8. Розв’язання задачі Коші для звичайних диференціальних рівнянь

8.1. Постановка задачі Коші

Задача Коші для звичайних диференціальних рівнянь використовується як математична модель при розв'язанні багатьох задач природознавства. Наприклад, задачі динаміки системи взаємодіючих тіл (у моделі руху матеріальних точок), задачі хімічної кінетики, електричних ланцюгів. Ряд важливих рівнянь у частинних похідних у випадках, що допускають розділення змінних, приводить до задач для звичайних диференціальних рівнянь. Це, як правило, крайові задачі (задачі про власні коливання пружних балок і пластин, визначення спектра власних значень енергії частинки у сферично-симетричних полях і багато інших).

Задача Коші для диференціального рівняння n-го порядку

$$y^{(n)} = f(x, y, y', y'', ..., y^{(n-1)}) \tag{8.1}$$

полягає у відшуканні функції $y = y(x)$, що задовольняє це рівняння і початкові умови:

$$y(x_0) = y_0,\ y'(x_0) = y'_0,\ \ldots,\ y^{(n-1)}(x_0) = y_0^{(n-1)}, \tag{8.2}$$

де $x_0, y_0, y'_0, y'', ..., y_0^{(n-1)}$ – задані числа.

Наприклад, можна розглянути 2-й закон Ньютона: в інерціальній системі відліку прискорення, яке отримує матеріальна точка з постійною масою, прямо пропорційно рівнодіючій всіх доданих до неї сил і обернено пропорційно її масі. Цей закон може бути записаний у вигляді формули

$$a = \frac{F}{m},$$

де a – прискорення матеріальної точки;

F – рівнодіюча всіх сил;

m – маса матеріальної точки.

Якщо припустити, що напрям сили не змінюється і позначити через y координату точки на прямій вздовж руху, то $a = y''$ і формула 2-го закону Ньютона приймає вигляд $a'' = \frac{F}{m}$, тобто є окремим випадком формули (8.1). Якщо відомі положення точки та її швидкість у початковий момент часу (8.2) і рівнодіюча сила F, то розв'язавши задачу Коші можна визначти її положення і швидкість у будь-який наступний момент часу.

Задача Коші для системи диференціальних рівнянь

$$\begin{cases} \dfrac{dy_1}{dx} = f_1(x, y_1, y_2, ..., y_m), \\ \dfrac{dy_2}{dx} = f_2(x, y_1, y_2, ..., y_m), \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ \dfrac{dy_m}{dx} = f_m(x, y_1, y_2, ..., y_m) \end{cases} \tag{8.3}$$

полягає у відшуканні функцій $y_1 = y_1(x)$, $y_2 = y_2(x)$, .., $y_m = y_m(x)$, що задовольняють цю систему і початкові умови

$$y_j(x_0) = y_{j0}, \ j = \overline{1,m}, \tag{8.4}$$

де x_0, y_{j0}, $j = \overline{1,m}$, – задані числа.

Систему, що містить похідні вищих порядків і розв'язану відносно старших похідних шуканих функцій, шляхом введення нових невідомих функцій можна привести до вигляду (8.3). Зокрема, диференціальне рівняння n-го порядку (8.1) приводиться до вигляду (8.3) за допомогою такої заміни:

$$y_1 = y, \ y_2 = y', \ y_3 = y'', \ \dots, \ y_n = y^{(n-1)},$$

що дає систему

$$\begin{cases} \dfrac{dy_1}{dx} = y_2, \\ \dfrac{dy_2}{dx} = y_3, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \dfrac{dy_n}{dx} = f(x, y_1, y_2, ..., y_n). \end{cases} \qquad (8.5)$$

Таким чином, задача (8.1), (8.2) є окремим випадком задачі (8.3), (8.4), тому чисельні методи розв'язання задачі Коші розроблені для більш загальної задачі (8.3), (8.4).

Варто зазначити, що в більшості практичних задач змінна x – це час, тобто в (8.3), (8.4) замість x можна застосовувати і позначення t.

Для задачі (8.3), (8.4) вводяться позначення:

$$Y = \begin{pmatrix} y_1(x) \\ y_2(x) \\ \cdots \\ y_m(x) \end{pmatrix}, \; Y_0 = \begin{pmatrix} y_{10} \\ y_{20} \\ \cdots \\ y_{m0} \end{pmatrix}, \; Y' = \begin{pmatrix} y_1'(x) \\ y_2'(x) \\ \cdots \\ y_m'(x) \end{pmatrix},$$

$$F(x,Y) = \begin{pmatrix} f_1(x, y_1, y_2, \ldots, y_m) \\ f_2(x, y_1, y_2, \ldots, y_m) \\ \cdots \\ f_m(x, y_1, y_2, \ldots, y_m) \end{pmatrix},$$

де Y – шуканий розв'язок;

Y_0 – вектор початкових умов;

$F(x,Y)$ – вектор правих частин системи (8.3).

Тоді задача Коші для системи диференціальних рівнянь у векторній формі має вигляд:

$$Y' = F(x,Y), \; Y(x_0) = Y_0. \qquad (8.6)$$

Надалі для простоти викладення буде розглянуто задачу Коші для одного звичайного диференціального рівняння виду

$$y' = f(x, y), \tag{8.7}$$

де y – скалярна змінна.

При цьому **задача Коші** полягає в такому: знайти функцію $y = y(x)$ на заданому відрізку $[a, b]$, що задовольняє рівнянню (8.7) і початковій умові

$$y(a) = y_0, \tag{8.8}$$

де y_0 задане.

Чисельні методи, що будуть розглянуті далі для задачі (8.7), (8.8), мають місце і для загальної задачі (8.6).

Чисельні методи розв'язання задачі (8.7), (8.8) 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 $x_0 = a$, $x_i = x_0 + i\,h$, $i = \overline{1, n}$, $h = \dfrac{b-a}{n}$, n – задане число розбиття відрізка $[a, b]$, y_i, $i = \overline{1, n}$ – знайдені наближені значення функції $y(x)$ в точках x_i (слід нагадати, що y_0 задане спочатку).

Варто розглянути декілька методів розв'язання задачі Коші для диференціального рівняння (8.7) на відрізку $[a, b]$.

8.2. Метод Ейлера та його модифікації

Ідея методу Ейлера заснована на тому, що шукана інтегральна крива $y = y(x)$, яка проходить через точку $M_0(x_0, y_0)$, відновлюється у вигляді кусочно-лінійної ламаної $M_0 M_1 M_2 \ldots M_n$ з вершинами $M_i(x_i, y_i)$ ($i = \overline{0, n}$) (рис. 8.1). Кожен відрізок $M_i M_{i+1}$ цієї ламаної має напрям, що співпадає з напрямом тієї інтегральної кривої рівняння (8.7), яка проходить через точку M_i.

Тоді $y_{i+1} - y_i = h \times tg(\alpha)$. Тангенс кута нахилу дотичної до $y(x)$ в точці x_i дорівнює $y'(x_i)$, а значить, згідно з (8.7), і дорівнює $f(x_i, y_i)$, звідки і випливає формула методу Ейлера.

