Metody numeryczne zadanie nr 3

Mateusz Miotk
Sylwia Kaczmarczyk
Michał Kulesz

December 18, 2012

1 Treść zadania

Zadanie 3.1 Zagadnienie różniczkowe: $y' = 2y^2 - 2x(x^3 - 1), y(1) = 1$
rozwiązać na przedziale $[1, 3]$ metodą Eulera oraz zmodyfikowaną metodą Eulera zwaną metodą punktu środkowego. Wyniki porównać z rozwiązaniem dokładnym $y(x) = x^2$.

2 Podstawy teoretyczne

2.1 Metoda Eulera

Niech będzie dane równanie różniczkowe zwyczajne $y' = f(x, y(x))$ z warunkiem początkowym $y(x_0) = y_0$
Metoda Eulera polega na zastąpieniu krzywej całkowej $y = y(x)$ przechodzącej przez punkt $M_0(x_0, y_0)$, odpowiadający warunkom początkowym, łamaną $M_0, M_1, M_2, ..,$ o wierzchołkach $M_i(x_i, y_i), i = 0, 1, 2, ...,$ składającą się z odcinków prostych.
Wykorzystywane jest tutaj dane rownanie rekurencyjne:

$$\begin{cases} y_0 = y(x_0) \\ y_1 = y_0 + hf(x_0, y_0) \\ y_{i+1} = y_i + hf(x_i, y_i) \end{cases}$$

gdzie h jest krokiem na osi x.

2.2 Zmodyfikowana metoda Eulera

Idea jest podobna ale wykorzystywany jest inny wzór rekurencyjny:

$$\begin{cases} y_0 = y(x_0) \\ y_1 = y_0 + hf(x_0 + \frac{h}{2}, y_0 + f(x_0, y_0) \cdot \frac{h}{2}) \\ y_{i+1} = y_i + hf(x_i + \frac{h}{2}, y_i + f(x_i, y_i) \cdot \frac{h}{2}) \end{cases}$$

3 Algorytm realizujący zadanie

3.1 Algorytm

1.Program będzie wymagał od użytkownika "dopóki mu się nie znudzi" parametru h gdzie $h \in [0, 1]$
2.Następnie dla danego parametru h w przedziale $[1, 3]$ będzie liczone rozwiązanie metodą Eulera oraz Zmodyfikowaną metodą Eulera według wzorów podanych powyżej.
3.Zostanie wypisana tabela ilustrująca poszczególne kroki metody a na końcu zostaną wypisane minimalne i maksymalne błędy osiągane przez obydwie metody.

3.2 Przykładowe rozwiązanie

Dla h = 0.5 rozwiązanie wynosi:

x_0	Y_{euler}	Y_{mid}	Dokładne	$Błąd_{euler}$	$Błąd_{midpoint}$
1.500000	2.000000	2.058594	2.250000	0.250000	0.191406
2.000000	2.437500	0.171691	4.000000	1.562500	3.828309
2.500000	-5.621094	23.217517	6.250000	11.871094	16.967517
3.000000	-10.586899	75298.618184	9.000000	19.586899	75289.618184

Dla h = 0.25 rozwiązanie wynosi:

x_0	Y_{euler}	Y_{mid}	Dokładne	$Błąd_{euler}$	$Błąd_{midpoint}$
1.250000	1.500000	1.542847	1.562500	0.062500	0.019653
1.500000	2.029297	2.136079	2.250000	0.220703	0.113921
1.750000	2.307070	2.309015	3.062500	0.755430	0.753485
2.000000	1.153902	-1.428743	4.000000	2.846098	5.428743
2.250000	-5.180353	-0.800476	5.062500	10.242853	5.862976
2.500000	-3.451778	5.506390	6.250000	9.701778	0.743610
2.750000	-15.775641	-9.136616	7.562500	23.338141	16.699116
3.000000	81.439086	-40.096835	9.000000	72.439086	49.096835

