Лабораторна робота №4.

студентки групи ПМО-21
Кравець Ольги
Варіант - 9

Постановка задачі розв'язування задачі Коші для звичайного диференціального рівняння першого порядку методами Рунге-Кутта.

Нехай на відрізку $[a, b]$ потрібно знайти чисельний розв'язок рівняння $y' = f(x, y)$ з початковою умовою $y(x_0) = y_0$. Розіб'ємо відрізок $[a, b]$ на n рівних частин точками $x_i = x_0 + ih$ $(i = \overline{0, n})$, де $h = \frac{b-a}{n}$ - крок інтегрування. Послідовні значення y_i шуканої ф-ї y визначаються за ф-лою $y_{i+1} = y_i + \Delta y_i$

Формула Рунге-Кутта другого порядку розв'язування задачі Коші для звичайного диф. р-ня першого порядку:

$y_{i+1} = y_i + \frac{k_1 + k_2}{2}$, де $k_1 = hf(x_i, y_i)$

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

Завдання 11.1

$h = 0,05$

$y' = \frac{y(2x^2 - y^2)}{2x^3}$; $x_0 = 2$; $y_0 = 2$

$y^* = \frac{x}{\sqrt{\ln\frac{x}{2} + 1}}$

$x \in [2; 3,5]$

Код програми (C++):

```
#include <iostream>

#include <cmath>

using namespace std;

double f(double x, double y)

{             return ((y * (pow(x, 2) - pow(y, 2)) / (2 * pow(x, 3))));   }

double y_func(double x)

{             return x / sqrt(log(x / 2) + 1);      }
```

```
void RungeKut(double a, double b, double h)
{
double xi = a;
double yi = 2;
double delta = 0;
double k1 = 0, k2 = 0;
int i = 0;
double y_s = 0;
double y_iplus1 = 0;
while (xi <= b + h)
{
k1 = h * f(xi, yi);
k2 = h * f(xi + h / 2.0, yi + k1 / 2.0);
y_iplus1 = yi + ((k1 + k2) / 2);
y_s = y_func(xi);
delta = y_s - (yi + ((k1 + k2) / 2));
cout << "xi [" << i << "] = " << xi << "\t yi = [" << i << "] = " << y_iplus1 << "\t y* = [" << i << "] = " << y_s << "\t delta = " << delta << '\n';
xi += h;
++i;
}
}
int main()
{
double a = 2, b = 3.5, h = 0.05;
cout << "Method Runge-Kutta:\n";
RungeKut(a, b, h);
cout << "\n";
system("pause");
return 0;
}
```

