20, 20, 18, 15, 12, 16.5, 18, 12)
```

Для розв'язання задачі скористаємося методом трапецій. Оскільки функції задані таблично, то процедуру обчислення методом трапецій можна представити в такому вигляді:

```
trapec = function(y,h,n)
{
  s = (y[1]+y[n])/2
  for (i in 2:(n-1))
  {
    s = s + y[i]
  }
  return (s*h)
}
```

Обчислюємо площу від берега до дальнього краю плями і площу від берега до ближнього краю плями. Площу забруднення визначаємо як різницю між отриманими значеннями.

Обчислення проводимо з використанням запрограмованого методу трапецій:

```
> S_bl = trapec(y_bl,h,n)
> S_bl
[1] 268
> S_dal = trapec(y_dal,h,n)
> S_dal
[1] 1745
> S = S_dal - S_bl
> S
[1] 1477
```

Відповідь: площа нафтової плями склала приблизно 1 477 м^2.

8.4. Порядок виконання роботи і варіанти завдань

8.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі чисельного інтегрування функцій;
- формули чисельного інтегрування функцій.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів формули трапецій і Сімпсона для аналітично заданої функції та функції, заданої таблично;
- запрограмувати у вигляді окремого модуля алгоритм за яким можна розрахувати значення визначеного інтеграла з заданою точністю;
- привести тексти складених програм;

- обчислити значення визначеного інтеграла (8.1) чисельно методами трапецій і Сімпсона для числа розбиття інтервалу інтегрування $n = 10$ і $n = 20$;
- обчислити значення визначеного інтеграла (8.1) чисельно методом Сімпсона с точністю $\varepsilon = 10^{-7}$.

8.4.2. Варіанти індивідуальних завдань

Варіант 1. $\int_1^2 \frac{\cos x dx}{x}$.

Варіант 2. $\int_2^5 \frac{dx}{(x-1)^2}$.

Варіант 3. $\int_1^4 \frac{dx}{(x-3)^3}$.

Варіант 4. $\int_4^8 \frac{dx}{(\sqrt{x}-1)}$.

Варіант 5. $\int_1^9 5t\sqrt{t}dt$.

Варіант 6. $\int_0^{1,2} \ln(1+x^2)dx$.

Варіант 7. $\int_1^9 3\sqrt{x}(1+\sqrt{x})dx$.

Варіант 8. $\int_2^5 \frac{dx}{(x^2+4)}$.

Варіант 9. $\int_1^9 4\sqrt{x}(6+\sqrt{x})dx$.

Варіант 10. $\int_0^1 \sin(e^x)dx$.

Варіант 11. $\int_1^2 x\ln x dx$.

Варіант 12. $\int_1^2 \frac{dx}{\sqrt{7x^3}}$.

Варіант 13. $\int_1^4 \frac{(1+t)}{\sqrt{2t}}dt$.

Варіант 14. $\int\limits_1^4 \frac{(1+t^2)}{\sqrt{2t}} dt$.

Варіант 15. $\int\limits_1^9 \sqrt{6x-5} dx$.

8.5. Контрольні запитання

1. Сформулюйте постановку задачі чисельного інтегрування функції.
2. У чому полягає ідея методу трапецій?
3. У чому полягає ідея методу Сімпсона?
4. Який з відомих вам методів чисельного інтегрування має більшу точність?
5. Від чого залежить точність обчислення визначеного інтеграла за будь-якою формулою?
6. Вкажіть алгоритм за яким можна розрахувати значення визначеного інтеграла з заданою точністю.

Лабораторна робота № 9
Чисельні методи розв'язання задачі Коші для звичайних диференціальних рівнянь

9.1. Мета роботи

Вивчення однокрокових і багатокрокових чисельних методів для практичного розв'язання задачі Коші для звичайних диференціальних рівнянь, придбання навичок використання цих методів для розв'язання задачі Коші із застосуванням комп'ютера.

9.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі Коші для звичайних диференціальних рівнянь; *вміти* розв'язувати цю задачу з використанням однокрокових і багатокрокових чисельних методів [1 – 6].

Задача Коші для звичайного диференціального рівняння виду

$$y' = f(x, y) \tag{9.1}$$

полягає в такому: знайти функцію $y = y(x)$ на заданому відрізку $[a, b]$, що задовольняє рівнянню (10.1) і початковій умові

$$y(a) = y_0, \tag{9.2}$$

де y_0 задано.

Чисельні методи розв'язання задачі (9.1) – (9.2) знаходять розв'язок (тобто функцію $y(x)$ на відрізку $[a, b]$) у табличному вигляді, а саме у вигляді набору точок (x_i, y_i), $i = \overline{0, n}$, де $x_0 = a$, $x_i = x_0 + i \times h$, $i = \overline{1, n}$, $h = \dfrac{b-a}{n}$, n – задане число розбиття відрізка $[a, b]$, y_i, $i = \overline{1, n}$ – знайдені наближені значення функції $y(x)$ у вузлах сітки x_i.

Розглянемо кілька методів розв'язання задачі Коші для диференціального рівняння (9.1) на відрізку $[a, b]$.

Метод Ейлера. Значення y_i обчислюються рекурентно за формулою:

$$y_{i+1} = y_i + h \times f(x_i, y_i), \; i = \overline{0, n-1}.$$

Похибка методу Ейлера, оцінювана для величини $|y_n - y(x_n)|$, має порядок $O(h)$ [1 – 3; 5; 6].

1-ша модифікація методу Ейлера. Значення y_i обчислюються рекурентно за формулами:

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x_i, y_i), \; y_{i+1} = y_i + h \times f(x_i + \frac{h}{2}, y_{i+\frac{1}{2}}), \; i = \overline{0, n-1}.$$

Похибка 1-ої модифікації методу Ейлера, оцінювана для величини $|y_n - y(x_n)|$, має порядок $O(h^3)$ [1 – 3; 5; 6].

2-га модифікація методу Ейлера. Значення y_i обчислюються рекурентно за формулами:

$$y_{i+1}^* = y_i + h \times f(x_i, y_i), \; y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)), \; i = \overline{0, n-1}.$$

Похибка 2-ої модифікації методу Ейлера, оцінювана для величини $|y_n - y(x_n)|$, має порядок $O(h^3)$ [1 – 3; 5; 6].

Метод Рунге – Кутта 4-го порядку. Значення y_i обчислюються рекурентно за формулами:

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),\ i = \overline{0, n-1},$$

де $k_1 = h \times f(x_i, y_i)$, $k_2 = h \times f(x_i + \frac{h}{2}, y_i + \frac{k_1}{2})$, $k_3 = h \times f(x_i + \frac{h}{2}, y_i + \frac{k_2}{2})$, $k_4 = h \times f(x_i + h, y_i + k_3)$.

Похибка методу Рунге – Кутта 4-го порядку, оцінювана для величини $|y_n - y(x_n)|$, має порядок $O(h^4)$ [1 – 3; 5; 6].

У багатокроковому **методі прогнозу та корекції** 4-го порядку при розв'язанні задачі Коші (10.1) значення y_i у вузлах сітки x_i, $i = \overline{0, n}$, обчислюються рекурентно за формулами:

$$y_{i+1}^{(pred)} = y_i + \frac{h}{24}(-9f_{i-3} + 37f_{i-2} - 59f_{i-1} + 55f_i),$$

$$f_{i+1}^{(pred)} = f(x_{i+1},\ y_{i+1}^{(pred)}),$$

$$y_{i+1} = y_i + \frac{h}{24}(f_{i-2} - 5f_{i-1} + 19f_i + 9f_{i+1}^{(pred)}),$$

де $f_i = f(x_i, y_i)$.

До початку розрахунків за **методом прогнозу та корекції** 4-го порядку необхідно 3 перші кроки зробити будь-яким однокроковим методом, наприклад, методом Рунге – Кутта.

Похибка методу прогнозу та корекції 4-го порядку, оцінювана для величини $\|y_n - y(x_n)\|$, має порядок $O(h^4)$ [1 – 3; 5; 6].

9.3. Контрольні приклади

Приклад 1. Розв'язати методом Ейлера задачу Коші: $y' = y - \frac{2x}{y}$, $y(0) = 1$ на відрізку [0,1] с числом розбиття відрізка $n = 80$.

Розв'язання.

Складемо процедуру для метода Ейлера в пакеті R:

