

Tarea 1

Física Computacional

Diego Sarceño

201900109

26 de septiembre de 2022

Problema 1

Dada la ecuación diferencial

$$y'(x) = y^2 + 1, \quad (1)$$

en la región $0 < x < 1$ con condiciones iniciales $y(0) = 0$. Utilizando el método de Euler, se encuentra la solución a (1) con diferentes tamaños de paso, con los que se encuentran el número de iteraciones¹.

Utilizando los resultados y la solución real de la ecuación diferencial se obtiene la figura 1.

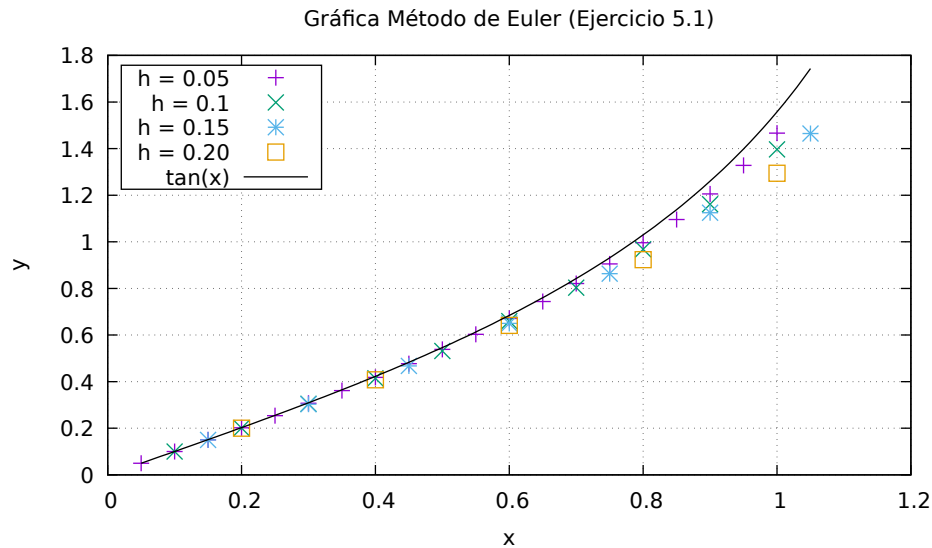


Figura 1: Gráfica para diferentes tamaños de paso.

El código creado para la solución es:

```
1 // Librerias
2 #include <iostream>
3 #include <fstream>
4
5 using namespace std;
6
7 double euler(double y, double x, double h);
8 double derivada(double y, double x);
9
```

¹Dado un tamaño de paso y un intervalo se encuentra el número de iteraciones con la fórmula $n = (b - a)/h$.

```
10
11 int main(){
12     const double y0 = 0.0;
13     const double x0 = 0.0;
14     const double h [4] = {0.05,0.10,0.15,0.20};
15     const int N [4] = {20,10,7,5};
16     ofstream salida_uno, salida_dos, salida_tres, salida_cuatro;
17
18
19     double y = y0;
20     double x = x0;
21     double y_new = 0.0;
22
23     // METODO DE EULER Y GENERACION DE ARCHIVOS
24     salida_uno.open("h1.dat", ios::out);
25     for(int i = 0; i <= N[0] - 1; i++){
26         y_new = euler(y, x, h[0]);
27
28         y = y_new;
29         x = x + h[0];
30
31         salida_uno << x << "\t" << y << endl;
32     } // END FOR
33     salida_uno.close();
34
35
36     y = y0;
37     x = x0;
38     y_new = 0.0;
39     salida_dos.open("h2.dat", ios::out);
40     for(int i = 0; i <= N[1] - 1; i++){
41         y_new = euler(y, x, h[1]);
42
43         y = y_new;
44         x = x + h[1];
45
46         salida_dos << x << "\t" << y << endl;
47     } // END FOR
48     salida_dos.close();
49
50
51     y = y0;
52     x = x0;
53     y_new = 0.0;
54     salida_tres.open("h3.dat", ios::out);
55     for(int i = 0; i <= N[2] - 1; i++){
56         y_new = euler(y, x, h[2]);
57
```



```
58     y = y_new;
59     x = x + h[2];
60
61     salida_tres << x << "\t" << y << endl;
62 } // END FOR
63 salida_tres.close();
64
65
66 y = y0;
67 x = x0;
68 y_new = 0.0;
69 salida_cuatro.open("h4.dat", ios::out);
70 for(int i = 0; i <= N[3] - 1; i++){
71     y_new = euler(y, x, h[3]);
72
73     y = y_new;
74     x = x + h[3];
75
76     salida_cuatro << x << "\t" << y << endl;
77 } // END FOR
78 salida_cuatro.close();
79
80 return 0;
81 } // END MAIN
82
83
84 double euler(double y, double x, double h){
85     return y + h*derivada(y,x);
86 } // END EULER
87
88
89 double derivada(double y, double x){
90     return y*y + 1;
91 } // END DERIVADA
```

Problema 2

Para la misma ecuación diferencial (1), se compararon los resultados entre el Método de Euler, el Método de Euler Modificado y el Método de Euler Mejorado. Esto, únicamente, con $h = 0.1$ como tamaño de paso. Entonces, la solución encontrada para cada uno de los métodos se muestra en la figura 2.