Результат виконання програми:

```
Method Runge-Kutta:
xi [0] = 2        yi = [0] = 2.0003       y* = [0] = 2    delta = -0.00030295
xi [1] = 2.05     yi = [1] = 2.00144      y* = [1] = 2.02515      delta = 0.0237128
xi [2] = 2.1      yi = [2] = 2.00244      y* = [2] = 2.05057      delta = 0.0481325
xi [3] = 2.15     yi = [3] = 2.00333      y* = [3] = 2.07623      delta = 0.0729059
xi [4] = 2.2      yi = [4] = 2.00411      y* = [4] = 2.1021       delta = 0.0979897
xi [5] = 2.25     yi = [5] = 2.00481      y* = [5] = 2.12816      delta = 0.123346
xi [6] = 2.3      yi = [6] = 2.00543      y* = [6] = 2.15437      delta = 0.148943
xi [7] = 2.35     yi = [7] = 2.00598      y* = [7] = 2.18073      delta = 0.174751
xi [8] = 2.4      yi = [8] = 2.00646      y* = [8] = 2.20721      delta = 0.200746
xi [9] = 2.45     yi = [9] = 2.00689      y* = [9] = 2.2338       delta = 0.226906
xi [10] = 2.5     yi = [10] = 2.00727     y* = [10] = 2.26048     delta = 0.25321
xi [11] = 2.55    yi = [11] = 2.00761     y* = [11] = 2.28725     delta = 0.279642
xi [12] = 2.6     yi = [12] = 2.00791     y* = [12] = 2.31409     delta = 0.306187
xi [13] = 2.65    yi = [13] = 2.00817     y* = [13] = 2.341       delta = 0.332831
xi [14] = 2.7     yi = [14] = 2.0084      y* = [14] = 2.36796     delta = 0.359562
xi [15] = 2.75    yi = [15] = 2.0086      y* = [15] = 2.39497     delta = 0.38637
xi [16] = 2.8     yi = [16] = 2.00878     y* = [16] = 2.42202     delta = 0.413244
xi [17] = 2.85    yi = [17] = 2.00893     y* = [17] = 2.44911     delta = 0.440177
xi [18] = 2.9     yi = [18] = 2.00906     y* = [18] = 2.47622     delta = 0.467161
xi [19] = 2.95    yi = [19] = 2.00918     y* = [19] = 2.50337     delta = 0.494188
xi [20] = 3       yi = [20] = 2.00928     y* = [20] = 2.53053     delta = 0.521253
xi [21] = 3.05    yi = [21] = 2.00936     y* = [21] = 2.55771     delta = 0.54835
xi [22] = 3.1     yi = [22] = 2.00943     y* = [22] = 2.5849      delta = 0.575473
xi [23] = 3.15    yi = [23] = 2.00948     y* = [23] = 2.6121      delta = 0.602619
xi [24] = 3.2     yi = [24] = 2.00953     y* = [24] = 2.63931     delta = 0.629784
xi [25] = 3.25    yi = [25] = 2.00956     y* = [25] = 2.66653     delta = 0.656963
xi [26] = 3.3     yi = [26] = 2.00959     y* = [26] = 2.69374     delta = 0.684153
xi [27] = 3.35    yi = [27] = 2.00961     y* = [27] = 2.72096     delta = 0.711351
xi [28] = 3.4     yi = [28] = 2.00962     y* = [28] = 2.74817     delta = 0.738554
xi [29] = 3.45    yi = [29] = 2.00962     y* = [29] = 2.77538     delta = 0.765761
xi [30] = 3.5     yi = [30] = 2.00962     y* = [30] = 2.80259     delta = 0.792968
xi [31] = 3.55    yi = [31] = 2.00961     y* = [31] = 2.82979     delta = 0.820173

Press any key to continue . . .
```

Постановка задачі розв'язування задачі Коші для системи двох звичайних диф. р-нь першого порядку методом Рунге-Кутта:

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

Треба відшукати розв. системи р-нь (*) $y = y(x)$, $z = z(x)$, який задовольняє початкові умови $(\cdot)$. Задача має єдиний розв. на відр. $[x_0, x_0+a]$, якщо в деякій обл. G $\{x_0 \le x \le x_0+a, |y-y_0| \le b, |z-z_0| \le c\}$ ф-ї $f(x,y,z)$ і $g(x,y,z)$ неперервні і задов. умови Ліпшиця по y і z.

Розрахункові ф-ли:
$$\begin{cases} y_{i+1} = y_i + hf(x_i, y_i, z_i) \\ z_{i+1} = z_i + hg(x_i, y_i, z_i), \end{cases}$$

де $y_i \approx y(x_i)$, $z_i \approx z(x_i)$, $i = 0, 1, \ldots, m-1$.

Ф-ла Рунге-Кутта другого порядку для розв. задачі Коші для системи двох звичайних дифер. р-нь першого порядку:

$$\begin{cases} y_{i+1} = y_i + \frac{k_1+k_2}{2} \\ z_{i+1} = z_i + \frac{l_1+l_2}{2}, \end{cases}$$

де $k_1 = hf(x_i, y_i, z_i)$, $k_2 = hf(x_i+h, y_i+k_1, z_i+l_1)$
$l_1 = hg(x_i, y_i, z_i)$, $l_2 = hg(x_i+h, y_i+k_1, z_i+l_1)$

Завдання 11.2.