```
> eiler = function(a, b, y0, n)
+ {
+   h = (b-a)/n
+   x = c(1:n)
+   y = c(1:n)
+   x[1] = a
+   y[1] = y0
+   for (i in 1:n)
+   {
+     x[i+1] = x[i]+h
+     y[i+1] = y[i]+h*FunPr(x[i],y[i])
+   }
+   return(cbind(x,y))
+ }
```

Вводимо дані для розв'язання задачі:

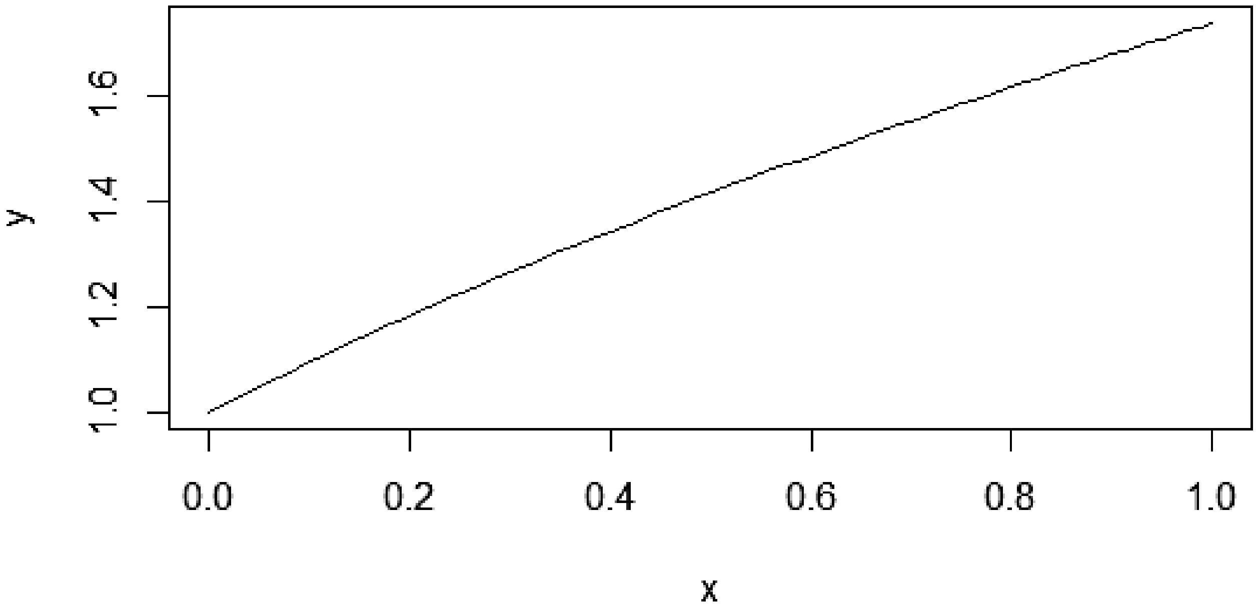
```
> # задаємо рівняння (функцію правої частини)
> FunPr = function(x,y)
+ {
+   return (y - 2*x/y)
+ }
> a = 0
> b = 1
> y0 = 1
> n = 80
```

Викликаємо записану процедуру і отримуємо розв'язок диференціального рівняння у вигляді таблично представленої функції:

```
> Y = eiler(a, b, y0, n)
> Y
           x        y
 [1,] 0.0000 1.000000
 [2,] 0.0125 1.012500
 [3,] 0.0250 1.024848
 [4,] 0.0375 1.037048
 [5,] 0.0500 1.049107
 [6,] 0.0625 1.061030
 [7,] 0.0750 1.072820
 [8,] 0.0875 1.084483
 [9,] 0.1000 1.096022
[10,] 0.1125 1.107441

………………………..

[80,] 0.9875 1.732003
[81,] 1.0000 1.739400
```

Будуємо графік отриманого розв'язку:

```
> x = Y[,1]
> y = Y[,2]
> plot (x,y,type = "l")
```



Приклад 2. Процес руху автомобіля на площині в найпростішому випадку може бути описаний системою диференціальних рівнянь:

$$\begin{cases} \dfrac{\partial x}{\partial t} = V\cos\theta \\ \dfrac{\partial y}{\partial t} = V\sin\theta \\ \dfrac{\partial \theta}{\partial t} = -\dfrac{V}{W\,Ctg\phi + \dfrac{w}{2}} \end{cases}, \qquad (9.3)$$

де (x, y) – координати точки M на площині xOy, θ – кут між повздовжньою віссю автомобіля й віссю Ox, V – швидкість, ϕ – кут повороту передніх коліс відносно повздовжньої осі автомобіля, W – відстань між передньою і задньою осями (колісна база), w – відстань між колесами автомобіля на задній осі (колія задніх коліс) (рис. 9.1). У моделі (9.3) всі кути вимірюються в радіанах.

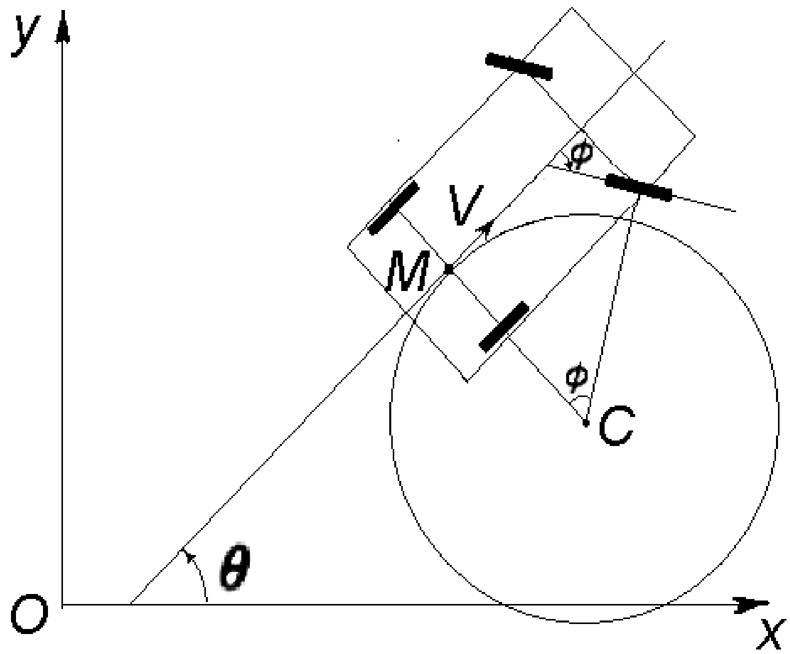


Рис. 9.1. **Модель руху автомобіля на площині**

Треба визначити траєкторію руху автомобіля (тобто точки M) за 5 секунд, якщо його швидкість V була постійна і дорівнювала 5 км/год., кут ϕ також був постійним і дорівнював 30^0, $W = 2.47$ м, $w = 1.456$ м, на початку руху точка M мала координати $(5, 2)$, а кут $\theta = 0$.

Розв'язання.

Оскільки в даному прикладі маємо три координати (x, y, θ), які змінюються у часі, то процедуру для метода Ейлера перепишемо в такому вигляді:

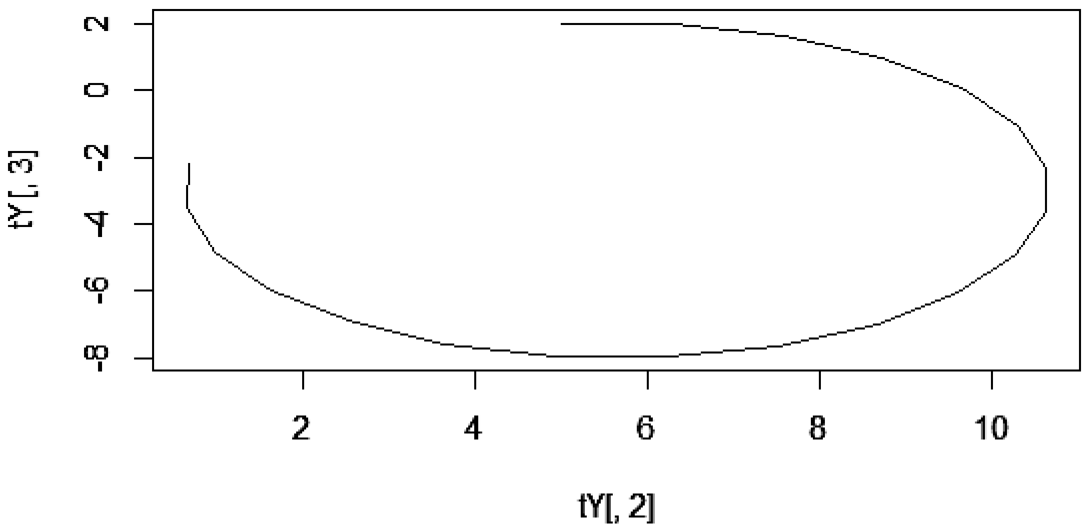
```
MetEiler = function(a, b, Y0, n)
{
  t = c(1:n)
  Y = matrix(0, nrow=n, ncol=3)
  h = (b-a)/(n-1)
  t[1] = a
  Y[1,] = Y0
  for (i in 1:(n-1))
  {
    t[i+1] = t[i] + h
    Y[i+1,] = Y[i,] + h*FunPr(t[i],Y[i,])
  }
  return(cbind(t,Y))
}
```

Вводимо дані для розв'язання задачі:

```
> W = 2.47; w = 1.456
> V = function(t){ return (5) }
> Fi = function(t){ return (pi/6) }
>
> # задаємо рівняння (функцію правої частини)
> FunPr = function(t,Y)
+ {
+   Yt = c(1:3)
+   Yt[1] = V(t)*cos(Y[3])
+   Yt[2] = V(t)*sin(Y[3])
+   Yt[3] = -V(t)/(W/tan(Fi(t))+w/2)
+   return (Yt)
+ }
>
> n = 20
> a = 0
> b = 5
> Y0 = c(5, 2, 0)
```

Викликаємо записану процедуру і отримуємо розв'язок. Будуємо графік отриманого розв'язку:

```
> tY = MetEiler(a, b, Y0, n)
> x = tY[,2]
> y = tY[,3]
> plot(x, y, type="l")
```



Приклад 3. Припустимо, що рух гоночного автомобіля описується рівнянням $y' = \dfrac{1.3y}{t}$, де y – відстань, пройдена за час t з моменту початку руху.

Нехай через півгодини автомобіль віддалився на 75 км. Необхідно визначити на яку відстань він від'їде через 4 години після початку руху.

Розв'язання. Для розв'язання скористаємося записаною в прикладі 1 процедурою:

```
> # задаємо рівняння (функцію правої частини)
> FunPr = function(x,y)
+ {
+   return (1.3*y/x)
+ }
> a = 0.5
> b = 4
> y0 = 75
> n = 100

