Laboratorium do wykładu 7

1 Zastosowanie wzoru Taylora

Ćwiczenie 1.

Będę stosował wzór Taylora ze str. 6. We wzorze tym występują pochodne z funkcji y od pierwszej do czwartej. Oznaczę je odpowiednio: d1y, d2y,d3y i d4y. Prawa strona wzoru przyjmuje teraz postać:

p:y+h*d1y+1/2*h^2*d2y+1/6*h^3*d3y+1/24*h^4*d4y

W zadaniu mam podany krok

```
→ h:0.05;
(h) 0.05
```

Definiję teraz poszczególne pochodne. Pierwszą pochodną mam daną w równaniu, które rozwiązuję

```
→ d1y:cos(x)−sin(y)+x^2;
```

(d1y) $-\sin(y)+\cos(x)+x^2$

Pozostałe pochodne muszę wyliczyć.

```
→ d2y:−sin(x)−d1y·cos(y)+2·x;
  d3y:−cos(x)−d2y·cos(y)+(d1y)^2·sin(y)+2;
  d4y:sin(x)−d3y·cos(y)+3·d1y·d2y·sin(y)+(d1y)^3·cos(y);
```

(d2y) $-\cos(y)\left(-\sin(y)+\cos(x)+x^2\right)-\sin(x)+2x$

(d3y) $\left(-\sin(y)+\cos(x)+x^2\right)^2\sin(y)-\cos(y)\left(-\cos(y)\left(-\sin(y)+\cos(x)+x^2\right)-\sin(x)+2x\right)-\cos(x)+2$

(d4y) $-\cos(y)\left(\left(-\sin(y)+\cos(x)+x^2\right)^2\sin(y)-\cos(y)\left(-\cos(y)\left(-\sin(y)+\cos(x)+x^2\right)-\sin(x)+2x\right)-\cos(x)+2\right)+3\left(-\cos(y)\left(-\sin(y)+\cos(x)+x^2\right)-\sin(x)+2x\right)\left(-\sin(y)+\cos(x)+x^2\right)\sin(y)+\cos(y)\left(-\sin(y)+\cos(x)+x^2\right)^3+\sin(x)$

```
→ i:0;
(i) 0
```

Zapisuję warunek początkowy

```
x[0]:−1;
y[0]:3;
```

(x[0]) −1

(y[0]) 3

Dla uproszczenia zapisu oznaczam te wartości początkowe przez x i y.

```
x:x[0];
y:y[0];
```

(x) −1

(y) 3

Pochodne teraz przyjmują konkretne wartości:

```
d1y:d1y,numer;
d2y:d2y,numer;
d3y:d3y,numer;
d4y:d4y,numer;
```

(d1y) 1.399182297808273

(d2y) 0.2266509610142562

(d3y) 1.960352651446951

(d4y) −1.478260311273203

Wartość funkcji y(x+h) obliczana ze wzoru Taylora przyjmuje teraz wartość

```
p:y+h·d1y+1/2·h^2·d2y+1/6·h^3·d3y+1/24·h^4·d4y;
```

(p) 3.070282884308297

Obliczam wartości rozwiązania w zależności od parametru i.

```
iter(i):=block([p],x:x[i],y:y[i],d1y:d1y,d2y:d2y,d3y:d3y,d4y:d4y,p:y+h·d1y+1/2·h^2·d2y+1/6·h^3·d
i:i+1,x[i]:x+h,y[i]:p,return(p))$
```

Wypisuję kolejnych 40ści wartości przybliżających rozwiązanie w zadanym przedziale.

```
for i:0 thru 39 do print(i+1,x[i+1],iter(i));
```

