

Métodos de un paso para PVI en MATLAB

Tomo 1

Hugo Sánchez
Israel Heredia
Jhon Jairo Cevallos



ESPOCH
2023

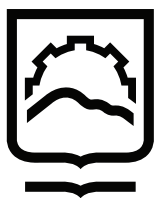
Métodos de un paso para PVI con MATLAB

Tomo1

Métodos de un paso para PVI con MATLAB

Tomo 1

Hugo Sánchez
Israel Heredia
John Jairo Cevallos



**Dirección de
Publicaciones**



espooh

Métodos de un paso para PVI con MATLAB. Tomo 1

© 2023 Hugo Sánchez, Israel Heredia, John Jairo Cevallos

© 2023 Escuela Superior Politécnica de Chimborazo

Panamericana Sur, kilómetro 1 ½
Instituto de Investigaciones
Dirección de Publicaciones Científicas
Riobamba, Ecuador
Teléfono: 593 (03) 2 998-200
Código Postal: EC0600155

Aval ESPOCH

Este libro se sometió a arbitraje bajo el sistema de doble ciego
(*peer review*)

Corrección y diseño:
Editorial Politécnica ESPOCH

Impreso en Ecuador

Prohibida la reproducción de este libro, por cualquier medio,
sin la previa autorización por escrito de los propietarios del
Copyright

CDU: 512 + 621.3

Métodos de un paso para PVI en Matlab

Riobamba: Escuela Superior Politécnica de Chimborazo

Dirección de Publicaciones, año 2023

157 pp. vol: 17,6 x 25 cm

ISBN: 978-9942-44-500-1

1. Álgebra
2. Electrónica

ÍNDICE GENERAL

INTRODUCCIÓN	8
CAPÍTULO I	10
1. INTRODUCCIÓN AL CÁLCULO NUMÉRICO	10
1.1. ERROR DE REDONDEO	11
1.2. ERROR DE TRUNCAMIENTO	13
1.3. DEFINICIÓN DEL ERROR	14
1.3.1. Errores que coincidan con la solución analítica	14
1.3.2. Error cuando no se conoce la solución analítica	14
1.4. APLICACIÓN DE MÉTODOS NUMÉRICOS A LA RESOLUCIÓN DE PROBLEMAS	15
1.4.1. Problemas de valor inicial	15
CAPÍTULO II	17
2. PROBLEMAS DE VALOR INICIAL MÉTODOS DE UN PASO	17
2.1. PVI DEFINIDOS POR ECUACIONES DIFERENCIALES DE PRIMER ORDEN	17
2.1.1. Ejemplo 1	18
2.2. PVI DEFINIDOS POR SISTEMAS DE ECUACIONES DIFERENCIALES DE PRIMER ORDEN	19
2.2.1. Ejemplo 2	21
2.3. PVI DEFINIDOS POR ECUACIONES DIFERENCIALES DE ORDEN SUPERIOR	22
2.3.1. Ejemplo 3	23

CAPÍTULO III	25
3. MÉTODO DE EULER	25
3.1. ALGORITMO DE EULER	26
3.2. ALGORITMO DE EULER PARA SISTEMAS DE ECUACIONES	28
3.3. PROBLEMAS PVI CON EULER	29
3.3.1. Problema 1	29
3.3.2. Problema 2	31
3.3.3. Problema 3	34
3.3.4. Problema 4	37
3.3.5. Problema 6	43
3.3.6. Problema 7	45
3.3.7. Problema 8	48
3.3.8. Problema 9	52
3.3.9. Problema 10	55
3.3.10. Problema 11	58
3.3.11. Problema 12	62
3.3.12. Problema 13	63
3.3.13. Problema 14	64
3.3.14. Problema 15	67
 CAPÍTULO IV	 69
4. MÉTODO DE HEUN	69
4.1. ALGORITMO DE HEUN	70
4.2. ALGORITMO DE HEUN PARA SISTEMAS DE ECUACIONES	71
4.3. PROBLEMAS PVI CON HEUN	72
4.3.1. Problema 1	72
4.3.2. Problema 2	74

4.3.3. Problema 3	76
4.3.4. Problema 4	78
4.3.5. Problema 5	80
4.3.6. Problema 6	83
4.3.7. Problema 7	84
4.3.8. Problema 8	86
4.3.9. Problema 9	90
4.3.10. Problema 10	93
4.3.11. Problema 11	96
4.3.12. Problema 12	100
4.3.14. Problema 14	105
4.3.15. Problema 15	107
4.3.16. Problema 16	110
4.3.17. Problema 17	112
4.3.18. Problema 18	113
4.3.19. Problema 19	114
4.3.20. Problema 20	116
 CAPÍTULO V	 119
5. MÉTODO DE RUNGE-KUTTA	119
5.1. ALGORITMO DE RUN	121
5.2. PROBLEMAS PVI CON RUNGE-KUTTA	123
5.2.1. Problema 1	123
5.2.2. Problema 2	124
5.2.3. Problema 3	126
5.2.4. Problema 4	127
5.2.5. Problema 5	128

5.2.6. Problema 6	130
5.2.7. Problema 7	132
5.2.8. Problema 8	133
5.2.9. Problema 9	135
5.2.10. Problema 10	136
5.2.12. Problema 12	142
5.2.13. Problema 13	144
5.2.14. Problema 14	146
5.2.15. Problema 15	149
5.2.16. Problema 16	151
5.2.17. Problema 17	153
 BIBLIOGRAFÍA	 155

INTRODUCCIÓN

Los métodos numéricos permiten obtener soluciones aproximadas a una serie de problemas. Antes de la revolución del transistor, que dio lugar a los circuitos integrados y, por ende, a las computadoras, los métodos numéricos requerían de un coste temporal tremendo para poder obtener las soluciones aproximadas de determinados problemas de la ciencia y la ingeniería sin conocer su solución analítica; sin embargo, en la segunda parte del siglo XX, los métodos numéricos tienen su mayor auge. El motivo fundamental es la aparición de las máquinas de computación. Si bien es cierto que no funcionaban con la velocidad que lo hace cualquier equipo doméstico de nuestros días, supusieron un gran salto cualitativo para desarrollar este tipo de técnicas.

Uno de los tipos de *software* comerciales que nos permite optimizar el tiempo que dedicamos a la resolución de problemas de forma numérica es Matlab. Originalmente, no era un lenguaje de programación, sino una calculadora matricial interactiva; de hecho, Matlab es un acrónimo de Matrix Laboratory (primera versión 1981). Pero la versión de la que disponemos hoy es mucho más avanzada. Cada semestre sale una versión en la que se mejora el código y se implementan nuevas funciones.

La parte teórica de este libro presenta los principales aspectos en los que se basan los métodos numéricos, para luego proceder a aplicarlos en el desarrollo de problemas, por lo que es una herramienta de trabajo de fácil entendimiento para los lectores.

A lo largo de este libro, se concentra la atención en los métodos numéricos de un paso (Euler, Heun, Runge Kutta). Estos métodos son fundamentales para el aprendizaje, de tal forma que se los puede emplear en el estudio de solución de ecuaciones diferenciales. A continuación, se hace un resumen de los contenidos de cada capítulo.

El capítulo I trata sobre los errores que se obtienen al resolver una ecuación diferencial al utilizar aproximaciones y las características de los problemas de valor inicial PVI.

El capítulo II trata sobre el estudio de problemas de valor inicial (PVI) definido por ecuaciones diferenciales, por sistemas de ecuación de primer orden y por sistemas de ecuación de diferenciales de orden superior.

En el capítulo III, se analiza y se construye el programa basado en Euler para la implementación en Matlab y luego se procede a utilizar para la solución de ecuaciones diferenciales en las áreas de matemáticas, física y química.

En el capítulo IV, se analiza y se construye el programa basado en Heun para la implementación en Matlab y luego se procede a utilizar para la solución de ecuaciones diferenciales en las áreas de matemáticas, física y química.

En el capítulo V, se analiza y se construye el programa basado en Runge Kutta para la implementación en Matlab y luego se procede a utilizar para la solución de ecuaciones diferenciales en las áreas de matemáticas, física y química.

CAPÍTULO I

1. INTRODUCCIÓN AL CÁLCULO NUMÉRICO

Los problemas matemáticos a los que se han enfrentado a lo largo de la historia y se enfrentan hoy matemáticos e ingenieros son múltiples. La forma ideal de darles solución es proporcionar una respuesta en forma de ecuación, de número, de resultado que siempre se cumpla bajo cualquier condición. No obstante, no siempre es posible obtener la solución analítica, o no siempre quien debe resolver los problemas tiene suficiente experiencia o antecedentes matemáticos específicos.

Por ejemplo, supongamos resolver la ecuación:

$$f(x) = x^2 + 2x - 3 = 0$$

Esta ecuación tiene una solución analítica y además es muy sencilla de obtener. De este modo, cualquier persona que tenga conocimientos en matemáticas, por mínimos que sean, sería capaz de indicar que la solución es $x = \{-3, 1\}$.

Sin embargo, si planteamos otro problema que parecería fácil como:

$$f(x) = x - e^x = 0$$

Podemos ver que su solución no se puede obtener por métodos analíticos. A no ser que la persona resolviera este problema por intuición o por una idea acerca de métodos numéricos, nunca sería capaz de adivinar que la respuesta está cerca del punto $x = 0.568$.

La alternativa a la resolución analítica de los problemas es la resolución numérica. Para poder abordarla, nos adentramos en el campo del análisis numérico. A lo largo de este libro, vamos a presentar los métodos que se utiliza para resolver problemas de valor inicial.

La característica fundamental del análisis numérico es tener soluciones aproximadas. De forma que siempre vamos a cometer un error que será inherente al proceso numérico. También existirán otros errores derivados de la toma de datos o de la precisión de las máquinas sobre las que trabajamos.

En este primer capítulo, nos centraremos en todos los errores que se van a dar en los métodos numéricos, tanto por obtener soluciones aproximadas como por las herramientas que vamos a utilizar.

1.1. ERROR DE REDONDEO

Estos errores vienen originados por no disponer de precisión infinita para la presentación de magnitudes.

Para esto, debemos introducir el concepto de **cifras significativas**. Cualquier magnitud que sea medible puede ser representada a partir de un número y su correspondiente unidad. Al instante de representarlo, se tiene que tomar una decisión acerca de **cuántas cifras vamos a utilizar**, de forma que el número con el que presentamos esa cantidad coincida con la que estamos midiendo.

La tabla 1.1 muestra diferentes casos en los que las cifras significativas tienen diferentes implicaciones en función del número que estamos presentando.

Tabla 1.1. Ejemplos de cifras significativas

Número	Cifra significativa	Características
135,23	5	Número decimal superior a 1
0,000475	3	Número decimal inferior a 1
$3,350 \times 10^3$	4	Notación científica

Las personas y las computadoras solemos trabajar con cifras significativas. Por ejemplo, el valor de π que utilizamos es 3.1416, pero en realidad tiene número infinito de decimales. Debido al uso de las cifras significativas, estamos cometiendo un error, conocido como error de redondeo.

Matlab trabaja por defecto con el formato *double*, que es de doble precisión. Este utiliza treinta y dos dígitos de precisión para almacenar los datos, aunque después puede presentarlos por pantalla en función de cómo se defina la instrucción *format*.

Ejemplo 1: con el formato *double*, el número π esta almacenado con treinta y dos dígitos significativos. Si lo presentamos con *format short*, vamos a obtener una representación con cinco dígitos.

```
>>format short
```

```
>>pi
```

```
Ans= 3.1416
```

Sin embargo, si lo representamos con *format long*, obtendremos una representación con quince dígitos.

```
>>format long
```

```
>>pi
```

```
Ans= 3.141592653589793
```

Matlab también puede trabajar con un mayor número de dígitos para que los errores de redondeo se minimicen. Este concepto se conoce como **aritmética de precisión variable**. Como inconveniente, tendremos unos cálculos más lentos. Para ello, es suficiente con ejecutar el comando *vpa* (dígitos), donde dígitos es la cantidad de dígitos significativos con los que vamos a trabajar. A partir de esto, la variable que vamos a trabajar será de formato *vpa* y será tratada como simbólica en lugar de numérica. Para volver a transformarla en variable numérica, deberemos introducirla como argumento en el comando *double*.

Ejemplo 2. Almacenamos el valor de π con los treinta y dos dígitos con que trabaja Matlab por defecto.

```
>>pi32= pi;
```

En lugar de trabajar con treinta y dos dígitos de precisión, queremos trabajar con cincuenta. Para ello le indicamos a Matlab que lo realice de la siguiente forma:

```
>>digits(50)
```

Almacenamos el valor de π con los cincuenta dígitos:

```
>>pi50= vpa(pi);
```

Otorgamos la diferencia entre ambos valores almacenados:

```
>>dpi= pi50 – pi32;
```

```
>>double(dpi)
```

```
Ans= -5.0926e – 60
```

1.2. ERROR DE TRUNCAMIENTO

Los errores de truncamiento son los que cometemos al realizar una aproximación matemática en lugar de un procedimiento exacto. Para obtener un procedimiento sobre las características de estos errores, se debe considerar una formulación matemática que se utiliza ampliamente en los métodos numéricos para expresar funciones de manera aproximada: la serie de Taylor.

La serie de Taylor nos permite utilizar **aproximaciones tan precisas como deseemos** de las funciones sobre las que vamos a trabajar, de forma que, sobre estas aproximaciones, utilizaremos los métodos numéricos para resolver problemas de forma aproximada.

Teorema 1. Teorema de Taylor. Sean funciones f y sus primeras $n+1$ derivadas continuas en un intervalo que contiene a y x . Entonces, el valor de la función en x viene dado por:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n$$

Donde R_n es el residuo, que se define por:

$$R_n = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}, \xi \in (a, x)$$

Al utilizar el polinomio de Taylor de grado 2, lo que estamos haciendo es tomar los términos hasta el segundo grado y depreciar el resto, de forma que tendríamos como residuo R_3 . La diferencia entre el valor exacto de la función y el residuo R_3 . La diferencia entre el valor exacto de la función y el residuo R_n es lo que se conoce como error de truncamiento.

Ejemplo 3. Se requiere realizar una aproximación de la función derivada. Para ello utilizamos el desarrollo en serie de Taylor de primer orden tomando como $a = x_i$ y como $x = x_{i+1}$, desarrollado:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) \leftrightarrow f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Al utilizar esta aproximación, habremos cometido un error de truncamiento.

$$R_2 = \int_{x_i}^{x_{i+1}} \frac{(x-t)^2}{2} f''(t) dt$$

1.3. DEFINICIÓN DEL ERROR

Una vez determinados los errores más comunes al utilizar métodos numéricos y representaciones o almacenamientos en computadoras, vamos a aportar una serie de definiciones de error que nos van a permitir evaluar la bondad de un método iterativo.

1.3.1. Errores que coinciden con la solución analítica

Cuando se conoce la solución analítica y_a y la numérica y_n , se puede obtener una serie de errores conocidos como verdaderos. Se trata de errores que se pueden medir y no de estimaciones.

El primer error que se presenta es el numérico ϵ_n , que se puede calcular como la diferencia entre ambas soluciones.

$$\epsilon_n = y_a - y_n$$

La magnitud de este error no va a indicar la buena o mala aproximación que se ha realizado. Para este propósito está el error relativo porcentual ϵ_r que nos indica en un valor porcentual la bondad de nuestra aproximación a partir de:

$$\epsilon_r [\%] = 100 \frac{\epsilon_n}{y_a}$$

1.3.2. Error cuando no se conoce la solución analítica

En los métodos iterativos, no se conoce la solución analítica del problema, pero sí se puede averiguar cómo una secuencia se va aproximando cada vez a la solución esperada. En este caso, los errores los obtenemos a partir de dos aproximaciones consecutivas. De este modo, uno de los errores más habituales que se utiliza es:

$$y_{(k+1)} - y_k$$

Y, si lo necesitamos de forma porcentual, tendríamos:

$$100 \frac{y_{k-1} - y_k}{y_{k+1}}$$

1.4. APLICACIÓN DE MÉTODOS NUMÉRICOS A LA RESOLUCIÓN DE PROBLEMAS

A lo largo de este libro, vamos a centrarnos en la resolución de determinados problemas de ingeniería a partir de métodos numéricos. A continuación, se describen los problemas que vamos a resolver y las técnicas que debemos utilizar.

1.4.1. Problemas de valor inicial

Un problema de valor inicial está compuesto por una ecuación diferencial de primer orden y una condición inicial. Es decir:

$$y'(t) = f(t, y(t)), t \in [a, b], y(a) = y_a$$

Existen problemas de valor inicial definidos a partir de un sistema de ecuaciones diferenciales de primer orden:

$$y'_1(t) = f_1(t, y_1(t), y_2(t), \dots, y_m(t))$$

$$y'_2(t) = f_2(t, y_1(t), y_2(t), \dots, y_m(t))$$

$$y'_m(t) = f_m(t, y_1(t), y_2(t), \dots, y_m(t))$$

Con $t \in [a, b]$, y una condición inicial sobre cada una de las ecuaciones:

$$y_i(a) = y_{i,a}, i = 1, 2, \dots, m$$

De esta manera, podemos resolver una ecuación diferencial de orden mayor a uno:

$$y^{(m)}(t) = f(t, y(t), y'(t), \dots, y^{(m-1)}(t))$$

Si conocemos los valores de las condiciones iniciales para la función incógnita $y(t)$ y sus derivadas de orden $m-1$:

$$y(a) = y_a, y'(a) = y'_a, \dots, y^{(m-1)}(a) = y^{(m-1)}_a$$

Los métodos que vamos a utilizar para resolver los tres tipos de problemas de valor inicial se clasifican como de un paso y multipaso. En ambos casos, las soluciones se obtienen por intervalos en orden creciente.

En el caso de los métodos de un paso, se utiliza la información del intervalo actual para obtener la solución en el intervalo siguiente. Este tipo de método describiremos en los capítulos siguientes, y profundizaremos sobre los métodos de Euler, Heun y Runge-Kutta.

Respecto de los métodos de multipaso, podemos distinguir principalmente entre dos tipos: los métodos explícitos y los métodos implícitos.

En los métodos explícitos, se obtiene la solución en el intervalo siguiente a partir de la información actual y de los anteriores.

CAPÍTULO II

2. PROBLEMAS DE VALOR INICIAL MÉTODOS DE UN PASO

Los problemas de valor inicial se pueden representar para diferentes esquemas de ecuaciones diferenciales.

2.1. PVI DEFINIDOS POR ECUACIONES DIFERENCIALES DE PRIMER ORDEN

Un PVI tiene como objetivo encontrar la solución $y(t)$ de la ecuación diferencial de primer orden:

$$y'(t) = f(t, y(t)), t \in [a, b]$$

Conociendo la condición inicial:

$$y(a) = y_a$$

Teorema 1. Si suponemos $D = \{(t, y): t \in [a, b], y \in \mathbb{R}\}$, y que $f(x, y)$ es continua en D . Si dados $(t_1, y_1), (t_2, y_2) \in D$, existe una constante $L > 0$ tal que f cumple la condición de Lipschitz:

$$|f(t_1, y_1) - f(t_2, y_2)| \leq L |y_1 - y_2|$$

Entonces, el PVI dado por:

$$y'(t) = f(t, y(t)), t \in [a, b], y(a) = y_a$$

Tienen una solución única $y(t), t \in [a, b]$.

A partir del teorema 1, las técnicas analíticas tratan de obtener la solución continua $y(t)$. En el caso que nos ocupa, que son las técnicas numéricas, vamos a obtener la solución discreta $y_i = y[(t)]_i, i = 0, 1, 2, \dots, n$, donde $t_i, i = 0, 1, 2, \dots, n$ son los nodos de la discretización.

Para ello, el primer paso que debemos dar consiste en discretizar la variable independiente t , dividiendo en intervalos completos en N subintervalos, como se ilustra en la figura 2.1.

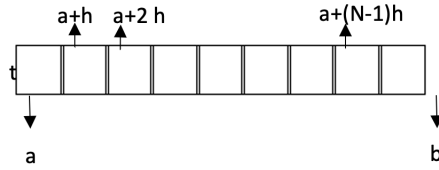


Figura 2.1. Discretización

Dividimos el intervalo $[a, b]$ en N subintervalos $[t_i, t_{(i+1)}]$, cuyos extremos son los nodos equiespaciados $t_i = a + ih$, $i = 0, 1, 2, \dots, N$, siendo $h = \frac{b-a}{N}$ siendo el paso de la partición. La solución numérica la obtendremos en los nodos t_i , de forma que $y_1 \approx y(t_i)$.

Una vez discretizada la variable independiente, veamos cómo resolver de forma genérica en Matlab este tipo de PVI. Para ello, utilizaremos el comando *ode23* de Matlab, que es un método numérico de bajo orden. En primer lugar, debemos definir la función $y'(t) = f(t, y(t))$ a través de una función. A continuación, definiremos los valores de t discretizados y, por último, indicaremos las condiciones iniciales $y(a) = y_a$.

2.1.1. Ejemplo 1

Resuelva el PVI:

$$y'(t) = (1 - 2t)y(t), t \in [0, 3], y(0) = 1$$

Use treinta subintervalos. Represente la función $y(t)$.

Definimos una función que implemente la función $f(t, y(t)) = (1 - 2t)y(t)$:

```
function dy = ejemplo(t,y)
```

```
dy = (1-2*t)*y;
```

```
end
```

A continuación, defines la variable independiente discretizada tomando treinta subintervalos:

```
>> a=0;  
>> b=3;  
>> N=30;  
>> h=(b-a)/N;  
>> td=a:h:b;  
Defines la function inicial.  
>> ya=1;  
Ejecutamos la function od23  
>> [t,y]=ode23('ejemplo1',td,ya);  
Representamos la solución  
>> plot(t,y,'o-')
```

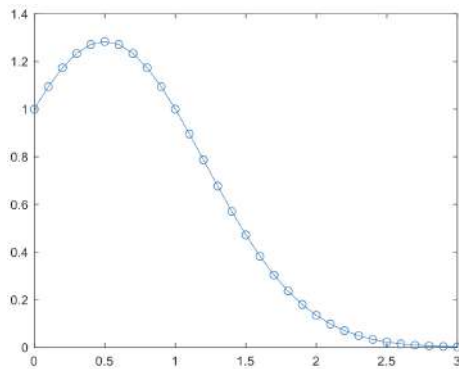


Figura 2.2. Representación solución Ejemplo 1

2.2. PVI DEFINIDOS POR SISTEMAS DE ECUACIONES DIFERENCIALES DE PRIMER ORDEN

Un sistema de ecuaciones diferenciales de primer orden expresa las relaciones que verifican varias funciones desconocidas y sus primeras derivadas respecto a la variable independiente. De este modo el objetivo es resolver:

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), \dots, y_m(t)) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \dots, y_m(t)) \\ \vdots \\ y_m'(t) = f_m(t, y_1(t), y_2(t), \dots, y_m(t)) \end{cases}$$

Si conocemos los valores de m funciones incógnitas en el instante inicial t_a es decir:

$$y_1(t_a) = y_{1a}, y_2(t_a) = y_{2a}, \dots, y_m(t_a) = y_{ma}$$

De esta manera, estamos ante un PVI.

La estructura de este sistema permite utilizar una notación compacta en términos vectoriales. De este modo, si escribimos las funciones incógnitas como $Y: \mathbb{R} \rightarrow \mathbb{R}^m$, tal que:

$$Y(t) = [y_1(t) \ y_2(t) \ \dots \ y_m(t)]^T$$

Y las ecuaciones como componente de una función vectorial de varias variables $F: \mathbb{R}^{(m+1)} \rightarrow \mathbb{R}^m$, tal que:

$$F(t, Y(t)) = \begin{cases} f_1(t, y_1(t), \dots, y_m(t)) \\ f_2(t, y_1(t), \dots, y_m(t)) \\ \vdots \\ f_m(t, y_1(t), \dots, y_m(t)) \end{cases}$$

Se puede escribir el sistema como:

$$Y'(t) = F(t, Y(t)),$$

Siendo la condición inicial del PVI:

$$Y_a = Y(t_a) = [y_1(t_a) \ y_2(t_a) \ \dots \ y_m(t_a)]^T = [y_{1a} \ y_{2a} \ \dots \ y_{ma}]^T$$

En la discretización \mathbf{e} , la variable independiente sigue la misma estructura que en el caso de las ecuaciones diferenciales de primer orden. Las modificaciones que se tienen que realizar se basan en las estructuras de la función que introducimos como *function* y en la adaptación del método numérico al caso vectorial. En cuanto a la función, tomará como parámetros de entrada las funciones $f_i(t, y_1(t), \dots, y_m(t))$, $i = 1, 2, \dots, m$.

Como parámetro de salida, tendremos un vector con los valores de las f_i evaluadas en los puntos a los que llame el método numérico correspondiente.

Para cada uno de los métodos numéricos que desarrollemos, veremos su implementación vectorial. Sobre el comando **ode23**, no hay que introducir modificaciones. No obstante, hay que destacar que, como parámetros de entrada, habrá que introducir, además de la función que implementa f_i y la variable independiente discretizada, las condiciones iniciales y_{ia} , $i = 1, 2, \dots, m$. El resultado el comando **ode23** nos aportará las soluciones y_i , $i = 1, 2, \dots, m$, una en cada columna.

2.2.1. Ejemplo 2

Obtenga la solución numérica del PVI

$$\begin{cases} y_1'(t) = y_2(t), \\ y_2'(t) = -y_1(t) - 2e^t + 1, \\ y_3'(t) = -y_1(t) - e^t + 1 \end{cases}$$

Con $t \in [0, 2]$, $y_1(0) = 1$, $y_2(0) = 0$, $y_3(0) = 1$, tomando como paso $h = 0.1$ y representar las soluciones $y_1(t)$, $y_2(t)$, e $y_3(t)$.

Definimos una function que implemente el sistema:

```
function Dy = ejemplo(t,y)
```

```
y1=y(1);
```

```
y2=y(2);
```

```
y3=y(3);
```

```
Dy=[y2; -y1-2*exp(t)+1; -y1-exp(t)+1];
```

```
end
```

A continuación, definimos las variables independientes discretizada, tomando el paso indicado:

```
>> a=0;
```

```
>> b=2;
```

```
>> h=0.1;
```

```
>> td=a:h:b;
```

Definimos el vector de condiciones iniciales:

```
>> ya=[1;0;1];
```

Ejecutamos la función *ode23*:

```
>> [t,y]=ode23('ejemplo',td,ya);
```

Representamos la solución:

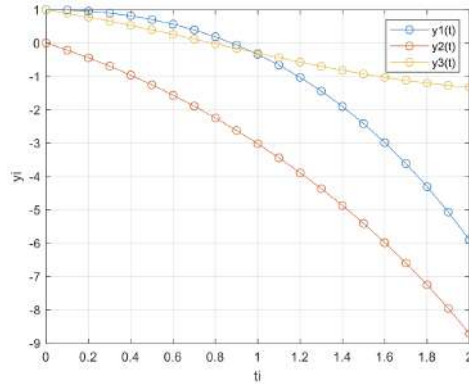


Figura 2.3. Representación de la solución Ejemplo 2

2.3. PVI DEFINIDOS POR ECUACIONES DIFERENCIALES DE ORDEN SUPERIOR

Las ecuaciones diferenciales de orden m son aquellas ecuaciones diferenciales en las que interviene la función incógnita y sus derivadas hasta el orden m . De forma general, podemos expresar una ecuación diferencial de orden m como:

$$y^m(t) = f(t, y(t), y'(t), \dots, y^{(m-1)}(t))$$

Si conocemos los valores de las condiciones iniciales de la función incógnita y las derivadas hasta el orden $m-1$, es decir:

$$y(t_a) = y_a, y'(t_a) = y_a', \dots, y^{(m-1)}(t_a) = y_a^{(m-1)}$$

Estamos ante un PVI.

El procedimiento para la resolución de estas ecuaciones diferenciales es similar al de los sistemas de ecuaciones diferenciales de primer orden por medio del cambio de variables:

$$y_1(t) = y(t), y_2(t) = y'(t), y_3(t) = y''(t), \dots, y_m(t) = y^{(m-1)}(t)$$

De este modo, la ecuación diferencial de orden m se ha convertido en el sistema de ecuaciones diferenciales:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ y_3(t) \\ \vdots \\ y_m(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ y'(t) \\ y''(t) \\ \vdots \\ y^{(m-1)}(t) \end{bmatrix} \rightarrow \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ y_3'(t) \\ \vdots \\ y_m'(t) \end{bmatrix} = \begin{bmatrix} y'(t) \\ y''(t) \\ y'''(t) \\ \vdots \\ y^{(m)}(t) \end{bmatrix} = \begin{bmatrix} y_2(t) \\ y_3(t) \\ y_4(t) \\ \vdots \\ f(t, y_1(t), \dots, y_m(t)) \end{bmatrix}$$

2.3.1. Ejemplo 3

Obtenga la solución del PVI dada por el péndulo de longitud $L=1\text{m}$ y la aceleración de la $\frac{d^2\theta(t)}{dt^2} - \frac{g}{L} \sin\theta(t) = 0$ gravedad $g = 9.8 \frac{m}{s^2}$, modelado por la ecuación diferencial:

viendo que $\theta(0) = \frac{\pi}{6} \text{rad}$, $\theta'(0) = \frac{0m}{s}$ y tomando como paso $h = 0.1$. Indique el valor del ángulo θ para $t = \{0, 1, 2\}$ y represente la evolución de $\theta(t)$.