> Y = eiler(a, b, y0, n)
> Y[n+1,2]
       y
1106.619
```

Відповідь: за 4 години гоночний автомобіль подолає відстань 1 106,619 км.

9.4. Порядок виконання роботи і варіанти завдань

9.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі Коші для звичайного диференціального рівняння 1-го порядку;
- чисельні однокрокові методи розв'язання задачі Коші для звичайного диференціального рівняння 1-го порядку;
- метод прогнозу та корекції розв'язання задачі Коші для звичайного диференціального рівняння 1-го порядку.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремих модулів метод Ейлера, 1-шу і 2-гу модифікації методу Ейлера та метод Рунге – Кутта 4-го порядку для розв'язання задачі Коші для рівняння (9.1);
- запрограмувати у вигляді окремого модуля алгоритм за яким можна розв'язати задачу Коші з заданою точністю;
- запрограмувати у вигляді окремого модуля метод прогнозу та корекції для розв'язання задачі Коші для рівняння (9.1);

- привести тексти складених програм;
- розв'язати задачу Коші для диференціального рівняння (9.1) чисельно методами Ейлера і його модифікаціями й методом Рунге – Кутта із числом розбиття відрізка $n = 10$ і $n = 20$;
- розв'язати задачу Коші для диференціального рівняння (9.1) чисельно методом прогнозу та корекції із числом розбиття відрізка $n = 10$ і $n = 20$;
- побудувати графіки отриманих розв'язків;
- розв'язати задачу Коші для диференціального рівняння (9.1) чисельно методом Ейлера с точністю $\varepsilon = 10^{-7}$.

9.4.2. Варіанти індивідуальних завдань

Варіант 1. $y' = x^2 + y^2, \quad [0,1], \quad y(0) = 0$.

Варіант 2. $y' = 1 + xy^2, \quad [0,1], \quad y(0) = 0$.

Варіант 3. $y' = \dfrac{y}{x+1} - y^2, \quad [0,1], \quad y(0) = 1$.

Варіант 4. $y' = \dfrac{y}{x}, \quad [1,25], \quad y(1) = 1$.

Варіант 5. $y' = \dfrac{x^2 - y^2}{xy}, \quad [2,25], \quad y(2) = 1$.

Варіант 6. $y' = \dfrac{1}{2}xy, \quad [0,1], \quad y(0) = 1$.

Варіант 7. $y' = \dfrac{xy}{x^2 - y^2}, \quad [2,25], \quad y(2) = 1$.

Варіант 8. $y' = \dfrac{x + y - 3}{x - y - 1}, \quad [2,25], \quad y(2) = 1$.

Варіант 9. $y' = \dfrac{2xy - 1}{4x + 2y + 5}, \quad [2,25], \quad y(2) = 1$.

Варіант 10. $y' = \dfrac{3x - y + 2}{2x - 3y + 1}, \quad [3,30], \quad y(3) = 5$.

Варіант 11. $y' = \dfrac{xy^3}{x^2 + y^2}, \quad [5,25] \quad y(5) = 7$.

Варіант 12. $y' = \dfrac{y^4 + y}{x^2}, \quad [2,34] \quad y(2) = 10$.

Варіант 13. $y' = \dfrac{2y + y^2}{x}$, $[2,22]$ $y(2) = 8$.

Варіант 14. $y' = \dfrac{3y^2 - y}{2 + x}$, $[3,33]$, $y(3) = 7$.

Варіант 15. $y' = \dfrac{5y + y^2}{22 - x}$, $[1,21]$, $y(1) = 18$.

10.5. Контрольні запитання

1. Сформулюйте постановку задачі Коші для звичайного диференціального рівняння 1-го порядку.
2. У чому полягає ідея методу Ейлера та його модифікацій?
3. У чому полягає ідея методу Рунге – Кутта 4-го порядку?
4. Чим відрізняються багатокрокові методи розв'язання задачі Коші від однокрокових?
5. У чому полягає ідея методу прогнозу та корекції?
6. Від чого залежить точність розв'язку задачі Коші будь-яким методом?

Лабораторна робота № 10 Чисельні методи розв'язання лінійної крайової задачі для звичайних диференціальних рівнянь 2-го порядку

10.1. Мета роботи

Вивчення чисельних методів інтегрування диференціальних рівнянь для практичного розв'язання крайової задачі для звичайних диференціальних рівнянь, придбання навичок використання цих методів для розв'язання крайової задачі із застосуванням комп'ютера.

10.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання крайової задачі для звичайних диференціальних рівнянь

2-го порядку; *вміти* розв'язувати цю задачу з використанням чисельного методу кінцевих різниць [1 – 6].

Крайова задача для звичайного диференціального рівняння 2-го порядку виду:

$$F(x,y,y',y'')=0 \tag{10.1}$$

полягає в такому: знайти функцію $y=y(x)$ на заданому відрізку $[a,b]$, що задовольняє рівнянню (10.1) і крайовим умовам:

$$\begin{cases}\phi_1(y(a),y'(a))=0\\ \phi_2(y(b),y'(b))=0\end{cases}, \tag{10.2}$$

де F, ϕ_1, ϕ_2 – задані неперервні функції відповідного числа аргументів.

Крайова задача (10.1) – (10.2) називається **лінійною**, якщо функції F, ϕ_1, ϕ_2 лінійні відносно y, y', y''. Таким чином, лінійна крайова задача для звичайного диференціального рівняння 2-го порядку полягає в наступному: знайти функцію $y=y(x)$ на заданому відрізку $[a,b]$, що задовольняє лінійному рівнянню виду:

$$y''+p(x)y'+q(x)y=f(x) \tag{10.3}$$

і лінійним крайовим умовам:

$$\begin{cases}\alpha_0 y(a)+\alpha_1 y'(a)=A\\ \beta_0 y(b)+\beta_1 y'(b)=B\end{cases}, \tag{10.4}$$

де $p(x)$, $q(x)$, $f(x)$ – задані неперервні функції від x, α_0, α_1, β_0, β_1, A, B – задані константи, причому $|\alpha_0|+|\alpha_1|\neq 0$, $|\beta_0|+|\beta_1|\neq 0$.

Чисельні методи розв'язання задачі (10.1) – (10.2), а зокрема й задачі (10.3) – (10.4), знаходять розв'язок (тобто функцію $y(x)$ на відрізку $[a,b]$) у табличному вигляді, а саме у вигляді набору точок (x_i,y_i), $i=\overline{0,n}$, де $x_0=a$, $x_i=x_0+i\times h$, $i=\overline{1,n}$, $h=\dfrac{b-a}{n}$, n – задане число розбиття відрізка $[a,b]$, y_i, $i=\overline{0,n}$ – обчислені наближені значення функції $y(x)$ у вузлах сітки x_i

Розглянемо **метод кінцевих різниць** розв'язання лінійної крайової задачі (10.3) – (10.4).

Замінимо приблизно в кожному внутрішньому вузлі x_i похідні $y'(x_i)$, $y''(x_i)$ кінцево-різницевими формулами:

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}, \ y''(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}, \qquad (10.5)$$

а на кінцях відрізка $[a,b]$ покладемо

$$y'(x_0) \approx \frac{y_1 - y_0}{h}, \ y'(x_n) \approx \frac{y_n - y_{n-1}}{h}. \qquad (10.6)$$

Також введемо позначення: $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$, $i = \overline{1,n}$.

Використовуючи формули (10.5) – (10.6), приблизно замінимо рівняння (10.3) і крайові умови (10.4) системою рівнянь:

$$\begin{cases} \dfrac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \dfrac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i \ (i = \overline{1,n-1}); \\ \alpha_0 y_0 + \alpha_1 \dfrac{y_1 - y_0}{h} = A; \\ \beta_0 y_n + \beta_1 \dfrac{y_n - y_{n-1}}{h} = B. \end{cases} \qquad (10.7)$$

Отримана система (10.7) є лінійною алгебраїчною системою $(n+1)$ рівнянь відносно $(n+1)$ невідомих y_i, $i = \overline{0,n}$. Розв'язавши її, якщо це можливо, і буде отримана таблиця наближених значень шуканої функції $y(x)$ на відрізку $[a,b]$.

Додамо, що систему (10.7) при розв'язанні краще записати у вигляді

$$\begin{cases} (\alpha_0 h - \alpha_1) y_0 + \alpha_1 y_1 = Ah; \\ (1 - \dfrac{p_i h}{2}) y_{i-1} + (q_i h^2 - 2) y_i + (1 + \dfrac{p_i h}{2}) y_{i+1} = f_i h^2 \ (i = \overline{1,n-1}); \\ -\beta_1 y_{n-1} + (\beta_0 h + \beta_1) y_n + = Bh. \end{cases} \qquad (10.8)$$

Похибка методу кінцевих різниць, оцінювана для величин $|y_i - y(x_i)|$, $i = \overline{0,n}$ має порядок $O(h^2)$ [1 – 3; 5; 6].

10.3. Контрольні приклади

Приклад 1. Розв'язати методом кінцевих різниць крайову задачу $x^2y'' + xy' = 1$, $y(1) = 0$, $y(1.4) = 0{,}05661$ з числом розбиття відрізка $n = 4$.

Розв'язання. Насамперед, необхідно привести дану крайову задачу до виду (10.3) – (10.4), а саме диференціальне рівняння потрібно записати так: $y'' + \frac{1}{x}y' = \frac{1}{x^2}$, тобто $p(x) = \frac{1}{x}$, $q(x) = 0$, $f(x) = \frac{1}{x^2}$.

Тому початкові дані для розв'язання задачі запишемо так:

```
> alpha = c(1,0)
> beta = c(1,0)
> a = 1
> b = 1.4
> A = 0
> B = 0.05661
> n = 4
> p = function(x) { return(1/x) }
> q = function(x) { return(0) }
> f = function(x) { return(1/(x^2)) }
```

Систему (10.8) будемо розв'язувати за допомогою процедури *solve* пакета R. Процедури для формування матриці коефіцієнтів C і вектора вільних членів D мають вигляд:

```
> matrC = function(alpha,beta,p,q,n,h,x)
+ {
+   C = matrix(0, nrow=(n+1), ncol=(n+1))
+   C[1,1] = (alpha[1])*h-alpha[2]
+   C[1,2] = alpha[2]
+   C[(n+1),n] = -beta[2]
+   C[(n+1),(n+1)] = (beta[1])*h+beta[2]
+   for (i in 2:n)
+   {
+     C[i,(i-1)] = 1-h*p(x[i])/2
+     C[i,i] = (h^2)*q(x[i])-2
+     C[i,(i+1)] = 1+h*p(x[i])/2
+   }
+   return (C)
+ }
```