1 −0.95 3.070282884308297
2 −0.8999999999999999 3.140565768616594
3 −0.8499999999999999 3.210848652924891
4 −0.7999999999999998 3.281131537233188
5 −0.7499999999999998 3.351414421541485
6 −0.6999999999999997 3.421697305849782
7 −0.6499999999999997 3.491980190158079
8 −0.5999999999999996 3.562263074466376
9 −0.5499999999999996 3.632545958774673
10 −0.4999999999999996 3.70282884308297
11 −0.4499999999999996 3.773111727391267
12 −0.3999999999999996 3.843394611699564
13 −0.3499999999999996 3.913677496007861
14 −0.2999999999999997 3.983960380316158
15 −0.2499999999999997 4.054243264624455
16 −0.1999999999999997 4.124526148932753
17 −0.1499999999999997 4.19480903324105
18 −0.09999999999999969 4.265091917549348
19 −0.04999999999999968 4.335374801857645
20 $3.191891195797325\ 10^{-16}$ 4.405657686165942
21 0.05000000000000032 4.47594057047424
22 0.1000000000000003 4.546223454782537
23 0.1500000000000003 4.616506339090835
24 0.2000000000000003 4.686789223399132
25 0.2500000000000003 4.75707210770743
26 0.3000000000000003 4.827354992015727
27 0.3500000000000003 4.897637876324024
28 0.4000000000000003 4.967920760632322
29 0.4500000000000003 5.038203644940619
30 0.5000000000000003 5.108486529248917
31 0.5500000000000004 5.178769413557214
32 0.6000000000000004 5.249052297865512
33 0.6500000000000005 5.319335182173809
34 0.7000000000000005 5.389618066482107
35 0.7500000000000006 5.459900950790404
36 0.8000000000000006 5.530183835098701
37 0.8500000000000006 5.600466719406999
38 0.9000000000000007 5.670749603715296
39 0.9500000000000007 5.741032488023594
40 1.000000000000001 5.811315372331891

(%o20) *done*

```
→      kill(all);
(%o0)  done
```

2 Zastosowanie wzoru Eulera

Ćwiczenie 2.

Stosuję wzór Eulera ze str. 9.

W zadaniu mam podany krok

```
→      h:0.01;
(h)    0.01
```

Zapisuję warunek początkowy

```
→      x[0]:0;
       y[0]:1;
(x[0]) 0
(y[0]) 1
```

```
→      x:x[0];
       y:y[0];
(x)    0
(y)    1
```

Wartość funkcji y(x+h) obliczana ze wzoru Eulera przyjmuje teraz wartość

```
→      p:y+h·(1−4·y);
(p)    0.97
```

```
→      iter(i):=block([p],x:x[i],y:y[i],p:y+h·(1−4·y),i:i+1,x[i]:x+h,y[i]:p,return(p))$
```

Wypisuję rozwiązanie przybliżone porównując je jednocześnie z dokładnym rozwiązaniem: y=1/4(3e^(-4x)+1).

```
for i:0 thru 199 do print(i+1,x[i+1],iter(i),1/4·(3·%e^(−4·x[i+1])+1));
1 0.01 0.97 0.9705920793642424
2 0.02 0.9411999999999999 0.9423372597899768
3 0.03 0.9135519999999999 0.9151903275378681
4 0.04 0.8870099199999999 0.8891078417246585
5 0.05 0.8615295231999999 0.8640480648084864
6 0.06 0.8370683422719999 0.839970895799915
7 0.07 0.81358560858112 0.816837806091794l
8 0.08 0.7910421842378752 0.7946117778052681
9 0.09 0.7694004968683602 0.7732572445532733
10 0.1 0.7486244769936258 0.7527400345267294
11 0.11 0.7286794979138808 0.733027315812356
12 0.12 0.7095323179973255 0.7140875438546057
13 0.13 0.6911510252774326 0.6958904109776458
14 0.14 0.6735049842663352 0.6784067978866112
15 0.15 0.6565647848956818 0.6616087270705199
16 0.16 0.6403021934998545 0.6454693180322864
17 0.17 0.6246901057598603 0.6299627442741922
18 0.18 0.6097025015294659 0.6150641919699787
19 0.19 0.5953144014682873 0.600749820257432
20 0.2 0.5815018254095558 0.5869967230879161
21 0.21 0.5682417523931735 0.5737828925718097
22 0.2200000000000001 0.5555120822974466
0.5610871837611859
23 0.2300000000000001 0.5432915990055488
0.5488892808133856
24 0.2400000000000001 0.5315599350453268
0.537169664481334
25 0.2500000000000001 0.5202975376435137
0.5259095808785816
26 0.2600000000000001 0.5094856361377732
0.515091011469085
27 0.2700000000000001 0.4991062106922623
0.5046966442337043
28 0.2800000000000001 0.4891419622645718
0.4947098459672795
29 0.2900000000000001 0.4795762837739889
0.4851146356619539
30 0.3000000000000001 0.4703932324230294
0.4758956589341515
31 0.3100000000000001 0.4615775031261082
0.4670381634542879
32 0.3200000000000001 0.4531144030010639
0.4585279753398955
33 0.3300000000000001 0.4449898268810213
0.4503514764743877
34 0.3400000000000001 0.4371902338057805
```

```
→ kill(all);
(%o0) done
```