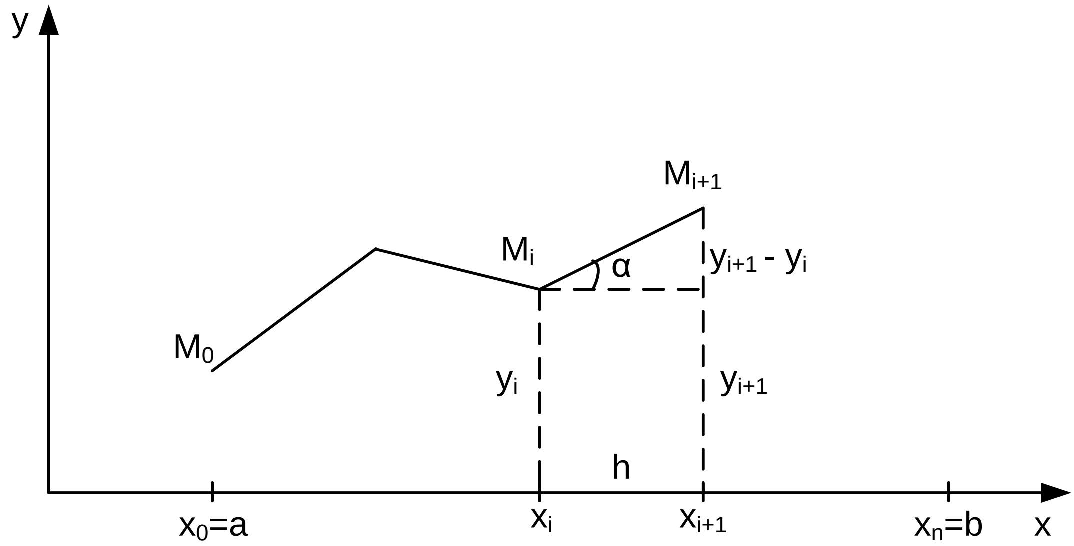


Рис. 8.1. **Графічна інтерпретація методу Ейлера**

Таким чином, у методі Ейлера значення y_i обчислюються рекурентно за формулою

$$y_{i+1} = y_i + h \times f(x_i, y_i),\ i = \overline{0, n-1}. \tag{8.9}$$

Погрішність методу Ейлера [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**], що оцінюється для величини $|y_n - y(x_n)|$, має порядок $O(h)$, тобто існує деяка константа $M > 0$ така, що

$$|y_n - y(x_n)| \leq M\,h. \tag{8.10}$$

Тут $y(x)$ – точний розв'язок задачі Коші (8.7), (8.8), а порівняння з наближеним розв'язком (x_i, y_i), $i = \overline{0, n}$, йде в крайній (правій) точці $x_n = b$ відрізка $[a, b]$, оскільки саме в ній погрішність теоретично буде максимальною, відносно проміжних точок x_i.

Варто зазначити, що оцінка (8.10) носить лише теоретичний характер, на практиці ж для оцінки отриманого розв'язку можна скористатися подвійним прорахунком. Для цього проводять повторні обчислення, але вже з кроком $\frac{h}{2}$, отримують вже точніший наближений розв'язок (x_i^*, y_i^*), $i=\overline{0,n^*}$, ($n^* = 2n$) і погрішність наближення оцінюють як

$$\left|y_{n^*}^* - y(x_{n^*})\right| \approx \left|y_{n^*}^* - y_n\right|.$$

Таким чином, для того щоб отримати розв'язок задачі Коші (8.7), (8.8) із заданою точністю $\varepsilon > 0$ (застосовуючи метод Ейлера) можна, починаючи з деякого початкового значення кроку h, проводити подвійний прорахунок (тобто зменшуючи вдвічі h) допоки не виконається нерівність $\left|y_{n^*}^* - y_n\right| \le \varepsilon$.

Приклад 8.1. Розв'язати методом Ейлера задачу Коші для диференціального рівняння $y' = x + y$ на відрізку $[0,\ 5]$, якщо $y(0)=1$.

Розв'язання в математичному пакеті R

Відправними даними задачі є диференціальне рівняння $F(x,y)$, межі відрізка A і B, початкове значення $Y0$, число розбиття n відрізка $[a,b]$:

```
# задаємо рівняння (функцію правої частини)
FunPr = function(x,y)
{
  return (x + y)
}
a = 0
b = 5
y0 = 1
n = 20
```

Процедура обчислення для розв'язання задачі Коші для диференціального рівняння методом Ейлера може бути записана так:

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

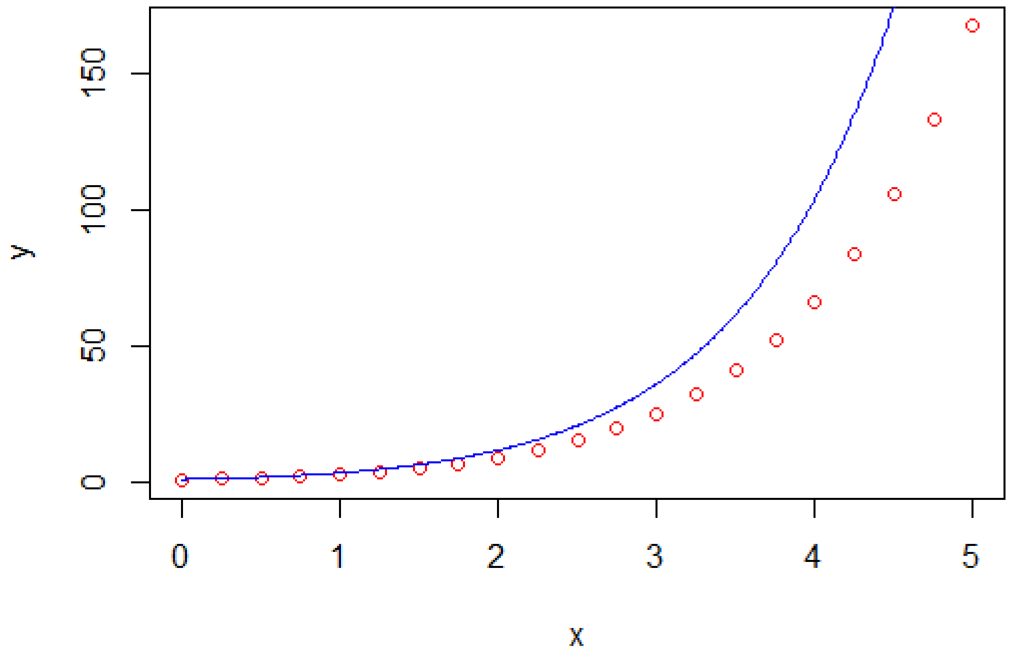
Результат обчислення з використанням записаної процедури:

```
> xy = MetEiler(FunPr, a, b, y0, n)
> xy
         x          y
 [1,] 0.00   1.000000
 [2,] 0.25   1.250000
 [3,] 0.50   1.625000
 [4,] 0.75   2.156250
 [5,] 1.00   2.882812
 [6,] 1.25   3.853516
 [7,] 1.50   5.129395
 [8,] 1.75   6.786743
 [9,] 2.00   8.920929
[10,] 2.25  11.651161
[11,] 2.50  15.126451
[12,] 2.75  19.533064
[13,] 3.00  25.103830
[14,] 3.25  32.129788
[15,] 3.50  40.974735
[16,] 3.75  52.093419
[17,] 4.00  66.054274
[18,] 4.25  83.567842
[19,] 4.50 105.522302
[20,] 4.75 133.027878
[21,] 5.00 167.472348
```

Аналітичним розв'язком даного диференціального рівняння є функція $YT(x) = 2e^{x} - x - 1$. Порівняти результати розв'язку, отримані чисельно методом Ейлера й аналітично, можна побудувавши графік:

```
x = xy[, 1]
y = xy[, 2]

# Будуємо графік чисельного розв'язку
plot(x, y, type = "p", col="red")
# Аналітичний розв'язок задачі
FunAn = function(x)
{
  return (2*exp(x) - x - 1)
}
x1 = seq(x[1], x[n+1], by=0.01)
y1 = FunAn(x1)
# додаємо в 1-у графічну підобласть ще лінію
lines(x1, y1, col="blue")
```



Як видно з наведеної оцінки погрішності (8.10), метод Ейлера дає невисоку точність при розв'язанні задачі Коші (8.7), (8.8), тому його рідко застосовують на практиці. Слід розглянути дві його модифікації.

1-а модифікація методу Ейлера. У 1-й модифікації методу Ейлера значення y_i обчислюються рекурентно за формулами $(i = \overline{0, n-1})$:

$$y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x_i, y_i); \tag{8.11}$$

$$y_{i+1} = y_i + h \times f(x_i + \frac{h}{2}, y_{i+\frac{1}{2}}). \tag{8.12}$$

Ідея цієї модифікації методу Ейлера полягає в тому, що спочатку обчислюється значення $y_{i+\frac{1}{2}}$ за формулою (8.11) у проміжній точці $x_{i+\frac{1}{2}} = x_i + \dfrac{h}{2}$ (середині) інтервала (x_i, x_{i+1}), а потім вже отримують y_{i+1} в точці x_{i+1} за формулою (8.12).

