

Wydział WIMiP	Imię i nazwisko: Zuzanna Będkowska	Rok: 2	Grupa: 1	Data: 20.05.2022
<b>Metody Numeryczne</b>	Temat: Rozwiązanie równań różniczkowych			

### Zadanie: Obliczenie wartości rozwiązania danego problemu Cauchy'ego

Zgodnie z instrukcją zaimplementowano 3 metody rozwiązywania równań różniczkowych:

- Metodę Eulera
- Metodę Rungego-Kutty rzędu drugiego
- Metodę Rungego-Kutty rzędu czwartego

Kod funkcji realizujących powyższe metody:

```

34 double Euler(int n, double x, double x0, double y0, double (f)(double, double))
35 {
36     double h = abs(x - x0) / n;
37     for (int i = 0; i < n; ++i)
38     {
39         double pomoc = f(x0, y0);
40         x0 += h;
41         y0 += h * pomoc;
42     }
43     return y0;
44 }
45
46 double RK2(int n, double x, double x0, double y0, double (f)(double, double))
47 {
48     double h = abs(x - x0) / n;
49     for (int i = 0; i < n; ++i)
50     {
51         double k1 = f(x0, y0);
52         double k2 = f(x0 + h, y0 + h * k1);
53         x0 += h;
54         y0 += h * (double)(1.0 / 2.0) * (k1 + k2);
55     }
56     return y0;
57 }
58

```

```

60 double RK4(int n, double x, double x0, double y0, double (f)(double, double))
61 {
62     double h = abs(x - x0) / n;
63     for (int i = 0; i < n; ++i)
64     {
65         double k1 = f(x0, y0);
66         double k2 = f(x0 + 1.0 / 2.0 * h, y0 + 1.0 / 2.0 * h * k1);
67         double k3 = f(x0 + 1.0 / 2.0 * h, y0 + 1.0 / 2.0 * h * k2);
68         double k4 = f(x0 + h, y0 + h * k3);
69         x0 += h;
70         y0 += h * (double)(1.0 / 6.0) * (k1 + 2.0 * k2 + 2.0 * k3 + k4);
71     }
72     return y0;
73 }
74

```

Każda z funkcji jako argument przyjmuje funkcję będącą implementacją podanego równania różniczkowego:

```

10 double rownanie1(double x, double y)
11 {
12     double dy = pow(x, 2) + y;
13     return dy;
14 }
15
16 double rownanie2(double x, double y)
17 {
18     double dy = x + y;
19     return dy;
20 }

```

Dodatkowo, w celu sprawdzenia wyników zaimplementowano funkcje będące rozwiązaniami podanych równań (obliczone przez kalkulator):

```

22 double rozwiazanie1(double x)
23 {
24     double y = (-1) * pow(x, 2) - 2 * x + 2.1 * pow(e, x) - 2;
25     return y;
26 }
27
28 double rozwiazanie2(double x)
29 {
30     double y = (-1)*x + 1.1 * pow(e, x) - 1;
31     return y;
32 }