Ćwiczenie 3.

Stosuję wzór Eulera ze str. 9.

W zadaniu mam podany krok

```
→ h:0.005;
(h) 0.005
```

Zapisuję warunek początkowy

```
→ x[0]:0;
  y[0]:3;
(x[0]) 0
(y[0]) 3
```

```
→ x:x[0];
  y:y[0];
(x) 0
(y) 3
```

Wartość funkcji y(x+h) obliczana ze wzoru Eulera przyjmuje teraz wartość (w pierwszym kroku):

```
→ p:y+h·(−y·log(y)),numer;
(p) 2.983520815669978
```

```
→ iter(i):=block([p],x:x[i],y:y[i],p:y+h·(−y·log(y)),i:i+1,x[i]:x+h,y[i]:p,return(p))$
```

Wypisuję rozwiązanie przybliżone.

```
for i:0 thru 199 do print(i+1,x[i+1],iter(i)),numer;
```

```
1 0.005 2.983520815669978
2 0.01 2.96721432171523
3 0.015 2.951078260490292
4 0.02 2.935110409601113
5 0.025 2.919308581271637
6 0.03 2.903670621723164
7 0.035 2.888194410566201
8 0.04 2.872877860204517
9 0.045 2.85771891525116
10 0.05 2.842715551956131
11 0.05499999999999999 2.827865777645489
12 0.05999999999999999 2.813167630171626
13 0.06499999999999999 2.798619177374462
14 0.06999999999999999 2.784218516553323
15 0.075 2.769963773949271
16 0.08 2.755853104237654
17 0.085 2.741884690030654
18 0.09000000000000001 2.728056741389617
19 0.09500000000000001 2.71436749534695
20 0.1 2.700815215437381
21 0.105 2.687398191238376
22 0.11 2.674114737919516
23 0.115 2.660963195800641
24 0.12 2.647941929918581
25 0.125 2.635049329602267
26 0.13 2.622283808056075
27 0.135 2.609643801951193
28 0.14 2.59712777102487
29 0.145 2.584734197687356
30 0.15 2.572461586636383
31 0.1550000000000001 2.560308464479025
32 0.1600000000000001 2.548273379360777
33 0.1650000000000001 2.536354900601698
34 0.1700000000000001 2.524551618339488
35 0.1750000000000001 2.512862143179319
36 0.1800000000000001 2.501285105850317
37 0.1850000000000001 2.489819156868524
38 0.1900000000000001 2.478462966206222
39 0.1950000000000001 2.467215222967482
40 0.2000000000000001 2.456074635069809
41 0.2050000000000001 2.445039928931748
42 0.2100000000000001 2.434109849166346
43 0.2150000000000001 2.423283158280323
44 0.2200000000000001 2.412558636378864
45 0.2250000000000001 2.401935080875886
46 0.2300000000000001 2.391411306209692
```

```
→ kill(all);
(%o0) done
```

3 Wzory Rungego-Kutty

Ćwiczenie 4.