Погрішність 1-ї модифікації метода Ейлера [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**], що оцінюється для величини $|y_n - y(x_n)|$, має порядок $O(h^3)$, тобто існує деяка константа $M > 0$ така, що $|y_n - y(x_n)| \le Mh^3$.

Як видно з наведеної оцінки погрішність 1-ої модифікації метода Ейлера на два порядки менша, ніж у звичайного метода Ейлера, хоча за це доводиться платити додатковими обчисленнями. А саме: функція $f(x, y)$ з правої частини диференціального рівняння (8.7) обчислюється на кожному кроці 1-ї модифікації (8.11), (8.12) двічі. Варто зазначити, що при розв'язанні практичних задач основні обчислювальні витрати припадатимуть саме на обчислення значень функції $f(x, y)$.

Приклад 8.2. Розв'язати задачу з прикладу 8.1, застосувавши 1-у модифікацію методу Ейлера.

Розв'язання в математичному пакеті R

Відправні дані – ті ж самі, що й у прикладі 8.1.

Процедура обчислення для розв'язання задачі Коші для диференціального рівняння з застосуванням 1-ї модифікації методу Ейлера може бути записана так:

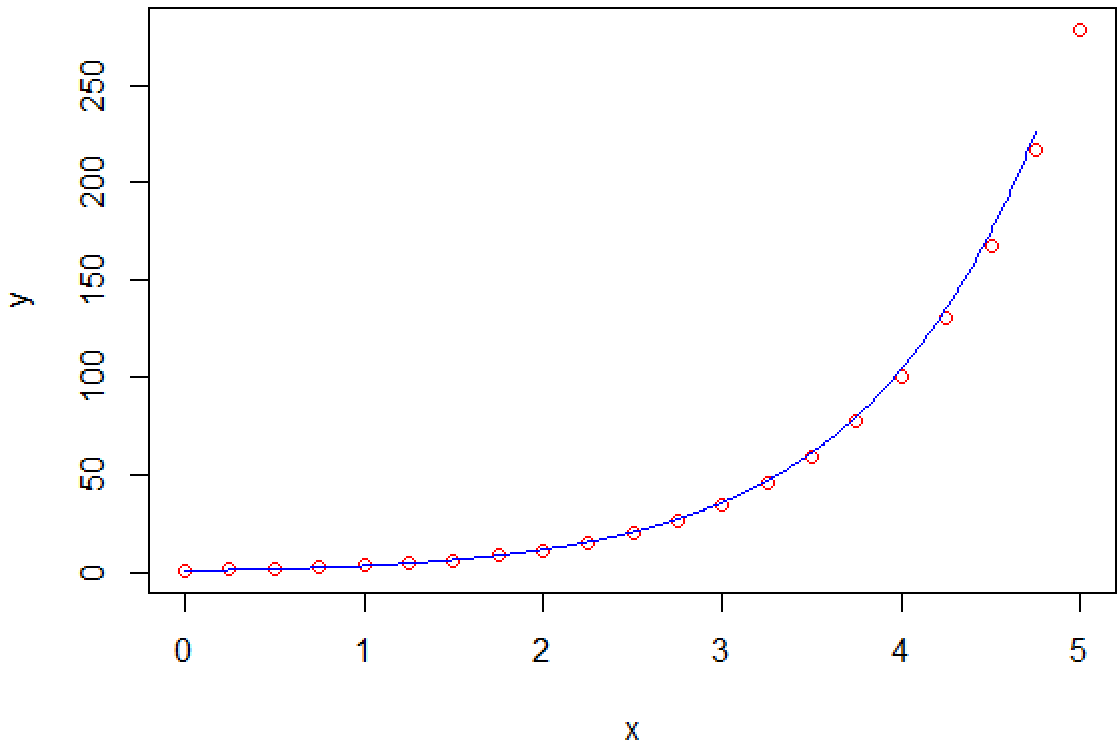
```
MetEiler1 = function(f, a, b, y0, n)
{
  h = (b-a)/n
  x = c(1:(n+1))
  y = c(1:(n+1))
  x[1] = a
  y[1] = y0
  for (i in 1:n)
  {
    x[i+1] = x[i] + h
    ypr = y[i] + h/2*f(x[i],y[i])
    y[i+1] = y[i] + h*f(x[i]+h/2,ypr)
  }
  return(cbind(x,y))
}
```

Результат обчислення з використанням записаної процедури:

```
> xy = MetEiler1(FunPr, a, b, y0, n)
> xy
         x          y
 [1,] 0.00   1.000000
 [2,] 0.25   1.312500
 [3,] 0.50   1.783203
 [4,] 0.75   2.456604
 [5,] 1.00   3.389711
 [6,] 1.25   4.655568
 [7,] 1.50   6.347759
 [8,] 1.75   8.586191
 [9,] 2.00  11.524494
[10,] 2.25  15.359508
[11,] 2.50  20.343433
[12,] 2.75  26.799398
[13,] 3.00  35.141416
[14,] 3.25  45.899940
[15,] 3.50  59.754610
[16,] 3.75  77.576219
[17,] 4.00 100.480468
[18,] 4.25 129.896850
[19,] 4.50 167.656902
[20,] 4.75 216.107281
[21,] 5.00 278.254641
```

Порівняти результати розв'язку, отримані чисельно з застосуванням 1-ї модифікації методу Ейлера й аналітично, можна побудувавши графік:

```
x = xy[, 1]
y = xy[, 2]
# Будуємо графік чисельного розв'язоку
plot(x, y, type = "p", col="red")
# Аналітичний розв'язок задачі
FunAn = function(x)
{
  return (2*exp(x) - x - 1)
}
x1 = seq(x[1], x[n], by=0.01)
y1 = FunAn(x1)
# додаємо в 1-у графічну підобласть ще лінію
lines(x1, y1, col="blue")
```



Як видно з цього графіка, 1-а модифікація методу Ейлера дає більш точний роз'язок задачі, ніж класичний метод Ейлера.

2-а модифікація методу Ейлера. У 2-й модифікації методу Ейлера значення y_i обчислюються рекурентно за формулами $(i = \overline{0, n-1})$:

$$y^*_{i+1} = y_i + h \times f(x_i, y_i); \qquad (8.13)$$

$$y_{i+1} = y_i + \frac{h}{2}(f(x_i, y_i) + f(x_{i+1}, y^*_{i+1})). \qquad (8.14)$$

Ідея цієї модифікації методу Ейлера полягає в тому, що спочатку обчислюється "грубе наближення" y^*_{i+1} за формулою (8.13) і за ним обчислюють значення $f(x_{i+1}, y^*_{i+1})$, а потім вже "остаточне наближення"

y_{i+1} отримують за формулою (8.14), в якому фігурує середнє значення $\frac{1}{2}(f(x_i, y_i) + f(x_{i+1}, y^*_{i+1}))$.

Погрішність 2-ї модифікації методу Ейлера, що оцінюється для величини $|y_n - y(x_n)|$, також має порядок $O(h^3)$, тобто існує деяка константа $M > 0$ така, що $|y_n - y(x_n)| \le Mh^3$. Значення функції $f(x, y)$з правої частини диференціального рівняння (8.7) також обчислюються двічі на кожному кроці цієї модифікації (8.13), (8.14).

8.3. Метод Рунге – Кутта четвертого порядку

Ідея методу Рунге – Кутта заснована на застосуванні, на відміну від методу Ейлера (кусочно-лінійна ламана), кривих вищого порядку для відновлення значень шуканої функції $y(x)$ на відрізку $[a, b]$.