Reescribimos la ecuación diferencial, dejando a un lado la derivada de mayor orden:

$$\theta''(t) - \frac{g}{L} \sin \theta(t) = 0 \leftrightarrow \theta''(t) = \frac{g}{L} \sin \theta(t)$$

Se trata de una ecuación de orden 2. Procedemos a realizar los cambios de variables:

$$\begin{bmatrix} \theta_1(t) \\ \theta_2(t) \end{bmatrix} = \begin{bmatrix} \theta(t) \\ \theta'(t) \end{bmatrix} \rightarrow \begin{bmatrix} \theta_1'(t) \\ \theta_2'(t) \end{bmatrix} = \begin{bmatrix} \theta'(t) \\ \theta''(t) \end{bmatrix} = \begin{bmatrix} \theta_2(t) \\ \frac{g}{L} \sin \theta_1(t) \end{bmatrix}$$

Definimos una function que implemente el sistema:

```
function Dy = ejemplo3(t,theta)
```

```
L=1;g=9.8;
```

```
theta1=theta(1);theta2=theta(2);
```

```
Dy=[theta2;g/L*sin(theta1)];
```

```
end
```

definimos las variables dependientes discretizadas, tomando el paso indicado:

```
>> a=0;
```

```
>> b=2;
```



```
>> h=0.1;
```

```
>> td=a:h:b;
```

Definimos el vector de condiciones iniciales:

```
>> theta_a=[pi/6;0];
```

Ejecutamos la función ode23:

```
>> [t,theta]=ode23('ejemplo3',td,theta_a);
```

Obtenemos los valores del $\theta(t)$ para $t = \{0,1,2\}$:

Tabla 1.1. Valores de $\theta(t)$ para $t = \{0,1,2\}$

t	$\theta(t)$
0	0,5236
1	3,8255
2	5,6092

Representamos la solución:

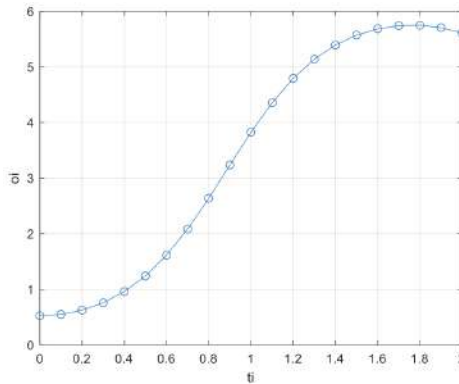


Figura 2.4. Representación de la solución Ejemplo 3

En el caso de las ecuaciones diferenciales de orden m , la solución aportada por los métodos numéricos en una matriz de m columnas. La columna i de la solución numérica representa la función solución $y^{(i-1)}(t)$, para $i = 1, 2, \dots, m-1$. Por eso, para representar la función solución $y(t)$, tomamos solo la primera columna, como ocurre en el ejemplo 3.

CAPÍTULO III

3. MÉTODO DE EULER

Hasta el momento, hemos resuelto diferentes tipos de PVI a partir del comando ode23 de Matlab. Sin embargo, hay métodos mucho más eficientes y sofisticados que, desde un punto de vista numérico y de cálculo computacional, mejoran considerablemente las prestaciones del comando ode23.

Comencemos por un método muy elemental e intuitivo. Haciendo el desarrollo en serie de Taylor de la función $y(t)$:

$$y(t+h) = y(t) + hy'(t) + \mathcal{O}(h^2)$$

Donde $\mathcal{O}(h^2)$ son los términos de orden mayor o igual. Podemos despejar la función derivada como:

$$y'(t) = \frac{y(t+h) - y(t)}{h}$$

y, reemplazando en la expresión de la ecuación diferencial de primer orden, obtenemos:

$$y'(t) = f(t, y(t)) \leftrightarrow \frac{y(t+h) - y(t)}{h} = f(t, y(t)) \leftrightarrow y(t+h) = y(t) + hf(t, y(t))$$

Siendo h muy próximo a cero. Si conocemos la condición inicial $y(t_a) = y_a$, podemos obtener los siguientes puntos:

$$y_{(k+1)} = y_k + hf(t_k, y_k), \quad k = 0, 1, \dots, N-1$$

Al realizar el desarrollo de la serie de Taylor, se descartaron en la expresión del error los términos de orden mayor o igual a dos y se obtuvo el error local de truncamiento:

$$e_{k+1} = y(t_{k+1}) - (y(t_k) + hy'(t_k)) = \frac{h^2}{2} y''(\xi_k) = \mathcal{O}(h^2), \quad \xi_k \in (t_k, t_{k+1})$$

Al ser $y''(t)$ continua:

$$\sum_{k=0}^{N-1} y''(\xi_k) = N y''(\xi), \quad \xi \in [a, b]$$

Después de N pasos, el error global acumulado es:

$$\sum_{k=0}^{N-1} \frac{h^2}{2} y''(\xi_k) = \frac{h^2}{2} N y''(\xi) = \frac{1}{2} (b-a) y''(\xi) h = \mathcal{O}(h), \xi \in [a, b]$$

Por tanto, el orden del método de Euler es 1. En el teorema 2, recogeremos este resultado:

Teorema 2. Sea f tal que $y'(t) = f(t, y(t))$ con condiciones iniciales $y(t_a) = y_a$. Si

$$\left| \frac{\partial f}{\partial y}(t, y) \right| \leq M_1, \text{ e } |y''(t)| \leq M_2, \quad \forall t \in [a, b]$$

Entonces la aproximación de **Euler** y_0, y_1, \dots, y_N convergen a la solución exacta $y(t)$ cuando $h \rightarrow 0$, y el error de discretización del método de Euler

$$|e_k| = |y(t_k) - y_k| \leq \frac{hM_2}{2M_1} (e^{M_1(t_k - t_a)} - 1)$$

Satisface $E(h) = \mathcal{O}(h)$.

Una manera de estimar el orden de un método conociendo la solución analítica es a partir de:

$$\log_2 \left(\lim_{N \rightarrow \infty} \frac{E_{N/2}}{E_N} \right)$$

Donde E_n es el error máximo entre las soluciones analíticas y numérica al usar N subintervalos.

En caso de no conocer la solución analítica, también es posible obtener una estimación numérica del orden del método a partir de:

$$\log_2 \left(\lim_{N \rightarrow \infty} \frac{E_{N/2}}{E_N} \right)$$

Donde $E_N = \|y_k^{(N)} - y_{1+2k}^{(2N)}\|$, $k = 0, 1, \dots, N$, e $y^{(N)}$ representa la solución discretizada del PVI al usar N subintervalos.

3.1. ALGORITMO DE EULER

Para la construcción del algoritmo de Euler, consideramos:

- Entradas

a, b extremos del intervalo

y_1 condicion inicial

N número de subintervalos

- Proceso

Cálculo del paso de integración, h

Obtención del vector nodos, x

Inicialización del vector solución y en a

Para k desde 1 hasta N

$$y_{(k+1)} = y_k + hf(x_k, y_k)$$

Fin par k

- Salida x, y

De esta manera, construimos el código en Matlab en que se implementa el método de Euler como se muestra a continuación:

```
function [x,y]=Euler_orden1_N(f,a,b,y0,N)
%Método de Euler de PVI orden 1
h=(b-a)/N;
x=a:h:b; % crea el vector como fila
x=x(:); % : cambia de fila a columna para trabajar
y=zeros(N+1,1); % : crea e vector y
y(1)=y0; % primera posición del vector y
for k=1:N
    K1=feval(f,x(k),y(k));
    y(k+1)=y(k)+h*K1;
end
```

3.2. ALGORITMO DE EULER PARA SISTEMAS DE ECUACIONES

Para la construcción del algoritmo de Euler para sistemas de ecuaciones, consideramos:

- Entradas

a, b extremos del intervalo

y_1 condicion inicial

N número de subintervalos

- Proceso

Cálculo del paso de integración, h

Obtención del vector nodos, x

Inicialización del vector solución y en a

Para k desde 1 hasta N para vectores

$$y_{(k+1)} = y_k + hf(x_k, y_k)$$

Fin par k

- Salida t, y

De esta manera construimos el código en Matlab que se implementa el método de Euler implícito como se muestra a continuación:

```
function [t,y]=Euler_orden1_Sistemas(f,a,b,y0,N)
% Método de Euler de PVI orden 1 con M sistemas de ecuaciones
h=(b-a)/N;
t=a:h:b; % crea el vector como fila
t=t(:); % : cambia de fila a columna para trabajar
y=zeros(N+1,length(y0));
```

```
y(1,:)=y0; % primera posición del vector
for k=1:N
    K1=feval(f,t(k),y(k,:))';
    y(k+1,:)=y(k,:)+h*K1;
end
```

3.3. PROBLEMAS PVI CON EULER

3.3.1. Problema 1

Sea el siguiente problema de valor inicial:

$$y'(x) = \cos(x) \times (y - x) \quad x \in [-\pi \quad \pi]$$

Con las condiciones iniciales $y(-\pi) = 0$.

a) Escriba una función que describa la ecuación diferencial de primer orden.

```
function [f,df] = Problema01(x,y)
f = cos(x)*y - x;
end
```

b) Resuelva el PVI para el intervalo $x \in [-\pi \quad \pi]$ con pasos $h = \frac{2\pi}{10}$ y represente $y(x)$

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$

Donde $N = \frac{\pi+\pi}{\frac{2\pi}{10}}$ obteniendo el número de iteraciones $N = 10$

Llamamos al método:

```
>> [x,y]=Euler_orden1_N('Problema01',-pi,pi,0,10)
```

Graficamos la solución

```
>> plot(x,y(:,1))
```

```
>> grid on
```

```
>> title('MÉTODO EULER PARA ECUACIONES')
>> xlabel('x')
>> ylabel('y')
```

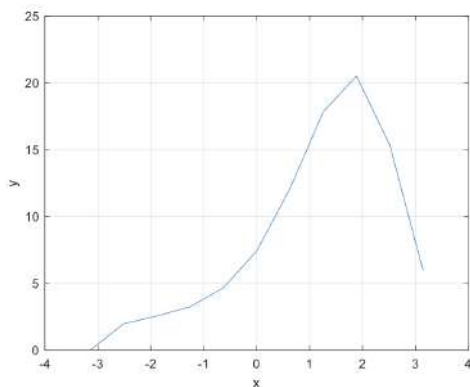


Figura 3.1. Representación de la solución Problema 1 con Euler

c) Indicar los valores de $y(x)$ para $x \in \{-\pi - 2\pi/5, 0, 3\pi/5, \pi\}$

Encontramos las posiciones requeridas:

$\frac{b-a}{N+1} = \frac{\pi - (-\pi - 2\pi/5)}{N+1} = \frac{3\pi/5}{N+1}$		
Posición - a	x	$N+1$
• π $\frac{2\pi}{-\pi + \pi} \frac{10+1}{x} = 1$	• $-2\pi/5$ $\frac{2\pi}{-2\pi/5 + \pi} \frac{11}{x} = 3.3 \rightarrow 4$	• 0 $\frac{2\pi}{0 + \pi} \frac{11}{x} = 5.5 \rightarrow 6$
• $3\pi/5$ $\frac{2\pi}{3\pi/5 + \pi} \frac{11}{x} = 8.6 \rightarrow 9$	• π $\frac{2\pi}{\pi + \pi} \frac{11}{x} = 10.8 \rightarrow 11$	

Hallamos las posiciones en Matlab:

```
>> x=[x(1) x(4) x(6) x(9) x(11)]
```

```
>> y=[y(1) y(4) y(6) y(9) y(11)]
```

x(t)	-3,1416	-1,2566	0	1,8850	3,1416
y	0	3,2390	7,4197	20,5001	5,9610

d) Indicar una estimación del orden de convergencia.

Código de Matlab para encontrar la convergencia:

```
>> [x,y1]=Euler_orden1_N('Problema01',-pi,pi,0,1*10);sol1=y1(:,1);
>> [x,y2]=Euler_orden1_N('Problema01',-pi,pi,0,2*10);sol2=y2(:,1);
>> [x,y3]=Euler_orden1_N('Problema01',-pi,pi,0,4*10);sol3=y3(:,1);
>> [x,y4]=Euler_orden1_N('Problema01',-pi,pi,0,8*10);sol4=y4(:,1);
>> [x,y5]=Euler_orden1_N('Problema01',-pi,pi,0,16*10);sol5=y5(:,1);
>> [x,y6]=Euler_orden1_N('Problema01',-pi,pi,0,32*10);sol6=y6(:,1);
>> [x,y7]=Euler_orden1_N('Problema01',-pi,pi,0,64*10);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
orden=log2(relacion)
```

Orden de aproximación:

0.5595	0.6952	0.8543	0.9226	0.9607
--------	--------	--------	--------	--------

Como podemos observar, el orden de convergencia se acerca a uno verificando que el programa es apto para la simulación con Euler.

3.3.2. Problema 2

Modelo de programación de una llama: denotamos $y(t)$ el radio de la bola que modeliza la llama.

$$y'(x) = y^2 - y^3 \quad x \in [0 \ 200]$$

Con las condiciones iniciales $y(0) = 0.01$:

a) Escriba una función que describa la ecuación diferencial de primer orden.

`function [f,y] = Problema02(x,y)`


```
f=y.^2-y.^3
```

```
end
```

b) Resuelva el PVI para el intervalo $x \in [0, 200]$ con pasos $h = 0.01$ y representar $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$

Donde $N = \frac{200-0}{0.01}$ obteniendo el número de iteraciones $N = 20\,000$

Comando para llamar al método:

```
>> [x,y]=Euler_orden1_N('Problema02',0,200,0.01,20000)
```

Comandos utilizados para graficar la solución:

```
>> plot(x,y(:,1))
```

```
>> grid on
```

```
>> title('METODO EULER PARA ECUACIONES')
```

```
>> xlabel('x')
```

```
>> ylabel('y')
```

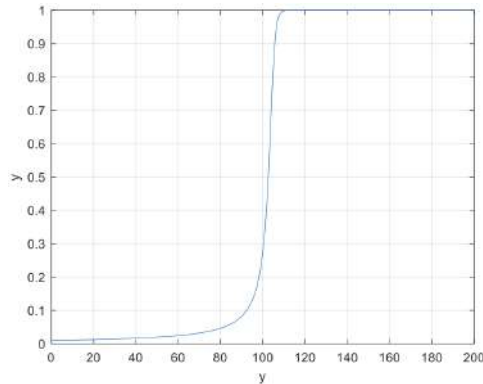


Figura 3.2. Representación de la solución Problema 2 con Euler

c) Indicar los valores de $y(x)$ para $x \in \{0, 30, 65, 200\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} - a = \frac{b-a}{N+1} x$$

<ul style="list-style-type: none"> • 0 $\begin{array}{r} 200 \\ 0-0 \end{array} \quad \begin{array}{r} 20001 \\ x \end{array} = 1 \rightarrow 1$	<ul style="list-style-type: none"> • 30 $\begin{array}{r} 200 \\ 30 \end{array} \quad \begin{array}{r} 20001 \\ x \end{array} = 3000,15 \\ \rightarrow 3001$	<ul style="list-style-type: none"> • 65 $\begin{array}{r} 200 \\ 65 \end{array} \quad \begin{array}{r} 20001 \\ x \end{array} = 65500.3 \\ \rightarrow 6501$
<ul style="list-style-type: none"> • 200 $\begin{array}{r} 200 \\ 200 \end{array} \quad \begin{array}{r} 20001 \\ x \end{array} = 20001$		

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(3001) x(6501) x(20001)]
```

```
>> y=[y(1) y(3001) y(6501) y(20001)]
```

x(t)	0	30	65	200
y	0.0100	0.0142	0.0277	1.0000

d) Indicar una estimación del orden de convergencia.

Código de Matlab para encontrar la convergencia:

```
>> [x,y1]=Euler_orden1_N('Problema02',0,200,0.01,1*20000);sol1=y1(:,1);
```

```
>> [x,y2]=Euler_orden1_N('Problema02',0,200,0.01,2*20000);sol2=y2(:,1);
```

```
>> [x,y3]=Euler_orden1_N('Problema02',0,200,0.01,4*20000);sol3=y3(:,1);
```

```
>> [x,y4]=Euler_orden1_N('Problema02',0,200,0.01,8*20000);sol4=y4(:,1);
```

```
>> [x,y5]=Euler_orden1_N('Problema02',0,200,0.01,16*20000);sol5=y5(:,1);
```

```
>> [x,y6]=Euler_orden1_N('Problema02',0,200,0.01,32*20000);sol6=y6(:,1);
```

```
>> [x,y7]=Euler_orden1_N('Problema02',0,200,0.01,64*20000);sol7=y7(:,1);
```

```
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
```

```
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
```

```
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
```

```
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/max(e5) max(e5)/max(e6) ]';
```

orden=log2(relacion)

Orden de aproximación

0.9997	0.9999	0.9999	1.0000	1.0000
--------	--------	--------	--------	--------

3.3.3. Problema 3

Modelo de programación de una llama: denotamos $y(t)$ el radio de la bola que modeliza la llama:

$$y'(x) = (1-2x)y(x) \quad x \in [0, 3]$$

Con las condiciones iniciales $y(0)=1$

a) Escriba una función que describa la ecuación diferencial de primer orden.

`function [f,df] = Problema03(x,y)`

`f = (1-2*x).*y;`

`end`

b) Resuelva el PVI para el intervalo $x \in [0,3]$ con dieciséis subintervalos y represente $y(x)$

Comando para llamar al método:

`>> [x,y]=Euler_orden1_N('Problema03',0,3,1,16)`

Comandos utilizados para graficar la solución:

`>> plot(x,y(:,1))`

`>> grid on`

`>> title('METODO EULER PARA ECUACIONES')`

`>> xlabel('x')`

`>> ylabel('y')`

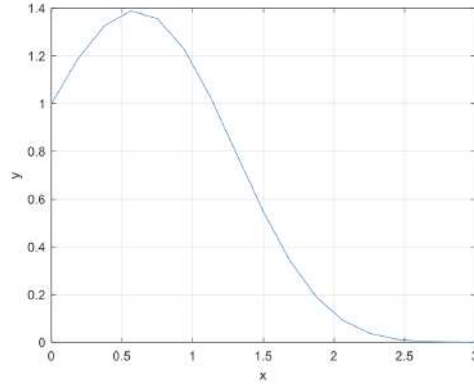


Figura 3.3. Representación de la solución $y(x)$ Problema 3 con Euler

c) Indicar los valores de $y(x)$ para $x \in \{0, 0.75, 2.25, 3\}$.

Encontramos las posiciones requeridas

$$\frac{b-a}{N+1} = \Delta x$$

<ul style="list-style-type: none"> 0 $\frac{3-0}{0-0} = 1 \rightarrow 1$ 	<ul style="list-style-type: none"> 0.75 $\frac{3-0.75}{0.75} = 4.25 \rightarrow 5$ 	<ul style="list-style-type: none"> 2.25 $\frac{3-2.25}{2.25} = 12.75 \rightarrow 13$
<ul style="list-style-type: none"> 3 $\frac{3-3}{3} = 17$ 		

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(5) x(13) x(17)]'
```

```
>> y=[y(1) y(5) y(13) y(17)]'
```

$x(t)$	0	0.7500	2.2500	3.0000
y	1.0000	1.3563	0.0380	0.0001

d) Indicar una estimación del orden de convergencia:

```
>> [x,y1]=Euler_orden1_N('Problema03',0,3,1,1*16);sol1=y1(:,1);
```

```
>> [x,y2]=Euler_orden1_N('Problema03',0,3,1,2*16);sol2=y2(:,1);
```

```
>> [x,y3]=Euler_orden1_N('Problema03',0,3,1,4*16);sol3=y3(:,1);
```

```
>> [x,y4]=Euler_orden1_N('Problema03',0,3,1,8*16);sol4=y4(:,1);
```

```
>> [x,y5]=Euler_orden1_N('Problema03',0,3,1,16*16);sol5=y5(:,1);
>> [x,y6]=Euler_orden1_N('Problema03',0,3,1,32*16);sol6=y6(:,1);
>> [x,y7]=Euler_orden1_N('Problema03',0,3,1,64*16);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';orden=log2(relacion)
```

e) Si la solución exacta es $y(x) = e^{\frac{1}{4} - (\frac{1}{2} - x)^2}$, compare gráficamente la solución exacta con dichas aproximaciones, grafique el error cometido y calcule su error máximo.

Código para calcular el error entre la solución exacta y la aproximada:

```
>> yexacta=exp(1/4-(1/2-x).^2);
>> Maxerror=max(abs(y-yexacta))
Maxerror = 0.1688
>> hold on
>> plot(x,yexacta)
>> legend('y_aproximado','y_error')
```

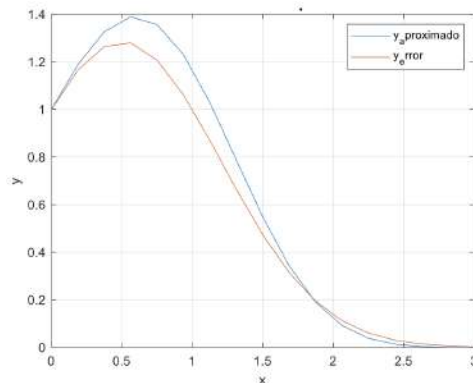


Figura 3.4. Representación de la solución exacta vs. la aproximada Problema 3

Código para encontrar la gráfica del error cometido:

```
>> plot(x,abs(y(:,1)-yexacta))
>> grid on
>> title('Grafica del error cometido')
```

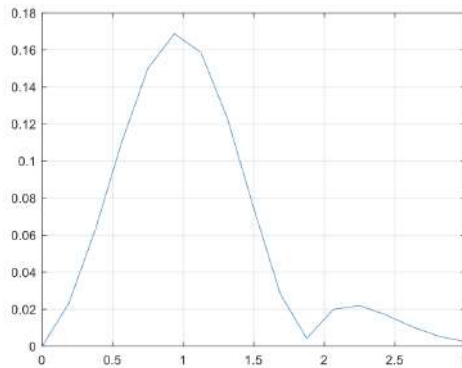


Figura 3.5. Representación del error cometido Problema 3

3.3.4. Problema 4

La velocidad de desintegración del radio es proporcional a la cantidad, donde $k = 4,1 \times 10^{-4}$, $y'(x) = -k \times y(x)$ $x \in [0 \ 1500]$. Con las condiciones iniciales de variación de la cantidad de radio existente en una respuesta de 10gr.

a) Escriba una función que describa la ecuación diferencial de primer orden.

```
function [f,df] = Problema04(x,y)
```

```
k = 4.1*1e-4;
```

```
f = -k*y;
```

```
end
```

b) Resuelva el PVI para el intervalo $x \in [0,1500]$ con diez subintervalos y presente $y(x)$

Comando para hallar el método:

```
>> [x,y]=Euler_orden1_N('Problema04',0,1500,10,10)
```

Comandos utilizados para graficar la solución:

```
>> plot(x,y(:,1))
>> grid on
>> xlabel('x')
>> ylabel('y')
>> title('METODO EULER PARA ECUACIONES')
```

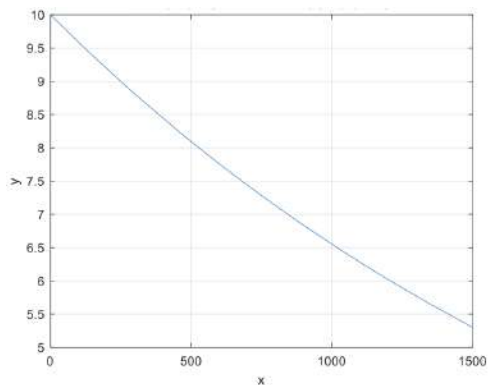


Figura 3.6. Representación de la solución $y(x)$ Problema 4 con Euler

c) Indicar los valores de $y(x)$ para $x \in \{0, 450, 1200, 1500\}$.

Encontramos las posiciones requeridas

	$\frac{b-a}{N+1}$ Posición - a	$\frac{N+1}{x}$
<ul style="list-style-type: none"> 0 $\frac{1500 - 0}{11} = 1 \rightarrow 1$ 	<ul style="list-style-type: none"> 450 $\frac{1500 - 450}{11} = 3.3 \rightarrow 4$ 	<ul style="list-style-type: none"> 1200 $\frac{1500 - 1200}{11} = 8.8 \rightarrow 9$
<ul style="list-style-type: none"> 1500 $\frac{1500 - 1500}{11} = 11$ 		

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(4) x(9) x(11)]'
>> y=[y(1) y(4) y(9) y(11)]'
```

$\mathbf{x(t)}$	0	450	1200	1500
\mathbf{y}	10	8.2661	6.0183	5.3008

d) Indicar una estimación del orden de convergencia.

Código de Matlab para encontrar la convergencia:

```
>> [x,y1]=Euler_orden1_N('Problema04',0,1500,10,1*10);sol1=y1(:,1);
>> [x,y2]=Euler_orden1_N('Problema04',0,1500,10,2*10);sol2=y2(:,1);
>> [x,y3]=Euler_orden1_N('Problema04',0,1500,10,4*10);sol3=y3(:,1);
>> [x,y4]=Euler_orden1_N('Problema04',0,1500,10,8*10);sol4=y4(:,1);
>> [x,y5]=Euler_orden1_N('Problema04',0,1500,10,16*10);sol5=y5(:,1);
>> [x,y6]=Euler_orden1_N('Problema04',0,1500,10,32*10);sol6=y6(:,1);
>> [x,y7]=Euler_orden1_N('Problema04',0,1500,10,64*10);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';orden=log2(relacion)
```

Orden de aproximación:

1.0352	1.0173	1.0086	1.0043	1.0021
--------	--------	--------	--------	--------

Si la solución exacta es $y(x) = 10 e^{-kx}$, compare gráficamente la solución exacta con dichas aproximaciones, grafique el error cometido y calcule su error máximo.

Código para calcular el error entre la solución exacta y la aproximada:

```
>> yexacta=10*exp(-4.1*1e-4*x);
>> Maxerror=max(abs(y-yexacta))
Maxerror = 0.1056
>> hold on
>> plot(x,yexacta)
```



```
>> legend('y_aproximado','y_error')
>> title('Solucion Exacta vs Aproximada')
```

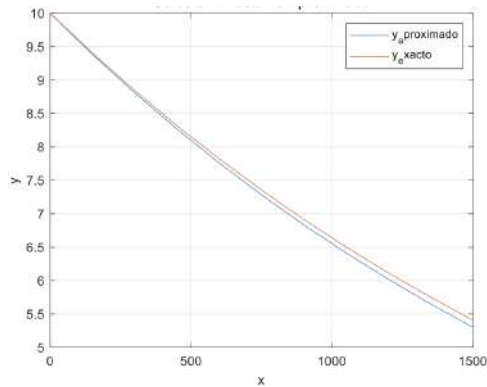


Figura 3.7. Representación de la solución exacta vs. la aproximada Problema 4

Código para encontrar la gráfica del error cometido:

```
>> plot(x,abs(y(:,1)-yexacta))
>> grid on
>> xlabel('x')
>> ylabel('y')
>>title('Grafica del Error cometido')
```

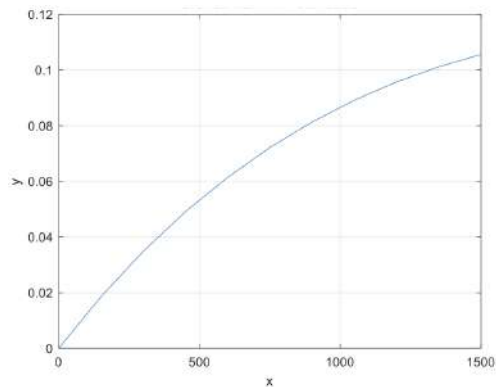


Figura 3.8. Representación del error cometido Problema 4

Sea el siguiente problema de valor inicial:

$$y'(x) = -\frac{y}{3} + \frac{e^x}{3y^3} \quad x \in [0 \ 5].$$

Con las condiciones iniciales $y(0) = -1$

a) Escriba una función que describa la ecuación diferencial de primer orden

```
function [f,df] = Problema05(x,y)
```

```
    f = -y/3+exp(x)./(3*y.^3);
```

```
end
```

b) Resuelva el PVI para el intervalo $x \in [0,5]$ con $N = 10$ subintervalos y represente $y(x)$

Llamamos al método:

```
>> [x,y]=Euler_orden1_N('Problema05',0,5,-1,10)
```

Comandos utilizados para graficar la solución aproximada:

```
>> plot(x,y(:,1))
```

```
>> grid on
```

```
>> xlabel('x')
```

```
>> ylabel('y')
```

```
>> title('METODO EULER PARA ECUACIONES')
```

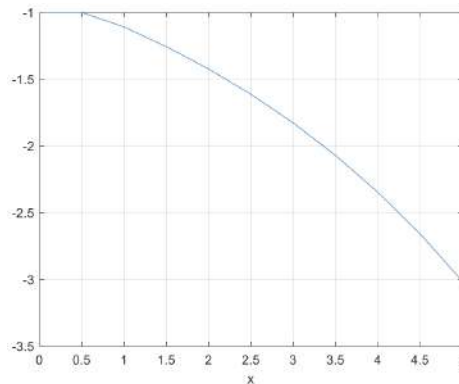


Figura 3.9. Representación de la solución $y(x)$ Problema 5 con Euler

c) Indicar los valores de $y(x)$ para $x \in \{0,1,4.5,5\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} - a \quad \frac{b-a}{x} \quad \frac{N+1}{x}$$

<ul style="list-style-type: none"> 0 $\frac{5}{0} \frac{11}{x} = 1 \rightarrow 1$	<ul style="list-style-type: none"> 1 $\frac{5}{1} \frac{11}{x} = 2.2 \rightarrow 3$	<ul style="list-style-type: none"> 4.5 $\frac{5}{4.5} \frac{11}{x} = 9.9 \rightarrow 10$
<ul style="list-style-type: none"> 5 $\frac{5}{5} \frac{11}{x} = 11$		

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(3) x(10) x(11)]'
```

```
>> y=[y(1) y(3) y(10) y(11)]'
```

x(t)	0	1	4.5	5
y	1	-1.1081	-2.6597	-3.0138

d) Indicar una estimación del orden de convergencia.

```
>> [x,y1]=Euler_orden1_N('Problema05',0,5,-1,1*10);sol1=y1(:,1);
```

```
>> [x,y2]=Euler_orden1_N('Problema05',0,5,-1,2*10);sol2=y2(:,1);
```

```
>> [x,y3]=Euler_orden1_N('Problema05',0,5,-1,4*10);sol3=y3(:,1);
```

```
>> [x,y4]=Euler_orden1_N('Problema05',0,5,-1,8*10);sol4=y4(:,1);
```

```
>> [x,y5]=Euler_orden1_N('Problema05',0,5,-1,16*10);sol5=y5(:,1);
```

```
>> [x,y6]=Euler_orden1_N('Problema05',0,5,-1,32*10);sol6=y6(:,1);
```

```
>> [x,y7]=Euler_orden1_N('Problema05',0,5,-1,64*10);sol7=y7(:,1);
```

```
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
```

```
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
```

```
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
```

```
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/  
max(e5) max(e5)/max(e6) ]';
```

>> orden=log2(relacion)

Orden de aproximación:

1.5620	1.1552	1.0803	1.0394	1.0186
--------	--------	--------	--------	--------

3.3.5. Problema 6

Sea el siguiente problema de valor inicial:

$$y''(x) = -y - 2\sin(x) \quad x \in [0, 2\pi].$$

Con las condiciones iniciales $y(0) = 0$, $y'(0) = 1$.

Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = -y - 2\sin(x) \quad x \in [0, 2\pi] \rightarrow z_2' = -z_1 - 2\sin(x) \quad x \in [0, 2\pi]$$

Con las condiciones iniciales $z_1(0) = 0$, $z_2(0) = 1$.

Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

`function f = Problema06(x,z)`

`f = [z(2); -z(1)-2*sin(x)];`

`end`

Resuelva el PVI para el intervalo $x \in [0, 2\pi]$ con paso de $h = \frac{\pi}{8}$ subintervalos y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2\pi-0}{\frac{\pi}{8}}$ obteniendo el número de iteraciones $N = 16$

Procedemos a llamar a la función:

```
>> [t,y]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],16)
```

Comandos utilizados para graficar $y(x)$, la solución aproximada:

```
>> plot(t,y(:,1))
```

```
>> grid on
```

```
>> xlabel('x')
```

```
>> ylabel('y')
```

```
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

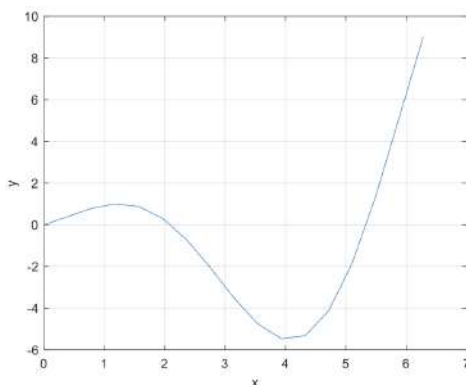


Figura 3.10. Representación de la solución $y(x)$ Problema 6 con Euler

d) Indicar los valores de $y(x)$ para $x \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\right\}$.

Encontramos las posiciones requeridas:

$\frac{b-a}{N+1} = \frac{2\pi - 0}{17} = \frac{2\pi}{17}$		
$\text{Posición} - a = \frac{N+1}{x} \cdot x$		
<ul style="list-style-type: none"> • 0 $\frac{2\pi}{0} \cdot \frac{17}{x} = 1 \rightarrow 1$	<ul style="list-style-type: none"> • $\frac{\pi}{2}$ $\frac{2\pi}{\frac{\pi}{2}} \cdot \frac{17}{x} = 4.25 \rightarrow 5$	<ul style="list-style-type: none"> • π $\frac{2\pi}{\pi} \cdot \frac{17}{x} = 8.5 \rightarrow 9$
<ul style="list-style-type: none"> • $\frac{3\pi}{2}$ $\frac{2\pi}{\frac{3\pi}{2}} \cdot \frac{17}{x} = 12.75 \rightarrow 13$	<ul style="list-style-type: none"> • 2π $\frac{2\pi}{2\pi} \cdot \frac{17}{x} = 17$	

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(5) t(9) t(13) t(17)]
```

```
>> y=[y(1) y(5) y(9) y(13) y(17)]
```

x(t)	0	0.0313	0.0625	0.0938	0.1250
y	0	0.8744	-3.4883	-4.1341	9.0370

e) Indicar una estimación del orden de convergencia.

```
>> [t,y1]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],1*16);sol1=y1(:,1);
>> [t,y2]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],2*16);sol2=y2(:,1);
>> [t,y3]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],4*16);sol3=y3(:,1);
>> [t,y4]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],8*16);sol4=y4(:,1);
>> [t,y5]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],16*16);sol5=y5(:,1);
>> [t,y6]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],32*16);sol6=y6(:,1);
>> [t,y7]=Euler_orden1_NM('Problema06',0,2*pi,[0 1],64*16);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ];
>> orden=log2(relacion)
```

Orden de aproximación:

1.4552	0.9135	1.0095	1.0145	1.0186
--------	--------	--------	--------	--------

3.3.6. Problema 7

El oscilador de van der Pol es un oscilador con amortiguamiento no lineal, cuya expresión es:

$$y''(x) = u(1 - y^2) y' - y \quad x \in [0 \ 20]$$

Con las condiciones iniciales $y(0) = 2$, $y'(0) = 0$.

Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = u(1-y^2)y' - y \quad x \in [0, 20] \rightarrow z_2' = u(1-z_1^2)z_2 - z_1 \quad x \in [0, 20]$$

Con las condiciones iniciales $z_1(0) = 2$, $z_2(0) = 0$.

Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema07(x,z)
```

```
u = 0; %caso no amortiguado u = 0, caso amortiguado u = 1
```

```
f = [z(2); u*(1-z(1).^2).*z(2)-z(1)];
```

```
end
```

c) Resuelva el PVI para el intervalo $x \in [0, 20]$ con paso de $h = 0.1$ subintervalos y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{20-0}{0.1}$ obteniendo el número de iteraciones $N = 200$.

Para este ejercicio, ocuparemos el programa para sistemas de ecuaciones con Euler.

Llamamos a la función:

```
>> [t,y]=Euler_orden1_NM('Problema07',0,20,[2 0],200);
```

Comandos utilizados para graficar la solución aproximada:

```
>> plot(t,y(:,1))
```

```
>> grid on
```

```
>> xlabel('x')
```

```
>> ylabel('y')  
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

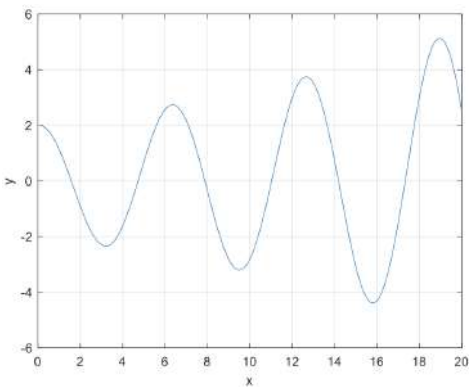


Figura 3.11. Representación de la solución $y(x)$ Problema 7 con Euler

d) Indicar los valores de $y(x)$ para $x \in \{0,8,14,16,20\}$.

Encontramos las posiciones requeridas:

$\begin{matrix} b-a & N+1 \\ \text{Posición} - a & x \end{matrix}$		
<ul style="list-style-type: none">• 0 $\begin{matrix} 20 & 201 \\ 0 & x \end{matrix} = 1 \rightarrow 1$	<ul style="list-style-type: none">• 8 $\begin{matrix} 20 & 201 \\ 8 & x \end{matrix} = 80.4 \rightarrow 81$	<ul style="list-style-type: none">• 14 $\begin{matrix} 20 & 201 \\ 14 & x \end{matrix} = 140.7 \rightarrow 141$
<ul style="list-style-type: none">• 16 $\begin{matrix} 20 & 201 \\ 16 & x \end{matrix} = 160.8 \rightarrow 161$	<ul style="list-style-type: none">• 20 $\begin{matrix} 20 & 201 \\ 20 & x \end{matrix} = 201$	

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(81) t(141) t(161) t(201)]  
>> y=[y(1) y(81) y(141) y(161) y(201)]
```

$\mathbf{x(t)}$	0	8	14	16	20
\mathbf{y}	2	-0.3550	0.7326	-4.3074	2.5298

e) Indicar una estimación del orden de convergencia.

```
>> [t,y1]=Euler_orden1_NM('Problema07',0,20,[2 0],200*1);sol1=y1(:,1);
>> [t,y2]=Euler_orden1_NM('Problema07',0,20,[2 0],200*2);sol2=y2(:,1);
>> [t,y3]=Euler_orden1_NM('Problema07',0,20,[2 0],200*4);sol3=y3(:,1);
>> [t,y4]=Euler_orden1_NM('Problema07',0,20,[2 0],200*8);sol4=y4(:,1);
>> [t,y5]=Euler_orden1_NM('Problema07',0,20,[2 0],200*16);sol5=y5(:,1);
>> [t,y6]=Euler_orden1_NM('Problema07',0,20,[2 0],200*32);sol6=y6(:,1);
>> [t,y7]=Euler_orden1_NM('Problema07',0,20,[2 0],200*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
>> orden=log2(relacion)
```

Orden de aproximación:

1.5268	1.2605	1.1295	1.0646	1.0322
--------	--------	--------	--------	--------

3.3.7. Problema 8

La distribución de potencial a lo largo de una línea satisface el problema de valor inicial.

$$y''(x) = y(x) \quad x \in [0 \quad L]$$

Con las condiciones iniciales $y(0) = 0.1$, $y'(0) = 0$.

Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = y(x) \quad x \in [0, L] \rightarrow z_2' = z_1 \quad x \in [0, L]$$

Con las condiciones iniciales $z_1(0) = 0.1$, $z_2(0) = 0$.

Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema08(x,z)
```

```
    f=[z(2); z(1)];
```

```
end
```

Resuelva el PVI cuando la longitud $L = 1m$, determinando el potencial cada 2 cm.

Represente $y(x)$.

Determinamos los intervalos si $L = 1m \Rightarrow \frac{100cm}{2cm}, h = 0.02m$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.02}$ obteniendo el número de iteraciones $N = 50$.

Para este ejercicio, ocuparemos el programa para sistemas de ecuaciones con Euler.

Llamamos a la función:

```
>>[t,y]=Euler_orden1_NM('Problema08',0,1,[0.1 0],50);
```

Comandos utilizados para graficar la solución aproximada.

```
>> plot(t,y(:,1))
```

```
>> grid on
```

```
>> xlabel('x')
```

```
>> ylabel('y')
```

```
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

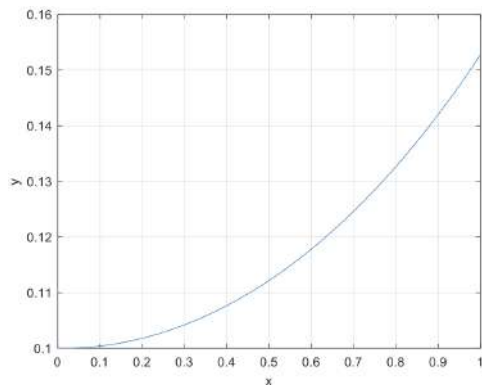


Figura 3.12. Representación de la solución $y(x)$ Problema 8 con Euler

d) Indicar los valores de $y(x)$ para $x \in \{0, 0.3, 0.5, 0.7, 1\}$.

Encontramos las posiciones requeridas:

$\frac{b-a}{N+1} = \frac{1-0}{51} = 0.0196 \approx 0.02$		
$\text{Posición} - a = \frac{N+1}{b-a} x$		
<ul style="list-style-type: none"> 0 $\frac{1}{0} \frac{51}{x} = 1 \rightarrow 1$	<ul style="list-style-type: none"> 0.3 $\frac{1}{0.3} \frac{51}{x} = 15.3 \rightarrow 16$	<ul style="list-style-type: none"> 0.5 $\frac{1}{0.5} \frac{51}{x} = 25.5 \rightarrow 26$
<ul style="list-style-type: none"> 0.7 $\frac{1}{0.7} \frac{51}{x} = 35.7 \rightarrow 36$	<ul style="list-style-type: none"> 1 $\frac{1}{1} \frac{51}{x} = 51$	

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(16) t(26) t(36) t(51)]
```

```
>> y=[y(1) y(16) y(26) y(36) y(51)]
```

$x(t)$	0	0.3	0.5	0.7	1
y	0.1	0.1042	0.1122	0.1246	0.1528

Si la solución exacta es $y(x) = 0.05e^x + 0.05e^{-x}$, compare gráficamente la solución con dichas aproximaciones, grafique el error cometido y calcule el error máximo.

Código para calcular el error entre la solución exacta y la aproximada:

```
>> yexacta=0.05*exp(t)+0.05*exp(-t);
```

```
>> Max_error=max(abs(yexacta-y(:,1)))

Max_error = 0.0015

>> plot(t,y(:,1))

>> hold on

>> plot(t,yexacta)

>> grid on

>> legend('y_aproximada','y_exacta')

>> title('Solucion Exacta vs Aproximada')

>> xlabel('x')

>> ylabel('y')
```

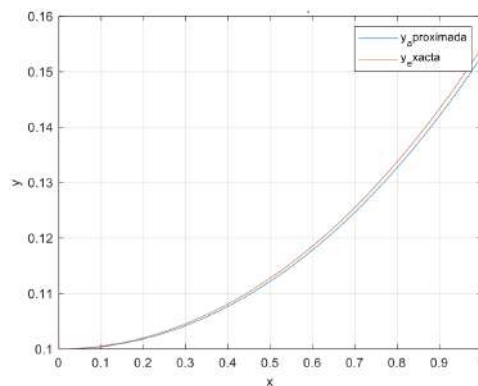


Figura 3.13. Representación de la solución exacta vs. la aproximada Problema 8

Código para encontrar la gráfica del error cometido:

```
>> figure

>> plot(t,abs(y(:,1)-yexacta))

>> grid on

>> xlabel('x')

>> ylabel('y')

>> title('Grafica del Error cometido')
```

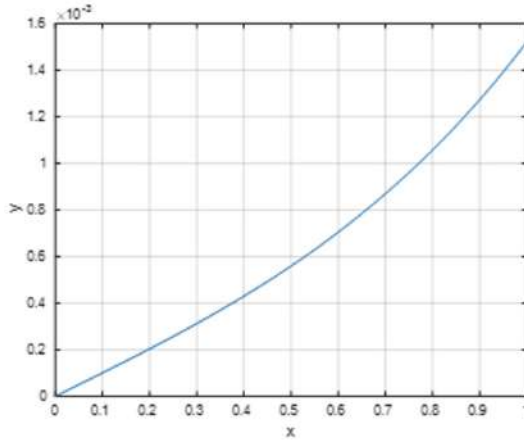


Figura 3.14. Representación del error cometido Problema 8

3.3.8. Problema 9

Una boya cilíndrica de peso de 100 kg, flota parcialmente sumergida en posición recta. Cuando es ligeramente separada de su posición de equilibrio, la boya sube y baja según la siguiente ecuación.

$$y''(x) = \frac{g}{100} [-16\pi y - cy'] \quad x \in [0 \ 12]$$

Con las condiciones iniciales $y(0) = 0.0$, $y'(0) = 1$.

a) Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = y(x) \quad x \in [0 \ 1] \rightarrow z_2' = \frac{g}{100} [-16\pi z_1 - cz_2] \quad x \in [0 \ 12]$$

Con las condiciones iniciales $z_1(0)=0.0$, $z_2(0)=1$.

b) Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

`function f = Probproblema09(x,z)`

`g=9.8; % fuerza de gravedad`

```
c=sqrt(15*pi);
f=[z(2); g/100*(-16*pi*z(1)-c*z(2))];
end
```

c) Resuelva el PVI para el intervalo $x \in [0, 12]$ con $N = 40$ subintervalos y presente $y(x)$.

Para este ejercicio ocuparemos el programa para sistemas de ecuaciones con Euler.

Llamamos a la función:

```
>> [t,y]=Euler_orden1_NM('Problema09',0,12,[0 1],40);
```

Comandos utilizados para graficar la solución aproximada:

```
>> plot(t,y(:,1))
```

```
>> grid on
```

```
>> xlabel('x')
```

```
>> ylabel('y')
```

```
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

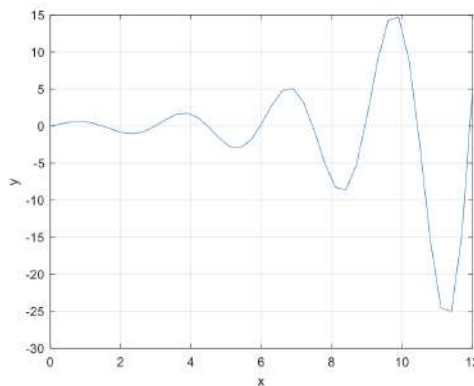


Figura 3.15. Representación de la solución $y(x)$ Problema 9 con Euler

d) Indicar los valores de $y(x)$ para $x \in \{0, 3.9, 6.9, 9.3, 12\}$.

Encontramos las posiciones requeridas:

	$\frac{b-a}{N+1}$ <i>Posición - a</i>	$\frac{N+1}{x}$
<ul style="list-style-type: none"> 0 $\frac{12}{0} \frac{41}{x} = 1 \rightarrow 1$	<ul style="list-style-type: none"> 3.9 $\frac{12}{3.9} \frac{41}{x} = 13.3 \rightarrow 14$	<ul style="list-style-type: none"> 6.9 $\frac{12}{6.9} \frac{41}{x} = 23.5 \rightarrow 24$
<ul style="list-style-type: none"> 9.3 $\frac{12}{9.3} \frac{41}{x} = 31.7 \rightarrow 32$	<ul style="list-style-type: none"> 12 $\frac{12}{12} \frac{41}{x} = 41$	

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(14) t(24) t(32) t(41)]
```

```
>> y=[y(1) y(14) y(24) y(32) y(41)]
```

x(t)	0	3.9	6.9	9.3	12
y	0	1.7403	5.0608	8.7799	4.9149

e) Indicar una estimación del orden de convergencia.

```
>> [t,y1]=Euler_orden1_NM('Problema09',0,12,[0 1],40*1);sol1=y1(:,1);
```

```
>> [t,y2]=Euler_orden1_NM('Problema09',0,12,[0 1],40*2);sol2=y2(:,1);
```

```
>> [t,y3]=Euler_orden1_NM('Problema09',0,12,[0 1],40*4);sol3=y3(:,1);
```

```
>> [t,y4]=Euler_orden1_NM('Problema09',0,12,[0 1],40*8);sol4=y4(:,1);
```

```
>> [t,y5]=Euler_orden1_NM('Problema09',0,12,[0 1],40*16);sol5=y5(:,1);
```

```
>> [t,y6]=Euler_orden1_NM('Problema09',0,12,[0 1],40*32);sol6=y6(:,1);
```

```
>> [t,y7]=Euler_orden1_NM('Problema09',0,12,[0 1],40*64);sol7=y7(:,1);
```

```
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
```

```
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
```

```
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
```

```
>> relacion_N_2N=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4)
max(e4)/max(e5) max(e5)/max(e6)]'
```

```
relacion_N_2N = 41.6528 7.6208 2.7318 2.2604 2.1245
```

```
>> orden=log2(relacion_N_2N)
```

```
orden = 5.3803 2.9299 1.4499 1.1766 1.0871
```

3.3.9. Problema 10

La temperatura $y(x)$ en un anillo circular de radio interior 1 y radio exterior 3 se define por:

$$y''(x) = \frac{-y'}{x} \quad x \in [1 \ 3]$$

Con las condiciones iniciales $y(1) = 1$, $y'(1) = \frac{1}{2 \log(\frac{1}{3})}$.

a) Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = \frac{-y'}{x} \quad x \in [1 \ 3] \rightarrow z_2' = \frac{-z_2}{x} \quad x \in [1 \ 3]$$

Con las condiciones iniciales $z_1(1) = 0.0$, $z_2(1) = \frac{1}{2 \log(\frac{1}{3})}$.

b) Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema10(x,z)
```

```
    f=[z(2); -z(2)./x];
```

```
end
```

Resuelva el PVI para el intervalo $x \in [1,3]$ con $N = 20$ subintervalos y presente $y(x)$.

```
>> [t,y]=Euler_orden1_NM('Problema10',0,12,[0 1],40);
```

Comandos utilizados para graficar la solución aproximada

```
>> plot(t,y(:,1))
```



```
>> grid on
>> xlabel('x')
>> ylabel('y')
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

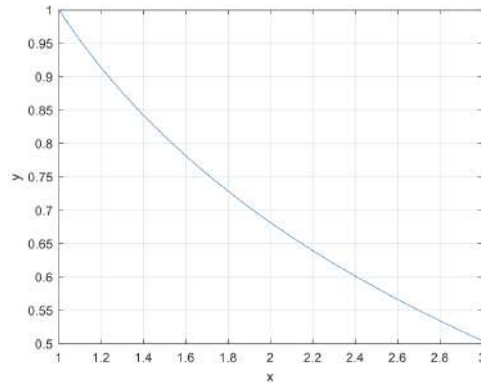


Figura 3.16. Representación de la solución $y(x)$ Problema 10 con Euler

d) Si la solución exacta es $y(x) = \frac{\log(\frac{x}{3}) - \frac{1}{2}\log x}{\log(\frac{1}{3})}$, compare gráficamente la solución con dichas aproximaciones, grafique el error cometido y calcule su error máximo.

```
>> yexacta=(log(t/3)-1/2*log(t))/log(1/3);
>> plot(t,y(:,1)),grid,xlabel('x')
>> hold on;
>> plot(t,yexacta,'red'),legend('y_a_p_r_o_x_i(x)','y_e_x_a_c(x)')
>> title('Solucion Exacta vs Aproximada')
```

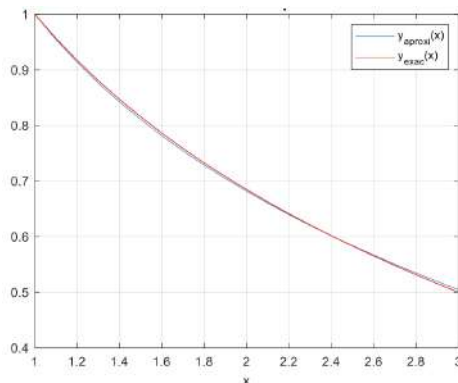


Figura 3.17. Representación de la solución exacta vs aproximada Problema 10

```
>> plot(t,abs(y(:,1)-yexacta)),grid,title('Gráfica del error cometido');
>> Max_error=max(abs(yexacta-y(:,1)))
Max_error = 0.0048
```

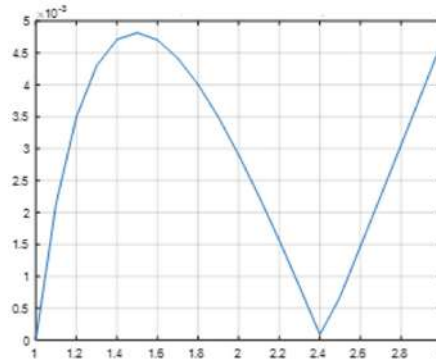


Figura 3.18. Representación del error cometido Problema 10

e) Calcular el error exacto en $x \in \{1.8, 2.4, 2.7\}$.

Encontramos las posiciones requeridas:

$$\frac{b-a}{\text{Posición} - a} = \frac{N+1}{x}$$

<ul style="list-style-type: none"> 1.8 $\frac{2}{0.8} \frac{21}{x} = 8.5 \rightarrow 9$	<ul style="list-style-type: none"> 2.4 $\frac{2}{1.4} \frac{21}{x} = 14.7 \rightarrow 15$	<ul style="list-style-type: none"> 2.7 $\frac{2}{1.7} \frac{21}{x} = 17.8 \rightarrow 18$
--	--	--

```
>> X=[t(9) t(15) t(18)]
```

```
>> Y=[abs(y(9)-yexacta(9)) abs(y(15)-yexacta(15)) abs(y(18)-yexacta(18))]
```

x	1.8	2.4	2.7
Error	0.0040	0.0001	0.0023

e) Indicar una estimación del orden de convergencia.

```
>> [t,y1]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*1);
sol1=y1(:,1);
```

```
>> [t,y2]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*2);
sol2=y2(:,1);
```

```
>> [t,y3]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*4);
sol3=y3(:,1);
```

```
>>[t,y4]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*8);-
sol4=y4(:,1);

>>[t,y5]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*16);-
sol5=y5(:,1);

>>[t,y6]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*32);-
sol6=y6(:,1);

>>[t,y7]=Euler_orden1_NM('Prob_Eq2_16',1,3,[1 1/(2*log(1/3))],20*64);-
sol7=y7(:,1);

>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));

>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));

>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));

>> relacion_N_2N=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4)
max(e4)/max(e5) max(e5)/max(e6)]

relacion_N_2N = 2.1781  2.0528  2.0140  2.0070  2.0035

>> orden=log2(relacion_N_2N)

orden = 1.1231  1.0376  1.0100  1.0051  1.0025
```

3.3.10. Problema 11

Sea el siguiente problema de valor inicial:

$$y''(x) = -2y' + x^2 y \quad x \in [0, 1]$$

Con las condiciones iniciales $y(0) = 1$, $y'(0) = -1$.

a) Transformar el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = -2y' + x^2 y \quad x \in [0, 1] \rightarrow z_2' = -2z_2 + x^2 z_2 \quad x \in [0, 1]$$

Con las condiciones iniciales $z_1(0) = 1$, $z_2(-1) = -1$.

b) Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema11(x,z)
```

```
f=[z(2); -2*z(2)+x.^2.*z(1)];
```

```
end
```

c) Resuelva el PVI para el intervalo $x \in [0,1]$ con pasos $h = 0.05$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.05}$ obteniendo el número de iteraciones $N = 20$.

Para este ejercicio, ocuparemos el programa para sistemas de ecuaciones con Euler.

Llamamos a la función:

```
>>[t,y]=Euler_orden1_NM('Problema11',0,1,[1 -1],20);
```

Comandos utilizados para graficar la solución aproximada:

```
>> plot(t,y(:,1)),grid, xlabel('x'),ylabel('y')
```

```
>> title('METODO EULER PARA SISTEMAS DE ECUACIONES')
```

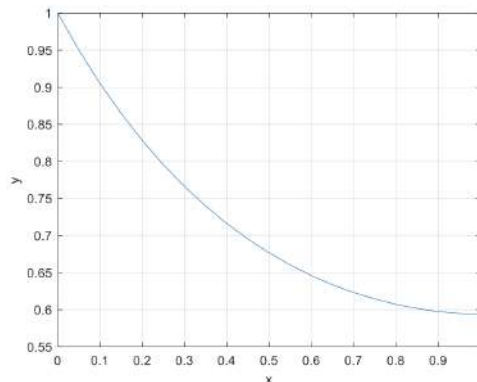


Figura 3.19. Representación de la solución $y(x)$ Problema 11 con Euler
Fuente: Autores

d) Indicar los valores de $y(x)$ para $x = 0.8$.

Encontramos las posiciones requeridas:

$$\frac{b-a}{N+1} = \frac{0.8-0}{17+1} = 0.0476$$

$$\frac{1}{0.8} \cdot 21 = 26.25 \rightarrow 27$$

Ingresamos los códigos en Matlab con la posición encontrada:

```
>> X=t(17)
```

```
X = 0.8000
```

```
>> Y=y(17)
```

```
Y = 0.6071
```

e) Indique una estimación del orden de convergencia.

```
>> [t,y1]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*1);sol1=y1(:,1);
```

```
>> [t,y2]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*2);sol2=y2(:,1);
```

```
>> [t,y3]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*4);sol3=y3(:,1);
```

```
>> [t,y4]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*8);sol4=y4(:,1);
```

```
>> [t,y5]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*16);sol5=y5(:,1);
```

```
>> [t,y6]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*32);sol6=y6(:,1);
```

```
>> [t,y7]=Euler_orden1_NM('Prob_Eq2_17',0,1,[1 -1],20*64);sol7=y7(:,1);
```

```
>> e1=norm(sol1-sol2(1:2:end)); e2=norm(sol2-sol3(1:2:end));
```

```
>> e3=norm(sol3-sol4(1:2:end)); e4=norm(sol4-sol5(1:2:end));
```

```
>> e5=norm(sol5-sol6(1:2:end)); e6=norm(sol6-sol7(1:2:end));
```

```
>> relacion_N_2N=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4)
max(e4)/max(e5) max(e5)/max(e6)]
```

```
relacion_N_2N = 1.4492 1.4314 1.4227 1.4184 1.4163
```

```
>> orden=log2(relacion_N_2N)
```

```
orden = 0.5352 0.5174 0.5087 0.5043 0.5022
```

f) Si la solución exacta es $y(x) = e^{(x-2)\frac{x}{2}}$, comparar gráficamente la solución con dichas aproximaciones, graficar el error cometido y calcular su error máximo.

Código para comprar la solución exacta con la aproximada:

```
>> yexacta=exp((t-2).*t/2);
```

```
>> plot(t,y(:,1)),grid,xlabel('x'), ylabel('y'),
```

```
>> hold on;
```

```
>> plot(t,yexacta,'red'),legend('y_a_p_r_o_x_i(x)','y_e_x_a_c(x)')
```

```
>> title('Solucion Exacta vs Aproximada')
```

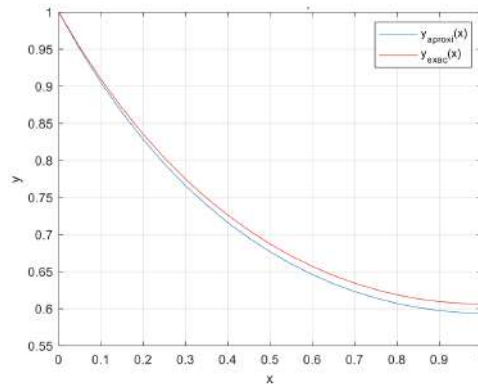


Figura 3.20. Representación de la solución exacta vs. la aproximada Problema 11

Código para calcular su error máximo:

```
>> Max_error=max(abs(yexacta-y(:,1)))
```

```
Max_error = 0,0127
```

Código para graficar el error cometido:

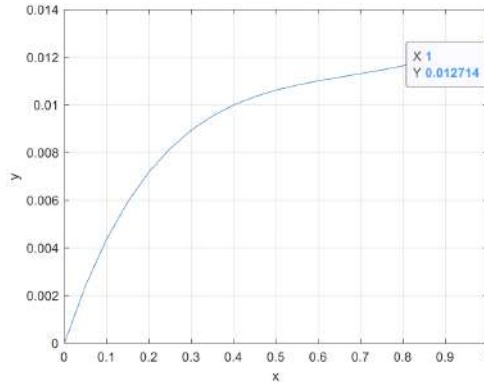


Figura 3.21. Representación del error cometido Problema 11

3.3.11. Problema 12

Un cuerpo que tiene una temperatura de 70°F es depositado (en el tiempo $t=0$) en un lugar donde la temperatura se mantiene a 40°F. Después de 3 min, la temperatura del cuerpo ha disminuido a 60°F.

a) Halle la ecuación que gobierna al sistema.

Si $T(t)$ es la temperatura del cuerpo en °F después de t minutos, entonces la ecuación diferencial que modela $T(t)$ es:

$$T'(t) = k[T(t) - T_a]$$

Donde $T_a = 40^\circ\text{F}$ es la temperatura fija del medio circulante.

b) Si las condiciones iniciales son $T(0) = 70$ y $k = -0.1352$ para este caso. ¿Cuál es la temperatura después de cinco minutos?

$$T'(t) = -0.1352[T(t) - 40]$$

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{5-0}{0.1}$ obteniendo el número de iteraciones $N = 50$.

Escribimos la función que se va a utilizar:

```
function f=Problema14(x,T)
```

```
f=-0.1352*(T-40);
```

end

Llamamos al método y graficamos la solución:

```
>> [x,y]=Euler_orden1_N('Problema14',0,5,[70],50);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

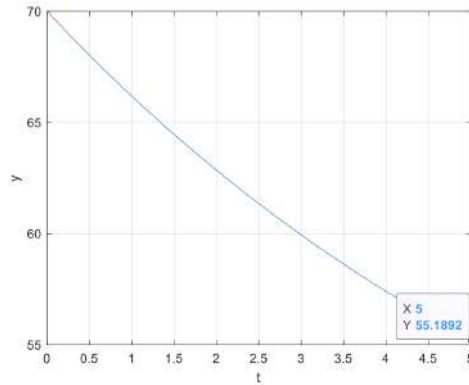


Figura 3.22. Representación de la solución T Problema 12 con Euler

La temperatura después de cinco minutos es de 55,1892°F.