```
> matrD = function(A,B,f,n,h,x)
+ {
+   d = c(1:(n+1))
+   d[1] = A*h
+   d[n+1] = B*h
+   for (i in 2:n)
+   {
+     d[i] = f(x[i])*h^2
+   }
+   return(d)
+ }
```

Процедура формування системи лінійних рівнянь (10.7) та її розв'язання має такий вигляд:

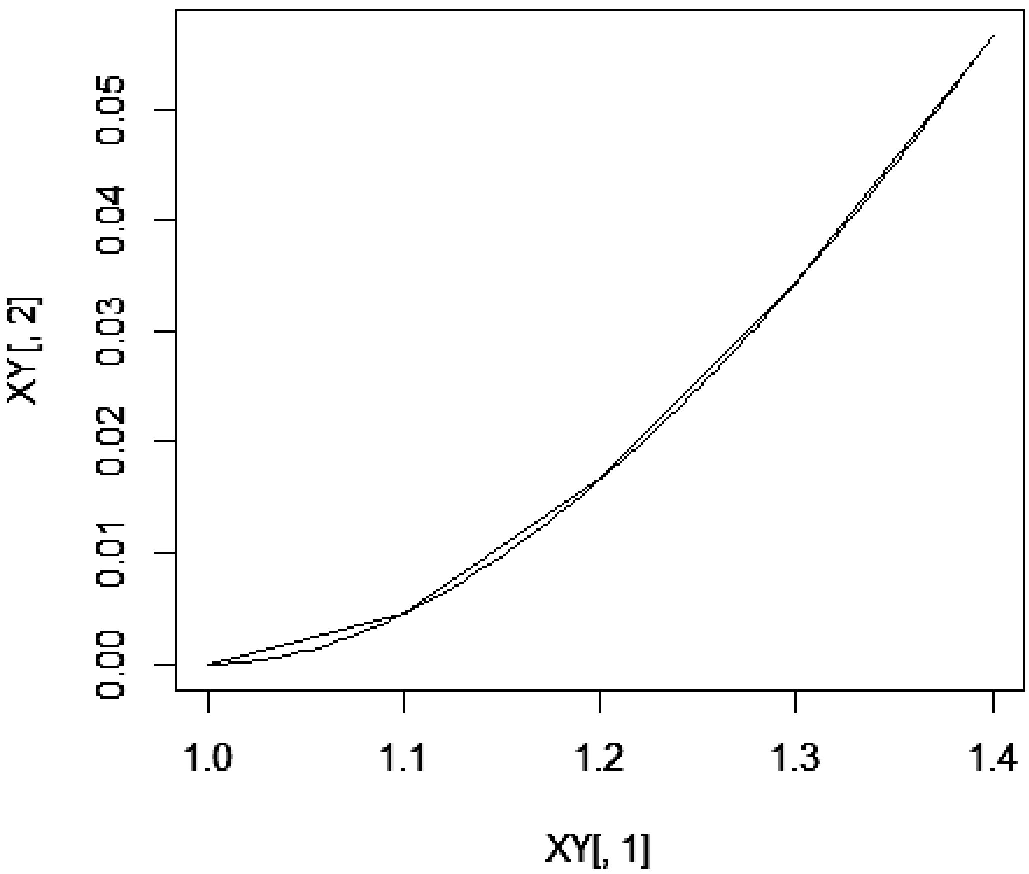
```
> LinDifUr = function(a,b,A,B,n)
+ {
+   h = (b-a)/n
+   x = c(1:(n+1))
+   for (i in 1:(n+1))
+   {
+     x[i] = a + (i-1)*h
+   }
+   C = matrC(alpha,beta,p,q,n,h,x)
+   D = matrD(A,B,f,n,h,x)
+   y = solve(C,D)
+   return(cbind(x,y))
+ }
```

Для отримання розв'язку викликається записана процедура:

```
> XY = LinDifUr(a,b,A,B,n)
```

Побудуємо графік і порівняємо отриманий чисельний розв'язок з аналітичним.