Найчастіше при розв'язанні практичних задач (8.7), (8.8) використовується **метод Рунге – Кутта четвертого порядку**. Згідно з цим методом значення y_i обчислюються рекурентно за формулою

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),\ i = \overline{0, n-1}, \qquad (8.15)$$

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

Погрішність методу Рунге – Кутта 4-го порядку, що оцінюється для величини $|y_n - y(x_n)|$, має порядок $O(h^4)$. Як видно, погрішність при використанні цього методу нижча, ніж при використанні методу Ейлера і його модифікацій, але за це доводиться платити додатковими обчисленнями. А саме: функція $f(x, y)$ з правої частини диферен-ціального рівняння (8.7) обчислюється на кожному кроці методу Рунге – Кутта 4-го порядку (8.15) чотири рази, що у випадках, коли обчислення значення функції $f(x, y)$ є дуже трудомістким, може істотно уповільнити пошук розв'язку.

Приклад 8.3. Розв'язати задачу з прикладу 8.1 методом Рунге – Кутта 4-го порядку.

Розв'язання в математичному пакеті R

Відправні дані – ті ж самі, що й у прикладі 8.1.

Процедура обчислення для розв'язання задачі Коші для диференціального рівняння методом Рунге – Кутта 4-го порядку може бути записана так:

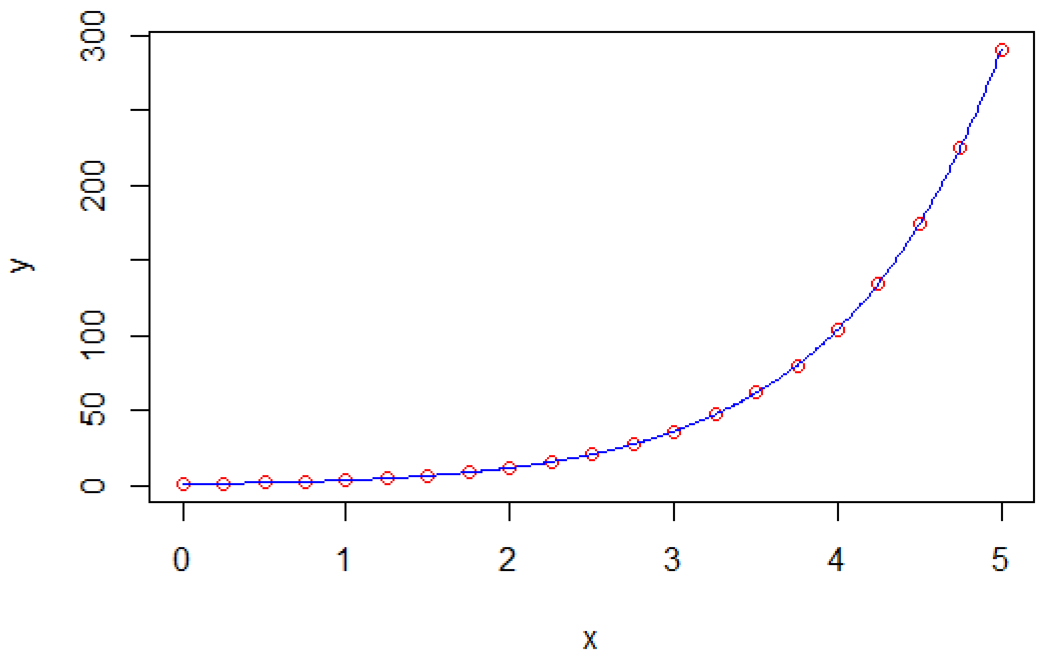
```
MetRungKut = function(f, a, b, y0, n)
{
  h = (b-a)/n
  x = c(1:(n+1))
  y = c(1:(n+1))
  x[1] = a
  y[1] = y0
  for (i in 1:n)
  {
    x[i+1] = x[i] + h
    k1 =  h*f(x[i],y[i])
    k2 =  h*f(x[i]+h/2,y[i]+k1/2)
    k3 =  h*f(x[i]+h/2,y[i]+k2/2)
    k4 =  h*f(x[i]+h,y[i]+k3)
    y[i+1] = y[i] + 1/6*(k1 + 2*k2 + 2*k3 + k4)
  }
  return(cbind(x,y))
}
```

Результат обчислення з використанням записаної процедури:

```
> xy = MetRungKut(FunPr, a, b, y0, n)
> xy
         x          y
 [1,] 0.00   1.000000
 [2,] 0.25   1.318034
 [3,] 0.50   1.797399
 [4,] 0.75   2.483916
 [5,] 1.00   3.436420
 [6,] 1.25   4.730455
 [7,] 1.50   6.463023
 [8,] 1.75   8.758673
 [9,] 2.00  11.777331
[10,] 2.25  15.724343
[11,] 2.50  20.863377
[12,] 2.75  27.532989
[13,] 3.00  36.167887
[14,] 3.25  47.326247
[15,] 3.50  61.724774
[16,] 3.75  80.283730
[17,] 4.00 104.184749
[18,] 4.25 134.945066
[19,] 4.50 174.512838
[20,] 4.75 225.389531
[21,] 5.00 290.787070
```

Порівняти результати розв'язку, отримані чисельно методом Рунге – Кутта 4-го порядку й аналітично, можна побудувавши графік:

```
> x = xy[, 1]
> y = xy[, 2]
> # Будуємо графік чисельного розв'язку
> plot(x, y, type = "p", col="red")
> # Аналітичний розв'язок задачі
> FunAn = function(x)
+ {
+   return (2*exp(x) - x - 1)
+ }
> x1 = seq(x[1], x[n+1], by=0.01)
> y1 = FunAn(x1)
> # додаємо в 1-у графічну підобласть ще лінію
> lines(x1, y1, col="blue")
```



Порівняння графіків розв'язків, отриманих чисельно в прикладах 8.1 – 8.3, з аналітичним розв'язком ще раз доводить, що з розглянутих методів розв'язання задачі Коші для диференціального рівняння найточнішим є метод методом Рунге – Кутта 4-го порядку (чисельно отриманий розв'язок збігається з аналітичним), а найгіршим – метод Ейлера.

8.4. Висновки

1. Математичні моделі процесів та явищ, зокрема моделі динамічних систем, у більшості випадків записуються у вигляді диференціальних рівнянь.
2. Задача Коші для звичайних диференціальних рівнянь має велике практичне значення.

8.5. Контрольні запитання та завдання

1. Що називається звичайним диференціальним рівнянням?
2. Сформулюйте постановку задачі Коші для звичайного диференціального рівняння. Що є її розв'язком? У якому вигляді подається розв'язок чисельним методом?
3. Який з відомих вам методів розв'язання задачі Коші для звичайного диференціального рівняння треба застосовувати в тих чи інших випадках?
4. Розв’яжіть задачу Коші для диференціального рівняння $y' = \dfrac{x+y-3}{x-y-1}$ на відрізку $[2, 25]$ (початкова умова $y(2)=1$) чисельно

методами Ейлера і його модифікаціями й методом Рунге – Кутта із числом розбиття відрізка $n = 20$. Побудуйте графіки отриманих розв’язків.

9. Багатокрокові методи розв'язання звичайних диференціальних рівнянь

9.1. Поняття багатокрокового методу

Розглянуті у розділі 10 чисельні методи розв'язання задачі Коші для звичайних диференціальних рівнянь відносяться до **однокрокових методів** тому, що при розрахунку поточного значення y_{i+1} на i-му кроці використовується тільки інформація на останньому відрізку $\left[x_i, x_{i+1}\right]$. Все, що робилось на попередніх кроках методу, явно не використовується. Навпаки, є **багатокрокові методи** розв’язання задачі Коші (8.7), які використовують те, що було отримано на попередніх кроках методу явно. Такими є, наприклад, методи: Адамса – Бошфорда, Адамса – Мултона, Гира – Брайтона [**Ошибка! Источник ссылки не найден.**].

9.2. Метод Адамса – Бошфорда

В основі методу Адамса – Бошфорда лежить формула Ньютона – Лейбніца:

$$\int_a^b f(x)dx = \Phi(b) - \Phi(a),$$

де $\Phi(x)$ – будь-яка первісна для підінтегральної функції $f(x)$ на відрізку $\left[a,b\right]$.

