Tarea 1

Física Computacional

Diego Sarceño

201900109

30 de septiembre de 2022

Problema 1

Dada la ecuación diferencial

$$y'(x) = y^2 + 1, \tag{1}$$

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 iteraciones[1].

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

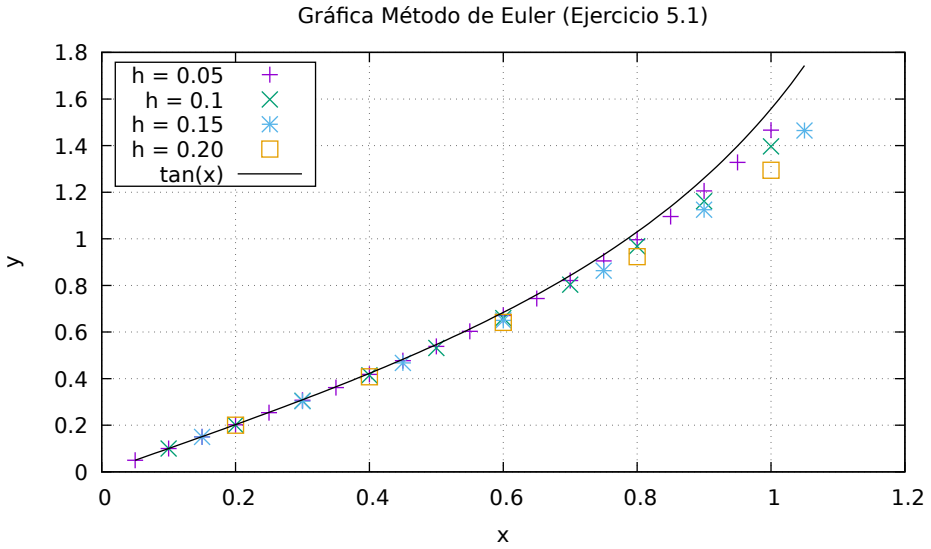

Figura 1: Gráfica para diferentes tamaños de paso.

El código creado para la solución es:

```
// Librerias
#include <iostream>
#include <fstream>

using namespace std;

double euler(double y, double x, double h);
double derivada(double y, double x);

```

[1]Dado 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.

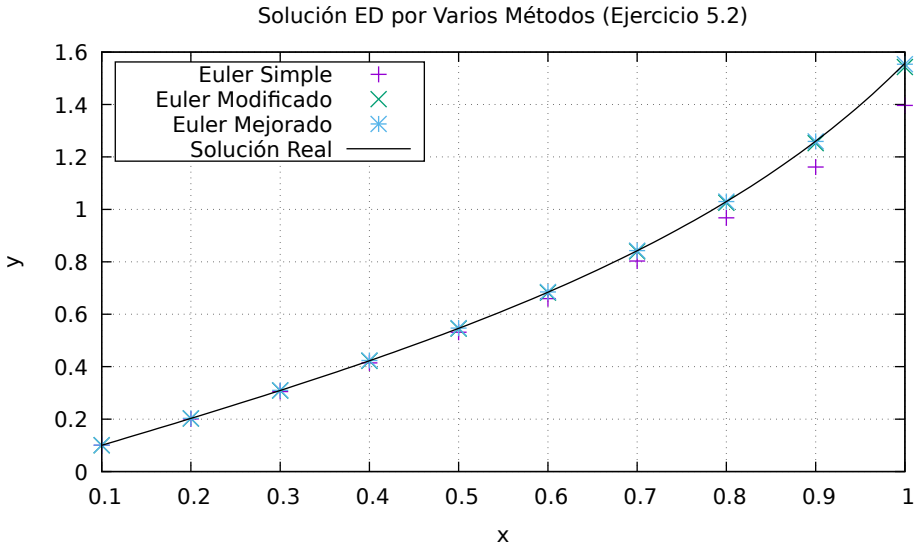

Figura 2: Gráfica de los distintos métodos de Euler para la ecuación (1).

El código creado para la solución es:

```
// Librerias
#include <iostream>
#include <fstream>

using namespace std;


double euler(double y, double x, double h);
double euler_modificado(double y, double x, double h);
double euler_mejorado(double y, double x, double h);
double derivada(double y, double x);


int main(){
  const double y0 = 0.0;
  const double x0 = 0.0;
  const double h = 0.10;
  const int N = 10;
  ofstream salida_simple, salida_mejorado, salida_modificado;

  double y = y0;
  double x = x0;
  double y_new = 0.0;


  salida_simple.open("simple.dat", ios::out);
  for (int i = 0; i <= N - 1; i++){
    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   return y_imas1;
84 } // END EULER_MEJORADO
85
86
87 double derivada(double y, double x){
88   return y*y + 1;
89 } // END DERIVADA
```

Problema 3

Dado el sistema de una masa en un resorte, se tiene la ecuación diferencial

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{kx}{m},$$

la cual se divide en las siguientes dos ecuaciones

$$\frac{\mathrm{d}v}{\mathrm{d}t} = -x,$$

$$\frac{\mathrm{d}x}{\mathrm{d}t} = v.$$

Estas se resuelven por medio del método de Euler Modificado y un paso $h = 0.1$. Con lo cual, se obtuvieron los resultados