```

Metody rozwiązywania uruchomiono dla danych z instrukcji oraz wykonano analizę zbieżności, a jej wyniki zapisano w macierzy:

```

76 int main()
77 {
78     int n1 = 10;
79     double y1 = 0.1;
80     double x1 = 0.0;
81     double punkt1 = 1.0;
82     cout << "Rozwiązanie rzeczywiste f1 w punkcie x = " << punkt1 << " : " << rozwiazanie1(punkt1) << "\n";
83     cout << "Rozwiązanie metoda eulera dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku początkowego y(0) = " << y1;
84     cout << " to: " << Euler(n1, punkt1, x1, y1, rownanie1) << "\n";
85     cout << "Rozwiązanie metoda RK2 dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku początkowego y(0) = " << y1;
86     cout << " to: " << RK2(n1, punkt1, x1, y1, rownanie1) << "\n";
87     cout << "Rozwiązanie metoda RK4 dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku początkowego y(0) = " << y1;
88     cout << " to: " << RK4(n1, punkt1, x1, y1, rownanie1) << "\n";
89     int n2 = 10;
90     double y2 = 0.1;
91     double x2 = 0.0;
92     double punkt2 = 1.0;
93     cout << "Rozwiązanie rzeczywiste f2 w punkcie x = " << punkt2 << " : " << rozwiazanie2(punkt2) << "\n";
94     cout << "Rozwiązanie metoda eulera dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku początkowego y(0) = " << y2;
95     cout << " to: " << Euler(n2, punkt2, x2, y2, rownanie2) << "\n";
96     cout << "Rozwiązanie metoda RK2 dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku początkowego y(0) = " << y2;
97     cout << " to: " << RK2(n2, punkt2, x2, y2, rownanie2) << "\n";
98     cout << "Rozwiązanie metoda RK4 dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku początkowego y(0) = " << y2;
99     cout << " to: " << RK4(n2, punkt2, x2, y2, rownanie2) << "\n";
100     cout << "Analiza zbieżności:\n";
101     vector <double> pomoc(6, 0.0);
102     vector <vector <double>> tabela(100, pomoc);
103     for (int i = 0; i < 100; ++i)
104     {
105         tabela[i][0] = Euler(i + 1, punkt1, x1, y1, rownanie1);
106         tabela[i][1] = RK2(i + 1, punkt1, x1, y1, rownanie1);
107         tabela[i][2] = RK4(i + 1, punkt1, x1, y1, rownanie1);
108         tabela[i][3] = Euler(i + 1, punkt1, x1, y1, rownanie2);
109         tabela[i][4] = RK2(i + 1, punkt1, x1, y1, rownanie2);
110         tabela[i][5] = RK4(i + 1, punkt1, x1, y1, rownanie2);
111     }
112
113     cout << setw(15) << right << "          n          E 1          RK2 1          RK4 1          E 2          RK2 2          RK4 2 \n";
114     for (int i = 0; i < 100; ++i)
115     {
116         cout << setw(15) << right << i + 1 << " ";
117         for (auto j : tabela[i])
118         {
119             cout << setw(15) << right << j << " ";
120         }
121         cout << "\n";
122     }

```

## Efekt działania programu:

```

Rozwiązanie rzeczywiste f1 w punkcie x = 1: 0.708392
Rozwiązanie metoda eulera dla f1 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.606233
Rozwiązanie metoda RK2 dla f1 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.707732
Rozwiązanie metoda RK4 dla f1 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.708391
Rozwiązanie rzeczywiste f2 w punkcie x = 1: 0.99011
Rozwiązanie metoda eulera dla f2 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.853117
Rozwiązanie metoda RK2 dla f2 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.985489
Rozwiązanie metoda RK4 dla f2 w punkcie x = 1 dla N = 10 i warunku początkowego y(0) = 0.1 to: 0.990108
Analiza zbieżności:

```

n	E 1	RK2 1	RK4 1	E 2	RK2 2	RK4 2
1	0.2	0.75	0.708333	0.2	0.75	0.979167
2	0.35	0.709375	0.708151	0.475	0.904687	0.989081
3	0.434568	0.706247	0.708319	0.607407	0.947102	0.989877
4	0.487305	0.706276	0.708364	0.685547	0.964341	0.990031
5	0.523136	0.706645	0.708379	0.737152	0.972979	0.990076
6	0.54902	0.706984	0.708385	0.773789	0.977908	0.990093
7	0.568578	0.70725	0.708388	0.80115	0.980981	0.990101
8	0.583871	0.707454	0.70839	0.822363	0.983025	0.990105
9	0.596153	0.70761	0.70839	0.839292	0.984453	0.990107
10	0.606233	0.707732	0.708391	0.853117	0.985489	0.990108
11	0.614654	0.707828	0.708391	0.864619	0.986265	0.990108
12	0.621794	0.707905	0.708391	0.874339	0.98686	0.990109
13	0.627923	0.707968	0.708391	0.882661	0.987327	0.990109
14	0.633243	0.708019	0.708392	0.889867	0.987701	0.990109
15	0.637904	0.708062	0.708392	0.896167	0.988004	0.99011
16	0.64202	0.708098	0.708392	0.901721	0.988253	0.99011
17	0.645683	0.708128	0.708392	0.906656	0.98846	0.99011
18	0.648962	0.708154	0.708392	0.911068	0.988635	0.99011
19	0.651916	0.708176	0.708392	0.915038	0.988783	0.99011
20	0.65459	0.708195	0.708392	0.918627	0.98891	0.99011
21	0.657022	0.708212	0.708392	0.92189	0.98902	0.99011
22	0.659244	0.708227	0.708392	0.924867	0.989115	0.99011
23	0.661282	0.70824	0.708392	0.927595	0.989198	0.99011
24	0.663158	0.708252	0.708392	0.930104	0.989272	0.99011
25	0.66489	0.708262	0.708392	0.93242	0.989336	0.99011
26	0.666494	0.708271	0.708392	0.934563	0.989394	0.99011
27	0.667984	0.708279	0.708392	0.936553	0.989445	0.99011
28	0.669372	0.708287	0.708392	0.938406	0.989491	0.99011
29	0.670668	0.708293	0.708392	0.940134	0.989533	0.99011
30	0.67188	0.7083	0.708392	0.941751	0.98957	0.99011
31	0.673017	0.708305	0.708392	0.943266	0.989604	0.99011
32	0.674085	0.70831	0.708392	0.944689	0.989635	0.99011
33	0.675091	0.708315	0.708392	0.946028	0.989663	0.99011
34	0.676039	0.708319	0.708392	0.947291	0.989688	0.99011
35	0.676935	0.708323	0.708392	0.948483	0.989712	0.99011
36	0.677783	0.708327	0.708392	0.949611	0.989733	0.99011
37	0.678586	0.70833	0.708392	0.950679	0.989753	0.99011
38	0.679348	0.708333	0.708392	0.951692	0.989772	0.99011
39	0.680072	0.708336	0.708392	0.952655	0.989789	0.99011