```
> # Будуємо графік розв'язку
> plot(XY[,1], XY[,2], type="l")
> # Порівнюємо отриманий розв'язок з точним
(аналітичним) розв'язком
> x = seq(a,  b, len=100)
> y = c(1:100)
> for(i in 1:100){ y[i] = 0.5*(log(x[i]))^2 }
> lines(x, y, type="l", col="red")
```



10.4. Порядок виконання роботи і варіанти завдань

10.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку крайової задачі для звичайного диференціального рівняння 2-го порядку;
- метод кінцевих різниць розв'язання лінійної крайової задачі для звичайного диференціального рівняння 2-го порядку.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремого модуля метод кінцевих різниць розв'язання лінійної крайової задачі для звичайного диференціального рівняння 2-го порядку;
- привести текст складеної програми;
- розв'язати методом кінцевих різниць крайову задачу із числом розбиття відрізка $n = 5$ і $n = 10$;
- побудувати графіки отриманих розв'язків.

10.4.2. Варіанти індивідуальних завдань

Варі-ант	Вид рівняння	Інтер-вал	Умова
1	$y'' - 2xy' - 2y = -4x$	[0,1]	$y(0) - y'(0) = 0$, $y(1) = 3.718$
2	$y'' + \frac{x}{2}y' + (1 + 2\pi^2 x^2)y = 4x$	[0,1]	$y(0) = 1,\ y(1) = 1.367$
3	$y'' + x^2 y' + (1 - x)y = \frac{x}{x^2 + 3}$	[0,1]	$y(0) = 0,\ y(1) = 0$
4	$y'' + (x - 1)y' + 3.125y = 4x$	[0,1]	$y(0) = 1,\ y(1) = 1.367$
5	$y'' + x^2 y' + (1.4 - x)y = \frac{x}{x^2 + 2.5}$	[0,1]	$y(0) = 0,\ y(1) = 0$
6	$y'' + 2xy' + 2y = \frac{2 \cdot (5 - 2x)}{(2 - x)^3}$	[0,1]	$y(0) = 1,\ y(1) = 1.367$
7	$y'' + (1 + x^3)y' + (1 - x^2)y = e^{1 - 2.5x^2}$	[0,1]	$y(0) = 0,\ y(1) = 0$
8	$y'' + (1.4 + x^3)y' + (1 - x^2)y = e^{1 - 3x^2}$	[0,1]	$y(0) = 0,\ y(1) = 0$
9	$y'' + x^2 y' + (1.8 - x)y = \frac{x}{x^2 + 3.5}$	[0,1]	$y(0) = 0,\ y(1) = 0$
10	$y'' + (1.8 + x^3)y' + (1 - x^2)y = e^{1 - 3.5x^2}$	[0,1]	$y(0) = 0,\ y(1) = 0$
11	$y'' + x^2 y' + (2.2 - x)y = \frac{x}{x^2 + 4}$	[0,1]	$y(0) = 0,\ y(1) = 0$
12	$y'' + (2.2 + x^3)y' + (1 - x^2)y = e^{1 - 4x^2}$	[0,1]	$y(0) = 0,\ y(1) = 0$
13	$y'' + y' \sin(x) + y = \frac{1}{2.5 + \sin^2(x)}$	[0,1]	$y(0) = 0,\ y(1) = 0$
14	$y'' + x^2 y' + (3 - x)y = \frac{x}{x^2 + 5}$	[0,1]	$y(0) = 0,\ y(1) = 0$
15	$y'' + \frac{y'}{\sqrt{x^2 + 2.5}} + y = x$	[0,1]	$y(0) = 0,\ y(1) = 0$

10.5. Контрольні запитання

1. Сформулюйте постановку крайової задачі для звичайного диференціального рівняння 2-го порядку.

2. У чому полягає ідея методу кінцевих різниць розв'язання крайової задачі?

3. Від чого залежить точність розв'язку крайової задачі методом кінцевих різниць?

Лабораторна робота № 11
Чисельні методи розв'язання інтегральних рівнянь

11.1. Мета роботи

Вивчення чисельних методів розв'язання інтегральних рівнянь, придбання навичок використання цих методів для пошуку розв'язків інтегральних рівнянь із застосуванням комп'ютера.

11.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання задачі розв'язання інтегрального рівняння; *вміти* розв'язувати цю задачу з використанням чисельних методів [1 – 3; 5; 6].

Інтегральними рівняннями 2-го роду називаються рівняння виду:

$$\phi(x) - \lambda \int_a^b K(x,t)\phi(t)\,dt = f(x), \qquad (11.1)$$

де $\phi(x)$ – шукана (невідома) функція, λ – заданий параметр рівняння, $K(x,t)$ – ядро рівняння, $f(x)$ – вільний член (права частина) рівняння, причому, $K(x,t)$ та $f(x)$ – задані (відомі) функції.

Інтегральними рівняннями 1-го роду називаються рівняння виду:

$$\int_a^b K(x,t)\phi(t)\,dt = f(x). \qquad (11.2)$$

Якщо ядро $K(x,t)$ неперервне для $\forall x \in [a,b]$ та $\forall t \in [a,b]$, або має такі розриви, що інтеграл $\int_a^b\int_a^b \left|K^2(x,t)\right| dx\, dt$ є обмеженим, то рівняння

(11.1) та (11.2) називаються **рівняннями Фредгольма**, відповідно 2-го та 1-го роду.

У **методі послідовних наближень** [6] при розв'язанні рівняння (11.1) будується ітераційна послідовність $\{\phi_k(x)\}, k = 0,1,2,\ldots,$ за рекурентною формулою

$$\phi_{k+1}(x) = f(x) + \lambda \int_a^b K(x,t)\,\phi_k(t)\,dt\ .$$

Початкове наближення, наприклад, можна взяти таким: $\phi_0(x) = f(x)$.

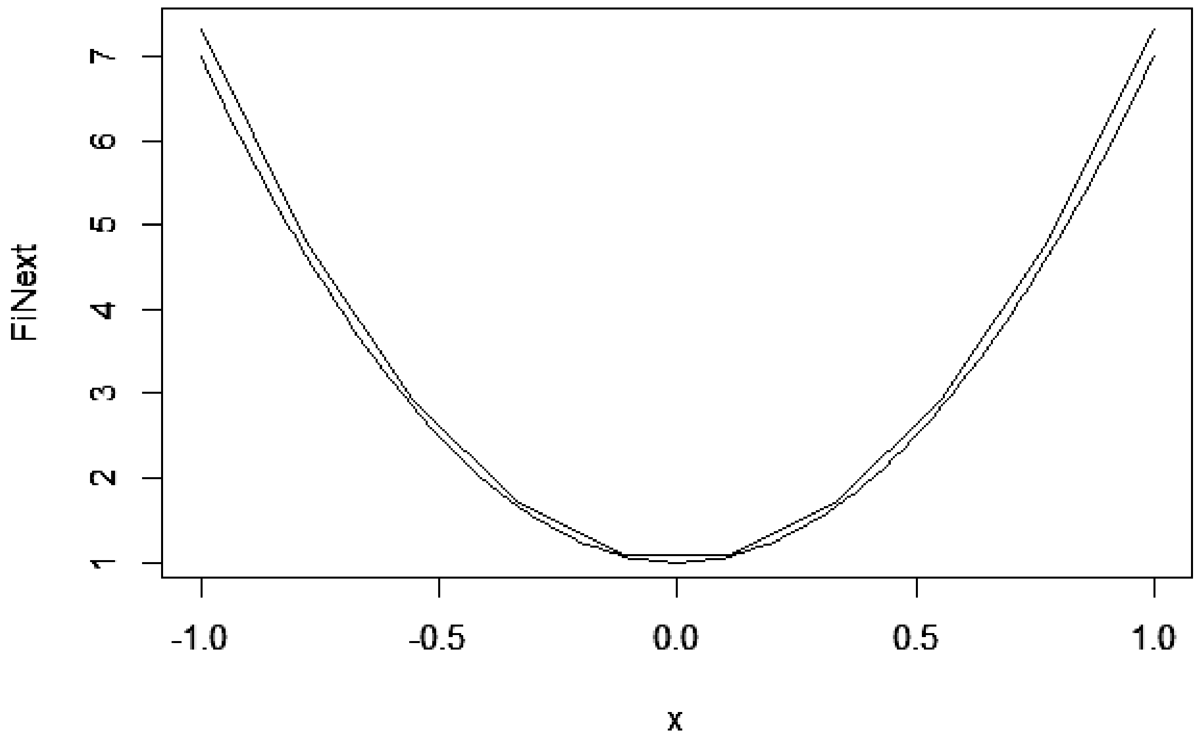
11.3. Контрольний приклад

Приклад. Розв'язати методом послідовних наближень інтегральне рівняння [6] $\phi(x) = 1 + \int_{-1}^{1}(x \times t + x^2)\phi(t)dt$ (аналітичний розв'язок $\phi(x) = 1 + 6x^2$).

Розв'язання. Для розв'язання поставленої задачі скористаємося засобами пакета *R*.

```
> # Задаємо границі інтегрування і параметр lambda
> a=-1; b=1; lambda=1
> # Задаємо функцію ядра
> FunCore = function(x, t){ return( x*t+x*x ) }
>
> # Задаємо функцію правої частини рівняння
> FunF = function(x){ return( 1 ) }

> # Програмуємо функцію обчислення інтеграла за формулою
трапецій
> # для таблично заданої функції
> MetTrap = function(x, y, n){
+   h = x[2] - x[1]
+   S = (y[1] + y[n])/2
+   for(i in 2:(n-1)){ S = S + y[i] }
+   return( h*S )
+ }
> # Програмуємо функцію обчислення норми вектора
> Norma = function(x){
+   return( sqrt(crossprod(x, x)) )
+ }
```

```
> # Задаємо число точок сітки на відрізку [a,b] і точність
> n = 10; eps = 1E-5
> # Визначаємо точки сітки
> h = (b-a)/(n-1)
> x = c(1:n)
> for(i in 1:n){ x[i] = a + h*(i-1) }
> t = c(1:n)
> t = x
> # Обчислюємо значення функції ядра в точках сітки
> K = matrix(0, ncol=n, nrow=n)
> for(i in 1:n){
+   for(j in 1:n){ K[i,j] = FunCore(x[i], t[j]) }
+ }
> Fi = c(1:n); FiNext = c(1:n)

> # Задаємо початкове наближення розв'язку
> for(i in 1:n){
+   Fi[i] = FunF(x[i]) }

> # Запускаємо ітераційний процес
> y = c(1:n)
> for(ki in 1:100)
+ {
+   # Обчислюємо наступне наближення розв'язку
+   for(i in 1:n)
+   {
+     for(j in 1:n){ y[j] = K[i,j]*Fi[j] }
+     FiNext[i] = FunF(x[i]) + lambda*MetTrap(t, y, n)
+   }
+   # Порівнюємо два наближення розв'язку
+   if( Norma(FiNext-Fi) <= eps)
+     break
+   Fi = FiNext
+ }
> ki
[1] 35