3.3.12. Problema 13

Un paracaidista de masa M kg salta desde un avión en $t = 0$. Consideremos que la velocidad vertical inicial del paracaidista es cero en $t = 0$ y que la caída es vertical. Si el arrastre aerodinámico está dado por $F_{aire} = cv^2$, donde c es una constante y v es la velocidad vertical (positiva hacia abajo), asuma $M = 70$ kg, $c = 0.27$ kg/m y $h = 0.1$. Para la primera ley de Newton, el equilibrio de las fuerzas satisface:

$$M \frac{dv(t)}{dt} = -F_{aire} + gM$$

a) Halle la velocidad del paracaidista para $t \leq 20$ s.

Sabemos que v es la velocidad del paracaidista en m/s (positiva hacia abajo) y g es la gravedad dada 9,8 m/s² obtenido la siguiente ecuación:

$$\frac{dv(t)}{dt} = -\frac{cv^2}{M} + g$$

$$\frac{dv(t)}{dt} = -\frac{0.27v^2}{70} + 9.8$$

Escribimos una función para Matlab:

```
function f=Problema15(x,v)
```

```
f=-0.27/70*v^2+9.8;
```

```
end
```

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{20-0}{0.1}$ obteniendo el número de iteraciones $N = 200$.

Procedemos a llamar al método y graficamos la solución:

```
>> [x,y]=Euler_orden1_N('Problema15',0,20,[0],200);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

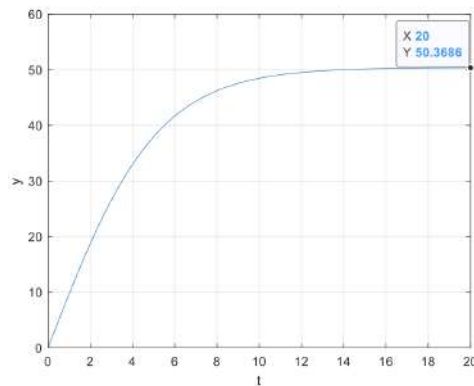


Figura 3.23. Representación de la solución Problema 13 con Euler

La velocidad del paracaidista para $t \leq 20s$ es de 50.36 m/s .

3.3.13. Problema 14

La velocidad de un cuerpo en caída se modeló mediante el problema de valor inicial:

$$m \frac{dv(t)}{dt} = mg - kv \quad v(0) = v_o$$

De acuerdo con la hipótesis de que la fuerza debida a la resistencia del aire es de $-kv$. Sin embargo, en ciertos casos, la fuerza originada por la resistencia del aire se comporta más como $-kv^r$ donde $r > 1$ dado por el modelo:

$$m \frac{dv(t)}{dt} = mg - kv^r \quad v(0) = v_o$$

a) Para estudiar el efecto de cambiar el parámetro de r , se debe considerar $m = 1$, $g = 9.81$, $k = 2$ y $v_o = 0$. Utilice $h = 0.2$ para aproximar la solución de la ecuación en el intervalo $0 \leq t \leq 5$ para $r = 1, 1.5$ y 2 .

Partiendo de la ecuación

$$\frac{dv(t)}{dt} = g - \frac{kv^r}{m} \quad v(0) = v_o$$

$$\frac{dv(t)}{dt} = 9.81 - \frac{2v^r}{1} \quad v(0) = v_o$$

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{5-0}{0.2}$ obteniendo el número de iteraciones $N = 25$.

Creamos la función para $r=1.5$

`function f=Problema17e(x,v)`

`f=-9.81-2*v;`

`end`

Llamamos al método y graficamos la solución:

`>> [x,y]=Euler_orden1_N('Problema17e',0,5,[0],25);`

`>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')`

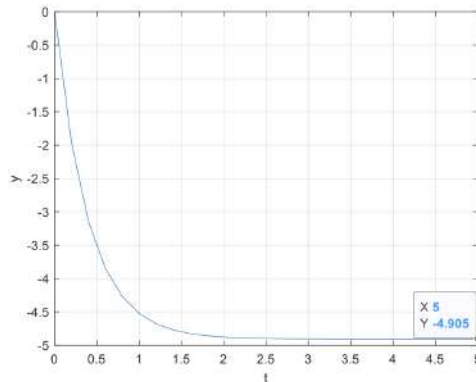


Figura 3.24. Representación de la solución Problema 14 para $r = 1$

Modificamos la función para $r = 1$:

```
function f=Problema17e(x,v)
```

```
    f=-9.81-2*v;
```

```
end
```

Llamamos a la función:

```
>> [x,y]=Euler_orden1_N('Problema17e',0,5,[0],25);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

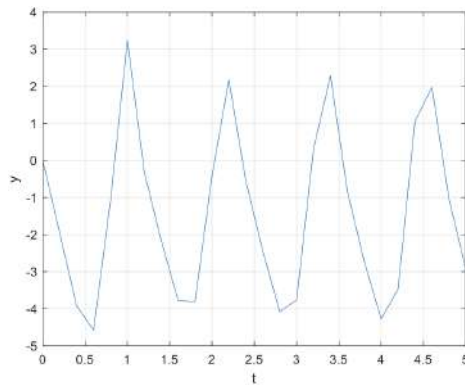


Figura 3.25. Representación de la solución Problema 14 para $r = 1.5$

Modificamos la función para $r = 2$:

```
function f=Problema17e(x,v)
```

```
    f=-9.81-2*v.^2;
```

```
end
```

Llamamos al método y graficamos la solución:

```
>> [x,y]=Euler_orden1_N('Problema17e',0,5,[0],25);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

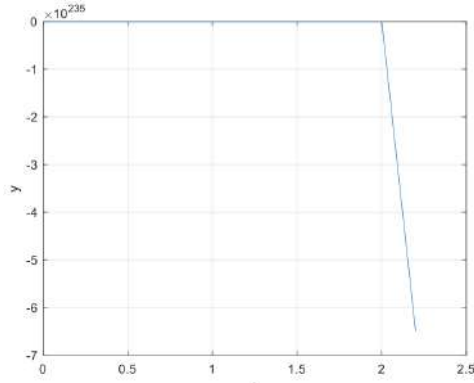


Figura 3.26. Representación de la solución Problema 14 para $r = 1.5$

3.3.14. Problema 15

En un circuito de voltaje impreso E que tiene la resistencia R , la inductancia L y la capacitancia C en paralelo, la corriente i satisface la ecuación diferencial.

$$\frac{di}{dt} = C \frac{d^2 E}{dt^2} + \frac{1}{R} \frac{dE}{dt} + \frac{1}{L} E$$

Supongamos que $C = 0.3$ faradios, $R = 1.4$ ohmios, $L = 1.7$ henrios y que el voltaje esta dado por $E(t) = e^{-0.06t} \text{sen}(2t - \pi)$.

a) Si $i(0)=0$, calcule la corriente i con los valores $h=0.1$, y grafique la solución.

$$\frac{dE}{dt} = -2 \cos(2t) e^{-0.06t} + 0.06 e^{-0.06t} \text{sen}(2t - \pi) e^{-0.06t}$$

$$\frac{d^2 E}{dt^2} = 0.24 \cos(2t) e^{-0.06t} + 3.9964 \text{sen}(2t) e^{-0.06t}$$

Remplazamos y encontramos la ecuación que gobierna el sistema:

$$\frac{di}{dt} = -1.35657 \cos(2t) e^{-0.06t} + 0.65354 \text{sen}(2t) e^{-0.06t}$$

Creamos la función en Matlab:

```
function f=Problema15(t,i)
```

```
    f=-1.35657.*cos(2*t).*exp(-0.06*t)+0.65354.*sin(2*t).*exp(-0.06*t);
```

```
end
```

Llamamos al método para graficar la solución:

```
>> [x,y]=Euler_orden1_N('Problema15',0,10,[0],100);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

```
>> [x,y]=Euler_orden1_N('Problema15',0,10,[0],100)
```

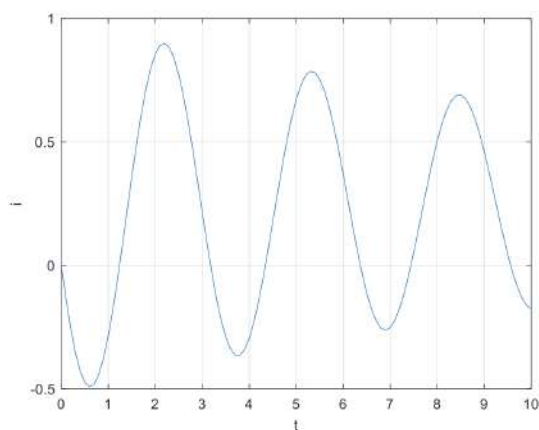


Figura 3.27. Representación de la solución Problema 15

CAPÍTULO IV

4. MÉTODO DE HEUN

Si, en el método de Euler, desarrollamos por Taylor la función $f(t)$ hasta los términos de orden 2, en este método, se tomará un método más.

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \mathcal{O}(h^2)$$

Donde $\mathcal{O}(h^3)$ son los términos de orden mayor o igual a 3 teniendo en cuenta que $y'(t) = f(t, y(t))$, y que $y''(t) = \frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} y'(t)$, el desarrollo de Taylor queda:

$$\begin{aligned} y(t, h) &= y(t) + hf(t, y) + \frac{h^2}{2} \left(\frac{\partial f(t, y(t))}{\partial t} + \frac{\partial f(t, y(t))}{\partial y} f(t, y) \right) + \mathcal{O}(h^3) = \\ &= y(t) + \frac{h}{2} f(t, y) + \frac{h}{2} \left(f(t, y) + h \frac{\partial f(t, y)}{\partial t} + hf(t, y) \frac{\partial f(t, y)}{\partial y} \right) + \mathcal{O}(h^3) \end{aligned}$$

Consideremos el desarrollo de Taylor para dos variables:

$$f(t+h, y+k) = f(t, y) + h \frac{\partial f(t, y)}{\partial t} + k \frac{\partial f(t, y)}{\partial y} + \mathcal{O}(h^2)$$

Reemplazando k por $hf(t, y)$:

$$f(t+h, y+hf(t, y)) = f(t, y) + h \frac{\partial f(t, y)}{\partial t} + hf(t, y) \frac{\partial f(t, y)}{\partial y} + \mathcal{O}(h^2)$$

Sustituyendo en la expresión de $y(t+h)$, se obtiene:

$$y(t+h) = y(t) + \frac{h}{2} f(t, y) + \frac{h}{2} f(t+h, y+hf(t, y)) + \mathcal{O}(h^3)$$

Si $t_k + h = t_{k+1}$ ey $(t_k + h) = t_{k+1}$, entonces :

$$y_{k+1} = y_k + \frac{1}{2} hf(t_k, y_k) + \frac{1}{2} hf(t_{k+1}, y_k + hf(t_k, y_k)) = y_k + \frac{1}{2} k_1 + \frac{1}{2} k_2$$

Donde $k_1 = hf(t_k, y_k)$ y $k_2 = hf(t_{k+1}, y_k + k_1)$.

Teorema 3. Sea f tal que $y'(t) = f(t, y(t))$ con condiciones iniciales $y(t_a) = y_a$. Si $y(t) \in \mathcal{C}^3$ $[a, b]$ y $\{(t_k, y_k)\}_{k \geq 0}$ es la sucesión de aproximaciones dadas por el método de Heun, entonces:

$$|e_k| \leq |y(t_k) - y_k| = \left| y(t_k) - y_{k-1} - \frac{1}{2}hf(t_{k-1}, y_{k-1}) - \frac{1}{2}hf(t_{k-1}, y_{k-1} + hf(t_k, y_k)) \right|$$

$$= \mathcal{O}(h^3)$$

Satisface $E(h) = \frac{1}{h_{1 \leq k \leq N}} \max |e_k| = \mathcal{O}(h^2)$

4.1. ALGORITMO DE HEUN

Para la construcción del algoritmo de Euler consideramos

Entradas

a, b extremos del intervalo

y_1 condicion inicial

N número de subintervalos

- Proceso

Cálculo del paso de integración, h y obtención x

Inicialización del vector solución y en a

Para k desde 1 hasta N

$$k_1 = hf(x_k, y_k)$$

$$k_2 = hf(x_{k+1}, y_k + k_1)$$

$$y_{k+1} = y_k + \frac{k_1 + k_2}{2}$$

Fin par k

- Salida x, y

Construimos el código en Matlab implementando el método de Heun como se muestra a continuación:

```
function [x,y] = Heun2_Ecu(fun,a,b,N,y0)

h=(b-a)/N;%paso

x=a:h:b;%vector de nodos x

x=x(:);% vector en forma de columna;

y=zeros(N+1,length(y0));% cuántas columnas va a tener la matriz

y(1,:)=y0;% almacenamos el vector la primera componente

for k=1:N % contador

    K1=h*feval(fun,x(k),y(k));% para evaluar funciones

    K2=h*feval(fun,x(k+1),y(k)+K1);

    y(k+1)=y(k)+(K1+K2)/2;% adaptamos a matrices

end
```

4.2. ALGORITMO DE HEUN PARA SISTEMAS DE ECUACIONES

Para la construcción del algoritmo de Euler consideramos

Entradas

a, b extremos del intervalo

y_1 condicion inicial

N número de subintervalos

- Proceso

Cálculo del paso de integración, h y obtención x

Inicialización del vector solución y en a

Para k desde 1 hasta N

$$k_1 = hf(x_k, y_k)$$

$$k_2 = hf(x_{k+1}, y_k + k_1)$$

$$y_{k+1} = y_k + \frac{k_1 + k_2}{2}$$

Fin par k

- Salida x, y

Construimos el código en Matlab implementando el método de Heun para sistemas de ecuaciones:

```
function [x,y] = Heun2_Sis(fun,a,b,N,y0)
h=(b-a)/N;% paso
x=a:h:b;% vector de nodos x
x=x(:);% vector en forma de columna;
y=zeros(N+1,length(y0));% length cuántas columnas va a tener
y(1,:)=y0;% almacenamos el vector la primera componente
for k=1:N % contador
    K1=h*feval(fun,x(k),y(k,:));% para evaluar funciones
    K2=h*feval(fun,x(k+1),y(k,:)+K1);
    y(k+1,:)=y(k,:)+(K1+K2)/2;% adaptamos a matrices
end
```

4.3. PROBLEMAS PVI CON HEUN

4.3.1. Problema 1

Un proyectil de masa $m = 0.11$ kg es lanzado verticalmente hacia arriba con una velocidad inicial $V(0) = 8$ m/s.

El proyectil disminuye su velocidad por efecto de la fuerza de gravedad $F_g = -mg$ y por la resistencia del aire $F_r = kv|v|$ donde $g = 9.8$ m/s² y $k = 0.002$ kg/m.

La ecuación diferencial de la velocidad está dada por:

$$m \frac{\delta v}{\delta t} = -mg - kv|v|$$

a) Escriba una función que describa la ecuación diferencial de primer orden.

$$m \frac{\delta v}{\delta t} = -mg - kv|v|$$

Se despeja:

$$\frac{\delta v}{\delta t} = -g - \frac{k}{m} v|v|$$

Estandarizamos la ecuación:

$$v' = -g - \frac{k}{m} v|v|$$

$$v' = -9.8 - \frac{0.002}{0.11} v|v|$$

`function f=Problema01(x,v)`

`f=-9.8-(0.002/0.11)*v.^2;`

`end`

b) Resuelva el PVI para el intervalo $t \in [0 \ 1]$ con subintervalo $h = 0,2$ y representar $y(t)$

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.2}$ obteniendo el número de iteraciones $N = 5$.

Llamamos al método Heun para ecuaciones:

`>> [x,y]= Heun2_Ecu ('Problema01',0,1,8,5);`

`>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')`

`>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')`

`>> legend('v(t)')`

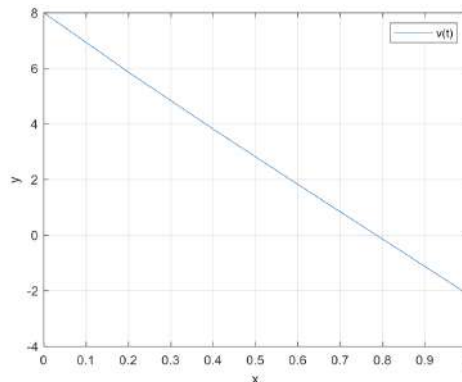


Figura 4.1. Representación de la solución $v(t)$ Problema 1 con Heun

4.3.2. Problema 2

Sea el siguiente problema de valor inicial:

$$y'(t) = (1 - 2t)y(t) \quad t \in [0 \ 3]$$

Con las condiciones iniciales $y(0) = 1$.

a) Escriba una función que describa la ecuación diferencial de primer orden.

```
function dy= Problema02(t,y)
```

```
dy=(1-2*t)*y;
```

```
end
```

b) Resuelva el PVI para el intervalo $t \in [0 \ 3]$ con subintervalo $N = 4$ y represente $y(t)$

Llamamos al método:

```
>>[x,y]= Heun2_Ecu ('Problema01',0,3,1,4)
```

Graficamos la solución:

```
>> plot(x,y(:,1)),grid on,xlabel('x'), ylabel('y')
```

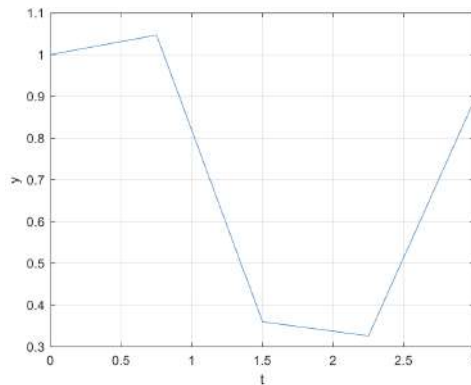


Figura 4.2. Representación de la solución $y(t)$ Problema 2 con Heun

c) Indicar los valores de $y(t)$ para $t \in \{0 \ 1.5 \ 2.25 \}$.

Encontramos las posiciones requeridas:

$$\text{Posición} - a = \frac{b - a}{N + 1} x$$

• 0	• 1,5	• 2,25
$\begin{matrix} 3 & 4 \\ 0 & x \end{matrix} = 1$	$\begin{matrix} 3 & 4 \\ 1,5 & x \end{matrix} = 2$	$\begin{matrix} 3 & 4 \\ 2,25 & x \end{matrix} = 3$

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(2) t(3) ]
```

```
>> y=[y(1) y(2) y(3) ]
```

x(t)	0	0.75	2.25
y	1	1.0469	0.3261

d) Indicar una estimación del orden de convergencia.

```
>> [x,y1]=Heun2_Ecu('Problema1',0,3,1,4*1);sol1=y1(:,1);
```

```
>> [x,y2]=Heun2_Ecu('Problema1',0,3,1,4*2);sol2=y2(:,1);
```

```
>> [x,y3]=Heun2_Ecu('Problema1',0,3,1,4*4);sol3=y3(:,1);
```

```
>> [x,y4]=Heun2_Ecu('Problema1',0,3,1,4*8);sol4=y4(:,1);
```

```
>> [x,y5]=Heun2_Ecu('Problema1',0,3,1,4*16);sol5=y5(:,1);
```

```
>> [x,y6]=Heun2_Ecu('Problema1',0,3,1,4*32);sol6=y6(:,1);
```

```
>> [x,y7]=Heun2_Ecu('Problema1',0,3,1,4*64);sol7=y7(:,1);
```

```
>> e1=norm(sol1-sol2(1:2:end)); e2=norm(sol2-sol3(1:2:end));
```

```
>> e3=norm(sol3-sol4(1:2:end)); e4=norm(sol4-sol5(1:2:end));
```

```
>> e5=norm(sol5-sol6(1:2:end)); e6=norm(sol6-sol7(1:2:end));
```

```
>> relacion_N_2N=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4)
max(e4)/max(e5) max(e5)/max(e6)]
```

```
relacion_N_2N =
```

```
8.9742 5.3577 3.4854 3.0651 2.9242
```

```
>> orden=log2(relacion_N_2N)
```

```
orden =
```

```
3.1658 2.4216 1.8013 1.6160 1.5480
```

4.3.3. Problema 3

Consideremos el problema de valor inicial:

$$y'(t) = \cos(2t) + \sin(3t) \quad t \in [0, 1]$$

Con las condiciones iniciales $y(0) = 1$, cuya solución exacta es $y(t) = \frac{1}{2}\sin(2t) - \frac{1}{3}\cos(3t) + \frac{4}{3}$.

a) Escriba una función que describa la ecuación diferencial de primer orden.

```
function dy= Problema03(t,y)
```

```
dy=cos(2*t)+sin(3*t);
```

```
end
```

b) Resuelva el PVI para el intervalo $t \in [0,1]$ con pasos $h = 0,1$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.1}$ obteniendo el número de iteraciones $N = 10$.

Llamamos al método y graficamos la solución:

```
>> [x,y]= Heun2_Ecu('Problema02',0,1,1,10);
```

```
>> plot(x,y(:,1)), xlabel('x'),ylabel('y'), grid on
```

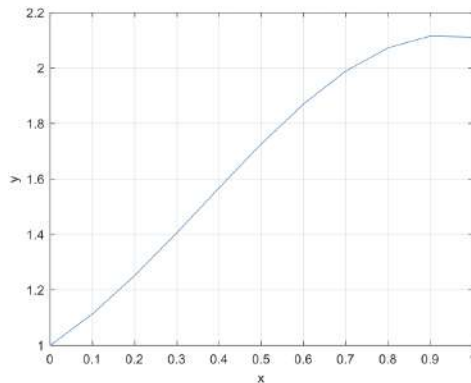


Figura 4.3. Representación de la solución $y(t)$ Problema 2 con Heun

c) Resuelva el PVI para el intervalo $t \in [0,1]$ con pasos $h = 0.1$ y represente $y(x)$ y su campo de direcciones.

```
>> t=1:1:1;
>> y=t;
>> yp=cos(2*T)+sin(3*T);
>> quiver(T,Y,ones(size(T)),yp)
>> hold on
>> [x,y]= Heun2_Ecu('Problema03',0,1,1,10);
>> plot(x,y(:,1)), label('x'),ylabel('y')
```

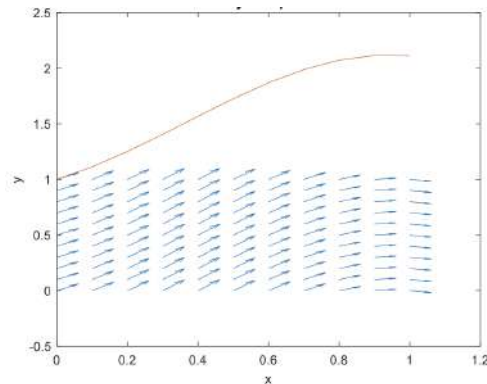


Figura 4.4. Representación de la solución $y(t)$ Problema 3 con Heun y campo de direcciones

d) Compare gráficamente la solución exacta con la solución aproximada, grafique el error cometido y calcule su error máximo.

```
>> [x,y]=Heun2_Ecu('Problema02',0,1,1,10);
>> plot(x,y(:,1)),grid on, xlabel('x'),ylabel('x')
>> yexacta=(1./2.*sin(2.*x)-1./3.*cos(3.*x)+4./3);
>> hold on
>> plot(x,yexacta,'red')
>> legend('y_a_p_r_o_x_i(x)','y_e_x_a_c(x)')
>> title('Solucion Aproximada vs Exacta')
```

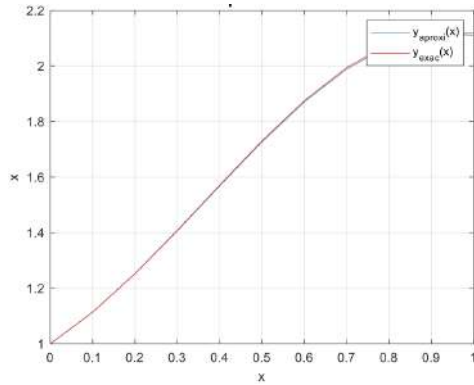


Figura 4.5. Representación de la solución exacta vs. la aproximada Problema 3

Calculamos su error máximo:

```
>> Max_error=max(abs(yexacta-y(:,1)))
```

Max_error =

0.0065

```
>> plot(x,abs(y(:,1)-yexacta)),grid,title('Gráfica del error cometido');
```

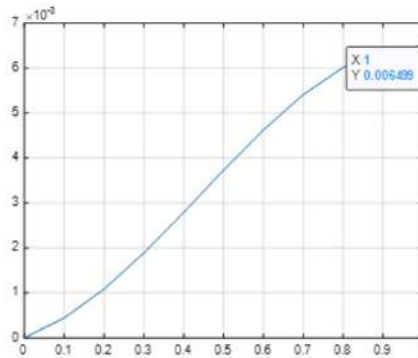


Figura 4.6. Representación del error cometido Problema 3

4.3.4. Problema 4

Consideremos el problema de valor inicial $\lambda = 50$:

$$y'(x) = -2\pi \sin(2\pi x) - \frac{1}{\epsilon}(y - \cos(2\pi x)) \quad x \in [0, 10]$$

Con las condiciones iniciales $y(0) = 1$, el problema se hace rígido cuando $\epsilon \rightarrow 0$. Su solución exacta es $y(t) = \cos(2\pi t)$.

a) Construya el método de Heun implícito.

```
function [x,y,maxiter] = Heun_Imp(f,a,b,N,y0,tol,maxiter)
```

```
%metodo Heun Implicito
```

```
h=(b-a)/N; x=a:h:b; x=x(:);
```

```
y=zeros(N+1,length(y0));
```

```
y(1)=y0;
```

```
for k=1:N
```

```
    yn0=y(k);
```

```
    [fk,dfk]=feval(f,x(k),y(k));
```

```
    incre=tol+1;
```

```
    iter=0;
```

```
    while incre>tol && iter<maxiter
```

```
        [fun,dfun]=feval(f,x(k+1),yn0);
```

```
        yn=yn0-(yn0-y(k)-h/2*(fk+fun))/(1-dfun*h/2);
```

```
        incre=abs(yn-yn0);
```

```
        yn0=yn;
```

```
        iter=iter+1;
```

```
    end
```

```
    y(k+1)=yn;
```

```
end
```

b) Escriba una función que describa la ecuación diferencial de primer orden.

Utilice $\epsilon=10^{-6}$

```
function [fun,dfun] = Problema04(x,y)
```

```
    eps=1.e-6;
```



```
fun=-2*pi*sin(2*pi*x)-1/eps*(y-cos(2*pi*x));  
dfun=-1/eps;
```

```
end
```

c) Resuelva el PVI para el intervalo $x \in [0,10]$ con pasos $N = 2$, una tolerancia e^{-6} y represente $y(x)$

```
>> [x,y,maxiter] = Heun_Imp('Problema04',0,10,20,1,1e-6,20)  
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')  
>> title('METODO HEUN IMPLICITO PARA ECUACIONES')
```

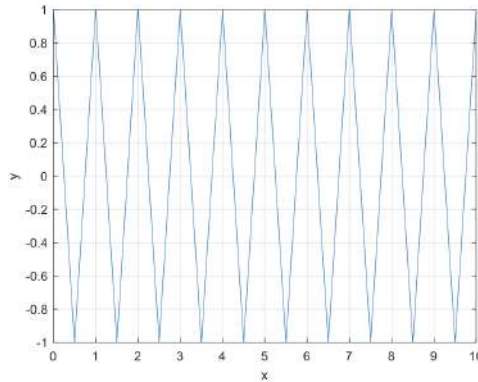


Figura 4.7. Representación de la solución $y(t)$ “Problema 4” con HEUN IMPLICITO

4.3.5. Problema 5

La velocidad de desintegración del radio es proporcional a la cantidad, siendo esta constante de proporcionalidad $k = 4,1 \cdot 10^{-4}$.

a) Plantee el problema de valor inicial de manera que podamos estudiar la variación de la cantidad de radio existente en una muestra de 10 g.

$$y'(t) = -ky(t) \quad y(0) = 10$$

De manera inmediata que se pueda encontrar la solución exacta:

$$y(t) = 10e^{-kt}$$

Que podríamos comparar con la solución aproximada.

b) Resuelva el método y represente $y(t)$ utilizando Heun con $n = 10$, la cantidad de radio que quedaría en la muestra al cabo de 1500 años.

Plateamos la función en Matlab:

```
function dy= Problema05(x,y)
```

```
    k=4.1*10^-4;
```

```
    dy=-k*y;
```

```
end
```

Llamamos al método:

```
>> [x,y]=Heun2_Ecu('Problema05',0,1500,10,10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

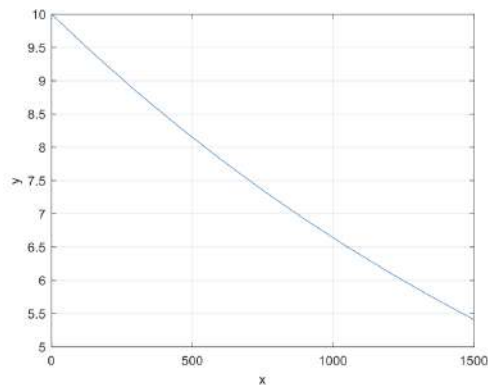


Figura 4.8. Representación de la solución $y(t)$ Problema 5 con Heun

c) Compare gráficamente la solución con dichas aproximaciones, grafique el error cometido y calcule su error máximo.