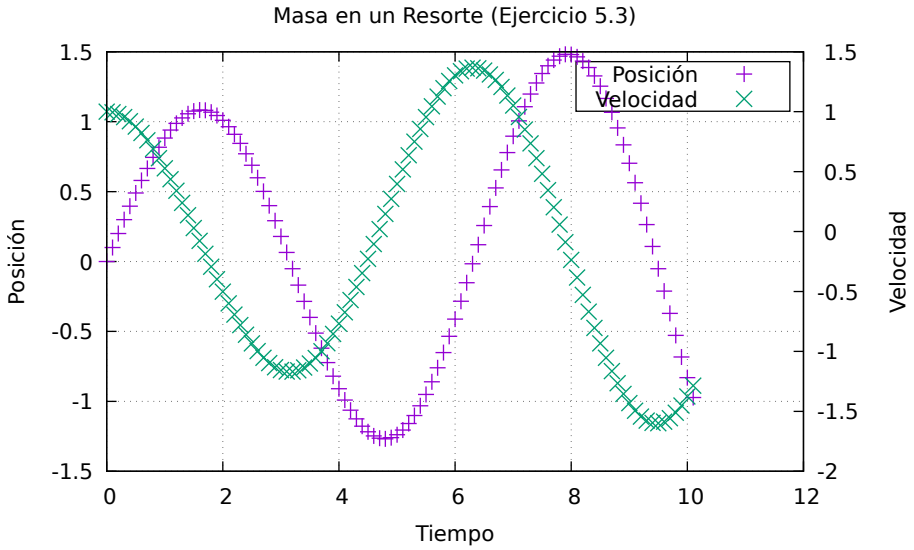

Figura 3: Posición y Velocidad en el tiempo.

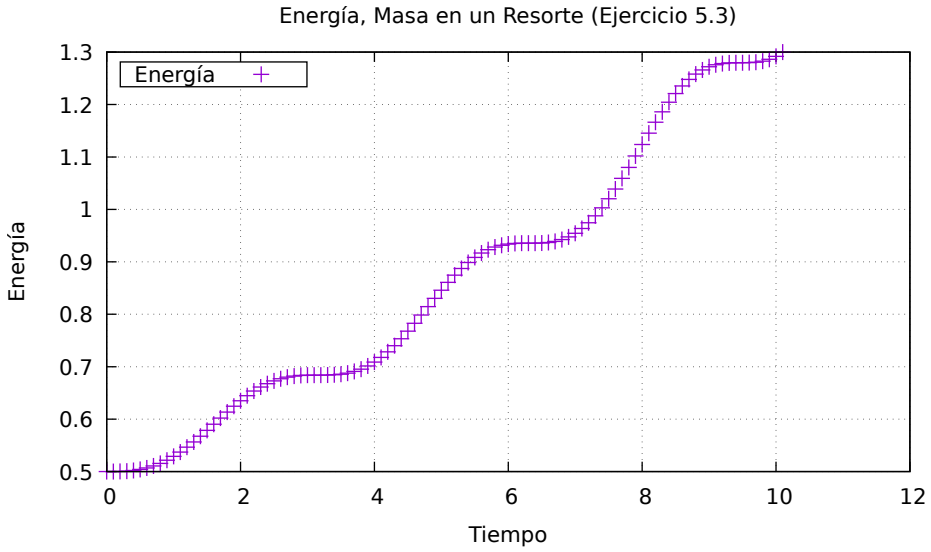

Figura 4: Energía.

```cpp
// Librerias
#include <iostream>
#include <fstream>

using namespace std;

double euler_modificado(double y, double v, double t, double h);
double euler_modificado2(double y, double v, double t, double h);
double derivada(double v, double t);
double derivada2(double y, double t);
double energia(double y, double v);


int main(){
  const double y0 = 0.0;
  const double v0 = 1.0;
  const double t0 = 0.0;
  const double h = 0.10;
  const double e0 = 0.5;
  const int N = 100;
  ofstream data;

  double y = y0;
  double v = v0;
  double t = t0;
  double E = e0;
  double y_new = 0.0;
  double v_new = 0.0;
  double E_new = 0.0;
```

```
30
31
32    data.open("data.dat", ios::out);
33    data << t << "\t" << y << "\t" << v << "\t" << E << endl;
34
35    for (int i = 0; i <= N; i++){
36      v_new = euler_modificado(y, v, t, h);
37      y_new = euler_modificado2(y, v, t, h);
38
39      y = y_new;
40      v = v_new;
41
42      E_new = energia(y, v);
43      E = E_new;
44
45      t = t + h;
46
47      data << t << "\t" << y << "\t" << v << "\t" << E << endl;
48    } // END FOR
49    data.close();
50
51    return 0;
52  } // END MAIN
53
54
55  double euler_modificado(double y, double v, double t, double h){
56    //double t_mid = t + h/2;
57    //double v_mid = v - 0.5*h*derivada2(y, t);
58    double y_mid = y + 0.5*h*derivada(v, t);
59
60    return v - h*y_mid;
61  } // END EULER_MODIFICADO
62
63  double euler_modificado2(double y, double v, double t, double h){
64    //double t_mid = t + h/2;
65    double v_mid = v - 0.5*h*derivada2(y, t);
66    //double y_mid = y + 0.5*h*derivada(v, t);
67
68    return y + h*v_mid;
69  } // END EULER_MODIFICADO
70
71  double derivada(double v, double t){
72    return v;
73  } // END DERIVADA
74
75
76  double derivada2(double y, double t){
77    return -y;
```

```
78 } // END DERIVADA
79
80
81 double energia(double y, double v){
82   return y*y/2 + v*v/2;
83 } // END ENERGIA
```