Pełny kod:

```
1  #include <iostream>
2  #include <iomanip>
3  #include <vector>
4  #include <cmath>
5
6  using namespace std;
7
8  constexpr auto e = 2.718281828459;
9
10 double rownanie1(double x, double y)
11 {
12     double dy = pow(x, 2) + y;
13     return dy;
14 }
15
16 double rownanie2(double x, double y)
17 {
18     double dy = x + y;
19     return dy;
20 }
21
22 double rozwiazanie1(double x)
23 {
24     double y = (-1) * pow(x, 2) - 2 * x + 2.1 * pow(e, x) - 2;
25     return y;
26 }
27
28 double rozwiazanie2(double x)
29 {
30     double y = (-1)*x + 1.1 * pow(e, x) - 1;
31     return y;
32 }
33
34 double Euler(int n, double x, double x0, double y0, double (f)(double, double))
35 {
36     double h = abs(x - x0) / n;
37     for (int i = 0; i < n; ++i)
```

```
38     {
39         double pomoc = f(x0, y0);
40         x0 += h;
41         y0 += h * pomoc;
42     }
43     return y0;
44 }
45
46 double RK2(int n, double x, double x0, double y0, double (f)(double, double))
47 {
48     double h = abs(x - x0) / n;
49     for (int i = 0; i < n; ++i)
50     {
51         double k1 = f(x0, y0);
52         double k2 = f(x0 + h, y0 + h * k1);
53         x0 += h;
54         y0 += h * (double)(1.0 / 2.0) * (k1 + k2);
55     }
56     return y0;
57 }
58
59
60 double RK4(int n, double x, double x0, double y0, double (f)(double, double))
61 {
62     double h = abs(x - x0) / n;
63     for (int i = 0; i < n; ++i)
64     {
65         double k1 = f(x0, y0);
66         double k2 = f(x0 + 1.0 / 2.0 * h, y0 + 1.0 / 2.0 * h * k1);
67         double k3 = f(x0 + 1.0 / 2.0 * h, y0 + 1.0 / 2.0 * h * k2);
68         double k4 = f(x0 + h, y0 + h * k3);
69         x0 += h;
70         y0 += h * (double)(1.0 / 6.0) * (k1 + 2.0 * k2 + 2.0 * k3 + k4);
71     }
72     return y0;
73 }
74 }
```

```

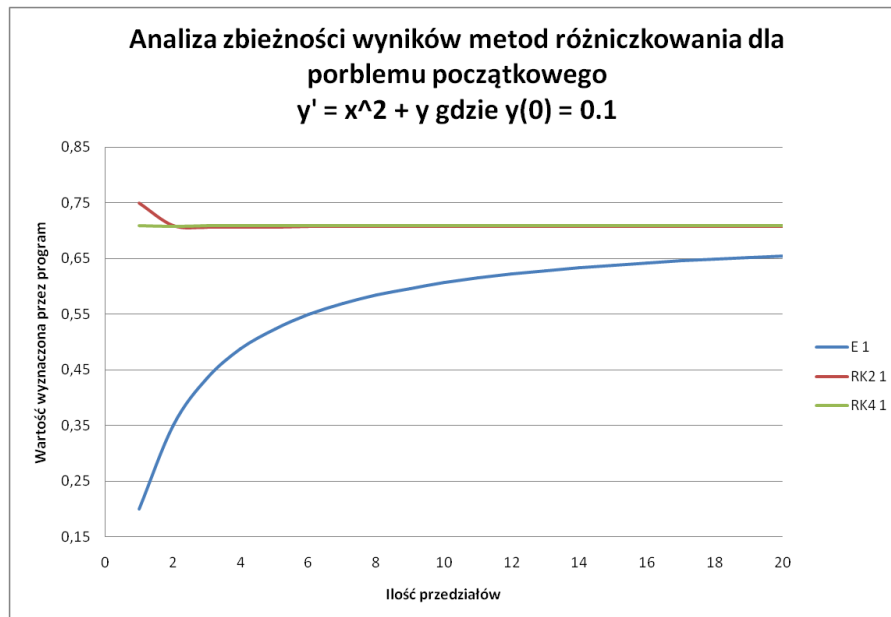
76 int main()
77 {
78     int n1 = 10;
79     double y1 = 0.1;
80     double x1 = 0.0;
81     double punkt1 = 1.0;
82     cout << "Rozwiazanie rzeczywiste f1 w punkcie x = " << punkt1 << " : " << rozwiazanie1(punkt1) << "\n";
83     cout << "Rozwiazanie metoda eulera dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku poczatkowego y(0) = " << y1;
84     cout << " to: " << Euler(n1, punkt1, x1, y1, rownanie1) << "\n";
85     cout << "Rozwiazanie metoda RK2 dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku poczatkowego y(0) = " << y1;
86     cout << " to: " << RK2(n1, punkt1, x1, y1, rownanie1) << "\n";
87     cout << "Rozwiazanie metoda RK4 dla f1 w punkcie x = " << punkt1 << " dla N = " << n1 << " i warunku poczatkowego y(0) = " << y1;
88     cout << " to: " << RK4(n1, punkt1, x1, y1, rownanie1) << "\n";
89     int n2 = 10;
90     double y2 = 0.1;
91     double x2 = 0.0;
92     double punkt2 = 1.0;
93     cout << "Rozwiazanie rzeczywiste f2 w punkcie x = " << punkt2 << " : " << rozwiazanie2(punkt2) << "\n";
94     cout << "Rozwiazanie metoda eulera dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku poczatkowego y(0) = " << y2;
95     cout << " to: " << Euler(n2, punkt2, x2, y2, rownanie2) << "\n";
96     cout << "Rozwiazanie metoda RK2 dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku poczatkowego y(0) = " << y2;
97     cout << " to: " << RK2(n2, punkt2, x2, y2, rownanie2) << "\n";
98     cout << "Rozwiazanie metoda RK4 dla f2 w punkcie x = " << punkt2 << " dla N = " << n2 << " i warunku poczatkowego y(0) = " << y2;
99     cout << " to: " << RK4(n2, punkt2, x2, y2, rownanie2) << "\n";
100    cout << "Analiza zbieznosci:\n";
101    vector<double> pomoc(6, 0.0);
102    vector<vector<double>> tabela(100, pomoc);
103    for (int i = 0; i < 100; ++i)
104    {
105        tabela[i][0] = Euler(i + 1, punkt1, x1, y1, rownanie1);
106        tabela[i][1] = RK2(i + 1, punkt1, x1, y1, rownanie1);
107        tabela[i][2] = RK4(i + 1, punkt1, x1, y1, rownanie1);
108        tabela[i][3] = Euler(i + 1, punkt1, x1, y1, rownanie2);
109        tabela[i][4] = RK2(i + 1, punkt1, x1, y1, rownanie2);
110        tabela[i][5] = RK4(i + 1, punkt1, x1, y1, rownanie2);
111    }
112    cout << setw(15) << right << "          n          E 1          RK2 1          RK4 1          E 2          RK2 2          RK4 2 \n";

113    for (int i = 0; i < 100; ++i)
114    {
115        cout << setw(15) << right << i + 1 << " ";
116        for (auto j : tabela[i])
117        {
118            cout << setw(15) << right << j << " ";
119        }
120        cout << "\n";
121    }
122 }