Stosuję wzór Rungego-Kutty ze str. 11.

W zadaniu mam podany krok

```
→ h:0.02;
(h) 0.02
```

Zapisuję warunek początkowy

```
→ x[0]:0;
  y[0]:0.5;
(x[0]) 0
(y[0]) 0.5
```

```
→ x:x[0];
  y:y[0];
(x) 0
(y) 0.5
```

Definiuję prawą stronę naszego równania:

```
→ f(x,y):=−y·log(y);
(%o6) f(x,y):=(−y) log(y)
```

Definiuję funkcje składowe:

```
→ F1(x,y):=h·f(x,y);
  F2(x,y):=h·f(x+h,y+F1(x,y));
(%o7) F1(x,y):=h f(x,y)
(%o8) F2(x,y):=h f(x+h,y+F1(x,y))
```

Wartość funkcji y(x+h) obliczana ze wzoru Rungego-Kutty drugiego rzędu przyjmuje w pierwszym kroku wartość:

```
→ p:y+1/2·(F1(x,y)+F2(x,y));
(p) 0.5069097241408236
```

```
iter(i):=block([p],x:x[i],y:y[i],p:y+1/2·(F1(x,y)+F2(x,y)),i:i+1,x[i]:x+h,y[i]:p,return(p))$
```

Wypisuję rozwiązanie przybliżone.

```
→ for i:0 thru 49 do print(i+1,x[i+1],iter(i)),numer;
1  0.02  0.5069097241408236
2  0.04  0.5137752924138591
3  0.06  0.5205951516620799
4  0.08  0.5273678196735259
5  0.1  0.534091884419145
6  0.12  0.5407660032114386
7  0.14  0.5473889017898301
8  0.16  0.5539593733385136
9  0.18  0.5604762774423674
10  0.2  0.5669385389863392
11  0.22  0.5733451470035275
12  0.24  0.5796951534769968
13  0.26  0.5859876721001746
14  0.28  0.592221877000484
15  0.3  0.5983970014306781
16  0.32  0.6045123364321412
17  0.34  0.6105672294742364
18  0.36  0.6165610830735813
19  0.3800000000000001  0.6224933533969476
20  0.4000000000000001  0.6283635488512883
21  0.4200000000000001  0.6341712286642153
22  0.4400000000000001  0.639916001458063
23  0.4600000000000001  0.6455975238205008
24  0.4800000000000001  0.6512154988744763
25  0.5000000000000001  0.6567696748501065
26  0.5200000000000001  0.6622598436609647
27  0.5400000000000001  0.6676858394870502
28  0.5600000000000002  0.6730475373665731
29  0.5800000000000002  0.6783448517985327
30  0.6000000000000002  0.6835777353579239
31  0.6200000000000002  0.6887461773252642
32  0.6400000000000002  0.6938502023319973
33  0.6600000000000003  0.6988898690232009
34  0.6800000000000003  0.7038652687388987
35  0.7000000000000003  0.7087765242151588
36  0.7200000000000003  0.7136237883060456
37  0.7400000000000003  0.7184072427273813
38  0.7600000000000003  0.7231270968231722
39  0.7800000000000004  0.7277835863554523
40  0.8000000000000004  0.7323769723182051
41  0.8200000000000004  0.7369075397759346
42  0.8400000000000004  0.7413755967273704
43  0.8600000000000004  0.7457814729947154
44  0.8800000000000004  0.7501255191387651
45  0.9000000000000005  0.7544081054001608
46  0.9200000000000005  0.7586296206669693
```

```
→ kill(all);
```
(%o0) *done*

Ćwiczenie 5.