4 Opis programu

4.1 Opis struktur danych oraz funkcji w programie

4.2 Opis wejścia-wyjścia

4.3 Treść programu

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
double abs_double(double x){
        if(x <0)
        return -1.0*x;
        else
        return x;
}
double f_dokladne(double x){
        return x*x;
}
double f(double x,double y){
        return 2*y*y - 2*x*x*x*x+2*x;
}
void Euler(double h){
double x_0,y_0;
double x_temp,y_temp,blad;
double y_0_mid,y_temp_mid,blad_mid;
double max_blad_Euler,max_blad_Mid;
double min_blad_Euler,min_blad_Mid;
max_blad_Euler = max_blad_Mid =0.0;
min_blad_Euler = min_blad_Mid = INFINITY;
x_0 = 1;
y_0 = 1;
y_0_mid=1;
printf("X_0      \t\t|Y          \t|Y_mid   \t|Dokladne\t|Blad_Euler\t|Blad_MidPoint\n");
printf("--------------------------------------------------------------------------------
while(1){
        x_temp = x_0 + h;
        if(x_temp <= 3){
        y_temp = y_0+h*f(x_0,y_0);
        y_temp_mid = y_0_mid+h*f(x_0+(h/2),y_0_mid+(f(x_0,y_0_mid)*h/2));
        x_0 = x_temp;
        y_0 = y_temp;
        y_0_mid=y_temp_mid;
        blad=abs_double(f_dokladne(x_0) - y_0);
        blad_mid=abs_double(f_dokladne(x_0)-y_0_mid);
        if(max_blad_Euler < blad){
        max_blad_Euler = blad;
        }
        if(max_blad_Mid < blad_mid){
        max_blad_Mid = blad_mid;
        }
        if(min_blad_Euler > blad){
```

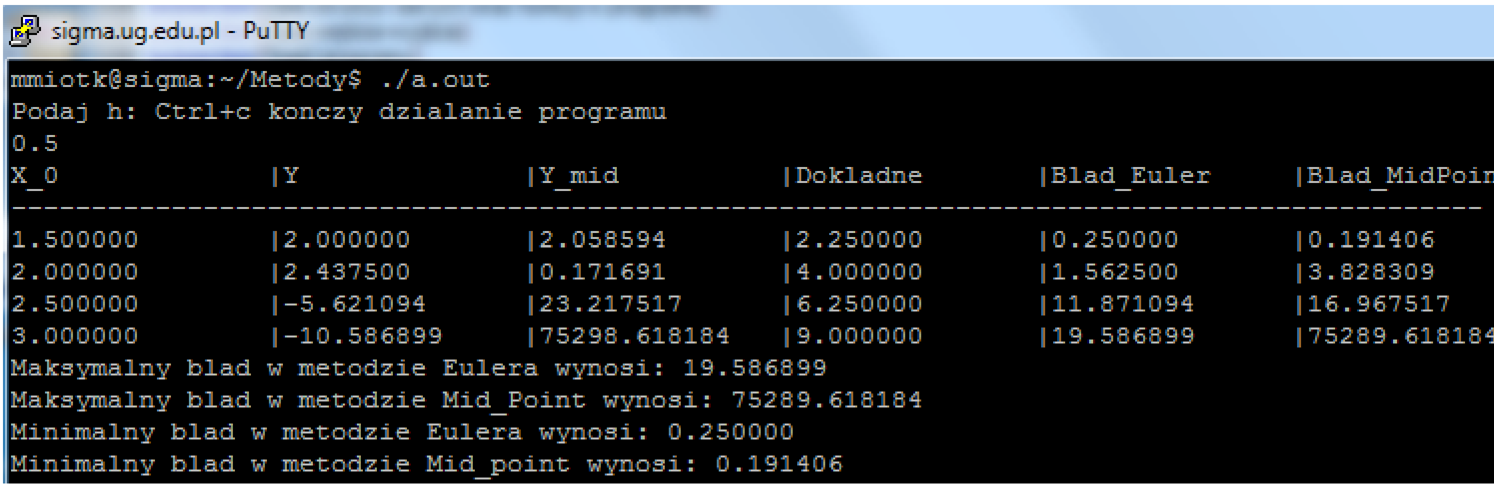
```
        min_blad_Euler=blad;
        }
        if(min_blad_Mid > blad_mid){
                min_blad_Mid = blad_mid;
        }
printf("%lf\t|%lf \t|%lf\t|%lf\t|%lf\t|%lf\n",x_0,y_0,y_0_mid,f_dokladne(x_0),blad,blad_mid);
        }
else{
printf("Maksymalny blad w metodzie Eulera wynosi: %lf\n",max_blad_Euler);
printf("Maksymalny blad w metodzie Mid_Point wynosi: %lf\n",max_blad_Mid);
printf("Minimalny blad w metodzie Eulera wynosi: %lf\n",min_blad_Euler);
printf("Minimalny blad w metodzie Mid_point wynosi: %lf\n",min_blad_Mid);
break;
}