Застосування формули Ньютона – Лейбніца до рівняння (8.7) на будь-якому відрізку $\left[x_i, x_{i+1}\right]$ приводить до рівняння:

$$y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} f(x, y(x))dx.$$

Тоді, якщо вже були отримані значення $y_{i-m+1}, y_{i-m+2}, \ldots, y_{i-1}, y_i$ в m попередніх точках (вузлах) $x_{i-m+1}, x_{i-m+2}, \ldots, x_{i-1}, x_i$, то підінтегральну функцію $f(x, y(x))$ можна замінити інтерполяційним поліномом Ньютона $H_{m-1}(x)$ (див. розділ 7.5), побудованим за цими m вузлами. Таким чином, отримуємо загальну рекурентну формулу методу Адамса – Бошфорда m-го порядку

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} H_{m-1}(x)dx. \qquad (9.1)$$

Оскільки $H_{m-1}(x)$ – поліном, то інтеграл від нього береться аналітично і таким чином отримують різницеву схему розв'язання задачі Коші для будь-якого m. Наприклад, якщо $m=1$, то $H_{m-1}(x)=f(x_i,y_i)$ і з (9.1) витікає

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

де $h = x_{i+1} - x_i$.

Таким чином, при $m=1$ метод Адамса – Бошфорда співпадає з методом Ейлера (див. розділ 10.2).

Найчастіше застосовують метод Адамса – Бошфорда 4-го порядку, який має таку різницеву схему [**Ошибка! Источник ссылки не найден.**]:

$$y_{i+1} = y_i + \frac{h}{24}(\,55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}), \qquad (9.2)$$

де $f_i = f(x_i,y_i)$.

Очевидно, що за формулою (9.2) можна проводити обчислення тільки при $i \geq 3$, тому що на i-му кроці методу треба знати значення $f_{i-3}, f_{i-2}, f_{i-1}, f_i$. Тому перш ніж застосовувати метод Адамса – Бошфорда 4-го порядку, необхідно спочатку обчислити значення f_0, f_1, f_2 для перших 3-х кроків. Зазвичай для цього застосовують метод Рунге – Кутта, бо він має достатньо велику точність.

Погрішність методу Адамса – Бошфорда 4-го порядку, що оцінюється для величини $|y_n - y(x_n)|$, має порядок $O(h^4)$ [**Ошибка! Источник ссылки не найден.**].

Приклад 9.1. Розв'язати задачу з прикладу 8.1 методом Адамса – Бошфорда 4-го порядку.

Розв'язання в математичному пакеті R

Відправні дані – ті ж самі, що й у прикладі 8.1.

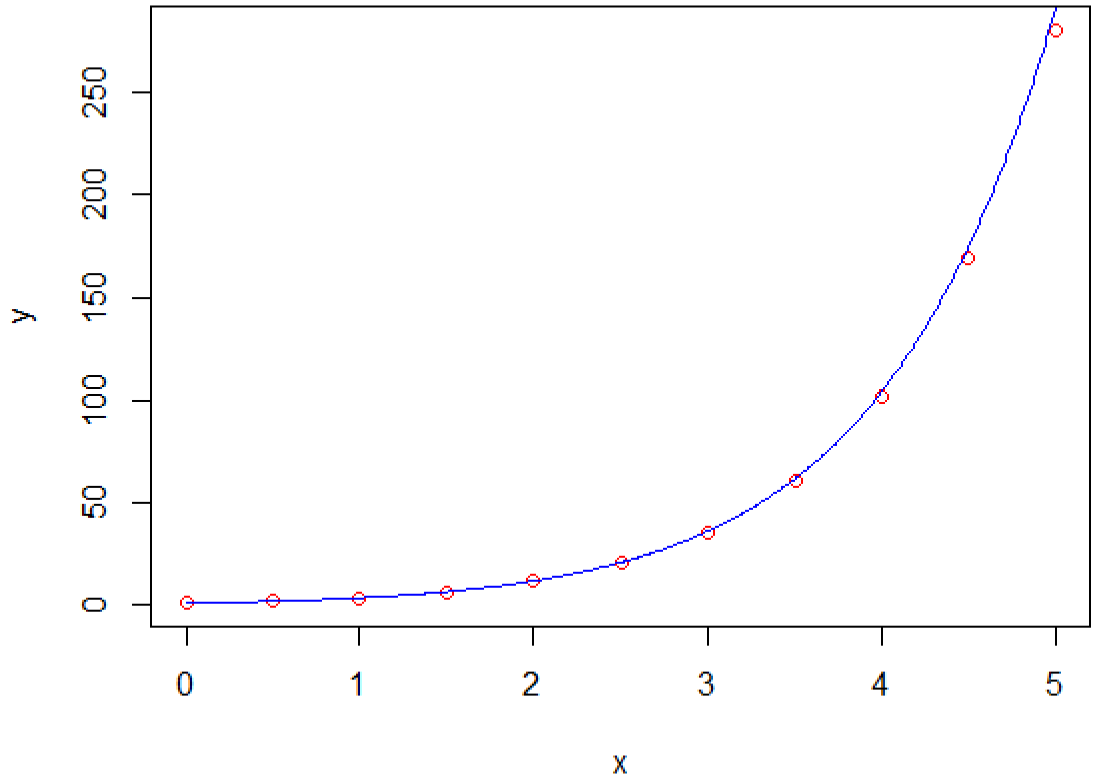
```
# задаємо рівняння (функцію правої частини)
FunPr = function(x,y)
{
  return (x + y)
}
a = 0
b = 5
y0 = 1
```

Процедура обчислення для розв'язання задачі Коші для диференціального рівняння методом Адамса – Бошфорда 4-го порядку може бути записана так:

```
MetAdamBosh4 = function(fun, a, b, y0, n)
{
  h = (b-a)/n
  x = c(1:(n+1))
  y = c(1:(n+1))
  f = c(1:(n+1))
  x[1] = a
  y[1] = y0
  f[1] =  fun(x[1], y[1])
  for (i in 1:3)
  {
    x[i+1] = x[i] + h
    k1 =  h*f[i]
    k2 =  h*fun(x[i]+h/2, y[i]+k1/2)
    k3 =  h*fun(x[i]+h/2, y[i]+k2/2)
    k4 =  h*fun(x[i]+h, y[i]+k3)
    y[i+1] = y[i] + 1/6*(k1 + 2*k2 + 2*k3 + k4)
    f[i+1] =  fun(x[i+1], y[i+1])
  }
  for (i in 4:n)
  {
    x[i+1] = x[i] + h
    y[i+1] = y[i] + h/24*(55*f[i] - 59*f[i-1] +
               + 37*f[i-2] - 9*f[i-3])
    f[i+1] =  fun(x[i+1], y[i+1])
  }
  return(cbind(x,y))
}
```

Результат обчислення з використанням записаної процедури та порівняння отриманого й аналітичного розв'язку:

```
> # Розв'язуємо систему диференційних рівнянь
> n = 10
> xy = MetAdamBosh4(FunPr, a, b, y0, n)
> x = xy[, 1]
> y = xy[, 2]
> # Будуємо графік чисельного розв'язоку
> plot(x, y, type = "p", col="red")
> # Аналітичний розв'язок задачі
> FunAn = function(x)
+ {
+   return (2*exp(x) - x - 1)
+ }
> x1 = seq(x[1], x[n+1], by=0.01)
> y1 = FunAn(x1)
> # додаємо в 1-у графічну підобласть ще лінію
> lines(x1, y1, col="blue")
```



Порівняння графіків розв'язків, отриманих чисельно в прикладах 8.3 і 9.1, з аналітичним розв'язком ще раз доводить, що з розглянутих методів розв'язання задачі Коші для диференціального рівняння метод Рунге–Кутта 4-го порядку і метод Адамса – Бошфорда 4-го порядку дають майже однакові результати.