Stosuję wzór Rungego-Kutty ze str. 11.

W zadaniu mam podany krok

```
→ h:1/128;
```
(h) $\frac{1}{128}$

Zapisuję warunek początkowy

```
→ x[0]:1;
y[0]:2;
```
(x[0]) 1
(y[0]) 2

```
→ x:x[0];
y:y[0];
```
(x) 1
(y) 2

Definiuję prawą stronę naszego równania:

```
→ f(x,y):=(x·y−y^2)/(x^2);
```
(%o6) $f(x,y):=\frac{x\,y-y^2}{x^2}$

Definiuję funkcje składowe:

```
→ F1(x,y):=h·f(x,y);
F2(x,y):=h·f(x+1/2·h,y+1/2·F1(x,y));
F3(x,y):=h·f(x+1/2·h,y+1/2·F2(x,y));
F4(x,y):=h·f(x+h,y+F3(x,y));
```
(%o7) $F1(x,y):=h\,f(x,y)$

(%o8) $F2(x,y):=h\,f\left(x+\frac{1}{2}h,y+\frac{1}{2}F1(x,y)\right)$

(%o9) $F3(x,y):=h\,f\left(x+\frac{1}{2}h,y+\frac{1}{2}F2(x,y)\right)$

(%o10) $F4(x,y):=h\,f(x+h,y+F3(x,y))$

Wartość funkcji y(x+h) obliczana ze wzoru Rungego-Kutty czwartego rzędu przyjmuje w pierwszym kroku wartość:

```
→ p:y+1/6·(F1(x,y)+2·F2(x,y)+2·F3(x,y)+F4(x,y)),numer;
(p) 1.984734041865838
```

```
→ iter(i):=block([p],x:x[i],y:y[i],p:y+1/6·(F1(x,y)+2·F2(x,y)+2·F3(x,y)+F4(x,y)),i:i+1,x[i]:x+h,y[i]:p,retu
```

Wypisuję rozwiązanie przybliżone.

```
for i:0 thru 255 do print(i+1,x[i+1],iter(i)),numer;
```

```
1  1.0078125  1.984734041865838
2  1.015625  1.970158587458953
3  1.0234375  1.956234594298531
4  1.03125  1.942925894996775
5  1.0390625  1.930198937560861
6  1.046875  1.91802255334111
7  1.0546875  1.906367749251739
8  1.0625  1.895207521352977
9  1.0703125  1.884516687275658
10  1.078125  1.874271735303228
11  1.0859375  1.864450688210971
12  1.09375  1.855032980206009
13  1.1015625  1.845999345520783
14  1.109375  1.837331717392581
15  1.1171875  1.829013136316847
16  1.125  1.821027666596032
17  1.1328125  1.813360320321977
18  1.140625  1.805996988030647
19  1.1484375  1.798924375355855
20  1.15625  1.79212994508516
21  1.1640625  1.785601864088026
22  1.171875  1.779328954644923
23  1.1796875  1.773300649757434
24  1.1875  1.767506952064616
25  1.1953125  1.761938396030653
26  1.203125  1.7565860131039
27  1.2109375  1.751441299578472
28  1.21875  1.74649618691694
29  1.2265625  1.74174301431706
30  1.234375  1.737174503327052
31  1.2421875  1.732783734333187
32  1.25  1.72856412476053
33  1.2578125  1.724509408842996
34  1.265625  1.720613618832486
35  1.2734375  1.716871067529086
36  1.28125  1.713276332025255
37  1.2890625  1.709824238566718
38  1.296875  1.70650984844159
39  1.3046875  1.703328444817187
40  1.3125  1.700275520451104
41  1.3203125  1.697346766209564
42  1.328125  1.694538060331861
43  1.3359375  1.691845458384944
44  1.34375  1.689265183856938
45  1.3515625  1.686793619342716
46  1.359375  1.684427298278475
```

```
--> kill(all);
(%o0) done
```