}
}
int poprawnosc_h(double h){
        if(h <= 0 || h>=1)
        return -1;
        else
        return 1;
}

void pobranie_danych(){
        double h;
        while(1){
                printf("Podaj h: Ctrl+c konczy dzialanie programu\n");
                scanf("%lf",&h);
                if(poprawnosc_h(h)==1){
                Euler(h);
                }
                else
                printf("Wartosc h nie jest w przedziale (0,1)");
        }
}

int main(){
        pobranie_danych();
        return EXIT_SUCCESS;
}
```

4.4 Zrzuty wybranego programu



```
sigma.ug.edu.pl - PuTTY
mmiotk@sigma:~/Metody$ ./a.out
Podaj h: Ctrl+c konczy dzialanie programu
0.5
X_0             |Y              |Y_mid          |Dokladne       |Blad_Euler     |Blad_MidPoi
-------------------------------------------------------------------------------------------
1.500000        |2.000000       |2.058594       |2.250000       |0.250000       |0.191406
2.000000        |2.437500       |0.171691       |4.000000       |1.562500       |3.828309
2.500000        |-5.621094      |23.217517      |6.250000       |11.871094      |16.967517
3.000000        |-10.586899     |75298.618184   |9.000000       |19.586899      |75289.61818
Maksymalny blad w metodzie Eulera wynosi: 19.586899
Maksymalny blad w metodzie Mid_Point wynosi: 75289.618184
Minimalny blad w metodzie Eulera wynosi: 0.250000
Minimalny blad w metodzie Mid_point wynosi: 0.191406
```

0.25

X_0	Y	Y_mid	Dokladne	Blad_Euler	Blad_MidPoin
1.250000	1.500000	1.542847	1.562500	0.062500	0.019653
1.500000	2.029297	2.136079	2.250000	0.220703	0.113921
1.750000	2.307070	2.309015	3.062500	0.755430	0.753485
2.000000	1.153902	-1.428743	4.000000	2.846098	5.428743
2.250000	-5.180353	-0.800476	5.062500	10.242853	5.862976
2.500000	-3.451778	5.506390	6.250000	9.701778	0.743610
2.750000	-15.775641	-9.136616	7.562500	23.338141	16.699116
3.000000	81.439086	-40.096835	9.000000	72.439086	49.096835

Maksymalny blad w metodzie Eulera wynosi: 72.439086
Maksymalny blad w metodzie Mid_Point wynosi: 49.096835
Minimalny blad w metodzie Eulera wynosi: 0.062500
Minimalny blad w metodzie Mid_point wynosi: 0.019653

X_0	Y	Y_mid	Dokladne	Blad_Euler	Blad_MidPoin
1.100000	1.200000	1.208899	1.210000	0.010000	0.001101
1.200000	1.415180	1.436856	1.440000	0.024820	0.003144
1.300000	1.641007	1.682771	1.690000	0.048993	0.007229
1.400000	1.868368	1.943931	1.960000	0.091632	0.016069
1.500000	2.078207	2.213163	2.250000	0.171793	0.036837
1.600000	2.229496	2.470183	2.560000	0.330504	0.089817
1.700000	2.232907	2.654230	2.890000	0.657093	0.235770
1.800000	1.899661	2.579451	3.240000	1.340339	0.660549
1.900000	0.881883	1.734919	3.610000	2.728117	1.875081
2.000000	-1.188993	-0.596606	4.000000	5.188993	4.596606
2.100000	-3.706252	-2.949693	4.410000	8.116252	7.359693
2.200000	-4.428611	-3.883213	4.840000	9.268611	8.723213
2.300000	-4.751212	-4.512884	5.290000	10.041212	9.802884
2.400000	-5.373229	-5.052722	5.760000	11.133229	10.812722
2.500000	-5.754431	-5.547062	6.250000	12.004431	11.797062
2.600000	-6.444236	-5.987198	6.760000	13.204236	12.747198
2.700000	-6.758121	-6.309295	7.290000	14.048121	13.599295
2.800000	-7.712502	-6.325395	7.840000	15.552502	14.165395
2.900000	-7.549085	-5.532252	8.410000	15.959085	13.942252

Maksymalny blad w metodzie Eulera wynosi: 15.959085
Maksymalny blad w metodzie Mid_Point wynosi: 14.165395
Minimalny blad w metodzie Eulera wynosi: 0.010000