Ingresamos el código para graficar la solución exacta vs. la aproximada

```
>> [x,y]= Heun2_Ecu('Problema05',0,1500,10,10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> hold on
```

```
>> k=4.1*10^-4;
```

```
>> yexacta=10.*exp(-k.*x);
>> plot(x,yexacta,'red')
>> legend('y_a_p_r_o_x_i(x)','y_e_x_a_c(x)')
>> title('Solucion Aproximada vs Exacta')
```

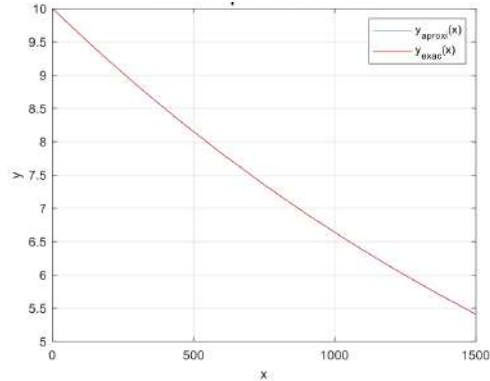


Figura 4.9. Representación de la solución exacta vs. aproximada Problema 5

El error máximo cometido:

```
>> Max_error=max(abs(yexacta-y(:,1)))
```

Max_error = 0.0022

Código para encontrar la gráfica del error cometido:

```
>> plot(x,abs(y(:,1)-yexacta)),grid,title('Gráfica del error cometido');
```

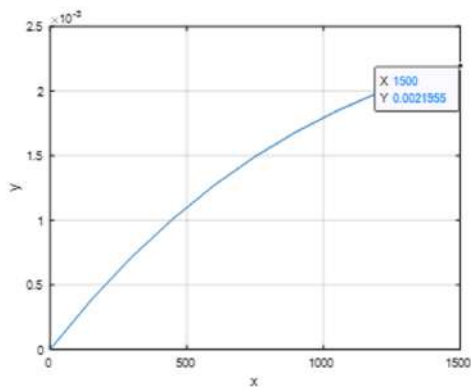
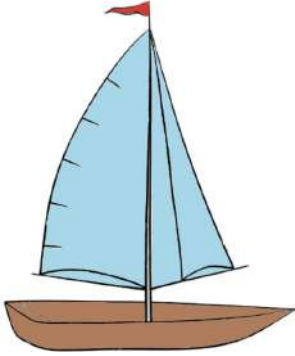


Figura 4.10. Representación del error cometido Problema 5

4.3.6. Problema 6

La ecuación siguiente se utiliza para modelar la deflexión del mástil de un bote expuesto a la fuerza del viento:

$$\frac{\delta^2 y}{\delta x^2} = \frac{f}{2EI} (L - x)^2$$



Donde:

f = fuerza

E = módulo de elasticidad

L = longitud del mástil

I = momento de inercia

Calcule la deflexión si $y = 0$ y $\frac{\delta y}{\delta x} = 0$ en $x = 0$

Para sus cálculos, considere $f=60$, $L=30$, $E=1.25 \times 10^8$ e $I=0.05$.

a) Encuentre el sistema de ecuaciones diferenciales equivalente a dicha ecuación.

Estandarizamos la ecuación:

$$y'' = \frac{f}{2EI} (L - x)^2 \quad x = 0, y = 0, y' = 0$$

Transformamos a un sistema de primer orden:

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = \frac{f}{2EI} (L - x)^2 \rightarrow z_2' = \frac{60}{2 \times 1.25 \times 10^8 \times 0.05} (30 - x)^2$$

$$z_2' = 4.8 \times 10^{-6} (30 - x)^2$$

b) Escriba una función que describa la ecuación diferencial de primer orden.

`function f=Problema06(x,z)`

`f=[z(2);0.0000048*(30-x).^2];`

`end`

c) Resuelva el método y represente $y(x)$ utilizando Heun con $h = 2$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{30-0}{2}$ obteniendo el número de iteraciones $N = 15$.

Llamamos al método y graficamos:

```
>> [t,y]= Heun2_Sis('Problema06',0,30,[0 0],15);
```

```
>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> legend('y(x)')
```

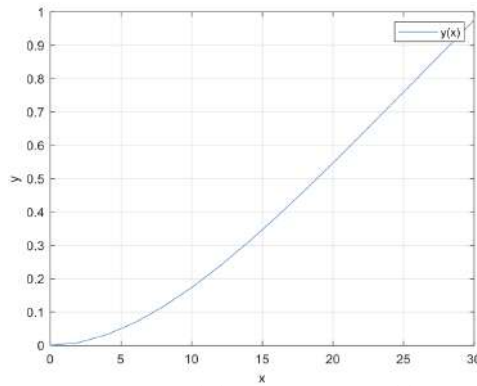
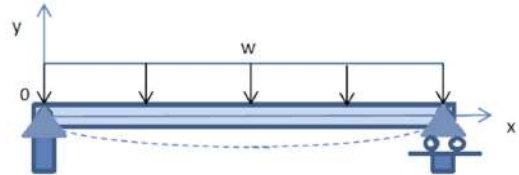


Figura 4.11. Representación de la solución $y(x)$ Problema 6 con Heun

4.3.7. Problema 7

La ecuación diferencial básica de la curva elástica para una viga con carga uniforme (ver figura) está dada por

$$EI \frac{\delta^2 y}{\delta x^2} = \frac{wLx}{2} - \frac{wx^2}{2}.$$



Donde E es el módulo de elasticidad, I momento de inercia.

Aplique los siguientes parámetros:

$$E = 30000 \text{ksi}, I = 800 \text{in}^2, w = 1 \text{ kip/in}, L = 10 \text{ft}$$

a) Encuentre el sistema de ecuaciones diferenciales equivalente a dicha ecuación.

$$y'' = \frac{EI(wLx - wx^2)}{2} \quad x = 0, y = 0, y' = 0$$

Transformamos a un sistema de primer orden:

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2'$$

$$y''(x) = \frac{EI(wLx - wx^2)}{2} \rightarrow z_2' = \frac{24\,000\,000(10x - x^2)}{2}$$

b) Escriba una función que describa la ecuación diferencial de primer orden.

```
function f=Problema07(x,z)
```

```
f=[z(2);(24000000*(10*x-x^2))/2];
```

```
end
```

Resuelva el método y represente $y(x)$ utilizando Heun con $h = 2$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{30-0}{2}$ obteniendo el número de iteraciones $N = 15$.

```
>> [t,y]=Heun2_Sis('Problema07',0,10,[0 0],15);
```

```
>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> legend('y(x)')
```

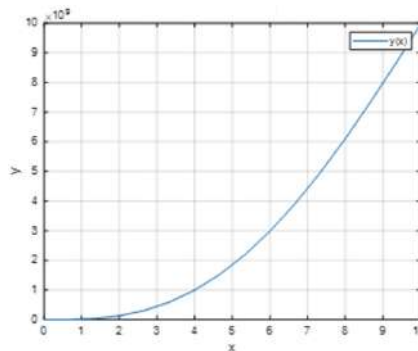
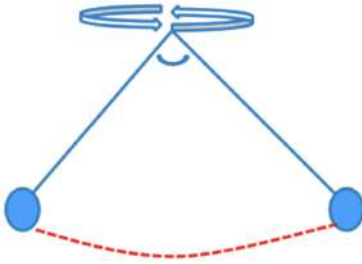


Figura 4.12. Representación de la solución $y(x)$ Problema 7 con Heun

4.3.8. Problema 8

Un péndulo de Foucault proporciona la demostración empírica de la rotación terrestre: el plano de la oscilación del péndulo gira con el tiempo de manera que, tras un cierto período que depende de la latitud del observador, vuelve a su posición original.



Las ecuaciones de movimiento son:

$$x'' - 2\omega \sin(\phi) y' + k^2 x = 0$$

$$y'' + 2\omega \sin(\phi) x' + k^2 y = 0$$

Donde ω es la velocidad angular de la rotación terrestre, $\phi = 40^\circ$ es la latitud del observador y la constante k depende de la aceleración de la gravedad g y la longitud del péndulo $l = 10$ de manera que $k^2 = \frac{g}{l}$. Para iniciar el movimiento, se sitúa el péndulo en la posición $x = -6$, $y = 1$ y se deja en movimiento libre, sin impulso inicial.

a) Transformar el sistema de ecuaciones de segundo orden en un sistema de ecuaciones diferenciales de primer orden.

$$x = z_1, \quad x' = z_1' = z_2, \quad x'' = z_1'' = z_2'$$

$$y = z_3, \quad y' = z_3' = z_4, \quad y'' = z_3'' = z_4'$$

Armamos el sistema de ecuaciones de primer orden:

$$z_1' = z_2,$$

$$z_2' = 2\omega \sin(\phi) z_4 + k^2 z_1$$

$$z_3' = z_4$$

$$z_4' = -2\omega \sin(\phi) z_2 + k^2 z_3$$

b) Calcule la posición del péndulo en el instante $t = 3600s$ con un paso temporal de 0,5 segundos y represente en una misma gráfica las variables $x(t)$ y $y(t)$.

Creamos la función:

```
function w= Problema08(t,z)
```

```
fi=40*pi/180;%latitud en radianes
```

```
om=7.29e-5;%velocidad angular terrestre (2*pi)/(24*3600)
```

```
k2=9.8/10;
```

```
w=[z(2);2*om*sin(fi)*z(4)-k2*z(1);z(4);-2*om*sin(fi)*z(2)-k2*z(3)]
```

```
end
```

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{3600-0}{0.5}$ obteniendo el número de iteraciones $N = 7200$.

Calculamos la posición del péndulo:

```
>> [t,y]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200);
```

```
>> sol1=y(end,:)
```

```
sol1 =
```

```
1.0e+24 *
```

```
1.0456 -0.1662 -0.5151 0.0612
```

Representación de la gráfica:

```
>> [t,y]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200);
```

```
>> plot(t,y(:,1),t,y(:,3)),grid on,xlabel('x'),ylabel('y')
```

```
>> plot(t,y(:,1),t,y(:,3)),grid on,xlabel('x'),ylabel('y')
```

```
>> legend('x(t)','y(t)')
```

```
>> title('grafica u(t),v(t) con HEUN');
```

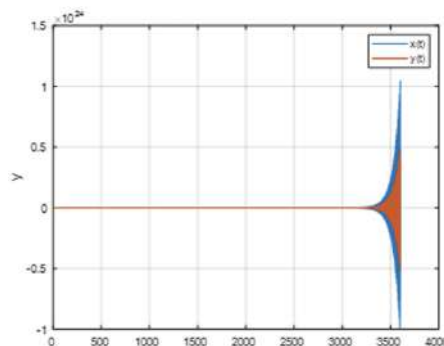


Figura 4.13. Representación de la solución de $x(t)$ y $y(t)$ con Heun

c) Observe el giro del plano de oscilación.

Para que se visualice el giro del plano de oscilación, representamos la evolución del péndulo en el plano YX.

```
>> [t,y]= Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200);
>> plot(y(:,1),y(:,3))
>> title('GIRO DEL PLANO DE OSCILACION');
```

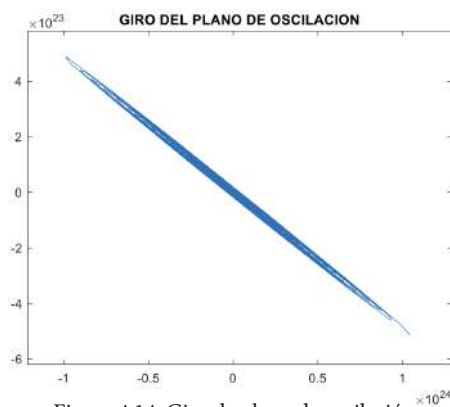


Figura 4.14. Giro de plano de oscilación

Represente las curvas $x(t)$ y $x'(t)$.

```
>> [t,y]= Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200);
>> plot(t(1:120),y(1:120,1:2))
>> title('Representación de las curvas con Heun ');
>> legend('x(t)','x'(t)'), grid on
```

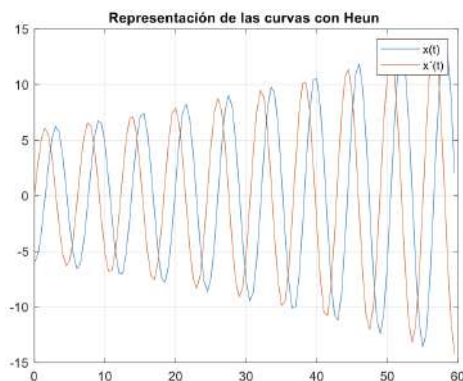


Figura 4.15. Representación de las curvas $x(t)$ y $x'(t)$

e) ¿Qué ocurre si el paso temporal es de un segundo? ¿Y si es de 0,1s?

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{3600-0}{1}$ obteniendo el número de iteraciones $N = 3600$.

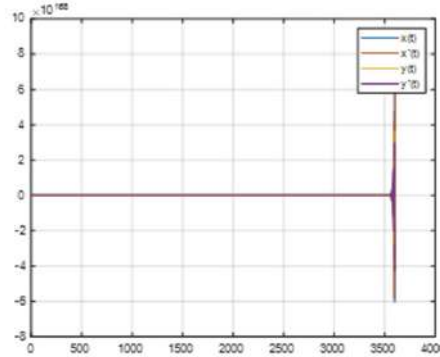


Figura 4.16. Gráfica con paso temporal 1s con Heun

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{3600-0}{0.1}$ obteniendo el número de iteraciones $N = 36\ 000$.

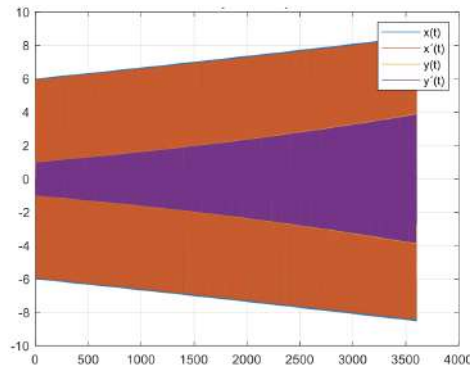


Figura 4.17. Gráfica con paso temporal 0,1s con Heun

f) Indicar una estimación del orden de convergencia.

```
>> [t,y1]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*1);sol1=y1(:,1);
```

```
>> [t,y2]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*2);sol2=y2(:,1);
```

```
>> [t,y3]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*4);sol3=y3(:,1);
```

```
>> [t,y4]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*8);sol4=y4(:,1);
```

```
>> [t,y5]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*16);sol5=y5(:,1);
>> [t,y6]= Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*32);sol6=y6(:,1);
>> [t,y7]=Heun2_Sis('Problema08',0,3600,[-6 0 1 0],7200*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ];
>> orden=log2(relacion)

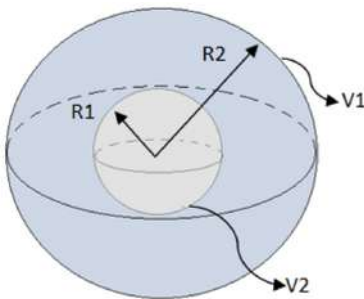
orden =

    67.6041    8.2901    0.7507    1.9073    2.0001
```

Alcanza al orden 2.

4.3.9. Problema 9

Sea $u(r)$ el potencial electrostático entre dos esferas metálicas concéntricas de radio $R1$ y $R2$, tales que el potencial en la esfera interior se mantiene constante a potencial $V1$ y el potencial en la superficie de la esfera exterior es $V2$.



El potencial en ambas esferas se describe por:

$$u''(r) + \frac{2}{r} \times u' \quad r \in [2 \quad 2R]$$

Con las condiciones iniciales: $u(R1) = V1$, $u'(R1) = -V1$ supongamos que $R = 2$, $V1 = 1$ voltios y $V2 = 0$ voltios.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$u = z_1$$

$$u' = z_1' = z_2$$

$$u'' = z_1'' = z_2'$$

$$u''(x) = -\frac{2}{r} \times u' \quad r \in [2 \ 2R] \rightarrow z'_2 = -\frac{2}{r} \times z_2 \quad r \in [2 \ 2R]$$

b) Escriba una función que describa la ecuación diferencial de segundo orden como un sistema de ecuaciones diferencial de primer orden.

```
function f = Problema09(r,z)
```

```
f=[z(2); -2./r.*z(2)];
```

```
end
```

c) Resuelva el PVI para el intervalo $r \in [2 \ 2R]$ con pasos $h = 0.05$ y represente $u(r)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{4-2}{0.05}$ obteniendo el número de iteraciones $N = 40$.

Condiciones iniciales $u_1(2) = 1$ y $u_1'(2) = -1$.

Procedemos a llamar al método para graficar:

```
>> [t,y]= Heun2_Sis('Problema09',2,4,[2 -1],40);
```

```
>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

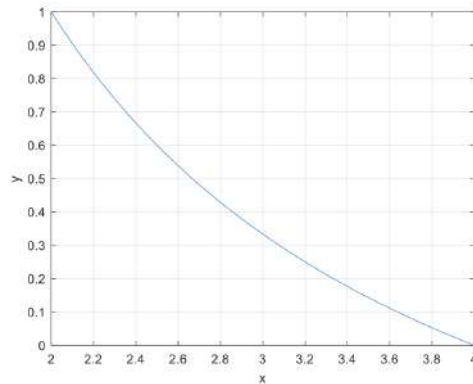


Figura 4.18. Representación de la solución de $u(r)$ con Heun

d) Indique los valores de $u(r)$ para $r = \{2 \ 3 \ 4\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} - a = \frac{b - a}{N + 1} \times x$$

<p>• 2</p> $\begin{matrix} 2 & 41 \\ 0 & x \end{matrix} = 1$	<p>• 3</p> $\begin{matrix} 2 & 41 \\ 1 & x \end{matrix} = 20.5 \rightarrow 21$	<p>• 4</p> $\begin{matrix} 2 & 41 \\ 2 & x \end{matrix} = 41$
--	--	---

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(21) t(41)]
```

```
>> y=[y(1) y(21) y(41)]
```

r	2	3	4
$u(r)$	1.0000	0.3336	0.0002

e) Indique una estimación del orden de convergencia. Posee un orden de convergencia 2.

```
>> [x,y1]=Heun2_Sis('Problrma09',2,4,[1 -1],40*1);sol1=y1(:,1);
>> [x,y2]=Heun2_Sis('Problrma09',2,4,[1 -1],40*2);sol2=y2(:,1);
>> [x,y3]=Heun2_Sis('Problrma09',2,4,[1 -1],40*4);sol3=y3(:,1);
>> [x,y4]=Heun2_Sis('Problrma09',2,4,[1 -1],40*8);sol4=y4(:,1);
>> [x,y5]=Heun2_Sis('Problrma09',2,4,[1 -1],40*16);sol5=y5(:,1);
>> [x,y6]=Heun2_Sis('Problrma09',2,4,[1 -1],40*32);sol6=y6(:,1);
>> [x,y7]=Heun2_Sis('Problrma09',2,4,[1 -1],40*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
>> orden=log2(relacion)

orden =

2.0255
```

2.0127

2.0064

2.0032

2.0016

Como podemos observar alcanza al orden de convergencia 2.

4.3.10. Problema 10

Sea el siguiente problema de valor inicial:

$$y'''(x) = -6y^4 \quad x \in [1, 9]$$

Con las condiciones iniciales $y(1) = -1$ $y'(1) = -1$ $y''(1) = -2$.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''' = z_1''' = z_2'' = z_3'$$

$$y'''(x) = -6y^4 \quad x \in [1, 9] \rightarrow z_3' = -6z_1^4 \quad x \in [1, 9]$$

b) Escriba una función que describa la ecuación diferencial de tercer orden como un sistema de ecuaciones diferencial de primer orden.

`function f = Problema10(x,z)`

`f=[z(2); z(3); -6*z(1).^4];`

`end`

Resuelva el problema con pasos $h=0.05$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1.9-1}{0.05}$ obteniendo el número de iteraciones $N = 18$.

Llamamos al método y graficamos

```
>> [x,y]= Heun2_Sis('Problema10',1,1.9,[-1 -1 -2],18)
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> title('METODO HEUN PARA ECUACIONES')
```

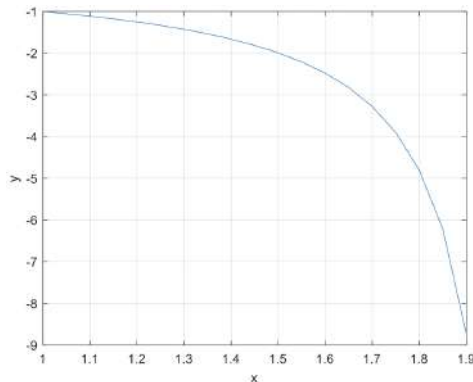


Figura 4.19. Representación de la solución de $y(x)$ con Heun

a) Indique los valores de $y(x)$ para $x \in \{1 \ 1.4 \ 1.9\}$.

Encontramos las posiciones requeridas:

$$\begin{matrix} b-a & N+1 \\ \text{Posición} - a & x \end{matrix}$$

<ul style="list-style-type: none"> 1 $\begin{matrix} 0.9 & 19 \\ 0 & x \end{matrix} = 1$	<ul style="list-style-type: none"> 1.4 $\begin{matrix} 0.9 & 19 \\ 0.4 & x \end{matrix} = 8.4 \rightarrow 9$	<ul style="list-style-type: none"> 1.9 $\begin{matrix} 0.9 & 19 \\ 0.9 & x \end{matrix} = 19$
---	---	--

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(9) x(19)]
```

```
>> y=[y(1) y(9) y(19)]
```

x	1	1.4	1.9
y(x)	-1.0000	-1.6621	-8.7347

d) Si la solución exacta es $y(x) = \frac{1}{x-2}$, comparar gráficamente la solución exacta con la aproximada; además, graficar el error cometido y calcular el error máximo.

Código para representar la solución exacta vs. la aproximada:

```
>> [x,y]= Heun2_Sis('Problema10',1,1.9,[-1 -1 -2],18);  
>> yexacta=1./(x-2);  
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')  
>> hold on  
>> plot(x,yexacta,'red'),legend('y_a_p_r_o_x_i_m_a_d_o','y_e_x_a_c_t_a')  
>> title('Solucion Aproximada vs Exacta')
```

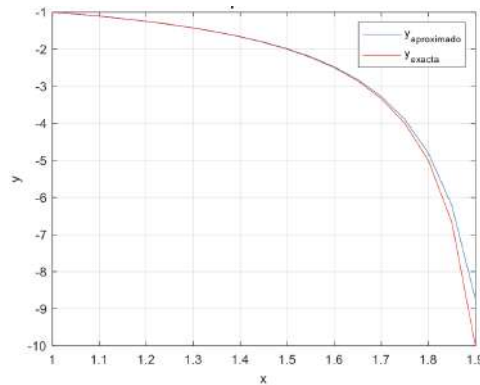


Figura 4.20. Representación de la solución exacta vs. aproximada

Cálculo del error máximo cometido:

```
>> Max_error=max(abs(yexacta-y(:,1)))  
  
Max_error = 1.2653
```

Ingresamos el código para obtener el error cometido:

```
>> figure  
>> plot(x,abs(y(:,1)-yexacta)),grid,title('grafica del error cometido');
```

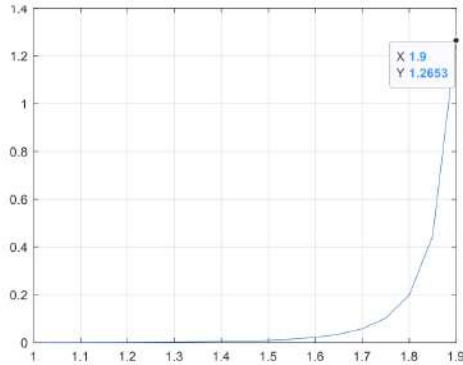



Figura 4.21. Representación del error cometido Problema 10

4.3.11. Problema 11

Sea el siguiente problema de valor inicial

$$y'''(x) = -\frac{36}{y} - \frac{60}{(x-1)^6} + 3(x-1)^5 \quad x \in [0 \ 0.5].$$

Con las condiciones iniciales $y(0) = -1$ $y'(0) = -3$ $y''(0) = -12$

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''' = z_1''' = z_2'' = z_3'$$

$$y'''(x) = -\frac{36}{y} - \frac{60}{(x-1)^6} + 3(x-1)^5 \quad x \in [1 \ 1.9]$$

$$z_3' = -\frac{36}{z_1} - \frac{60}{(x-1)^6} + 3(x-1)^5 \quad x \in [1 \ 1.9]$$

b) Escriba una función que describa la ecuación diferencial de tercer orden como un sistema de ecuaciones diferenciales de primer orden.

`function f = Problema11(x,z)`

`f = [z(2); z(3); -36./z(1)-60./(x-1).^6+3*(x-1).^5];`

`end`

c) Resuelva el ejercicio con pasos de $h = 0,1$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{0.5-0}{0.1}$ obteniendo el número de iteraciones $N = 5$.

Llamamos al método y graficamos:

```
>> [x,y]= Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

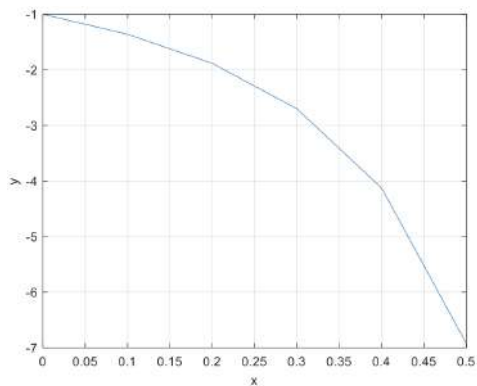


Figura 4.22. Representación de la solución de $y(x)$ con Heun

d) Indique los valores de $y(x)$ para $x \in \{0 \ 0.2 \ 0.5\}$.

Encontramos las posiciones requeridas:

$\frac{b-a}{\text{Posición} - a} \quad \frac{N+1}{x}$		
<ul style="list-style-type: none">0 $\frac{0.5}{0} \frac{6}{x} = 1$	<ul style="list-style-type: none">0.2 $\frac{0.5}{0.2} \frac{6}{x} = 2.4 \rightarrow 3$	<ul style="list-style-type: none">0.5 $\frac{0.5}{0.5} \frac{6}{x} = 6$

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(3) x(6)]
```

```
>> y=[y(1) y(3) y(6)]
```

x	1	0.2	0.5
y(x)	-1.0000	-1.8820	-6.9225

e) Si la solución exacta es $y(x) = \frac{1}{(x-1)^3}$, comparar gráficamente la solución exacta con la aproximada. Además, graficar el error cometido y calcular el error máximo.

```
>> [x,y]= Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5);
>> yexacta=1./(x-1).^3;
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
>> hold on
>> plot(x,yexacta,'red'),legend('y_a_p_r_o_x_i_m_a_d_o','y_e_x_a_c_t_a')
>> title('Solucion Aproximada vs Exacta')
```

Código para representar la solución exacta vs. la aproximada:

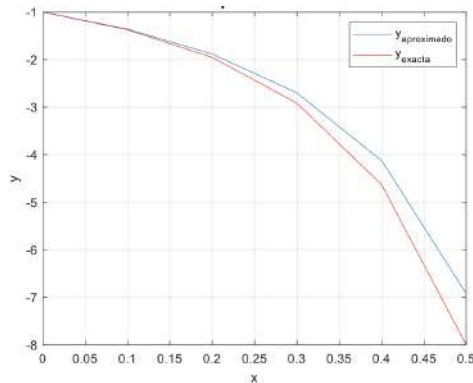


Figura 4.23. Representación de la solución exacta vs. la aproximada

```
>> Max_error=max(abs(yexacta-y(:,1)))
```

Max_error = 1.0775

Ingresamos el código para obtener el error cometido:

```
>> figure
```

```
>> plot(x,abs(y(:,1)-yexacta)),grid,title('grafica del error cometido');
```

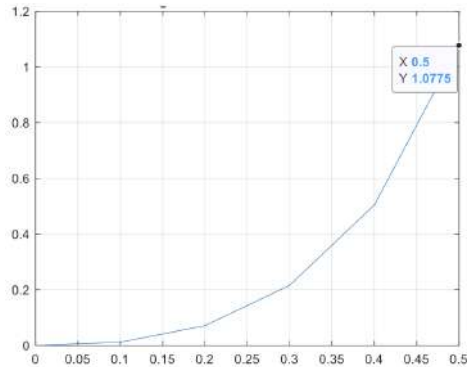


Figura 4.24. Representación del error cometido Problema 11

f) Indique una estimación del orden de convergencia. Posee un orden de convergencia 2.

```
>> [x,y1]=Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*1);sol1=y1(:,1);
>> [x,y2]=Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*2);sol2=y2(:,1);
>> [x,y3]=Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*4);sol3=y3(:,1);
>> [x,y4]=Heun_orden2_NM('Problema11',0,0.5,[-1 -3 -12],5*8);sol4=y4(:,1);
>> [x,y5]= Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*16);sol5=y5(:,1);
>> [x,y6]= Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*32);sol6=y6(:,1);
>> [x,y7]= Heun2_Sis('Problema11',0,0.5,[-1 -3 -12],5*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
>> orden=log2(relación)

orden =

1.5734
```

1.7832

1.8918

1.9461

1.9731

Se aproxima al orden 2.

4.3.12. Problema 12

Sea el siguiente problema de valor inicial:

$$y'''(x) = -2xy'' - y' - y + xy \quad x \in [1, 2]$$

Con las condiciones iniciales $y(1) = -1$ $y'(1) = 0$ $y''(1) = 1$.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''' = z_1''' = z_2'' = z_3'$$

$$y'''(x) = -2xy'' - y' - y + xy \quad x \in [1, 2]$$

$$z_3' = -2xz_3 - z_2 - z_1 + xz_1 \quad x \in [1, 2]$$

b) Escriba una función que describa la ecuación diferencial de tercer orden como un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema12(x,z)
```

```
f=[z(2); z(3); x.*z(1)-2*x.*z(3)-z(2)-z(1)];
```

```
end
```

c) Resuelva el problema con pasos $h = 0,1$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-1}{0.1}$ obteniendo el número de iteraciones $N = 10$.