9.3. Метод Адамса – Мултона

Слід зазначити, що в методі Адамса – Бошфорда інтерполяційний поліномом Ньютона $H_{m-1}(x)$ застосовується для екстраполяції функції $f(x, y(x))$ на відрізку $[x_i, x_{i+1}]$, оскільки він будується за вузлами $x_{i-m+1}, x_{i-m+2}, \ldots, x_{i-1}, x_i$. У методі Адамса – Мултона також застосовується інтерполяційний поліномом Ньютона $H_{m-1}(x)$, але не для екстраполяції, а для інтерполяції функції $f(x,y(x))$на відрізку $[x_i, x_{i+1}]$, оскільки він будується за вузлами $x_{i-m+2}, x_{i-m+3}, \ldots, x_i, x_{i+1}$.

Тому для метода Адамса – Мултона 4-го порядку отримано таку різницеву схему [**Ошибка! Источник ссылки не найден.**]:

$$y_{i+1} = y_i + \frac{h}{24}(\, 9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}\,), \tag{9.3}$$

де $f_{i+1} = f(x_{i+1}, y_{i+1})$.

На відміну від схеми (9.2), в схемі (9.3) у правій частині фігурує ще не відоме значення y_{i+1}. Такі методи носять назву **неявних методів**. Тому для пошуку y_{i+1} треба розв'язати нелінійне рівняння (9.3) відносно y_{i+1}. Слід зазначити, що в загальній постановці задачі Коші виду (8.3), (8.4) при застосуванні методу Адамса – Мултона доведеться на кожному кроці розв'язувати систему нелінійних рівнянь.

Погрішність методу Адамса – Мултона 4-го порядку, що оцінюється для величини $|y_n - y(x_n)|$, має порядок $O(h^5)$ [**Ошибка! Источник ссылки не найден.**].

9.4. Метод прогнозу та корекції

Різницеву схему (9.3) застосовують і для уточнення значення y_{i+1}, що було розраховане за різницевою схемою (9.2). Така комбінація методів Адамса – Бошфорда і Адамса – Мултона 4-го порядку має назву **методу прогнозу та корекції** [**Ошибка! Источник ссылки не найден.**].

У багатокроковому **методі прогнозу та корекції** 4-го порядку у ході розв'язання задачі Коші (8.7), (8.8) значення y_i у вузлах сітки x_i, $i = \overline{1,n}$, обчислюються рекурентно за формулами:

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

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

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

де $f_i = f(x_i, y_i)$.

Похибка методу прогнозу та корекції 4-го порядку, оцінювана для величини $|y_n - y(x_n)|$, має порядок $O(h^5)$ [**Ошибка! Источник ссылки не найден.**; **Ошибка! Источник ссылки не найден.**].

Якщо порівнювати однокрокові та багатокрокові методи розв'язання задачі Коші для звичайних диференціальних рівнянь, треба підкреслити, що в багатокрокових методах крок h можна обирати більшим, ніж у однокрокових методах. Це дає можливість значно зменшити кількість кроків, а значить і трудомісткість розв’язання задачі в цілому.

Приклад 9.2. Розв'язати задачу з прикладу 8.1 методом прогнозу та корекції 4-го порядку.

Розв'язання в математичному пакеті R

Відправні дані – ті ж самі, що й у прикладі 8.1.

```
# задаємо рівняння (функцію правої частини)
FunPr = function(x,y)
{
  return (x + y)
}
a = 0
b = 5
y0 = 1
```

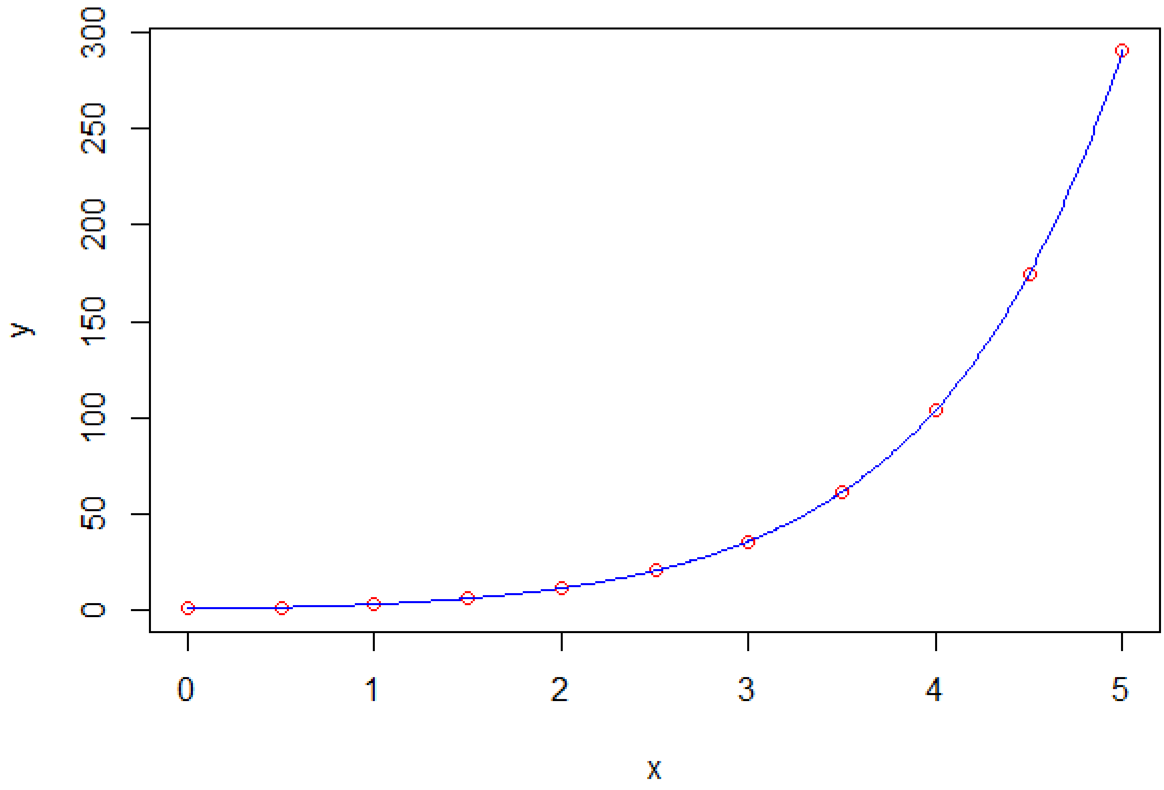
Процедура обчислення для розв'язання задачі Коші для диференціального рівняння методом прогнозу та корекції 4-го порядку може бути записана так:

```
MetPredCor4 = function(fun, a, b, y0, n)
{
     h = (b-a)/n
     x = c(1:(n+1))
     y = c(1:(n+1))
     f = c(1:(n+1))
     x[1] = a
     y[1] = y0
     f[1] = fun(x[1], y[1])
     for(i in 1:3)
     {
          x[i+1] = x[i] + h
          k1 = h*f[i]
          k2 = h*fun(x[i]+h/2, y[i]+k1/2)
          k3 = h*fun(x[i]+h/2, y[i]+k2/2)
          k4 = h*fun(x[i]+h, y[i]+k3)
          y[i+1] = y[i]+(1/6)*(k1+2*k2+2*k3+k4)
          f[i+1] = fun(x[i+1], y[i+1])
     }
     for(i in 4:n)
     {
          x[i+1] = x[i] + h
          ypred = y[i] + h/24*(55*f[i] - 59*f[i-1] + 37*f[i-2] - 9*f[i-3])
          fpred = fun(x[i+1], ypred)
          y[i+1] = y[i] + h/24*(9*fpred + 19*f[i]- 5*f[i-1] + f[i-2])
          f[i+1] = fun(x[i+1], y[i+1])
     }
     return(cbind(x, y))
}
```

Результат обчислення з використанням записаної процедури та порівняння отриманого й аналітичного розв'язку:

```
> n = 10
>
> # Розв'язуємо задачу Коші для диференціального рівняння
> xy = MetPredCor4(FunPr, a, b, y0, n)
> xy
         x          y
 [1,] 0.0   1.000000
 [2,] 0.5   1.796875
 [3,] 1.0   3.434692
 [4,] 1.5   6.458751
 [5,] 2.0  11.765654
 [6,] 2.5  20.836144
 [7,] 3.0  36.109983
 [8,] 3.5  61.607877
 [9,] 4.0 103.956680
[10,] 4.5 174.078571
[11,] 5.0 289.975092
```

```
> # Будуємо графік чисельного розв'язку
> x = xy[,1]
> y = xy[,2]
> plot(x, y, type = "p", col= "red")
>
> # Аналітичний розв'язок задачі
> FunAn = function(x)
+ {
+ return(2*exp(x) - x - 1)
+ }
> # Будуємо графік аналітичного розв'язку
> x1 = seq(a, b, by=0.01)
> y1 = FunAn(x1)
> # додаємо в 1-у графічну підобласть ще лінію
> lines(x1, y1, col="blue")
```



9.5. Висновки

1. При застосуванні багатокрокового методу необхідно спочатку декілька кроків виконати однокроковим методом.