$$\begin{cases} y' = 2z - 3y \\ z' = z - 2y \end{cases} \qquad y(0) = 2, \ z(0) = 3$$
$$[0; 0,1]; \quad h = 0,01$$

$$\begin{cases} y^* = 2(1+x)e^{-x} \\ z^* = (3+2x)e^{-x} \end{cases}$$

Код програми (C++):

```
#include <iostream>
#include <cmath>
using namespace std;

double f(double x, double y, double z)
{
return 2 * z - 3 * y;
}

double g(double x, double y, double z)
{
return z - 2 * y;
}

double y_func(double x)
{
return 2 * (1 + x) * exp(-x);
}

double z_func(double x)
{
return (3 + 2 * x) * exp(-x);
}

void RungeKut(double a, double b, double h)
{
double xi = a;
double yi = 2;
double zi = 3;
double delta_y = 0, delta_z = 0;
double k1 = 0, k2 = 0;
double l1 = 0, l2 = 0;
int i = 0;
double y_s = 0;
double z_s = 0;
double y_iplus1 = 0;
double z_iplus1 = 0;

while (xi <= b)
{
k1 = h * f(xi, yi, zi);
l1 = h * g(xi, yi, zi);
k2 = h * f(xi + h, yi + k1, zi + l1);
l2 = h * g(xi + h, yi + k1, zi + l1);
y_iplus1 = yi + ((k1 + k2) / 2);
z_iplus1 = zi + ((l1 + l2) / 2);
y_s = y_func(xi);
z_s = z_func(xi);
delta_y = y_s - y_iplus1;
delta_z = z_s - z_iplus1;
cout << "xi [" << i << "] = " << xi << "\t yi = [" << i << "] = " << y_iplus1 << "\t zi = [" << i << "] = " <<
z_iplus1;
cout << "\t y* = [" << i << "] = " << y_s << "\t z* = [" << i << "] = " << z_s;
cout << "\t  delta_y = " << delta_y << "\t  delta_z = " << delta_z << '\n';
xi += h;
yi += h;
zi += h;
++i;
}

}
int main()
```

```
{
double a = 0, b = 0.1, h = 0.01;
cout << "Method Runge-Kutta:\n";
RungeKut(a, b, h);


cout << "\n";
system("pause");
return 0;
}
```

Результат виконання програми:

```
Method Runge-Kutta:
xi [0] = 0        yi = [0] = 1.9999        zi = [0] = 2.98995
xi [1] = 0.01     yi = [1] = 2.0098        zi = [1] = 2.99985
xi [2] = 0.02     yi = [2] = 2.0197        zi = [2] = 3.00975
xi [3] = 0.03     yi = [3] = 2.0296        zi = [3] = 3.01965
xi [4] = 0.04     yi = [4] = 2.0395        zi = [4] = 3.02955
xi [5] = 0.05     yi = [5] = 2.0494        zi = [5] = 3.03945
xi [6] = 0.06     yi = [6] = 2.0593        zi = [6] = 3.04935
xi [7] = 0.07     yi = [7] = 2.0692        zi = [7] = 3.05925
xi [8] = 0.08     yi = [8] = 2.0791        zi = [8] = 3.06915
xi [9] = 0.09     yi = [9] = 2.089         zi = [9] = 3.07905
xi [10] = 0.1     yi = [10] = 2.0989       zi = [10] = 3.08895
```

```
y* = [0] = 2        z* = [0] = 3        delta_y = 0.0001        delta_z = 0.01005
y* = [1] = 1.9999        z* = [1] = 2.98995        delta_y = -0.00989984
y* = [2] = 1.99961       z* = [2] = 2.9798         delta_y = -0.0200957
y* = [3] = 1.99912       z* = [3] = 2.96956        delta_y = -0.0304837
y* = [4] = 1.99844       z* = [4] = 2.95923        delta_y = -0.04106
y* = [5] = 1.99758       z* = [5] = 2.94881        delta_y = -0.0518207
y* = [6] = 1.99654       z* = [6] = 2.93831        delta_y = -0.0627622
y* = [7] = 1.99532       z* = [7] = 2.92772        delta_y = -0.0738807
y* = [8] = 1.99393       z* = [8] = 2.91705        delta_y = -0.0851727
y* = [9] = 1.99237       z* = [9] = 2.9063         delta_y = -0.0966345
y* = [10] = 1.99064      z* = [10] = 2.89548       delta_y = -0.108263
```

```
          delta_z = -0.0099
delta_z = -0.029947
delta_z = -0.0500882
delta_z = -0.0703205
delta_z = -0.0906413
delta_z = -0.111048
delta_z = -0.131537
delta_z = -0.152106
delta_z = -0.172753
delta_z = -0.193475
```