Llamamos al método y graficamos:

```
>> [x,y]= Heun2_Sis('Problema12',1,2,[-1 0 1],10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> title('METODO HEUN PARA ECUACIONES')
```

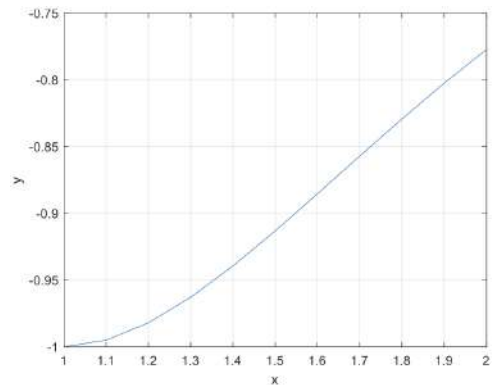


Figura 4.25. Representación de la solución de $y(x)$ con Heun

d) Indique los valores de $y(x)$ para $x \in \{1 \ 1.4 \ 2\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} = a + \frac{b-a}{N+1}x$$

<ul style="list-style-type: none">1 $\frac{1}{0} \frac{11}{x} = 1$	<ul style="list-style-type: none">1.4 $\frac{1}{0.4} \frac{11}{x} = 4.4 \rightarrow 5$	<ul style="list-style-type: none">2 $\frac{1}{1} \frac{11}{x} = 11$
--	--	---

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(5) x(11)]
```

```
>> y=[y(1) y(5) y(11)]
```

x	1	1.4	2
$y(x)$	-1.0000	-0.9394	-0.7773

Sea el siguiente problema de valor inicial:

$$y'''(x) = -\frac{2}{x}y'' - 8y + \frac{8}{x}e^{-2x} + 8\log(x) \quad x \in [1 \ 3].$$

Con las condiciones iniciales $y(1) = e^{-2}$ $y'(1) = 1 - 2e^{-2}$ $y''(1) = -1 + 4e^{-2}$.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''' = z_1''' = z_2'' = z_3'$$

$$y'''(x) = -\frac{2}{x}y'' - 8y + \frac{8}{x}e^{-2x} + 8\log(x) \quad x \in [1 \ 3]$$

$$z_3' = -\frac{2}{x}z_3 - 8z_1 + \frac{8}{x}e^{-2x} + 8\log(x) \quad x \in [1 \ 3]$$

b) Escriba una función que describa la ecuación diferencial de tercer orden como un sistema de ecuaciones diferenciales de primer orden.

`function f = Problema13(x,z)`

`f = [z(2); z(3); -2./x.*z(3)-8*z(1)+8./x.*exp(-2*x)+8*log(x)];`

`end`

c) Resuelva el PVI para el intervalo $x \in [1 \ 3]$ con paso $h = 0,1$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{3-1}{0.1}$ obteniendo el número de iteraciones $N = 20$.

Procedemos a llamar al método en Matlab y graficamos:

```
>> [x,y]=Heun_orden2_NM('Problema13',1,3,[exp(-2) 1-2*exp(-2) -1+4*exp(-2)],20);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

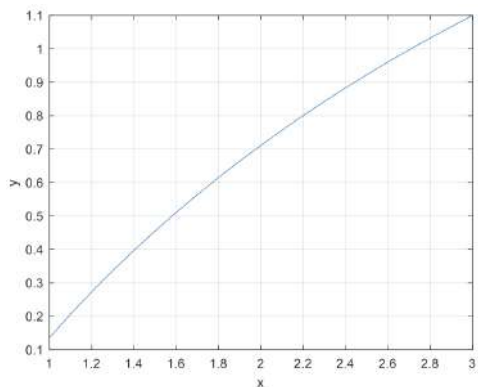


Figura 4.26. Representación de la solución de $y(x)$ con Heun

d) Indique los valores de $y(x)$ para $x \in \{1 \ 1.5 \ 2 \ 2.5 \ 3\}$.

Encontramos las posiciones requeridas:

$$\begin{matrix} b-a & N+1 \\ \text{Posición} - a & x \end{matrix}$$

<ul style="list-style-type: none">1 $\begin{matrix} 2 & 21 \\ 0 & x \end{matrix} = 1$	<ul style="list-style-type: none">1.5 $\begin{matrix} 2 & 21 \\ 0.5 & x \end{matrix} = 5.25 \rightarrow 6$	<ul style="list-style-type: none">2 $\begin{matrix} 2 & 21 \\ 1 & x \end{matrix} = 10.5 \rightarrow 11$
<ul style="list-style-type: none">2.5 $\begin{matrix} 2 & 21 \\ 1.5 & x \end{matrix} = 15.75 \rightarrow 16$	<ul style="list-style-type: none">3 $\begin{matrix} 2 & 21 \\ 2 & x \end{matrix} = 21$	

Ingresamos los códigos en Matlab:

```
>> x=[x(1) x(6) x(11) x(16) x(21)]
```

```
>> y=[y(1) y(5) y(11) y(16) y(21)]
```

x	1	1.5	2	2.5	3
$y(x)$	0.1353	0.3971	0.7113	0.9224	1.0995

e) Indique una estimación del orden de convergencia. Posee un orden de convergencia 2.

```
>>[x,y1]=Heun2_Sis('Problema13',1,3,[exp(-2) 1-2*exp(-2) -1+4*exp(-2)],20*1);sol1=y1(:,1);
```



```
>>[x,y2]=Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*2);sol2=y2(:,1);

>>[x,y3]=  Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*4);sol3=y3(:,1);

>>[x,y4]=  Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*8);sol4=y4(:,1);

>>[x,y5]=  Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*16);sol5=y5(:,1);

>>[x,y6]=  Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*32);sol6=y6(:,1);

>>[x,y7]=  Heun2_Sis('Problema13',1,3,[exp(-2)    1-2*exp(-2)    -1+4*exp(-2)],20*64);sol7=y7(:,1);

>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));

>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));

>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));

>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/max(e5) max(e5)/max(e6) ]';

>> orden=log2(relacion)

orden =

    1.9969

    1.9988

    1.9994

    1.9997

    1.9998
```

4.3.14. Problema 14

Sea el siguiente problema de valor inicial:

$$y'''(x) = 12y' + y + 1 - x[\cos(2x) + 16 \sin(2x)] \quad x \in [0, 2].$$

Con las condiciones iniciales $y(0) = -1$ $y'(0) = 1$ $y''(0) = 0$.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''' = z_1''' = z_2'' = z_3'$$

$$y'''(x) = 12y' + y + 1 - x[\cos(2x) + 16 \sin(2x)] \quad x \in [0, 2]$$

$$z_3' = 12z_2 + z_1 + 1 - x[\cos(2x) + 16 \sin(2x)] \quad x \in [1, 3]$$

b) Escriba una función que describa la ecuación diferencial de tercer orden como un sistema de ecuaciones diferenciales de primer orden.

`function f = Problema14(x,z)`

`f=[z(2); z(3); -12*z(2)+z(1)+1-x.*(cos(2*x)+16*sin(2*x))];`

`end`

c) Resuelva el PVI para el intervalo $x \in [0, 2]$ con paso $h = 0,1$ y represente $y(x)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0,1}$ obteniendo el número de iteraciones $N = 20$.

Procedemos a llamar al método en Matlab y graficamos:

`>> [t,y]= Heun2_Sis('Problema14',0,2,[-1 1 0],20);`

`>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')`

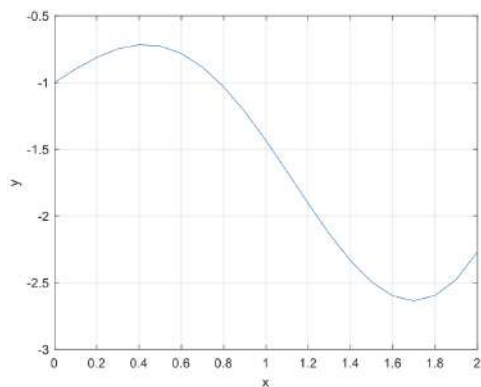


Figura 4.27. Representación de la solución de $y(x)$ con Heun

d) Indique los valores de $y(x)$ para $x \in \{0 \ 0.5 \ 1 \ 1.5 \ 2\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} - a \quad \frac{b-a}{N+1} \quad x$$

<ul style="list-style-type: none"> 0 $\frac{2}{0} \frac{21}{x} = 1$	<ul style="list-style-type: none"> 0.5 $\frac{2}{0.5} \frac{21}{x} = 5.25 \rightarrow 6$	<ul style="list-style-type: none"> 1 $\frac{2}{1} \frac{21}{x} = 10.5 \rightarrow 11$
<ul style="list-style-type: none"> 1.5 $\frac{2}{1.5} \frac{21}{x} = 15.75 \rightarrow 16$	<ul style="list-style-type: none"> 2 $\frac{2}{2} \frac{21}{x} = 21$	

Ingresamos los códigos en Matlab:

```
>> t=[t(1) t(6) t(11) t(16) t(21)]
```

```
>> y=[y(1) y(5) y(11) y(16) y(21)]
```

x	0	0.5	1	1.5	2
$y(x)$	-1.0000	-0.7170	-1.4351	-2.4957	-2.2720

e) Indique una estimación del orden de convergencia. Posee un orden de convergencia 2.

```
>> [t,y1]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*1);sol1=y1(:,1);
```

```
>> [t,y2]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*2);sol2=y2(:,1);
>> [t,y3]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*4);sol3=y3(:,1);
>> [t,y4]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*8);sol4=y4(:,1);
>> [t,y5]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*16);sol5=y5(:,1);
>> [t,y6]=Heun2_Sis('Problema14',0,2,[-1 1 0],20*32);sol6=y6(:,1);
>> [t,y7]= Heun2_Sis('Problema14',0,2,[-1 1 0],20*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
>> orden=log2(relacion)

orden =

    1.9345
    1.9599
    1.9801
    1.9904
    1.9953
```

4.3.15. Problema 15

Consideremos el problema de valor inicial definido mediante la ecuación:

$$y''(x) = 2y'(x) - 5y(x)$$

Sujeto a las condiciones iniciales:

$$y\left(-\frac{1}{2}\right) = 0, \quad y'\left(-\frac{1}{2}\right) = 2e^{-\frac{1}{2}}$$

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$y = z_1$$

$$y' = z_1' = z_2$$

$$y'' = z_1'' = z_2' = z_3$$

$$y''(x) = 2y'(x) - 5y(x)$$

$$z_2' = 2z_2 - 5z_1$$

b) Escriba una función que implemente el sistema de ecuaciones diferenciales de primer orden.

```
function f=Problema15(x,y)
```

```
    f=[z(2);2*z(2)-5*z(1)];
```

```
end
```

c) Use el método de Heun de orden 2 para resolver el PVI en el intervalo $\left[-\frac{1}{2}, \frac{1}{2}\right]$ con 40 subintervalos.

Procedemos a llamar al método en Matlab y graficamos:

```
>> [t,y]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40);
```

```
>> plot(t,y(:,1)),grid on,xlabel('x'),ylabel('y')
```

```
>> legend('y(x)')
```

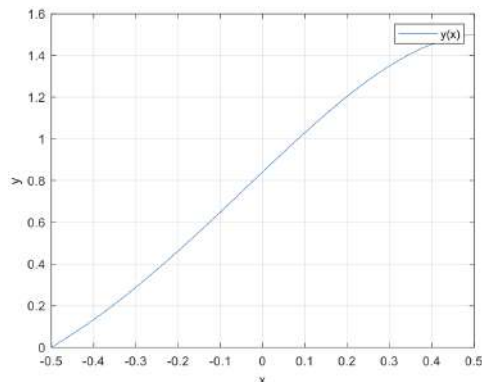


Figura 4.28. Representación de la solución de $y(x)$ con Heun

d) Indique en una tabla los valores de $y(x)$ para $x \in \left\{-\frac{1}{2}, -\frac{1}{4}, -\frac{1}{8}, 0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}\right\}$.

Encontramos las posiciones requeridas:

$\begin{array}{c} b-a \\ \text{Posición} - a \end{array} \quad \begin{array}{c} N+1 \\ x \end{array}$		
$\bullet -\frac{1}{2}$ $\frac{1}{0} \frac{41}{x} = 1$	$\bullet -\frac{1}{4}$ $\frac{1}{0.25} \frac{41}{x} = 10.25 \rightarrow 11$	$\bullet -\frac{1}{8}$ $\frac{1}{0.375} \frac{41}{x} = 15.37 \rightarrow 16$
$\bullet 0$ $\frac{1}{0.5} \frac{41}{x} = 20.5 \rightarrow 21$	$\bullet \frac{1}{8}$ $\frac{1}{0.625} \frac{41}{x} = 25.6 \rightarrow 26$	$\bullet \frac{1}{2}$ $\frac{1}{1} \frac{41}{x} = 41$
$\bullet \frac{1}{4}$ $\frac{1}{0.75} \frac{41}{x} = 30.75 \rightarrow 31$		

Ingresamos los códigos en Matlab:

```
>> x=[t(1) t(11) t(16) t(21) t(26) t(31) t(41)]
```

```
>> y=[y(1) y(11) y(16) y(21) y(26) y(31) y(41)]
```

x	-0.5	-0.25	-0.125	0	0.125	0.25	0.5
$y(x)$	0	0.3735	0.6018	0.8420	1.0761	1.2819	1.5008

e) Indique una estimación del orden de convergencia. Posee un orden de convergencia 2.

```
>> [t,y1]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*1);-
sol1=y1(:,1);
```

```
>> [t,y2]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*2);-
sol2=y2(:,1);
```

```
>> [t,y3]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*4);-
sol3=y3(:,1);
```

```
>> [t,y4]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*8);-
sol4=y4(:,1);

>> [t,y5]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*16);-
sol5=y5(:,1);

>> [t,y6]=Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*32);-
sol6=y6(:,1);

>> [t,y7]= Heun2_Sis('Problema15',-1/2,1/2,[0 2*exp(-1/2)],40*64);-
sol7=y7(:,1);

>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));

>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));

>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));

>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';

>> orden=log2(relación)

orden =

2.0054

2.0029

2.0015

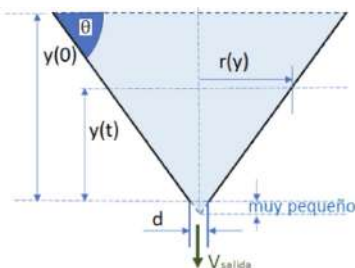
2.0008

2.0004
```

4.3.16 Problema 16

Los embudos cónicos se usan en la industria de bebidas, por ejemplo, para el llenado de botellas y tanques de almacenamiento.

Para la sección correspondiente al embudo cónico mostrado en la figura, se tiene como nivel



inicial $y(0) = 150$ mm, diámetro de salida $d = 10$ mm, la gravedad es $9,8$ m/s², $\theta = \pi/4$.

a) Hallar la ecuación diferencial que gobierne el embudo cónico.

Usando los conceptos de flujo volumétrico $q = AV_{\text{salida}}$, siendo A el área transversal del embudo, $\Delta V = q\Delta t$, la pérdida de volumen $\Delta V = -(\pi r^2)\Delta y$, que $\tan \theta = \frac{y}{r}$, con la fórmula de Bernoulli $V_{\text{salida}} = \sqrt{2gy}$.

Al sustituir se obtiene:

$$-\pi y(t)^2 \Delta y = \frac{\pi d^2}{4} \sqrt{2gy(t)} \Delta t$$

Se obtiene la ecuación ordinaria:

$$\frac{\delta y(t)}{\delta t} + \frac{d^2}{4} \sqrt{2gy(t)} \left[\frac{\tan \theta}{y(t)} \right]^2 = 0$$

Reemplazamos los valores:

$$\frac{\delta y(t)}{\delta t} = -\frac{0.01^2}{4} \sqrt{2 \times (9.8) \times y(t)} \left[\frac{\tan \pi/4}{y(t)} \right]^2$$

$$\frac{\delta y(t)}{\delta t} = -(1.1068e - 4) \sqrt{y(t)} \left[\frac{1}{y(t)} \right]^2$$

$$y'(t) = -(1.1068e - 4) \frac{\sqrt{y(t)}}{y(t)^2}$$

b) Escriba una función que describa la ecuación diferencial.

`function f=Problema16(x,y)`

`f=-(1.1068.*10.^-4).*(sqrt(y)/y.^2);`

`end`

c) Con $h = 0,5$, represente la grafica de $y(t)$ e indique los valores de $y(t)$ para $t \in \{0, 0,5, 1, 1,5\}$. $x \in [0, 1,5]$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1.5-0}{0.5}$ obteniendo el número de iteraciones $N = 3$.

Transformamos la condiconia inicial en metros $y(0) = 0.15$ m.

Llamamos al método y graficamos:


```
>> [x,y]= Heun2_Ecu('Problema16',0,1.5,0.15,3)
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
>> legend('y(t)')
```

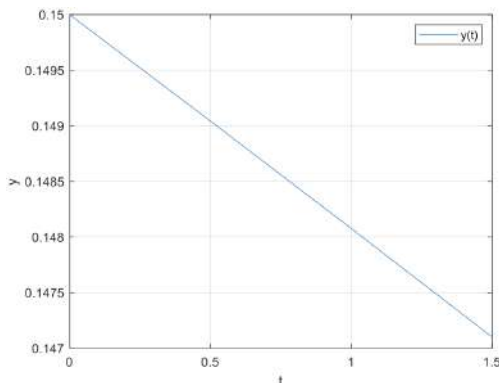


Figura 4.29. Representación de la solución de $y(t)$ con Heun

t	0	0.5	1	1.5
$y(t)$	0.1500	0.1490	0.1481	0.1471

4.3.17. Problema 17

Supongamos una mañana calurosa dentro de una tienda. Las personas están trabajando con el aire acondicionado, manteniéndose la temperatura interior a 20°C . A mediodía, se apaga el aire acondicionado y la gente se va a sus casas. La temperatura exterior permanece constante a 35°C . Si la constante de tiempo del edificio es de cuatro horas:

a) Halle una ecuación diferencial que represente al problema 17.

$$T'(t) = \frac{1}{4}(35 - T(t))$$

b) Dado que $H(t) = U(t) = 0$, junto con la condición inicial $T(0) = 20$, que se corresponde con la temperatura al mediodía, ¿cuál será la temperatura del edificio a las dos de la tarde?

Escribimos la función que se va a utilizar:

`function f=Problema17(x,T)`

```
f=1/4.*(35-T);
```

```
end
```

Llamamos al método y graficamos $y(t)$

```
>> [x,y]= Heun2_Ecu('Problema17',0,2,20,25);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

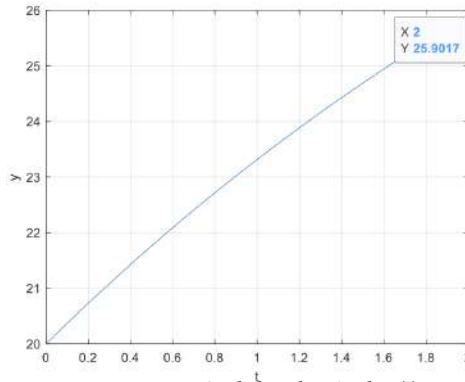


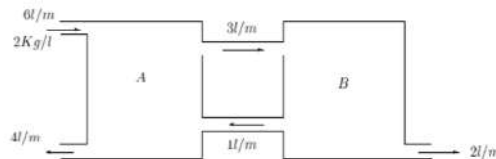
Figura 4.30. Representación de la solución de $y(t)$ con Heun

La temperatura a las dos de la tarde es de 25.9017°C .

4.3.18. Problema 18

Un calentador solar de agua consta de un tanque de agua y un panel solar. El tanque se encuentra bien aislado y tiene una constante de tiempo de sesenta y cuatro horas. El panel solar genera 2000 kilocalorías por hora durante el día y el tanque tiene una capacidad calorífica de 2°C por cada 1000 kilocalorías. Si el agua se encuentra inicialmente a 30°C y la temperatura ambiente es de 20°C .

En este caso



$$U(t) = 2^{\circ}\text{C}/1000\text{Kcal} \times 2000\text{Kcal}/h = 4^{\circ}\text{C}/h$$

a) Halle una ecuación diferencial que represente el problema 18.

$$T'(t) = \frac{1}{64}(20 - T(t)) + 4$$

b) Si la condición inicial $T(0) = 30^{\circ}\text{C}$ se corresponde con la temperatura al mediodía, ¿cuál será la temperatura del tanque al cabo de doce horas de luz solar?

Creamos la función:

```
function f=Problema18(x,T)
```

```
    f=1/64.*(20-T)+4;
```

```
end
```

procedemos a ejecutar el método y graficamos la solución:

```
>> [x,y]=Heun2_Ecu('Problema18',0,12,30,20);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

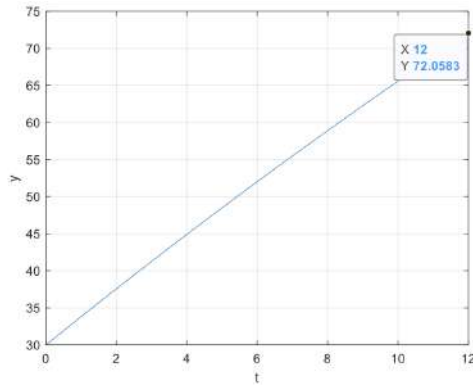


Figura 4.31. Representación de la solución de $y(t)$ con Heun

La temperatura del tanque al cabo de doce horas de luz solar 72.0583°C .

4.3.19. Problema 19

Dos grandes tanques, cada uno con cien litros de líquido, se encuentran interconectados por medio de tubos. El líquido fluye del tanque A hacia el tanque B a razón de 3 l/m y de B hacia A a razón de 1 l/m . El líquido contenido en el interior de cada tanque se mantiene bien agitado. Una solución de salmuera con una concentración de 2 kg/l fluye del exterior hacia el tanque A a razón de 6 l/m . La solución (diluída) fluye hacia el exterior del tanque A a razón de 4 l/m del tanque B a 2 l/m . Si inicialmente el tanque A contenía agua pura y el B 200 kg de sal, determinar la cantidad de sal en cada instante.

a) Encuentre las ecuaciones diferenciales que gobiernan al sistema

Consideramos $x(t)$ e $y(t)$ las cantidades de sal en cada instante en los tanques A y B, respectivamente, la variación de cantidad de sal en el tanque A es:

$$x'(t) = v_e - v_s$$

Donde v_e es la velocidad de entrada de la sal y v_s es la velocidad de salida.

Para el tanque A se tiene que:

$$v_e = 6 \text{ l/m} \times 2 \text{ kg/l} + 1 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}$$

$$v_s = (4 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}) + (3 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}) = 7 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}$$

Encontrando la ecuación diferencial :

$$x'(t) = -\frac{7}{100}x(t) + \frac{1}{100}y(t) + 12$$

Procedemos de igual manera con el tanque B:

$$y'(t) = v_e - v_s$$

Donde

$$v_e = 3 \text{ l/m} \times \frac{x(t)}{100} \text{ kg/l}$$

$$v_s = (1 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}) + (2 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}) = 3 \text{ l/m} \times \frac{y(t)}{100} \text{ kg/l}$$

Obteniendo la ecuación:

$$y'(t) = -\frac{3}{100}x(t) - \frac{3}{100}y(t)$$

Obteniendo el siguiente sistema de ecuaciones que gobierna a los tanques.

$$\begin{cases} x'(t) = -\frac{7}{100}x(t) + \frac{1}{100}y(t) + 12 \\ y'(t) = -\frac{3}{100}x(t) - \frac{3}{100}y(t) \end{cases}$$

b) Escriba una función que implemente el sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema18(t,Sistema)

x=Sistema(1);%aqui se guardan los resultados de x

y=Sistema(2); %aqui se guardan los resultados de y

f=[-7/100*x+1/100*y+12;3/100*x-3/100*y];

end
```

Si consideramos las condiciones iniciales $x(0) = 0$ y $y(0) = 200$ con paso de $h(0) = 0,3$, represente la solución de $x(t)$ y $y(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{12-0}{0.3}$ obteniendo el número de iteraciones $N = 40$.

```
>> [t,y]=Heun_orden2_NM('Problema18',0,12,[0 18],40);
```

```
>> plot(t,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

```
>> hold on
```

```
>> plot(t,y(:,2))
```

```
>> legend('x(t)','y(t)')
```

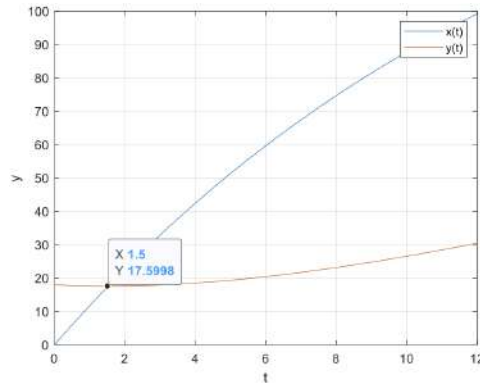
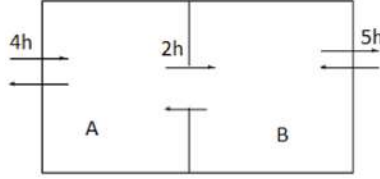


Figura 4.32. Representación de la solución $y(t)$ y $x(t)$ con Heun

4.3.20. Problema 20

Un edificio consta de dos zonas A y B. La zona A es calentada por un calefactor que genera 80 000 Kcal/h. La capacidad calorífica de la zona A es de $1/4^{\circ}\text{C}$ por cada 1000 Kcal. Las constantes de tiempo de transferencia de calor son, entre la

zona A y el exterior, cuatro horas; dos horas entre las zonas A y B; y cinco horas entre la zona B y el exterior. Si la temperatura exterior es de 0°C , determinar la temperatura de cada zona.



a) Encuentre las ecuaciones diferenciales que gobiernan al sistema.

Consideramos $x(t)$ e $y(t)$ a las temperaturas de las zonas A y B respectivamente:

$$x'(t) = \frac{1}{4}(0 - x(t)) + \frac{1}{2}(y(t) - x(t)) + U(t)$$

Donde

$$U(t) = \frac{1}{4} 100^{\circ}\text{C}/1000 \text{ Kcal} \times 80\,000 \text{ Kcal/h} = 20^{\circ}\text{C/h}$$

Obteniendo:

$$x'(t) = -\frac{3}{4}x(t) + \frac{1}{2}y(t) + 20$$

Para la zona B, obtenemos:

$$y'(t) = \frac{1}{5}(0 - y(t)) + \frac{1}{2}(x(t) - y(t))$$

Obteniendo:

$$y'(t) = \frac{1}{2}x(t) - \frac{7}{10}y(t)$$

Obteniendo el siguiente sistema de ecuaciones que gobierna al sistema de calefacción:

$$\begin{cases} x'(t) = -\frac{3}{4}x(t) + \frac{1}{2}y(t) + 20 \\ y'(t) = \frac{1}{2}x(t) - \frac{7}{10}y(t) \end{cases}$$

b) Escriba una función que implemente el sistema de ecuaciones diferenciales de primer orden.

`function f = Problema19(t,Sistema)`

`x=Sistema(1);`

```

y=Sistema(2);
f=[-3/4*x+1/2*y+20;1/2*x-7/10*y];
end

```

Si consideramos las condiciones iniciales con paso de $h(0) = 0.2$ represente la solución de $x(t)$ y $y(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{5-0}{0.2}$ obteniendo el número de iteraciones $N = 25$.

Llamamos al método:

```

>> [t,y]=Heun_orden2_NM('Problema19',0,5,[0 0.25],25);
>> plot(t,y(:,1)),grid on,xlabel('t'),ylabel('y')
>> hold on
>> plot(t,y(:,2))
>> legend('x(t)','y(t)')

```

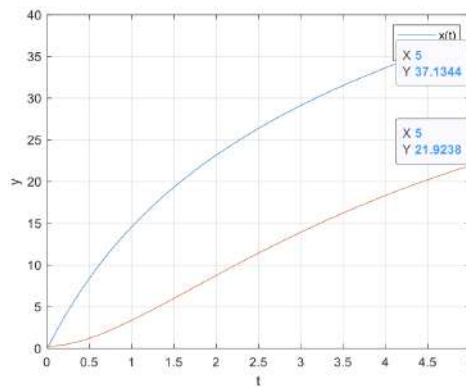


Figura 4.33. Representación de la solución $y(t)$ y $x(t)$ con Heun

La temperatura en la zona $A = 37.1344$ °C; y en la zona $B = 21.9238$ °C.

CAPÍTULO V

5. MÉTODO DE RUNGE-KUTTA

Los métodos descritos en los capítulos anteriores se basan en el desarrollo en la serie de Taylor de la función $y(t)$. Podrían haber sido deducidos a partir de determinadas fórmulas de integración numérica como la vamos a hacer para este caso.