2. У багатокрокових методах крок h можна обирати більшим, ніж у однокрокових методах.

9.6. Контрольні запитання та завдання

1. Які чисельні методи розв'язання задачі Коші для звичайних диференціальних рівнянь називаються багатокроковими методами?

2. Які особливості мають багатокрокові методи відносно однокрокових методів?

3. Наведіть загальну схему всіх методів розв'язання задачі Коші для системи звичайних диференціальних рівнянь.

10. НЕЯВНІ МЕТОДИ РОЗВ'ЯЗАННЯ ЖОРСТКИХ ЗАДАЧ КОШІ

10.1. Поняття жорсткої системи диференціальних рівнянь

Поняття жорсткої задачі Коші вводиться для задачі виду (8.3), (8.4) (в матричній формі (8.6)), тобто для системи диференціальних рівнянь.

Для підвищення точності та адекватності математичних моделей складних об’єктів і процесів при побудові моделей доводиться враховувати велику кількість факторів та параметрів. При цьому в математичній моделі, що описується системою диференціальних рівнянь, опиняються складові з великими і малими значеннями похідних від шуканих функцій $y_j(x)$. Це і призводить до так званої **жорсткої системи диференціальних рівнянь.** Тут треба зазначити, що жорсткість є властивістю самої математичної задачі, а не чисельного метода її розв’язання. Застосування описаних (розділи 10, 11) явних чисельних методів для жорстких систем диференціальних рівнянь дає велику похибку у розв’язку, тому для них розроблені так звані **неявні методи** [**Ошибка! Источник ссылки не найден.**].

Слід розглянути спочатку (замість (8.6)) лінійну систему диференціальних рівнянь з незалежною від x матрицею $A \in R^{m \times m}$:

$$Y' = A \times Y(x). \tag{10.1}$$

Нехай $\lambda_i, i = \overline{1,m}$ – множина власних чисел матриці A, $K = \dfrac{\max\limits_i |\mathrm{Re}(\lambda_i)|}{\min\limits_i |\mathrm{Re}(\lambda_i)|}$, де $\mathrm{Re}(\lambda_i)$ – дійсна частина власного числа.

Визначення. Система диференціальних рівнянь (10.1) називається **жорсткою**, якщо [**Ошибка! Источник ссылки не найден.**]:

$$\mathrm{Re}(\lambda_i) < 0, i = \overline{1,m} \text{ і } K >> 1.$$

При цьому K називається **числом жорсткості системи** (10.1).

Це поняття жорсткої системи узагальнюється і на систему (8.6). Роль матриці A при цьому відіграє матриця Якобі

$$A(x) = F_Y'(x,Y) = \begin{pmatrix} \dfrac{\partial f_1(x,Y)}{\partial y_1} & \dots & \dfrac{\partial f_1(x,Y)}{\partial y_m} \\ \dots & \dots & \dots \\ \dfrac{\partial f_n(x,Y)}{\partial y_1} & \dots & \dfrac{\partial f_n(x,Y)}{\partial y_m} \end{pmatrix},$$

яка в загальному випадку вже буде залежати від x, тобто змінюватись у часі буде і число жорсткості системи (8.6).

Як вже згадувалося раніше, застосування явних чисельних методів для жорстких систем диференціальних рівнянь дає велику похибку у розв'язку, якщо крок обчислень не є досить малим, а саме він повинен задовольняти обмеженню [**Ошибка! Источник ссылки не найден.**]:

$$h < \frac{c}{|\lambda_{\max}|},$$

де c – константа, яка залежить від умов задачі (8.6), $\lambda_{\max}$ – максимальне за модулем власне число матриці Якобі.

10.2. Неявні методи Ейлера і Рунге – Кутта

Поняття неявного методу було введено в розділі 10.3. Це методи, що застосовують схему (наприклад, як в схемі (9.3) методу Адамса – Мултона 4-го порядку), в якій в лівій і правій частині фігурує ще не відоме значення Y_{i+1}.

Неявний метод Ейлера. Явна схема методу Ейлера має вид (8.9).

У розділі 10.2 було зазначено, що цей метод Ейлера співпадає з методом Адамса – Бошфорда 1-го порядку (див. розділ 10.2). В той же час, метод Адамса – Мултона дає алгоритм переходу від явного методу до неявного, шляхом заміни виду інтерполяційного поліному Ньютона. Так, при $m = 1$, якщо побудувати інтерполяційний поліном Ньютона за вузлом x_{i+1}, то він буде мати вигляд: $H_{m-1}(x) = F(x_{i+1}, Y_{i+1})$. Таким чином, з (9.1) отримується схема неявного методу Ейлера:

$$Y_{i+1} = Y_i + h\,F(x_{i+1}, Y_{i+1}). \qquad (10.2)$$

Слід зазначити, що для визначення Y_{i+1} треба розв'язати систему нелінійних рівнянь (10.2). Система (10.2) нелінійна оскільки в загальному випадку векторна функція $F(x,Y)$ нелінійна відносно Y Розв'язати систему нелінійних рівнянь (10.2) можна, наприклад, методом Ньютона (див. розділ 5.2).

Неявний метод Рунге – Кутта. У неявному методі Рунге – Кутта 4-го порядку для розв'язання жорстких систем звичайних диференціальних рівнянь (8.6) значення Y_i у вузлах сітки x_i, $i=\overline{1,n}$, рекурентно обчислюються, як розв'язання системи рівнянь [**Ошибка! Источник ссылки не найден.**]:

$$Y_{i+1} = Y_i + \frac{1}{6}(K_1^* + 2K_2^* + 2K_3^* + K_4^*),\ i=\overline{0,n-1}, \tag{10.3}$$

де $K_1^* = hF(x_{i+1}, Y_{i+1})$, $K_2^* = hF(x_{i+1} - \frac{h}{2}, Y_{i+1} - \frac{h}{2}K_1^*)$,

$K_3^* = hF(x_{i+1} - \frac{h}{2}, Y_{i+1} - \frac{h}{2}K_2^*)$, $K_4^* = hF(x_{i+1}, Y_{i+1} - hK_3^*)$.

Глобальна похибка неявного методу Рунге – Кутта 4-го порядку, оцінювана для величини $\|Y_n - Y(x_n)\|$, має порядок $O(h^4)$, а локальна – $O(h^5)$ [**Ошибка! Источник ссылки не найден.**].

Приклад 10.1. Процес руху автомобіля на площині в найпростішому випадку може бути описаний системою диференціальних рівнянь [**Ошибка! Источник ссылки не найден.**]:

$$\begin{cases} \dfrac{\partial x}{\partial t} = V\cos\theta, \\ \dfrac{\partial y}{\partial t} = V\sin\theta, \\ \dfrac{\partial \theta}{\partial t} = -\dfrac{V}{W\,Ctg\phi + \dfrac{w}{2}}. \end{cases}, \tag{10.4}$$

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. 10.1).

У моделі (10.3) всі кути вимірюються в радіанах.