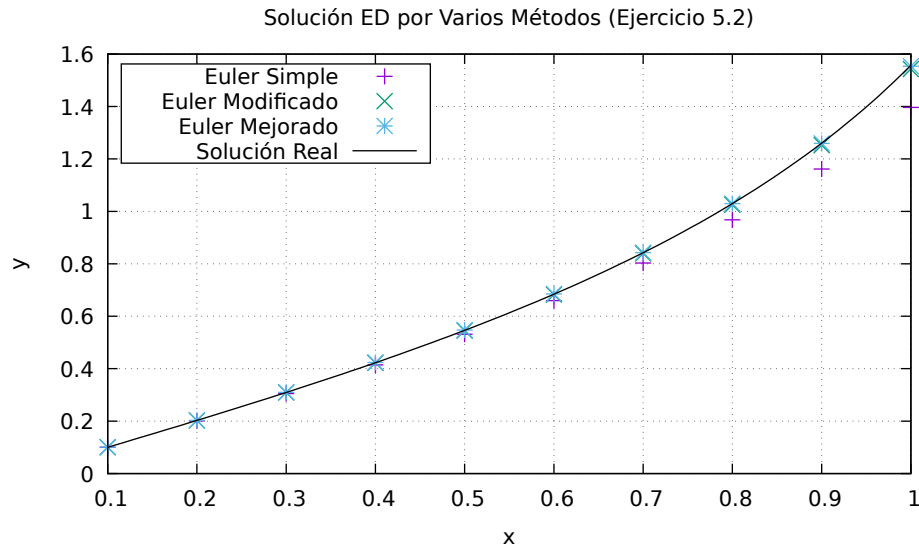


Figura 2: Gráfica de los distintos métodos de Euler para la ecuación (1).

El código creado para la solución es:

```

1 // Librerias
2 #include <iostream>
3 #include <fstream>
4
5 using namespace std;
6
7
8 double euler(double y, double x, double h);
9 double euler_modificado(double y, double x, double h);
10 double euler_mejorado(double y, double x, double h);
11 double derivada(double y, double x);
12
13
14 int main(){
15     const double y0 = 0.0;
16     const double x0 = 0.0;
17     const double h = 0.10;
18     const int N = 10;
19     ofstream salida_simple, salida_mejorado, salida_modificado;
20
21     double y = y0;
22     double x = x0;
23     double y_new = 0.0;
24
25
26     salida_simple.open("simple.dat", ios::out);
27     for (int i = 0; i <= N - 1; i++){
28         y_new = euler(y, x, h);

```



```
29
30     y = y_new;
31     x = x + h;
32
33     salida_simple << x << "\t" << y << endl;
34 } // END FOR
35 salida_simple.close();
36
37
38 y = y0;
39 x = x0;
40 y_new = 0;
41 salida_modificado.open("modificado.dat", ios::out);
42 for (int i = 0; i <= N - 1; i++){
43     y_new = euler_modificado(y, x, h);
44
45     y = y_new;
46     x = x + h;
47
48     salida_modificado << x << "\t" << y << endl;
49 } // END FOR
50 salida_modificado.close();
51
52
53 y = y0;
54 x = x0;
55 y_new = 0;
56 salida_mejorado.open("mejorado.dat", ios::out);
57 for (int i = 0; i <= N - 1; i++){
58     y_new = euler_mejorado(y, x, h);
59
60     y = y_new;
61     x = x + h;
62
63     salida_mejorado << x << "\t" << y << endl;
64 } // END FOR
65 salida_mejorado.close();
66
67 return 0;
68 } // END MAIN
69
70 double euler(double y, double x, double h){
71     return y + h*derivada(y,x);
72 } // END EULER
73
74 double euler_modificado(double y, double x, double h){
75     double x_mid = x + 0.5*h;
76     double y_mid = y + 0.5*h*derivada(y, x);
```



```
77     return y + h*derivada(y_mid, x_mid);
78 } // END EULER_MODIFICADO
79
80 double euler_mejorado(double y, double x, double h){
81     double y_tilde = y + h*derivada(y, x);
82     double y_imas1 = y + 0.5*h*( derivada(y, x) + derivada(y_tilde, x + h)
83         );
84     return y_imas1;
85 } // END EULER_MEJORADO
86
87 double derivada(double y, double x){
88     return y*y + 1;
89 } // END DERIVADA
```

Problema 3