Si partimos de PVI:

$$y'(t) = f(t, y(t)), t \in [a, b], y(a) = y_a$$

Se puede obtener la solución a partir de la integración:

$$y(t) = y_a + \int_a^t f(\tau, y(\tau)) d\tau$$

Si integramos en cada uno de los subintervalos:

$$\begin{aligned} y(t_1) &= y(a) + \int_a^{t_1} f(\tau, y(\tau)) d\tau \\ y(t_2) &= y(t_1) + \int_{t_1}^{t_2} f(\tau, y(\tau)) d\tau \\ &\vdots \\ y(b) &= y(t_N) + \int_{t_N}^b f(\tau, y(\tau)) d\tau \end{aligned}$$

De forma general se obtiene:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(\tau, y(\tau)) d\tau$$

Aproximando la integral por la fórmula de Simpson

$$\int_{t_k}^{t_{k+1}} f(\tau, y(\tau)) d\tau \approx \frac{t_{k+1} - t_k}{6} \left(f(t_k, y_k) + 4f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right) + f(t_{k+1}, y_{k+1}) \right)$$

Obtenemos:

$$y_{k+1} = y_k + \frac{h}{6} \left(f(t_k, y_k) + 4f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right) + f(t_{k+1}, y_{k+1}) \right)$$

El inconveniente es que no conocemos los términos $f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right)$ y f . Para solucionar, realizamos las siguientes aproximaciones:

$$f\left(t_{k+\frac{1}{2}}, y_{k+\frac{1}{2}}\right) = f\left(t_k + \frac{h}{2}, y\left(t_k + \frac{h}{2}\right)\right) \approx \frac{1}{2}(f_2 + f_3)$$

Donde:

$$f_1 = f(t_k, y_k)$$

$$f_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right)$$

$$f_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}f_2\right)$$

De modo que:

$$y_{k+1} \approx y_k + \frac{h}{6}(f_1 + 2f_2 + 3f_3 + f(t_{k+1}, y_k + hf_3))$$

La expresión general de Runge-Kutta es:

$$y_{k+1} \approx y_k + \frac{h}{6}(k_1 + 2k_2 + 3k_3 + k_4)$$

Donde

$$k_1 = f(t_k, y_k)$$

$$k_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_{k+1}, y_k + k_3)$$

Teorema 4. Sea f tal que $y'(t) = f(t, y(t))$ con condiciones iniciales $y(t_a) = y_a$. Si $y(t) \in \mathcal{C}^5$ $[a, b]$ y $\{(t_k, y_k)\}_{k \geq 0}$ es la sucesión de aproximaciones dadas por el método de Runge-Kutta, entonces:

$$|e_k| \leq |y(t_k) - y_k| = \mathcal{O}(h^5)$$

Y:

$$\text{Satisface } Eh = \frac{1}{h_{1 \leq k \leq N}} \max |e_k| = \mathcal{O}(h^4).$$

5.1. ALGORITMO DE RUN

Para la construcción del algoritmo de Euler consideramos:

- Obtención de la variable independiente discretizada
- Inicialización del vector posición
- Para $k=1:N$
 - Valor de k_1
 - Valor de k_2
 - Valor de k_3
 - Valor de k_4
 - Valor de $y(k+1)$
- Fin par k
- Salida x,y

Construimos el código en Matlab implementando el método de Runge-Kutta como se muestra a continuación

Runge-Kutta para sistemas de orden 2 para ecuaciones diferenciales:

```
function [x,y] = RK2_Ecu(fun,a,b,y0,N)
h=(b-a)/N;% paso
x=a:h:b;% vector de nodos x
x=x(:);% vector en forma de columna;
y=zeros(N+1,length(y0));% length cuántas columnas va a tener
y(1)=y0;% almacenamos el vector la primera componente
%fun=@(x,y) (1-2*x)*y; % función
%BUCLE
```

```

for k=1:N % contador

    K1=h*feval(fun,x(k),y(k));% para evaluar funciones Feval

    K2=h*feval(fun,x(k+1),y(k)+K1);

    y(k+1)=y(k)+(K1+K2)/2;% adaptamos a matrices

end

```

Runge-Kutta para sistemas de orden 4 para ecuaciones diferenciales:

```

function [x,y] = RK4_Ecu(fun,a,b,y0,N)

h=(b-a)/N;

x=a:h:b;

x=x(:);

y=zeros(N+1,length(y0));

y(1)=y0;

for k=1:N

    k1=feval(fun,x(k),y(k));

    k2=feval(fun,x(k)+h/2,y(k)+h*k1/2);

    k3=feval(fun,x(k)+h/2,y(k)+h*k2/2);

    k4=feval(fun,x(k+1),y(k)+h*k3);

    y(k+1)=y(k)+h*(k1+2*k2+2*k3+k4)/6;

end

```

Runge-Kutta para sistemas de ecuaciones:

```

function [x,y] = RK4_Sis(fun,a,b,y0,N)

h=(b-a)/N;%paso

x=a:h:b;%vector de nodos x

x=x(:);% vector en forma de columna;

```

```

n=length(y0);
y=zeros(N+1,n);
y(1,:)=y0;% almacenamos el vector la primera componente
for k=1:N % contador
    K1=feval(fun,x(k),y(k,:))';% para evaluar funciones Feval
    K2=feval(fun,x(k)+h/2,y(k,:)+h/2*K1)';
    K3=feval(fun,x(k)+h/2,y(k,:)+h/2*K2)';
    K4=feval(fun,x(k+1),y(k,:)+h*K3)';
    y(k+1,:)=y(k,:)+h*(K1+2*K2+2*K3+K4)/6;% adaptamos a matrices
end

```

5.2. PROBLEMAS PVI CON RUNGE-KUTTA

5.2.1. Problema 1

Una pieza metálica con una masa de 0,1 kg y 25°C se calienta internamente de forma eléctrica a razón de $q = 3000$ W. La ecuación diferencial de temperatura que se obtiene es:

$$\frac{dT}{dt} = 20 - t^2, \text{ si } T(0) = 298$$

a) Grafique $T(1)$ empleando $h = 0.01$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{25}{0.01}$ obteniendo el número de iteraciones $N = 2500$.

Creamos la función:

```

function f=Problema01(t,T)
    f=20-t^2;
end

```

llamamos al método y graficamos su solución:

```
>> [x,y]=RK_Ecu2('Problema01',0,25,298,2500);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('T')
```

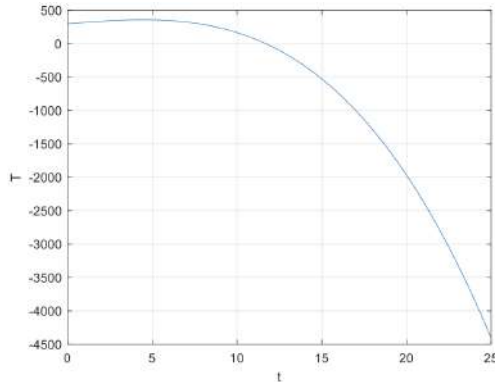


Figura 5.1. Representación de la solución con Runge-Kutta orden 2

5.2.2. Problema 2

La Ley de Newton del enfriamiento establece que la razón de cambio de la temperatura $T(t)$ es proporcional a la diferencia entre la temperatura del medio $M(t)$ y la temperatura del cuerpo. Esto es:

$$\frac{dT}{dt} = K[M(t) - T(t)]$$

Donde K es una constante $K = 1 \text{ l/min}$ y supóngase que la temperatura del medio es constante $M(t) = 70^\circ\text{F}$. Ai el cuerpo esta inicialmente a 100°F , use el método con $h = 0.1$ para aproximar la temperatura del cuerpo al cabo de:

a) Un minuto.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.1}$ obteniendo el número de iteraciones $N = 10$.

Creamos la función:

```
function f=Problema02(t,T)
```

```
    f=1*(70-T);
```

```
end
```

Procedemos a llamar a la función para hallar la solución:

```
>> [x,y]=RK2_Ecu('Problema02r',0,1,100,10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('T')
```

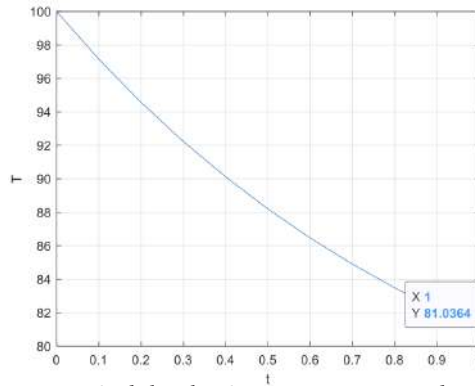


Figura 5.2. Representación de la solución con Runge-Kutta orden 2 en un minuto\

b) Dos minutos.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0.1}$ obteniendo el número de iteraciones $N = 20$.

```
>> [x,y]=RK2_Ecu('Problema02r',0,2,100,20);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('T')
```

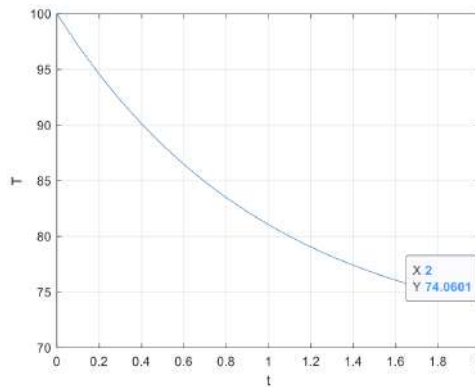


Figura 5.3. Representación de la solución con Runge-Kutta orden 2 en dos minutos

5.2.3. Problema 3

La ley de Stefan de radiación establece que la relación de cambio de la temperatura de un cuerpo a $T(t)$ grados, que se encuentra en un medio a $M(t)$ grados es, proporcional a $M^4 - T^4$. Es decir

$$\frac{dT}{dt} = K[M^4 - T^4]$$

Donde K es una constante. Sea $k = 40^{-4}$ y suponga que el medio de la temperatura es constante $M(t) = 70$. Si $T(0) = 1000$ con $h = 0,1$ para aproximar la temperatura del cuerpo al cabo de

a) Dos minutos.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0.1}$ obteniendo el número de iteraciones $N = 20$.

Creamos la función:

```
function f=Problema03r(t,T)
```

```
f=40^-4.*(70.^4-T.^4);
```

```
end
```

Llamamos al método para graficar la solución:

```
>> [x,y]=RK2_Ecu('Problema03r',0,2,100,20);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('T')
```

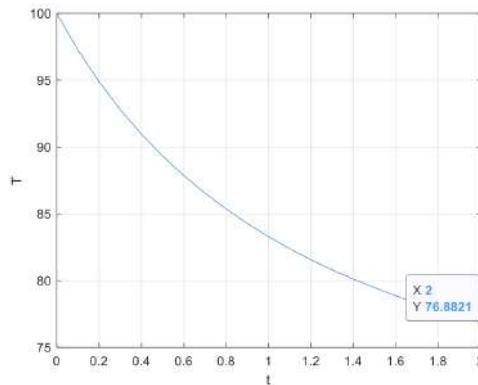


Figura 5.4. Representación de la solución con Runge-Kutta orden 2 en dos minutos

5.2.4. Problema 4

En un circuito eléctrico, se dispone de un condensador con capacidad constante $C = 1.1$ faradios y una resistencia $R = 2.1$ ohmios. Se le aplica un voltaje:

$$E(t) = e^{-0.06\pi t} \sin(2t - \pi)$$

La ecuación que rige el comportamiento de un circuito sin inducción es:

$$R \frac{dI}{dt} + \frac{I}{C} = \frac{dE}{dt}$$

Supongamos que la intensidad en el instante inicial es $I(0) = 1$ amperio.

a) Hallar la ecuación diferencial que gobierna el sistema.

Reemplazamos los valores numéricos en el modelo matemático, se obtiene:

$$2.1 \frac{dI}{dt} + \frac{I}{1.1} = (-0.06\pi t) e^{-0.06\pi t} \sin(2t - \pi) + 2e^{-0.06\pi t} \cos(2t - \pi)$$

Despejando:

$$\frac{dI}{dt} = -\frac{I}{2.31} - \frac{0.06\pi}{2.1} e^{-0.06\pi t} \sin(2t - \pi) + \frac{2}{2.1} e^{-0.06\pi t} \cos(2t - \pi)$$

b) Halle el valor aproximado de la intensidad cada 0,2 segundos los dos primeros segundos.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0.2}$ obteniendo el número de iteraciones $N = 10$.

Creamos la función:

```
function f=Problema04r(t,I)
```

```
f=-I/2.31-(0.06*pi/2.1)*exp(-0.06*pi*t)*sin(2*t-pi)+2/2.1*exp(-0.06*pi*t)*cos(2*t-pi);
```

```
end
```

Procedemos a llamar al método para graficar la solución:

```
>> [x,y]=RK2_Ecu('Problema04r',0,2,1,10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('I')
```

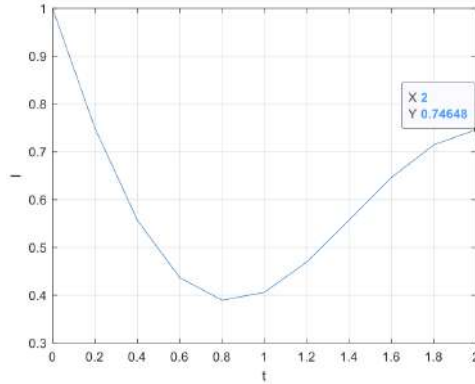



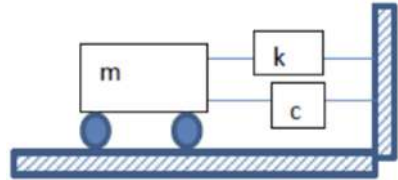
Figura 5.5. Representación de la solución con Runge-Kutta orden 2 en dos minutos

El resultado que obtenemos es $I(2) = \approx 0.7464$ amperios.

5.2.5. Problema 5

El movimiento de un sistema acoplado masa-resorte está descrito por la ecuación diferencial ordinaria que sigue:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$



Donde

x = el desplazamiento desde la posición de equilibrio (m)

t = tiempo (s)

$m = 20$ kg masa

$c = 5$ (N s/m) coeficiente de amortiguamiento

$k = 20$ (N/m) constante del resorte

La velocidad inicial es cero y el desplazamiento inicial es 1 m.

a) Transforme el PVI en un sistema de ecuaciones diferenciales de primer orden.

$$20x'' + 5x' + 20x = 0$$

$$x'' = -\frac{x'}{4} - x$$

Transformamos en un sistema de ecuaciones de primer orden:

$$x = z_1$$

$$x' = z_1' = z_2$$

$$x'' = z_1'' = z_2'$$

$$x''(t) = -\frac{x'}{4} - x \rightarrow z_2' = -\frac{z_2}{4} - z_1$$

b) Encuentre la solución para $x(t)$ y $x'(t)$ para $0 \leq t \leq 15s$ con paso $h = 0.1s$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{15-0}{0.1}$ obteniendo el número de iteraciones $N = 150$.

Creamos la función:

```
function f = Problema4r(t,z)
```

```
f=[z(2);-z(2)/4-z(1)];
```

```
end
```

Llamamos al método para encontrar la solución:

```
>> [x,y]=RK4_Sis('Problema4r',0,15,[0 1],150);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('x')
```

```
>> hold on
```

```
>> plot(x,y(:,2))
```

```
>> legend('x','x(2)')
```

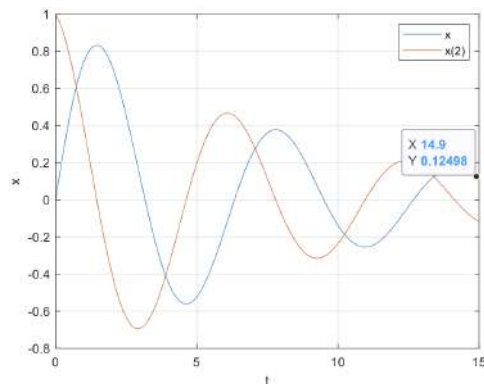


Figura 5.6. Representación de la solución de para $x(t)$ y $x'(t)$ con Runge-Kutta

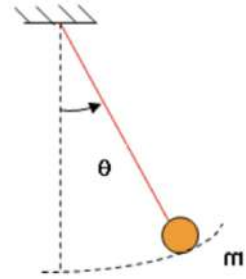
5.2.6. Problema 6

Suponga que un péndulo tiene 0.6 m de longitud, se desplaza θ desde la posición vertical de equilibrio.

$$\frac{d^2\theta}{dt^2} + \frac{g}{L}\sin(\theta) = 0$$

Donde

$$0 < t < 1, g = 9.81 \text{ m/s}^2 \text{ y } \theta(0) = \frac{\pi}{6} \text{ y } \theta'(0) = 0$$



a) Aproxime la solución de la ecuación para $h = 0.2$ y represente $\theta(t)$.

$$\theta'' = -\frac{g}{L}\sin(\theta) \rightarrow \theta'' = -\frac{9.8}{0.6}\sin(\theta)$$

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0.2}$ obteniendo el número de iteraciones $N = 5$.

Transformamos en un sistema de ecuaciones de primer orden:

$$\theta = z_1$$

$$\theta' = z_1' = z_2$$

$$\theta'' = z_1'' = z_2'$$

$$\theta''(t) = -\frac{9.8}{0.6}\sin(\theta) \rightarrow z_2' = -\frac{9.8}{0.6}\sin(z_1)$$

Creamos la función:

```
function f = Problema5r(t,z)
```

```
    f=[z(2);-16.33*sin(z(1))];
```

```
end
```

Llamamos al método para graficar la solución aproximada:

```
>> [x,y]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('x')
```

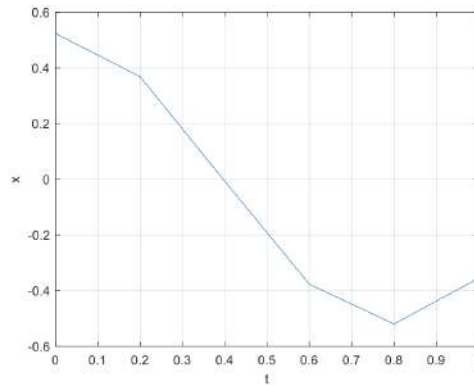


Figura 5.7. Representación de la solución de para $\theta(t)$ con Runge-Kutta

b) Indique una estimación del orden de convergencia.

```
>> [x,y]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5);
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('x')
>> [x,y1]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*1);sol1=y1(:,1);
>> [x,y2]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*2);sol2=y2(:,1);
>> [x,y3]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*4);sol3=y3(:,1);
>> [x,y4]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*8);sol4=y4(:,1);
>> [x,y5]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*16);sol5=y5(:,1);
>> [x,y6]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*32);sol6=y6(:,1);
>> [x,y7]=RK_orden4_NM('Problema5r',0,1,[pi/6 0],5*64);sol7=y7(:,1);
>> e1=abs(sol1-sol2(1:2:end)); e2=abs(sol2-sol3(1:2:end));
>> e3=abs(sol3-sol4(1:2:end)); e4=abs(sol4-sol5(1:2:end));
>> e5=abs(sol5-sol6(1:2:end)); e6=abs(sol6-sol7(1:2:end));
>> relacion=[max(e1)/max(e2) max(e2)/max(e3) max(e3)/max(e4) max(e4)/
max(e5) max(e5)/max(e6) ]';
>> orden=log2(relacion)
```

orden =

4.1815

3.9340

3.8359

3.9280

3.9663

5.2.7. Problema 7

En un tanque cilíndrico vertical, al abrir una válvula en la base, el agua fluirá rápidamente cuando el tanque esté lleno; conforme el tanque se vacíe, irá fluyendo más lentamente.

Si la rapidez a la que disminuye el nivel del agua es:

$$\frac{dy}{dt} = -k\sqrt{y}$$

Donde k es una constante que depende del área de la sección transversal del tanque y del orificio de salida. La profundidad del agua y se mide en pies; y el tiempo t en minutos. Si $k = 0.5$ e inicialmente el nivel del fluido es de nueve pies.

a) Resolver este problema con $h = 0.5$ minutos y $t \in [0, 0.5]$. ¿Cuál es el tiempo mínimo para que la altura del tanque sea inferior a seis pies?

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{0.5-0}{0.5}$ obteniendo el número de iteraciones $N = 1$.

Creamos la función:

```
function f = Problema6r(t,y)
```

```
f=-0.5*sqrt(y);
```

```
end
```

Llamamos al método para encontrar el tiempo mínimo:

```
>> [x,y]=RK_orden4_N('Problema6r',0,0.5,9,1);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

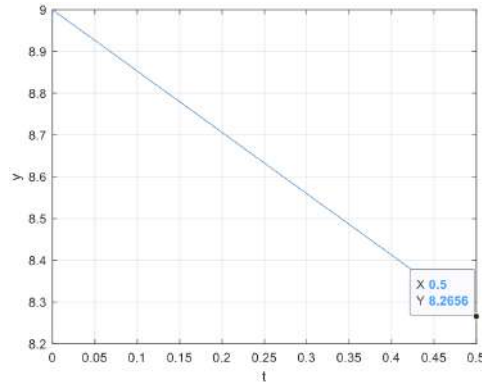
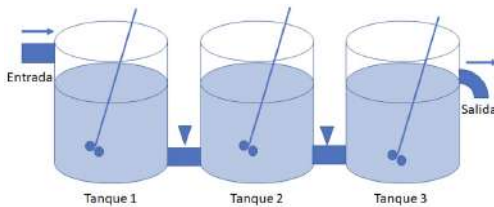


Figura 5.8. Representación de la solución con Runge-Kutta

El tiempo mínimo es de 8.2656 minutos.

5.2.8. Problema 8

Tres tanques perfectamente aislados, completamente llenos con una solución de concentración $C_i(0)$ g/L.



Los tanques están interconectados en serie de tal forma que, de añadir solución al primero, se transfiere la misma cantidad por la conexión al segundo y al tercero del cual rebosa hacia afuera del sistema. El tercer tanque tiene una salida por rebose que mantiene constante el volumen 3 en cada tanque. Desde un tiempo $t(0) = 0$, al primer tanque, se le añade una solución que tiene una concentración 50 g/l, a razón de 300 l/min.

Considere $C_i(0) = 30$ g/l y el volumen de cada tanque de 1000 l. En cada tanque, entre lo que recibe y se transfiere al siguiente tanque, se obtienen las siguientes ecuaciones:

$$\begin{aligned}\frac{dC_1}{dt} &= \frac{300}{1000} (50) - 0,3C_1 \\ \frac{dC_2}{dt} &= 0,3C_1 - 0,3C_2 \\ \frac{dC_3}{dt} &= 0,3C_2 - 0,3C_3\end{aligned}$$

a) Determine la concentración en cada tanque durante los tres primeros minutos de iniciar el experimento con $N =$

Creamos la función:

```
function f = Problema7r(t,Sistema)

x=Sistema(1); % aquí se guardan los resultados de x
y=Sistema(2); % aquí se guardan los resultados de y
z=Sistema(3); % aquí se guardan los resultados de z

f=[15-0.3*x;0.3*x-0.3*y;0.3*y-0.3*z];

end
```

Llamamos al método para graficar la solución aproximada de cada tanque:

```
>> [x,y]=RK_orden4_NM('Problema7r',0,3,[30 30 30],20);
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
>> hold on
>> plot(x,y(:,2))
>> plot(x,y(:,3))
>> legend('C1','C2','C3')
```

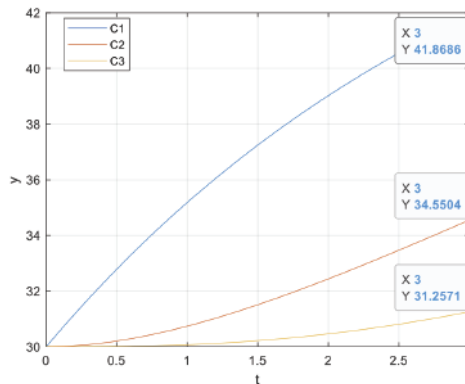
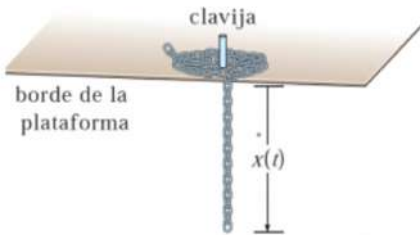


Figura 5.9. Representación de la solución de cada tanque con Runge-Kutta

5.2.9. Problema 9

Cadena cayendo. Una parte de una cadena de $L = 8$ pies de longitud está enrollada sin apretar alrededor de una clavija en el borde de una plataforma horizontal y la parte restante de la cadena cuelga descansando sobre el borde de la plataforma. Por simplicidad, use $g = 32$ pies/s².



Suponga que la longitud de la cadena que cuelga es de $x(0) = 3$ pies, que la cadena pesa 2 lb/pie y que la dirección positiva es hacia abajo.

Comenzando en $t = 0$ segundos, el peso de la cadena que cuelga causa que, sobre la plataforma, el resto se desenrolle suavemente y caiga al piso.

Si $x(t)$ denota la longitud de la cadena que cuelga de la mesa al tiempo $t = 0$, entonces $v = dx/dt$ es su velocidad $v(0) = 0$. Cuando se desprecian todas las fuerzas de resistencia, se puede demostrar que un modelo matemático que relaciona a v con x está dado por la ecuación mostrada.

$$\frac{d^2x}{dt^2} - \frac{g}{L}x = 0 \quad 0 \leq t \leq L$$

a) Resuelva el problema con $h = 0.05$.

$$x'' = \frac{32}{8}x \quad 0 \leq t \leq 8$$

Condiciones $x(0) = 3$, $v = \frac{dx}{dt} = 0 = x'$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{8-0}{0.05}$ obteniendo el número de iteraciones $N = 160$.

Transformamos en sistema de ecuaciones diferenciales:

$$x = z_1$$

$$x' = z_1' = z_2$$

$$x'' = z_1'' = z_2'$$

$$x''(t) = 4x \rightarrow z_2' = 4z_1$$

Creamos la función:

```
function f = Problema8(t,z)

f=[z(2);4*z(1)];

end
```

Llamamos al método para graficar la aproximación de la solución:

```
>> [x,y]=RK_orden4_NM('Problema8r',0,8,[3 0],160);
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('x')
```

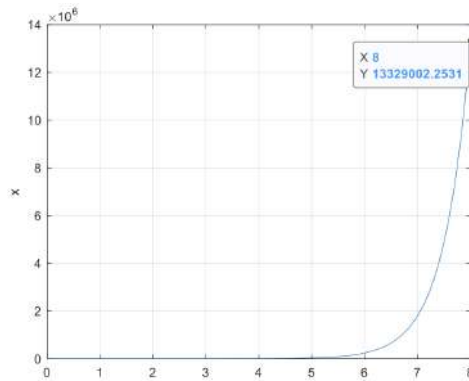


Figura 5.10. Representación de la solución con Runge-Kutta

5.2.10. Problema 10

Sea el siguiente problema de valor inicial:

$$x'(t) = 3x + 2y - (2t^2 + 1) e^{2t}$$

$$y'(t) = 4x + y - (t^2 + 2t - 4) e^{2t}$$

Con las condiciones iniciales $x(0) = 1$, $y(0) = 1$.

a) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema9(t,Sistema)

x=Sistema(1);

y=Sistema(2);
```

```
f=[3*x+2*y-(2*t.^2+1).*exp(2*t); 4*x+y+(t.^2+2*t-4).*exp(2*t)];
```

```
end
```

Resuelva el problema con $t \in [0, 1]$ y pasos $h = 0,1$. Represente $x(t)$ y $y(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{1-0}{0,1}$ obteniendo el número de iteraciones $N = 10$.

Llamamos al método para graficar la aproximación de la solución:

```
>> [x,y]=RK_orden4_NM('Problema9',0,1,[1 1],10);
```

```
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('y')
```

```
>> hold on
```

```
>> plot(x,y(:,2))
```

```
>> legend('x(t)','y(t)')
```

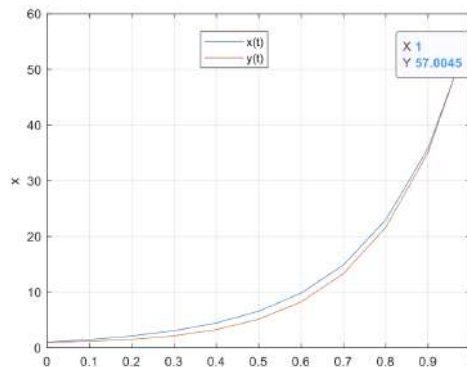


Figura 5.11. Representación de la solución $x(t)$ y $y(t)$ con Runge-Kutta

c) Indicar los valores de $x(t)$ y $y(t)$ para $t \in \{0,0.2,0.4,0.6,0.8,1\}$.

Encontramos las posiciones requeridas

$$\text{Posición} - a = \frac{b-a}{N+1} x$$

<ul style="list-style-type: none"> • 0 $\begin{matrix} 1 & 11 \\ 0 & x \end{matrix} = 1$	<ul style="list-style-type: none"> • 0.2 $\begin{matrix} 1 & 11 \\ 0.2 & x \end{matrix} = 2.2 \rightarrow 3$	<ul style="list-style-type: none"> • 0.4 $\begin{matrix} 1 & 11 \\ 0.4 & x \end{matrix} = 4.4 \rightarrow 5$
<ul style="list-style-type: none"> • 0.6 $\begin{matrix} 1 & 11 \\ 0.6 & x \end{matrix} = 6.6 \rightarrow 7$	<ul style="list-style-type: none"> • 0.8 $\begin{matrix} 1 & 11 \\ 0.8 & x \end{matrix} = 8.8 \rightarrow 9$	<ul style="list-style-type: none"> • 1 $\begin{matrix} 1 & 11 \\ 1 & x \end{matrix} = 11$

Ingresamos los códigos en Matlab:

```
>> Z=[x(1) x(3) x(5) x(7) x(9) x(11)]
>> X=[y(1,1) y(3,1) y(5,1) y(7,1) y(9,1) y(11,1)]
>> Y=[y(1,2) y(3,2) y(5,2) y(7,2) y(9,2) y(11,2)]
```

t	0	0.2	0.4	0.6	0.8	1
$x(t)$	1	1.2864	1.7815	2.5654	3.7652	5.5755
$y(t)$	1	1.9026	3.1136	4.8295	7.3285	11.0157

d) Compare gráficamente la solución exacta; con dicha aproximación, calcule un error máximo si las soluciones exactas son $x(t) = \frac{e^{5t}}{3} - \frac{e^{-t}}{3} + e^{2t}$ $y(t) = \frac{e^{5t}}{3} + \frac{2e^{-t}}{3} + t^2 e^{2t}$.

Comparamos gráficamente con los siguientes comandos en Matlab:

```
>> [x,y]=RK_orden4_NM('Problema9',0,1,[1 1],10);
>> plot(x,y(:,1)),grid on,xlabel('t'),ylabel('x')
>> hold on;
>> plot(x,y(:,2))
>> legend('x(t)','y(t)')
>> xexacta=1/3.*exp(5*x)-1/3.*exp(-x)+exp(2*x);
>> yexacta=1/3.*exp(5*x)+2/3.*exp(-x)+exp(2*x).*(x.^2);
>> plot(x,xexacta,'black');
```

```
>> plot(x,yexacta,'red'),legend('x_a(t)','y_a(t)','x_e(t)','y_e(t)')
```

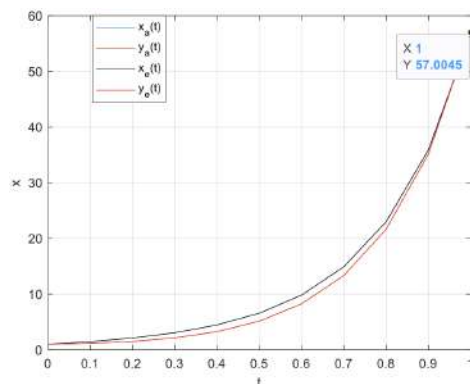


Figura 5.12. Comparación de la solución exacta con la aproximada

Calculamos el error:

```
>> Max_errorX=max(abs(xexacta-y(:,1)))
```

```
Max_errorX = 0.1010
```

```
>> Max_errorY=max(abs(yexacta-y(:,2)))
```

```
Max_errorY = 0.1009
```

Problema 11

Sea el siguiente problema de valor inicial:

$$x'(t) = y$$

$$y'(t) = -x - 2e^t + 1$$

$$z'(t) = -x - e^t + 1$$

Con las condiciones iniciales $x(0) = 1$, $y(0) = 0$, $z(0) = 1$.

a) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden

```
function f = Problema10(t,Sistema)
```

```
x=Sistema(1);
```

```
y=Sistema(2);
```

```
z=Sistema(3);  
f=[y; -x-2*exp(t)+1; -x-exp(t)+1];
```

```
end
```

b) Resuelva el problema con $t \in [0, 2]$ y pasos $h = 0.1$. Represente $x(t)$, $y(t)$ y $z(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0.1}$ obteniendo el número de iteraciones $N = 20$.