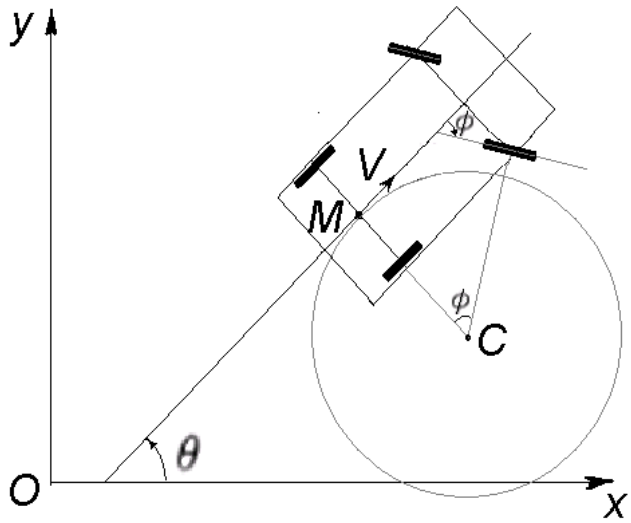


Рис. 10.1. **Модель руху автомобіля на площині**

Треба визначити траєкторію руху автомобіля (тобто точки M) протягом 5 секунд, якщо його швидкість V була постійна і дорівнювала -3 км/год (задній хід), кут ϕ змінювався відповідно функції $\phi(t) = \dfrac{-1.606}{\pi} arctg(a_0 t + a_1)$, $a_0 = 11771.1$, $a_1 = -13.9164$, $W = 2.47$ м, $w = 1.456$ м, на початку руху точка M мала координати $(0, 2)$, а кут $\theta = 0$.

Цим прикладом імітується паралельна парковка автомобіля заднім ходом.

Розв’язання.

Оскільки в даному прикладі є три координати (x, y, θ), які змінюються у часі, то процедуру, що реалізує неявний метод Рунге – Кутта 4-го порядку, можна записати в такому вигляді:

```
# Задаємо функцію системи нелінійних рівнянь (12.3)
# для неявного методу Рунке-Кутта
SysFun = function(Y)
{
     K1 = h*DEfunPr(ti1, Y)
     K2 = h*DEfunPr(ti1-h/2, Y-h/2*K1)
     K3 = h*DEfunPr(ti1-h/2, Y-h/2*K2)
     K4 = h*DEfunPr(ti1, Y-h/2*K3)
     F = Yi + 1/6*(K1 + 2*K2 + 2*K3 + K4) - Y
     return(F)
}

MetRKipml = function(a, b, Y0, n)
{
     t = numeric(n+1)
     Y = matrix(0, nrow=(n+1), ncol=3)
     h = (b-a)/n
     assign("h", h, envir = .GlobalEnv)
     t[1] = a
     Y[1,] = Y0
     for( i in 1:n)
     {
          t[i+1] = t[i] + h
          assign("ti1", t[i+1], envir = .GlobalEnv)
          assign("Yi", Y[i,], envir = .GlobalEnv)
# Розвязуємо систему нелінійних рівнянь за допомогою
# функції пакета "nleqslv"
          Z = nleqslv(Y[i,], SysFun, control=list(btol=.01))
          Y[i+1,] = Z$x
     }
     return(cbind(t, Y))
}
```

Вводяться дані для розв’язання задачі:

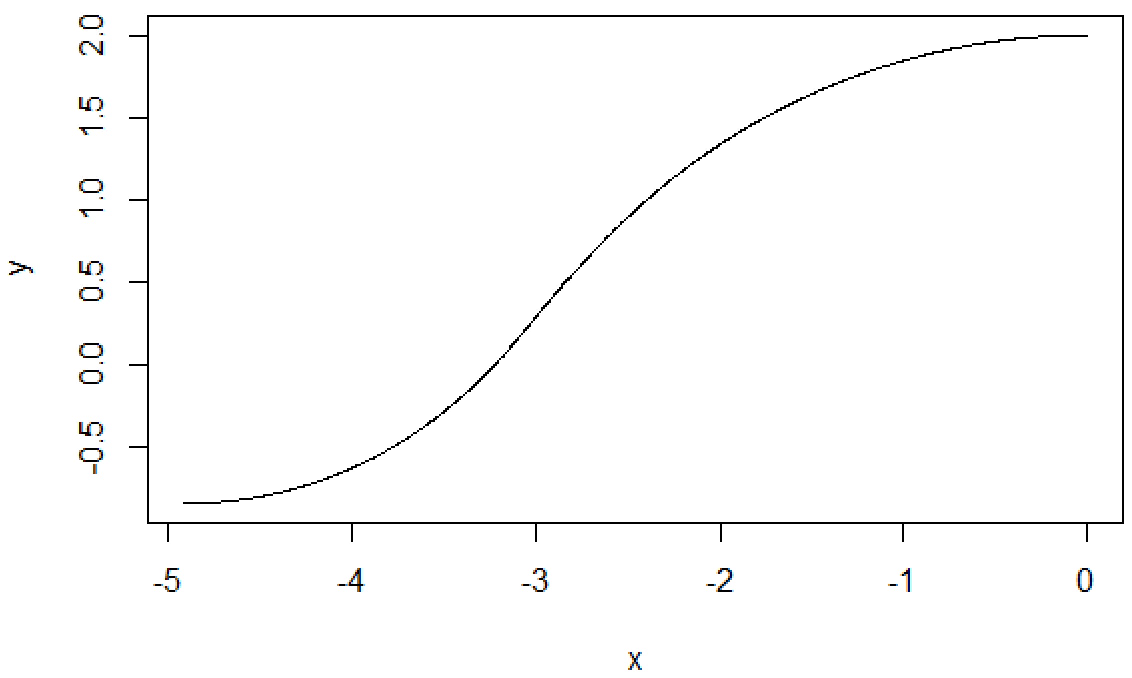
```
# Вводимо дані для розв’язання задачі
W = 2.47                # колісна база
w = 1.456               # колія задніх коліс
V = function(t){return(-3000)}# постійна швидкість заднім ходом
Fi = function(t){             # змінний кут повороту коліс
   a1 = -13.9164
   a0 = 11771.1
   return(-1.606/pi*atan(a0*t+a1))
}
a = 0                         # початковий момент часу
b = 0.00197          # кінцевий момент часу в годинах
# положення автомобіля у початковий момент часу
Y0 = c(0, 2, 0)
```

```
# Задаємо праву частину системи диференціальних рівнянь (12.4)
DEfunPr = function(t, Y)
{
      Ft = numeric(3)
      Ft[1] = V(t)*cos(Y[3])
      Ft[2] = V(t)*sin(Y[3])
      Ft[3] = -V(t)/(W/tan(Fi(t)) + w/2)
      return(Ft)
}
```

Викликається записана процедура й отримується розв'язок. Будується графік отриманого розв'язку:

```
> # Підключаємо бібліотеку для розв'язання
> # систем нелінійних рівнянь виду F(x)=0
> library("nleqslv")

> # Розв'язуємо систему диференціальних рівнянь
> n = 1000
> tY = MetRKipml(a, b, Y0, n)
> x = tY[, 2]
> y = tY[, 3]
> # Будуємо графік траєкторії руху автомобіля
> plot(x, y, type = "l")
```



Таким чином, виконана імітація руху автомобіля при паралельній парковці автомобіля заднім ходом.

10.3. Висновки

1. На практиці задача Коші як математична модель деяких динамічних процесів є найчастіше жорсткою.
2. У неявних методах крок h можна обирати більшим, ніж в явних методах.

10.4. Контрольні запитання та завдання

1. У чому полягає суть жорсткої задачі Коші для системи звичайних диференціальних рівнянь?
2. Які методи розв'язання задачі Коші для звичайних диференціальних рівнянь називаються неявними методами?
3. Які особливості мають неявні методи відносно явних методів?
4. Сформулюйте постановку задачі Коші для системи звичайних диференціальних рівнянь. Що є її розв'язком? У якому вигляді подається розв'язок чисельним методом?
5. Який з відомих вам методів розв'язання задачі Коші для системи звичайних диференціальних рівнянь треба застосовувати в тих чи інших випадках?