```

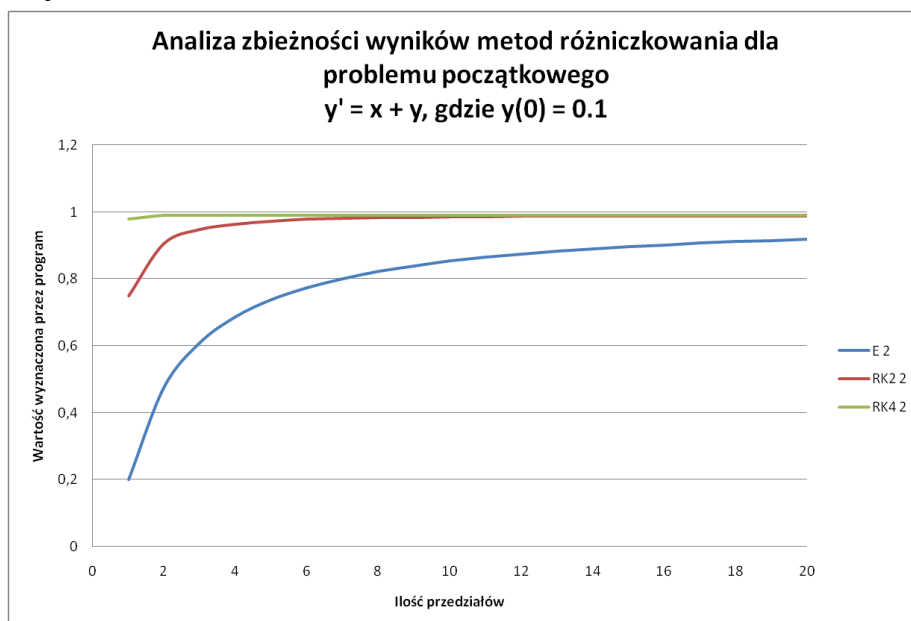
Analiza zbieżności - wykonano analizę dla ilości przedziałów z zakresu od 1 do 20:

- Funkcja  $y'(x) = x^2 + y(x)$ ,  $y(0) = 0.1$



Można zauważyć, że największą zbieżność z wynikiem wykazuje metoda Rungego-Kutty czwartego rzędu. Zgodnie z zawartością macierzy rozwiązań, wynik pokrywa się z rzeczywistym przy  $n = 14$ . Metod Rungego-Kutty czwartego drugiego rzędu wykazuje zbieżność dla  $N = 520$ . Najmniejszą zbieżność wykazuje metoda Eulera (brak zbieżności po przeprowadzeniu testów dla 10000 przedziałów).

- Funkcja  $y'(x) = x + y(x)$ ,  $y(0) = 0.1$



Można zauważyć, że największą zbieżność z wynikiem wykazuje metoda Rungego-Kutty czwartego rzędu. Zgodnie z zawartością macierzy rozwiązań, wynik pokrywa się z rzeczywistym przy  $n = 14$ . Metod Rungego-Kutty czwartego drugiego rzędu wykazuje zbieżność dla  $N = 987$ . Najmniejszą zbieżność wykazuje metoda Eulera (brak zbieżności po przeprowadzeniu testów dla 10000 przedziałów).