Llamamos al método para graficar la aproximación de la solución

```
>> [x,y]=RK_orden4_NM('Problema10',0,2,[1 0 1],20);
```

```
>> plot(x,y),grid on,xlabel('t'),ylabel('x')
```

```
>> legend('x(t)','y(t)','z(t)')
```

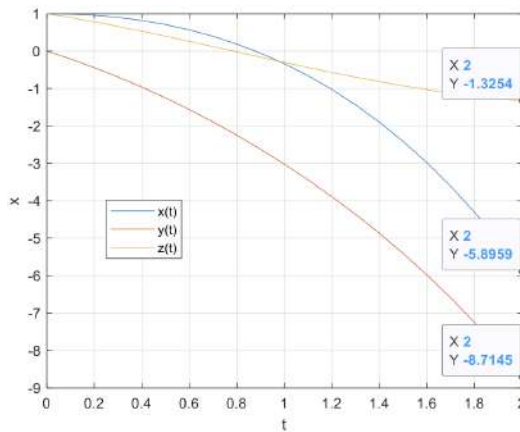


Figura 5.13. Representación de la solución $x(t)$, $y(t)$ y $z(t)$ con Runge-Kutta

c) Indicar los valores de $x(t)$, $y(t)$ y $z(t)$ para $t \in \{0, 0.6, 1, 1.4, 1.8, 2\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} = a + \frac{b-a}{N+1}x$$

<ul style="list-style-type: none"> 0 $\begin{matrix} 2 & 21 \\ 0 & x \end{matrix} = 1$	<ul style="list-style-type: none"> 0.6 $\begin{matrix} 2 & 21 \\ 0.6 & x \end{matrix} = 6.3 \rightarrow 7$	<ul style="list-style-type: none"> 1 $\begin{matrix} 2 & 21 \\ 1 & x \end{matrix} = 10.5 \rightarrow 11$
<ul style="list-style-type: none"> 1.4 $\begin{matrix} 2 & 21 \\ 1.4 & x \end{matrix} = 14.7 \rightarrow 15$	<ul style="list-style-type: none"> 1.8 $\begin{matrix} 2 & 21 \\ 1.8 & x \end{matrix} = 18.9 \rightarrow 19$	<ul style="list-style-type: none"> 2 $\begin{matrix} 2 & 21 \\ 2 & x \end{matrix} = 21$

Ingresamos los códigos en Matlab:

```
>> T=[x(1) x(7) x(11) x(15) x(19) x(21)]
>> X=[y(1,1) y(7,1) y(11,1) y(15,1) y(19,1) y(21,1)]
>> Y=[y(1,2) y(7,2) y(11,2) y(15,2) y(19,2) y(21,2)]
>> Z=[y(1,3) y(7,3) y(11,3) y(15,3) y(19,3) y(21,3)]
```

t	0	0.6	1	1.4	1.8	2
$x(t)$	1	0.5679	-0.3365	-1.8998	-4.3030	-5.8959
$y(t)$	0	-1.5614	-3.0194	-4.8707	-7.2507	-8.7145
$z(t)$	1	0.2607	-0.3012	-0.8155	-1.2010	-1.3254

d) Compare gráficamente la solución exacta; con dicha aproximación, calcule un error máximo si las soluciones exactas son $x(t) = \cos(t) + \sin(t) - e^t + 1$, $y(t) = -\sin(t) + \cos(t) - e^t$, $z(t) = \cos(t) - \sin(t)$.

Comparamos gráficamente con los siguientes comandos en Matlab:

```
>> [x,y]=RK_orden4_NM('Problema10',0,2,[1 0 1],20);
>> plot(x,y),grid on,xlabel('t'),ylabel('x')
>> hold on
>> xexacta=cos(x)+sin(x)-exp(x)+1;
>> yexacta=-sin(x)+cos(x)-exp(x);
>> zexacta=cos(x)-sin(x);
>> plot(x,xexacta,'cyan');
```

```
>> plot(x,yexacta,'magenta');

>> plot(x,zexacta,'yellow'),legend('x_a(t)','y_a(t)','z_a(t)','x_e(t)','y_e(-t)','z_e(t)')
```

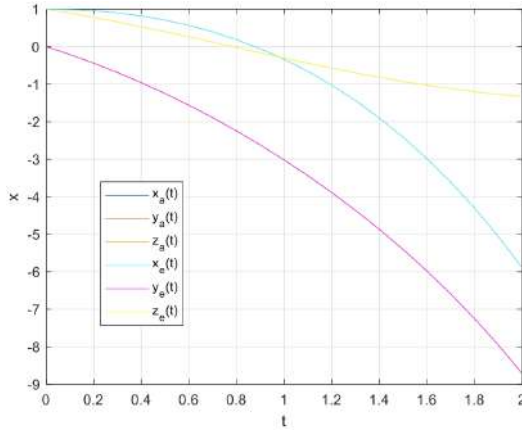


Figura 5.14. Comparación de la solución exacta con la aproximada

Calculamos el error:

```
>> Max_errorX=max(abs(xexacta-y(:,1)))
```

Max_errorX = 3.4091e-06

```
>> Max_errorY=max(abs(yexacta-y(:,2)))
```

Max_errorY = 7.8729e-06

```
>> Max_errorZ=max(abs(zexacta-y(:,3)))
```

Max_errorZ = 8.0947e-06

5.2.12. Problema 12

Uno de los modelos de propagación de virus está dado por el siguiente sistema de ecuación:

$$x'(t) = \frac{1}{10}(10 - x - xz)$$

$$y'(t) = \frac{1}{10}(xz - zy)$$

$$z'(t) = \frac{1}{10}(10y - 3z)$$

Con las condiciones iniciales $x(0) = 10$, $y(0) = 0$, $z(0) = 1$, donde x es la población sana; z población, afectada por el virus; y la población que está contagianda.

a) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema11(t,Sistema)

    x=Sistema(1);

    y=Sistema(2);

    z=Sistema(3);

    f=[1/10*(10-x-x.*z); 1/10*(x.*z-2*y); 1/10*(10**y-3*z)];

end
```

b) Resuelva el problema con $t \in [0,20]$ y pasos $h = 0.1$. Represente $x(t)$, $y(t)$ y $z(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{20-0}{0.1}$ obteniendo el número de iteraciones $N = 200$.

Llamamos al método y graficamos la solución:

```
>> [x,y]=RK_orden4_NM('Problema11',0,20,[10 0 1],200);

>> plot(x,y),grid on,xlabel('t'),ylabel('x')

>> plot(x,y),grid on,xlabel('t'),ylabel('x')

>> legend('x(t)','y(t)','z(t)')
```

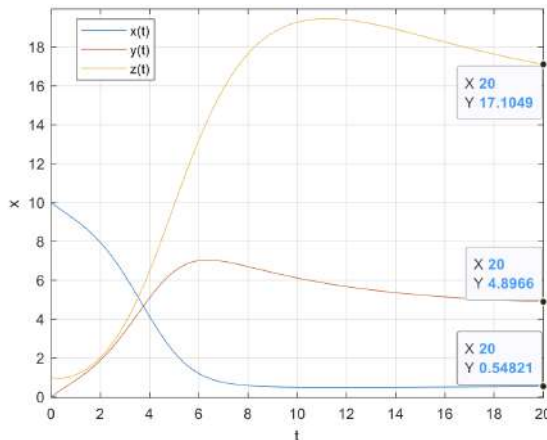


Figura 5.15. Representación de la solución $x(t)$, $y(t)$ y $z(t)$ con Runge-Kutta

c) Indicar los valores de $x(t)$ $y(t)$ y $z(t)$ para $t \in \{0, 4, 5, 11.3, 11.8, 20\}$.

Encontramos las posiciones requeridas:

$\begin{matrix} b-a & N+1 \\ \text{Posición} - a & x \end{matrix}$		
<ul style="list-style-type: none"> 0 $\begin{matrix} 20 & 201 \\ 0 & x \end{matrix} = 1$	<ul style="list-style-type: none"> 4 $\begin{matrix} 20 & 201 \\ 4 & x \end{matrix} = 40.2 \rightarrow 41$	<ul style="list-style-type: none"> 5 $\begin{matrix} 20 & 201 \\ 5 & x \end{matrix} = 50.25 \rightarrow 51$
<ul style="list-style-type: none"> 11.3 $\begin{matrix} 20 & 201 \\ 11.3 & x \end{matrix} = 113.56 \\ \rightarrow 114$	<ul style="list-style-type: none"> 11.8 $\begin{matrix} 20 & 201 \\ 11.8 & x \end{matrix} = 118.59 \\ \rightarrow 119$	<ul style="list-style-type: none"> 20 $\begin{matrix} 20 & 201 \\ 20 & x \end{matrix} = 201$

Ingresamos los códigos en Matlab:

```
>> T=[x(1) x(41) x(51) x(114) x(119) x(201)]
>> X=[y(1,1) y(41,1) y(51,1) y(114,1) y(119,1) y(201,1)]
>> Y=[y(1,2) y(41,2) y(51,2) y(114,2) y(119,2) y(201,2)]
>> Z=[y(1,3) y(41,3) y(51,3) y(114,3) y(119,3) y(201,3)]
```

t	0	4	5	11.3	11.8	20
$x(t)$	10	4.1324	2.2896	0.4901	0.4895	0.5482
$y(t)$	0	5.1011	6.4601	5.8156	5.7153	4.8966
$z(t)$	1	6.4971	9.8874	19.4574	19.4231	17.1049

5.2.13. Problema 13

El modelo de Lorenz-Haken para un láser de dos niveles, viene dado por el sistema de ecuaciones siguiente:

$$F'(t) = s(P - F)$$

$$P'(t) = -P + dF$$

$$d'(t) = b(r - d - FP)$$

Con las condiciones iniciales $F(0) = 4.5$, $P(0) = 4.5$, $d(0) = 1$.

a) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden.

```
function f = Problema12(t,Sistema)
```

```
F=Sistema(1);
```

```
P=Sistema(2);
```

```
d=Sistema(3);
```

```
s=3; b=1; r=17;
```

```
f=[s*(P-F); -P+d.*F; b*(r-d.*F)];
```

```
end
```

b) Resuelva el problema con $t \in [0, 2]$ y pasos $h = 0.1$. Represente $F(t)$, $P(t)$ y $d(t)$.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{2-0}{0.1}$ obteniendo el número de iteraciones $N = 20$.

Llamamos al método y graficamos la solución de las aproximaciones:

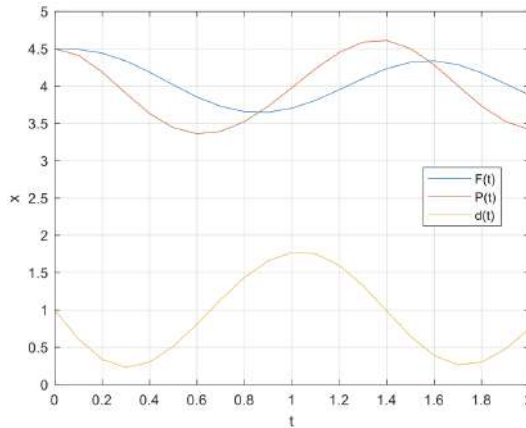


Figura 5.16. Representación de la solución $F(t)$, $P(t)$ y $d(t)$ con Runge-Kutta

c) Indicar los valores de $F(t)$, $P(t)$ y $d(t)$ para $\{0, 0.5, 1.5, 2\}$.

Encontramos las posiciones requeridas:

$$\frac{b-a}{\text{Posición} - a} = \frac{N+1}{x}$$

<p>• 0.5</p> $\frac{2}{0.5} = \frac{21}{x} = 5.25 \rightarrow 6$	<p>• 1.5</p> $\frac{2}{1.5} = \frac{21}{x} = 15.75 \rightarrow 16$	<p>• 2</p> $\frac{2}{2} = \frac{21}{x} = 21$
--	--	--

Ingresamos los códigos en Matlab:

```
>> T=[x(1) x(6) x(11) x(16) x(21)]
```

```
>> F=[y(1,1) y(6,1) y(11,1) y(16,1) y(21,1)]
```

```
>> P=[y(1,2) y(6,2) y(11,2) y(16,2) y(21,2)]
```

```
>> d=[y(1,3) y(6,3) y(11,3) y(16,3) y(21,3)]
```

t	0	0.5	1	1.5	2
$F(t)$	4.5000	4.0158	3.7044	4.3201	3.8842
$P(t)$	4.5000	3.4430	3.9791	4.5034	3.4191
$d(t)$	1.0000	0.5131	1.7710	0.6486	0.7439

5.2.14. Problema 14

El péndulo de Foucault: las ecuaciones empíricas del movimiento terrestre son:

$$x''(t) = 2\omega \sin(\phi) y' - k^2 x$$

$$y''(t) = -2\omega \sin(\phi) x' - k^2 y$$

Con las condiciones iniciales $x(0) = -6$, $x'(0) = 0$, $y(0) = 1$, $y'(0) = 0$.

Donde ω es la velocidad angular de rotación terrestres, $\phi = 40^\circ$ latitud del observados. $k^2 = \text{grav}/\text{long}$ del péndulo $I = 10$.

a) Transforme en un sistema de ecuaciones diferenciales de primer orden.

$$x = z_1$$

$$x' = z_1' = z_2$$

$$x'' = z_1'' = z_2'$$

$$x''(t) = 2\omega \sin(\phi) y' - k^2 x \rightarrow z_2' = 2\omega \sin(\phi) z_4 - k^2 z_1$$

$$y = z_3$$

$$y' = z_3' = z_4$$

$$y'' = z_3'' = z_4'$$

$$x''(t) = -2\omega \sin(\phi) x' - k^2 y \rightarrow z_4' = -2\omega \sin(\phi) z_2 - k^2 z_3$$

b) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden.

```
function f = Prob_Sys2_41(x,z)
w = 7.29e-5;    % velocidad angular terrestre (2*pi)/(24*3600)
k2 = 9.8/10;
fi = 40*pi/180; % latitud en radianes
f = [z(2); 2*w*sin(fi)*z(4)-k2*z(1); z(4); -2*w*sin(fi)*z(2)-k2*z(3)];
end
```

c) Resuelva el problema con $t \in [0, 3600]$ con un paso temporal de 0.5.

Calculamos el valor de N . Partiendo de $h = \frac{b-a}{N}$ y despejando, se obtiene $N = \frac{b-a}{h}$.

Donde $N = \frac{3600-0}{0.5}$ obteniendo el número de iteraciones $N = 7200$.

Llamamos al método y graficamos la solución:

```
>> [x,y]=RK_orden4_NM('Problema13',0,3600,[-6 0 1 0],7200);
>> plot(y(:,1),y(:,3)),grid on,xlabel('t'),ylabel('x')
```

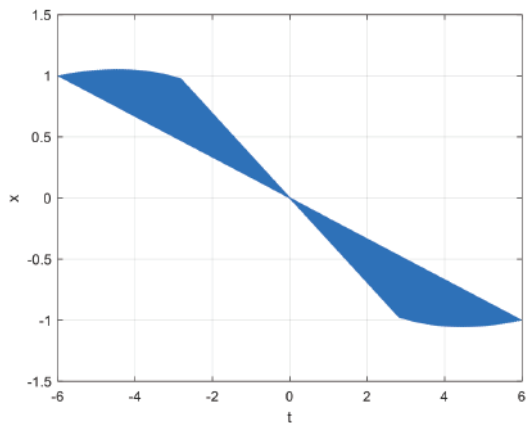


Figura 5.17. Representación de la solución de oscilación del plano (x,y)

d) Indicar los valores de $x(t)$ y $y(t)$ para $\{0,2000,3600\}$.

Encontramos las posiciones requeridas:

$$\text{Posición} = a + \frac{b-a}{N+1}x$$

<ul style="list-style-type: none"> 0 $\frac{3600 - 0}{7201 - 0}x = 1$	<ul style="list-style-type: none"> 2000 $\frac{3600 - 2000}{7201 - 2000}x = 4000.5$ $\rightarrow 4001$	<ul style="list-style-type: none"> 3600 $\frac{3600 - 3600}{7201 - 3600}x = 7201$
--	---	--

Ingresamos los códigos en Matlab:

```
>> T=[x(1) x(4001) x(7201)]
```

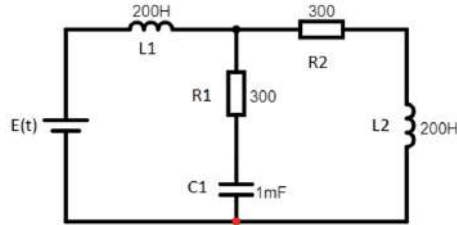
```
>> X=[y(1,1) y(4001,1) y(7201,1)]
```

```
>> Y=[y(1,3) y(4001,3) y(7201,3)]
```

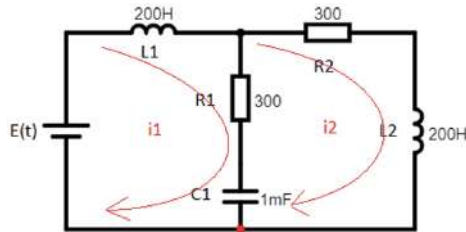
t	0	2000	3600
$x(t)$	-6.0000	-3.8708	-2.6196
$y(t)$	1.0000	1.0243	0.9077

5.2.15. Problema 15

El circuito que se presenta a continuación tiene las condiciones iniciales $h(0) = h'(0) = 0$, $i_2(0) = 0$, $i_2'(0) = 0$.



a) Determinar las ecuaciones diferenciales que gobiernan el circuito con $E(t) = 1v$.



Malla I

$$L_1 i_1' + R_1 (i_1 - i_2) + \frac{1}{C} \int_0^t (i_1(\tau) - i_2(\tau)) d\tau - E(t)$$

$$200i_1' + 300(i_1 - i_2) + 1000 \int_0^t (i_1(\tau) - i_2(\tau)) d\tau = E(t)$$

Derivamos:

$$200i_1'' + 300(i_1' - i_2') + 1000[i_1(t) - i_2(t)] = E'(t)$$

Malla II

$$\frac{1}{C} \int_0^t (i_1(\tau) - i_2(\tau)) d\tau + R_1 (i_2 - i_1) + R_2 i_2 + L_2 i_2' = 0$$

$$1000 \int_0^t (i_1(\tau) - i_2(\tau)) d\tau + 300(i_2 - i_1) + 300i_2 + 200i_2' = 0$$

Derivamos:

$$1000[i_2(t) - i_1(t)] + 300(i_2' - i_1') + 300i_2' + 200i_2'' = 0$$

Obtenemos las ecuaciones diferenciales que gobiernan el circuito:

$$i_1'' = \frac{E'(t)}{200} - \frac{3}{2}(i_1' - i_2') - 5(i_1 - i_2)$$

$$i_2'' = -\frac{3}{2}(i_2' - i_1') - \frac{3}{2}i_2' - 5(i_2 - i_1)$$

b) Escriba un sistema de ecuaciones diferenciales de primer orden.

$$i_1 = z_1$$

$$i_1' = z_1' = z_2$$

$$i_1'' = z_1'' = z_2'$$

$$i_1'' = \frac{E'(t)}{200} - \frac{3}{2}(i_1' - i_2') - 5(i_1 - i_2) \rightarrow z_2' = \frac{E'(t)}{200} - \frac{3}{2}(z_2 - z_4) - 5(z_1 - z_3)$$

$$i_2 = z_3$$

$$i_2' = z_3' = z_4$$

$$i_2'' = z_3'' = z_4'$$

$$i_2'' = -\frac{3}{2}(i_2' - i_1') - \frac{3}{2}i_2' + 5(i_2 - i_1) \rightarrow z_4' = -\frac{3}{2}(z_4 - z_2) - \frac{3}{2}z_4 - 5(z_3 - z_1)$$

c) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden; supóngase que la fuente de tensión es $E(t) = \sin(4t)$.

$$E'(t) = 4\cos(4t)$$

`function f = Problema14(x,z)`

$$dE = 4*\cos(4*x);$$

$$f = [z(2); dE/200+3/2*(z(4)-z(2))+5*(z(3)-z(1));...$$

$$z(4); 3/2*(z(2)-z(4))-3/2*z(4)+5*(z(1)-z(3))];$$

`end`

d) Resuelva el problema con $t \in [0,5]$ con un paso temporal de $N = 50$.

Llamamos al método para graficar la aproximación a la solución:

```
>> [x,y]=RK_orden4_NM('Prob_Sys2_43',0,5,[0 0 0 0],50);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('t')
```

>> hold on

>> plot(x,y(:,3))

>> legend('i1','y2')

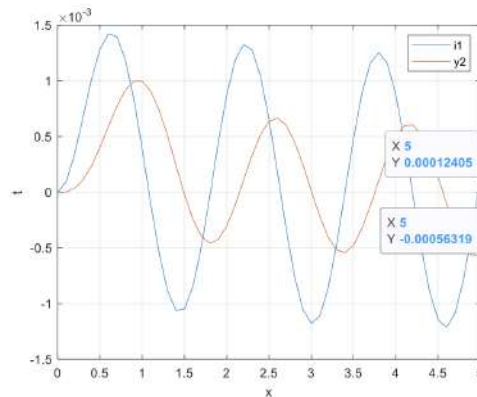
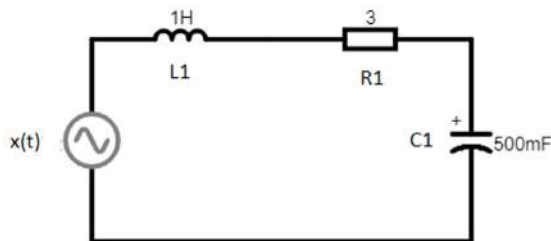


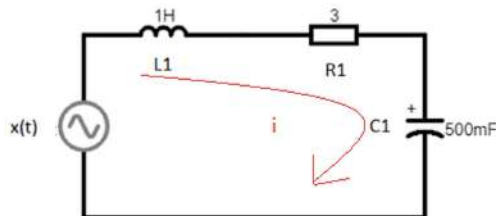
Figura 5.18. Representación de la solución $i1$, y $i2$ con Runge-Kutta

5.2.16. Problema 16

Para un circuito eléctrico mostrado en la figura, conocido también como un sistema LTIC (lineal continua invariante en el tiempo) con las condiciones iniciales $i(0) = 0$ $i'(0) = -5$:



a) Determinar las ecuaciones diferenciales que gobiernan el circuito.



Malla I

$$L_1 i_1' + R_1 i_1 + \frac{1}{c} \int_0^t i_1(\tau) d\tau = x(t)$$

$$i_1' + 3i_1 + 2 \int_0^t i_1(\tau) d\tau = x(t)$$

Derivamos:

$$i_1'' + 3i_1' + 2i_1(t) = x'(t)$$

b) Escriba un sistema de ecuaciones diferenciales de primer orden.

$$i_1 = z_1$$

$$i_1' = z_1' = z_2$$

$$i_1'' = z_1'' = z_2'$$

$$i_1'' = x'(t) - 3i_1' - 2i_1 \rightarrow z_2' = x'(t) - 3z_2 - 2z_1$$

c) Escriba una función que describa un sistema de ecuaciones diferenciales de primer orden. Supóngase que la fuente de tensión es $x(t) = \sin(4t)$.

$$x'(t) = 4\cos(4t)$$

`function f = Problema15(x,z)`

$$dx = 4\cos(4x);$$

$$f = [z(2); dx - 3*z(2) - 2*z(1)];$$

`end`

d) Considere como de interés el intervalo de tiempo entre $[0,6]$ con al menos 60 tramos. Y represente la solución.

```
>> [x,y]=RK_orden4_NM('Problema15r',0,6,[0 -5],60);
```

```
>> plot(x,y(:,1)),grid on,xlabel('x'),ylabel('t')
```

```
>> legend('i1')
```

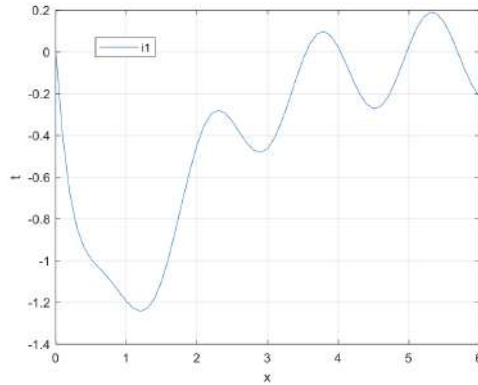


Figura 5.19. Representación de la solución $i1$ con Runge-Kutta

5.2.17. Problema 17

La temperatura $u(r)$ en un anillo circular de radio interior 1 y radio exterior 3 se define por:

$$ru'' + u' = 0, r \in [1,3], u(1) = 1, u'(1) = \frac{1}{2 \log\left(\frac{1}{3}\right)}$$

a) Calcule la solución del problema con veinte subintervalos. Cuál es la temperatura con $r = 2.4$. Represente gráficamente la temperatura en función de la distancia al origen.

Transformamos en sistema de ecuaciones diferenciales de primer orden:

$$u'' = -\frac{u'}{r}$$

$$u = z_1$$

$$u' = z_1' = z_2$$

$$u'' = z_1'' = z_2'$$

$$u'' = -\frac{u'}{r} \rightarrow z_2' = -\frac{z_2}{r}$$

Escribimos una función:

`function f = Problema16r(x,z)`

`r=2.4`

`f = [z(2);-z(2)/r];`

`end`

Llamamos al método para graficar la solución aproximada:

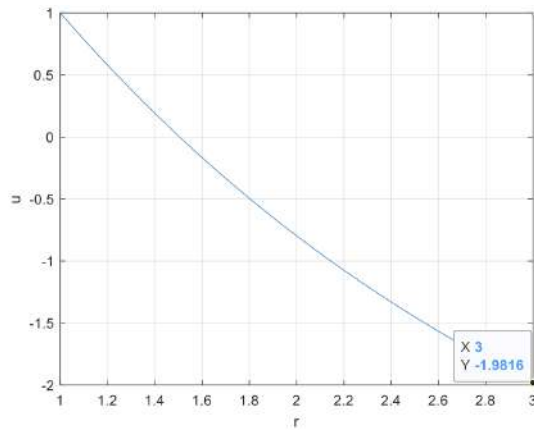


Figura 5.20. Representación de la solución u' con Runge-Kutta

BIBLIOGRAFÍA

- Agud, L. y Pla, M. L. (2015). *Matlab para matemáticas en ingenierías*. Universitat Politècnica de València.
- Bravo, J. L., Souto, A. y, Cantón, A. (2012). *Curso básico de programación en Matlab*. Tébar Flores.
- Burden, R.L. y Faires, J. D. (2011). *Numerical Analysis* (9.^a ed.). Brooks/Cole-CENGAGE learning.
- Chapra, S. (2007). *Métodos numéricos para ingenieros* (5.^a ed.). McGraw-Hill.
- Özis, T. y M.Ö. (2017). Derivation of Three-Derivative Runge-Kutta Methods. *Numerical Algorithms* 74, 247-265.
- Cordero, A., Hueso, J. L., Martínez, E. y Torregrosa, J. R. (2004). *Cálculo numérico*. Universitat Politècnica de València.
- . (2005). *Métodos numéricos con Matlab*. Universidad Politécnica de Valencia.
- . (2006). *Problemas resueltos de métodos numéricos*. Thomson.
- Cristina, M., & Fernández, C. (2013). *Manual básico de Matlab*.
- Gil, M. (2015). *Introducción rápida a Matlab y Simulink*. Díaz de Santos.
- Mathews, J. H., Fink, K. D. (2000). *Métodos numéricos con Matlab*. Prentice-Hall.
- Moreno, C. (2007). *Introducción al cálculo numérico*. UNED.
- Zill, D. G. (2016). *Advanced Engineering Mathematics*. Jones & Bartlett Learning.

Los métodos numéricos permiten obtener soluciones aproximadas a una serie de problemas. Antes de la revolución del transistor, que dio lugar a los circuitos integrados y, por ende, a las computadoras, los métodos numéricos requerían de un coste temporal tremendo para poder obtener las soluciones aproximadas de determinados problemas de la ciencia y la ingeniería sin conocer su solución analítica; sin embargo, en la segunda parte del siglo XX, los métodos numéricos tienen su mayor auge. El motivo fundamental es la aparición de las máquinas de computación. Si bien es cierto que no funcionaban con la velocidad que lo hace cualquier equipo doméstico de nuestros días, supusieron un gran salto cualitativo para desarrollar este tipo de técnicas.

Uno de los tipos de *software* comerciales que nos permite optimizar el tiempo que dedicamos a la resolución de problemas de forma numérica es Matlab. Originalmente, no era un lenguaje de programación, sino una calculadora matricial interactiva; de hecho, Matlab es un acrónimo de Matrix Laboratory (primera versión 1981). Pero la versión de la que disponemos hoy es mucho más avanzada. Cada semestre sale una versión en la que se mejora el código y se implementan nuevas funciones.

La parte teórica de este libro presenta los principales aspectos en los que se basan los métodos numéricos, para luego proceder a aplicarlos en el desarrollo de problemas, por lo que es una herramienta de trabajo de fácil entendimiento para los lectores.

A lo largo de este libro, se concentra la atención en los métodos numéricos de un paso (Euler, Heun, Runge Kutta). Estos métodos son fundamentales para el aprendizaje, de tal forma que se los puede emplear en el estudio de solución de ecuaciones diferenciales. A continuación, se hace un resumen de los contenidos de cada capítulo.

El capítulo I trata sobre los errores que se obtienen al resolver una ecuación diferencial al utilizar aproximaciones y las características de los problemas de valor inicial PVI.