> plot(x, FiNext, type="l", col="blue")
>
> # Порівнюємо отриманий розв'язок з точним (аналітичним)
розв'язком
> t = seq(a,  b, len=100)
> for(i in 1:100){ y[i] = 1+6*t[i]*t[i] }
> lines(t, y, type="l", col="red")
```



11.4. Порядок виконання роботи і варіанти завдань

11.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку задачі розв'язання інтегрального рівняння 2-го роду;
- метод послідовних наближень для розв'язання інтегрального рівняння 2-го роду.

У практичній частині роботи необхідно:

- запрограмувати у вигляді окремого модуля метод послідовних наближень для розв'язання інтегрального рівняння 2-го роду;
- привести текст складеної програми;
- розв'язати методом послідовних наближень інтегральне рівняння 2-го роду;
- побудувати графіки отриманих розв'язків.

11.4.2. Варіанти індивідуальних завдань

Варіант 1. $\phi(x) = 3 + \int\limits_{-2}^{2}(x \cdot t + x^2)\phi(t)dt$.

Варіант 2. $\phi(x) = \int\limits_{0}^{1}(x + t)\phi(t)dt$.

Варіант 3. $\phi(x) = \int_0^{\pi} (\cos^2(x)\cos(2t) + \cos^3(t)\cos(3x))\phi(t)dt$.

Варіант 4. $\phi(x) = 8 + \int_2^6 \frac{2 + x^2 t}{t}\phi(t)dt$.

Варіант 5. $\phi(x) = (1 - xe^{2x})\cos(1) - e^{2x}\sin(1) + \int_0^{2.5} (1 - (x - t)e^{2x})\phi(t)dt$

(точний розв'язок [6] $\phi(x) = e^x(\cos(e^x) - e^x\sin(e^x))$).

Варіант 6. $\phi(x) = \frac{5}{6}x + \frac{1}{2}\int_0^1 xt\,\phi(t)dt$.

Варіант 7. $\phi(x) = 1 + \frac{1}{2}\int_0^7 \phi(t)dt$ (точний розв'язок [6] $\phi(x) = e^x$).

Варіант 8. $\phi(x) = e^x - x - \int_0^1 x(e^{xt} - 1)\phi(t)dt$.

Варіант 9. $\phi(x) = 1 + \int_0^1 xt^2\phi(t)dt$.

Варіант 10. $\phi(x) = x + \cos x + \int_0^1 x(\sin xt - 1)\phi(t)dt$.

Варіант 11. $\phi(x) = 1 + \frac{4}{3}x + \int_{-1}^1 (xt^2 - x)\phi(t)dt$.

Варіант 12. $\phi(x) = 1 - x(e^x - e^{-x}) + \int_{-1}^1 x^2 e^{xt}\phi(t)dt$.

Варіант 13. $\phi(x) = \frac{1}{2}(e^{-x} + 3x - 1) + \int_0^1 (\exp(-xt^2) - 1)x\phi(t)dt$.

Варіант 14. $\phi(x) = 1 + \int_{-1}^1 (xt + x^2)\phi(t)dt$.

Варіант 15. $\phi(x) = \frac{x}{2} + \frac{1}{2}\sin x + \int_0^1 (1 - \cos xt^2)x\phi(t)dt$.

11.5. Контрольні запитання

1. Які рівняння називаються інтегральними?
2. Наведіть класифікацію інтегральних рівнянь.
3. Які є методи розв'язання інтегральних рівнянь? У чому їх ідея?

Лабораторна робота № 12
Методи математичної фізики. Розв'язання диференціальних рівнянь з частинними похідними

12.1. Мета роботи

Вивчити методи розв'язання диференціальних рівнянь у частинних похідних. Ознайомитися з прикладами практичних задач, що зводяться до розв'язання диференціальних рівнянь у частинних похідних.

12.2. Методичні рекомендації щодо організації самостійної роботи

За темою лабораторної роботи студент повинен: *знати* загальне формулювання крайових задач для диференціального рівняння із частинними похідними; *вміти* розв'язувати ці задачі з використанням методу кінцевих різниць [1 – 3; 5; 6].

12.2.1. Рівняння параболічного типу

Розглядається **рівняння теплопровідності** (дифузії)

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} + f(x,t)\,,\ 0 \le x \le L\,,\ t \ge 0\,, \qquad (12.1)$$

де $u(x,t)$ – шукана функція двох змінних (x – координата, t – час), a – коефіцієнт теплопровідності (якщо u – температура) чи коефіцієнт масо-перенесення (якщо u – концентрація речовини), $f(x,t)$ – функція, яка задає внутрішні джерела тепла (викиду речовини).

Рівняння теплопровідності описує: процеси поширення тепла, процеси дифузії.

Для рівняння (12.1) задаються:
початкові умови (при $t = 0$)

$$u(x,0) = \varphi(x)\,,\ 0 \le x \le L\,, \qquad (12.2)$$

і граничні умови (при $x = 0$ и $x = L$)

$$u(0,t) = \psi(t)\,,\ u(L,t) = \xi(t)\,,\ t \ge 0\,. \qquad (12.3)$$

Граничні умови (12.3) також можуть бути задані у вигляді

$$\frac{\partial u(0,t)}{\partial x} = \psi(x), \frac{\partial u(L,t)}{\partial x} = \xi(x), \tag{12.4}$$

чи комбінації (12.3) і (12.4).

В умовах (12.2) – (12.4) $\varphi(x)$, $\psi(t)$, $\xi(t)$ – задані функції.

12.2.2. Рівняння еліптичного типу

Розглядається **рівняння Пуассона**

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \ (x,y) \in G, \tag{12.5}$$

де $u(x,y)$ – шукана функція двох змінних (x, y – координати), G – задана область (тіло), $f(x,t)$ – функція, яка задає внутрішні джерела енергії (чи відтоків).

Якщо $f(x,t) = 0 \quad \forall (x,y) \in G$, то рівняння (12.5) називається **рівнянням Лапласа**.

Рівняння Лапласа і Пуассона описують: потоки ідеальних рідин в стаціонарних потоках, стаціонарний розподіл температури або напруженості електричних і магнітних полів. Рівняння Лапласа описує ці процеси за відсутності внутрішніх джерел енергії (чи відтоків).

Для рівняння (12.5) задаються лише граничні умови (Γ – межа області G):

1) 1-ша крайова задача (**задача Діріхле**):

$$u\big|_{\Gamma} = \phi(x,y), \quad (x,y) \in \Gamma; \tag{12.6}$$

2) 2-га крайова задача (**задача Неймана**):

$$\left.\frac{\partial u}{\partial n}\right|_{\Gamma} = \phi(x,y), \quad (x,y) \in \Gamma; \tag{12.7}$$

3) 3-тя крайова задача (комбінована):

$$\alpha u\big|_{\Gamma} + \beta \left.\frac{\partial u}{\partial n}\right|_{\Gamma} = \phi(x,y), \quad (x,y) \in \Gamma. \tag{12.8}$$

В умовах (12.7) і (12.8) n – зовнішня нормаль відносно області G в точці $(x,y) \in \Gamma$, $\phi(x,y)$ – задана функція на межі Γ області G.

12.2.3. Рівняння гіперболічного типу

Розглядається **хвильове рівняння**

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} = f(x,t),\ 0 \le x \le L,\ t \ge 0, \tag{12.9}$$

де $u(x,t)$ – шукана функція двох змінних (x – координата, t – час), a – швидкість розповсюдження збурювання (u – відхилення від рівноваги), $f(x,t)$ – функція, яка задає внутрішні джерела енергії.

Хвильове рівняння описує: малі подовжні коливання стержня, поперечні коливання струни, процеси поширення малих акустичних коливань тощо.

Для рівняння (12.9) задаються:
початкові умови (при $t = 0$)

$$u(x,0) = \varphi_1(x),\ \frac{\partial u(x,0)}{\partial t} = \varphi_2(x),\ 0 \le x \le L \tag{12.10}$$

і граничні умови (при $x = 0$ і $x = L$)

$$u(0,t) = \psi(t),\ u(L,t) = \xi(t),\ t \ge 0. \tag{12.11}$$

В умовах (12.10), (12.11) $\varphi_1(x)$, $\varphi_2(x)$, $\psi(t)$, $\xi(t)$ – задані функції.

Граничні умови (12.11) також можуть бути задані у вигляді

$$\frac{\partial u(0,t)}{\partial x} = \psi(x),\ \frac{\partial u(L,t)}{\partial x} = \xi(x), \tag{12.12}$$

чи комбінації (12.11) і (12.12).

12.3. Контрольний приклад

Вимагається знайти стаціонарний розподіл температури $u(x,y)$ у квадратній однорідній пластині розміру 1 х 1, для якої задані такі граничні умови: $u(0,y) = 0$, $u(1,y) = 100$, $u(x,0) = 100x$, $u(x,1) = 100x^2$. Температура вимірюється у градусах Цельсія.

Розв'язання. Оскільки процес стаціонарний і немає внутрішніх джерел тепла, то розподіл температури в пластині описується рівнянням Лапласа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

Для того, щоб сформулювати і розв'язати задачу методом кінцевих різниць введемо на пластині двовимірну сітку з відстанню між вузлами $h = 0.25$ (рис. 12.1).

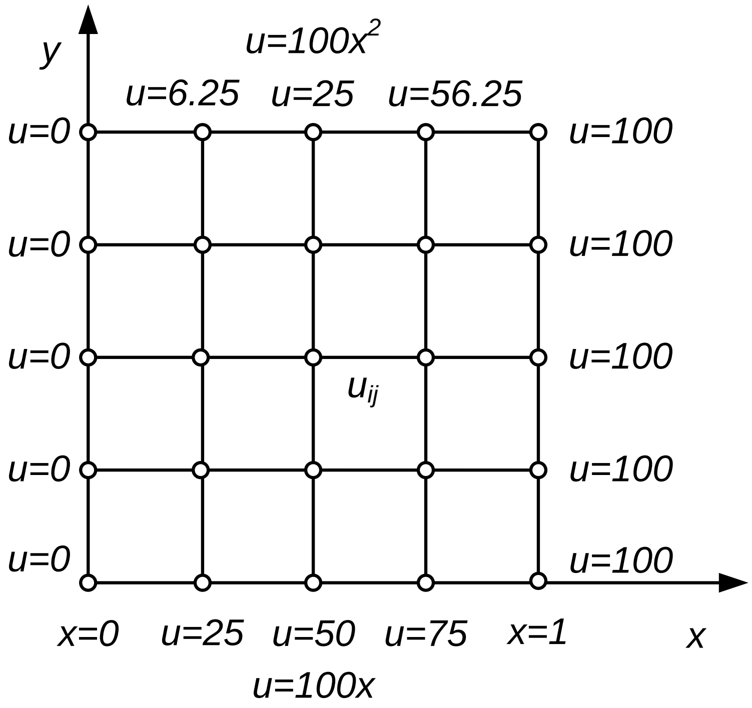


Рис. 12.1. **Двовимірна сітка на пластині**

Сітка складається з 25 вузлів, у 16 з яких температура відома з граничних умов. Таким чином, задача полягає у визначенні температури в 9 внутрішніх вузлах.

Замінюючи у вузлах (i, j) частинні похідні $\dfrac{\partial^2 u}{\partial x^2}, \dfrac{\partial^2 u}{\partial y^2}$ на симетричні кінцево-різницеві формули

$$\frac{\partial^2 u}{\partial x^2} \approx= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \quad \frac{\partial^2 u}{\partial y^2} \approx= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2},$$

отримуємо співвідношення

$$u_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4},$$

з якого і виходить формула для обчислень за методом ітерацій:

$$u_{i,j}^{(k+1)} = \frac{u_{i-1,j}^{(k)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} + u_{i,j+1}^{(k)}}{4}.$$

Початкові значення температури $u_{i,j}^{(0)}$ у вузлах (i, j) можна задати за допомогою лінійної інтерполяції.

12.4. Порядок виконання роботи

12.4.1. Зміст звіту

У теоретичній частині роботи необхідно коротко описати:

- постановку крайової задачі для диференційного рівняння в частинних похідних;
- метод кінцевих різниць.

У практичній частині роботи необхідно:

- виконати контрольний приклад;
- привести текст розробленої програми та результати її виконання;
- побудувати графік отриманого розв'язку задачі.

12.5. Контрольні запитання

1. Запишіть загальний вигляд диференціального рівняння в частинних похідних другого порядку з двома змінними. Які типи класифікації використовуються для цих рівнянь?
2. Приведіть класифікацію диференціальних рівнянь у частинних похідних другого порядку з двома змінними залежно від фізичного сенсу розв'язуваних з їх допомогою задач.
3. Приведіть класифікацію диференціальних рівнянь у частинних похідних другого порядку з двома змінними залежно від їх математичної природи.
4. Сформулюйте постановку задачі Діріхле для рівняння Пуассона.
5. Сформулюйте постановку задачі розв'язання диференціального рівняння в частинних похідних другого порядку з двома змінними для рівняння параболічного типу.

Використана література

1. Бахвалов Н. С. Численные методы / Бахвалов Н. С. – М. : Наука, 2000. – 286 с.

2. Волков Е. А. Численные методы / Волков Е. А. – М. : Высшая школа, 1987. – 311 с.

3. Демидович Б. П. Основы вычислительной математики / Демидович Б. П. – М. : Наука, 1994. – 664 с.

4. Задачин В. М. Чисельні методи в інформатиці: навчальний посібник / Задачин В. М., Конюшенко І. Г. – Х. : Вид. ХНЕУ, 2008. – 188 с.

5. Турчак Л. И. Основы численных методов / Турчак Л. И. – М. : Наука, 1997. – 320 с.

6. Фельдман Л. П. Чисельні методи в інформатиці / Фельдман Л. П., Петренко А. І., Дмитрієва О. А. – К. : Видавнича група BHV, 2006. – 480 с.

ЗМІСТ

Загальні положення 3
Позначення.. 5
Лабораторна робота № 1. Вступ у систему R 6
Лабораторна робота № 2. Чисельні методи розв'язання систем лінійних алгебраїчних рівнянь 17
Лабораторна робота № 3. Чисельні методи розв'язання нелінійних рівнянь з одним невідомим.. 27
Лабораторна робота № 4. Чисельні методи розв'язання систем нелінійних рівнянь.. 34
Лабораторна робота № 5. Чисельні методи знаходження власних значень і власних векторів матриці.. 42
Лабораторна робота № 6. Чисельні методи наближення функцій. Апроксимація та інтерполяція функцій 45
Лабораторна робота № 7. Чисельне диференціювання функцій............ 57
Лабораторна робота № 8. Чисельне інтегрування функцій.................... 62
Лабораторна робота № 9. Чисельні методи розв'язання задачі Коші для звичайних диференціальних рівнянь.. 67
Лабораторна робота № 10. Чисельні методи розв'язання лінійної крайової задачі для звичайних диференціальних рівнянь 2-го порядку. 76
Лабораторна робота № 11. Чисельні методи розв'язання інтегральних рівнянь.. 83
Лабораторна робота № 12. Методи математичної фізики. Розв'язання диференціальних рівнянь з частинними похідними 88
Використана література 93

НАВЧАЛЬНЕ ВИДАННЯ

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "ЧИСЕЛЬНІ МЕТОДИ"

для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання

Укладачі: **Задачин** Віктор Михайлович
Конюшенко Ірина Григорівна

Відповідальний за випуск **Пономаренко В. С.**

Редактор **Пушкар І. П.**

Коректор **Бриль В. О.**

План 2012 р. Поз. № 295.
Підп. до друку Формат 60×90 1/16. Папір MultiCopy. Друк Riso.
Ум.-друк. арк. 6,0. Обл.-вид. арк. 7,5. Тираж прим. Зам. №

Видавець і виготівник – видавництво ХНЕУ, 61166, м. Харків, пр. Леніна, 9а

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № 481 від 13.06.2001 р.

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни

"ЧИСЕЛЬНІ МЕТОДИ"

для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання