

MODELADO Y SIMULACIÓN DE SISTEMAS DINÁMICOS

Práctica 3: Métodos de Integración Numérica

Sebastián Giulianelli y María Agustina Torrano

Problema 1. Absorción de un Fármaco

1. Queremos obtener la solución analítica de la ecuación diferencial

$$\dot{x}(t) = -a \cdot x(t)$$

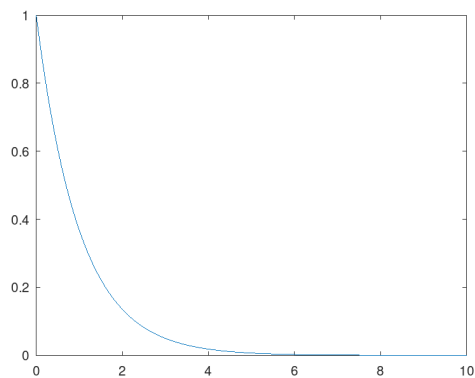
a partir de la condición inicial $x(0) = x_0$. Notemos que en este caso tenemos un sistema lineal, estacionario y autónomo, pues no hay una entrada externa. Luego, es equivalente a la ecuación $\dot{x}(t) = Ax(t)$ que presenta el apunte, con $A = -a$. Por lo tanto, la solución analítica es

$$x(t) = e^{-a \cdot t} \cdot x_0$$

2. La función hecha en Octave es la siguiente:

```
function x = solfarmaco(a,x0,t)
    x = exp(-a * t) * x0
end
```

3. Graficamos la solución analítica de la ecuación para $a = 1$ y $x_0 = 1$.



Problema 2. Solución analítica de sistemas LTI

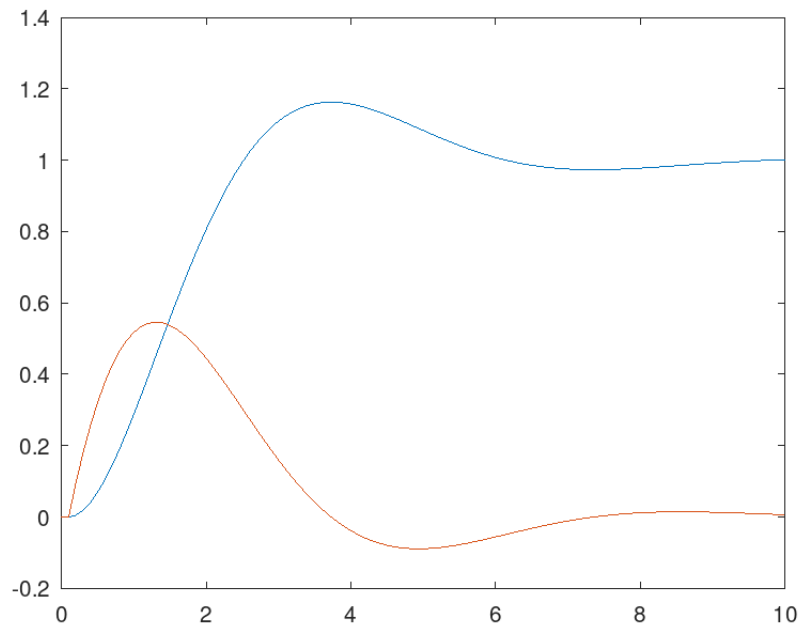
Escribiremos una función en Octave `ltiSolve(A,B,u,x0,t)` que permita obtener la solución analítica de un sistema LTI de la forma

$$\dot{x}(t) = Ax(t) + Bu$$

La función es la siguiente:

```
function x = ltiSolve(A,B,u,x0,t)
    x = zeros(length(x0),length(t))
    x(:,1) = x0
    I = eye(length(x0))
    iA = inv(A)
    for k = 1:length(t) - 1
        eAt = expm(A * t(k))
        x(:,k+1) = eAt * x0 + iA * (eAt - I) * B * u
    endfor
end
```

Obtenemos la siguiente simulación:



Problema 3. Sistema Masa-Resorte

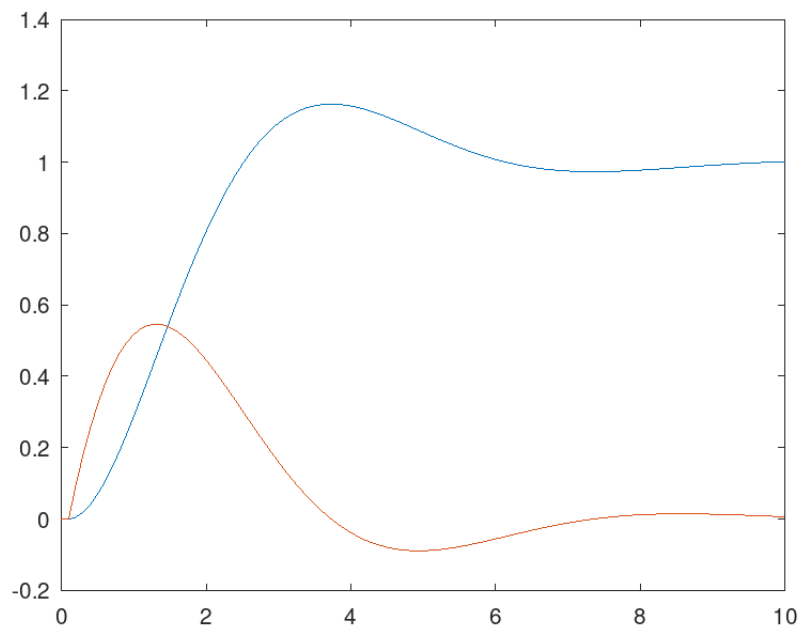
A partir del siguiente modelo

$$\begin{aligned}\dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\frac{k}{m}x_1(t) - \frac{b}{m}x_2(t) + F(t)/m\end{aligned}$$

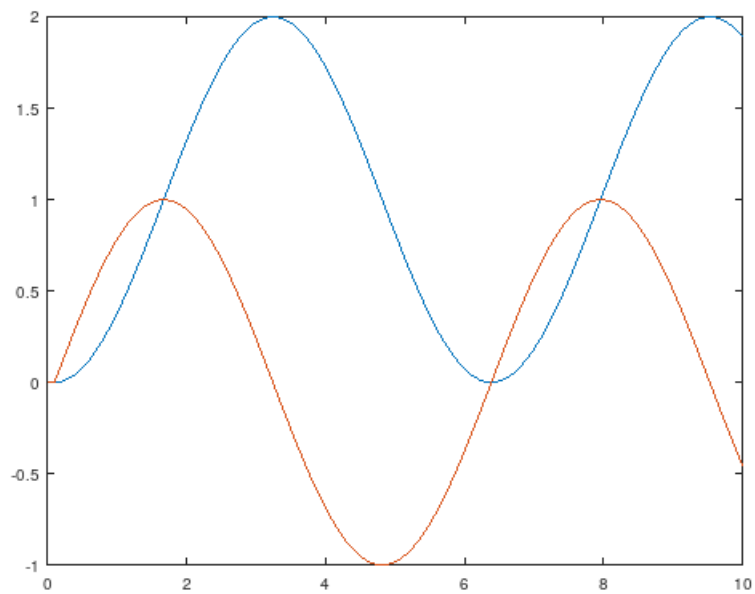
Obtenemos el siguiente sistema LTI:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ -\frac{1}{m} \end{bmatrix} F(t)$$

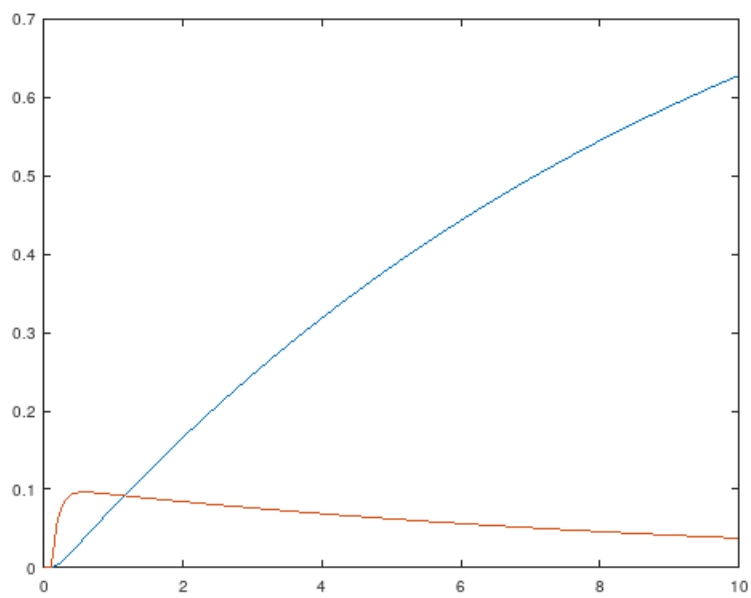
1. Simulamos para $m = k = b = 1$ y $F(t) = 1$.



2. Simulamos ahora para $b = 0$ y notamos que el método se mantiene inestable.



y con $b = 10$.

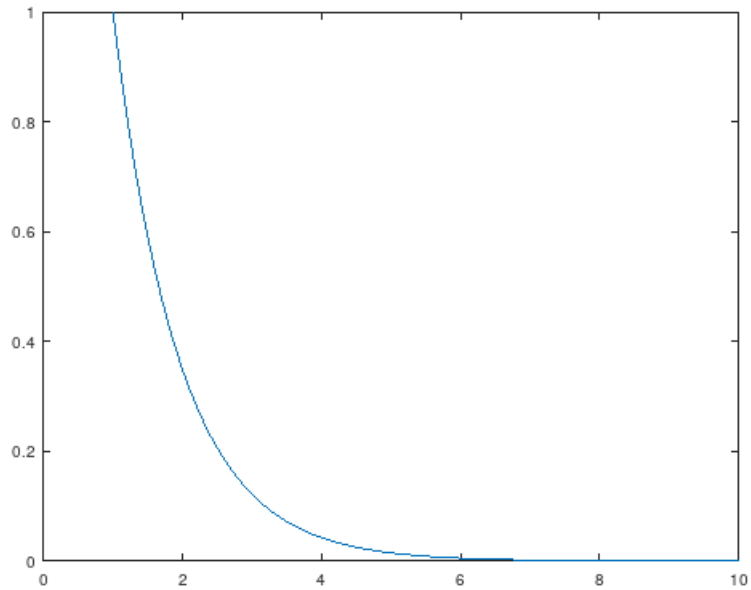


Problema 4. Método de Forward Euler

1. Para simular el modelo del Problema1 implementamos **f** como sigue.

```
function dx=farmaco(x,t)
    ra=1;
    dx=-ra*x;
end
```

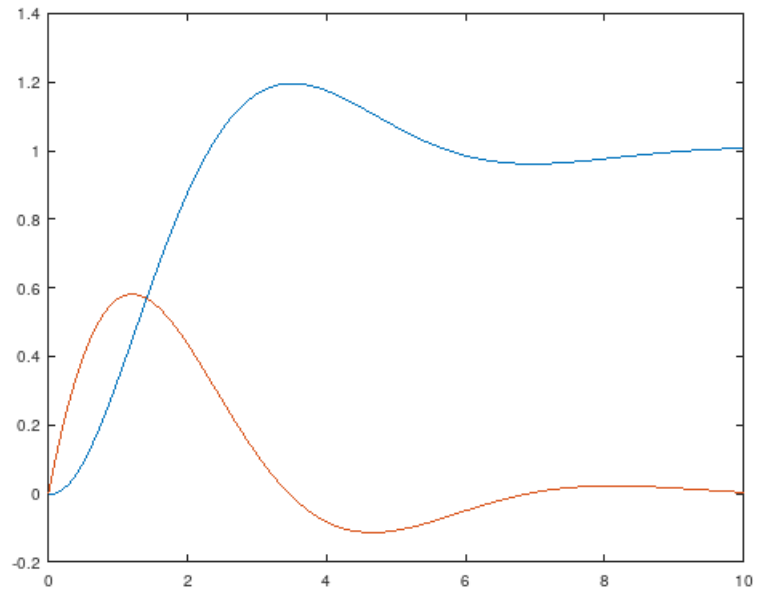
Luego realizamos la simulación.



2. Para simular el modelo del Problema2 implementamos **f** como sigue.

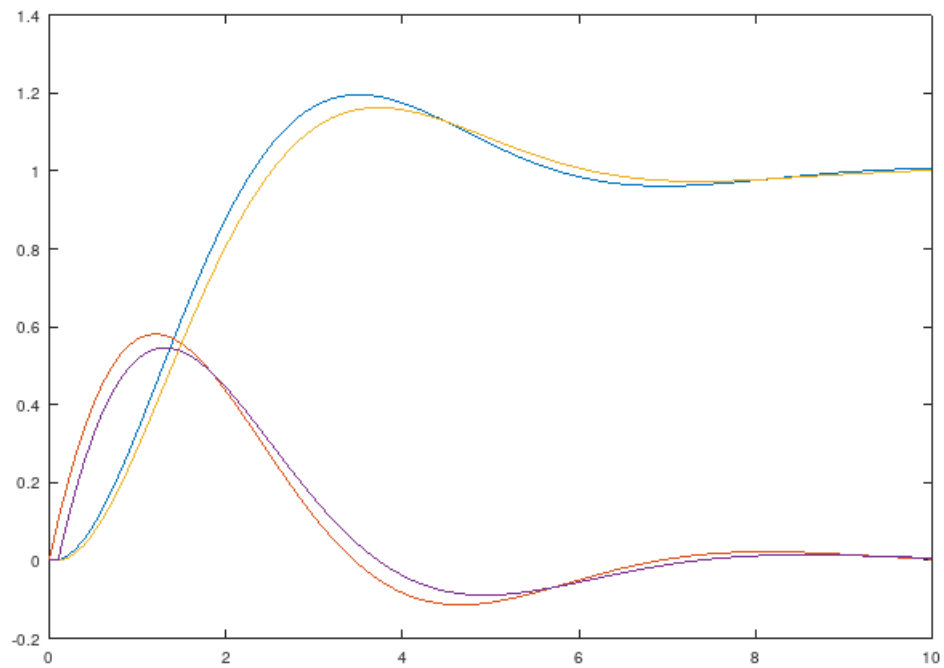
```
function dx=masares(x,t)
    k=1;
    m=1;
    b=1;
    F=1;
    dx=[x(2); 1/m*(-k*x(1)-b*x(2)+F)];
end
```

Luego, realizamos la simulación.



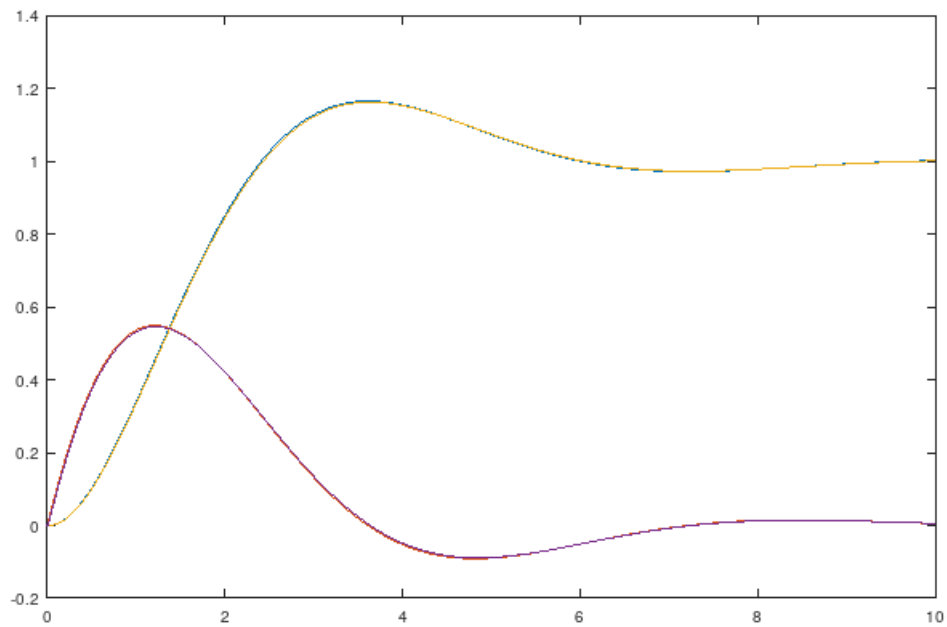
Problema 5. Precisión del Método de Forward Euler

Simulamos el sistema del problema 3 utilizando el método de Forward Euler con paso $h = 0.1$.



Luego, el error de primer paso calculado dió $9.8e - 03 \approx 0.1^2$ mientras que el error máximo 0.066139

Ahora simulamos con paso $h = 0.01$.



Luego, el error de primer paso calculado dió 0.000098 mientras que el error máximo 0.00614

Se puede observar que en ambos casos, el error global resulta proporcional al paso de integración h y que el error de primer paso es del orden de h^2 .

Problema 6. Estabilidad del Método de Forward Euler

1. Usando parámetros $m = k = b = F = 1$ y probando para distintos valores de h concluimos que para valores $h < 1$ el método de Forward Euler preserva la estabilidad numérica; para valores mayores se hace inestable.

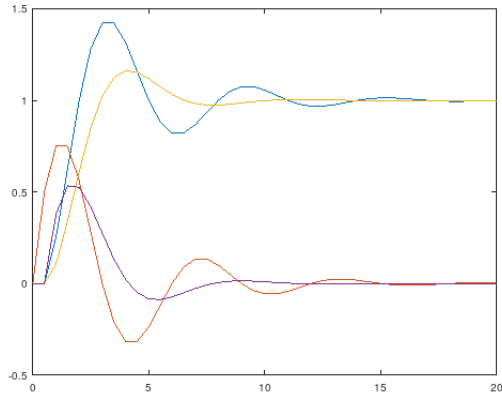


Figura 1: $h = 0.5$

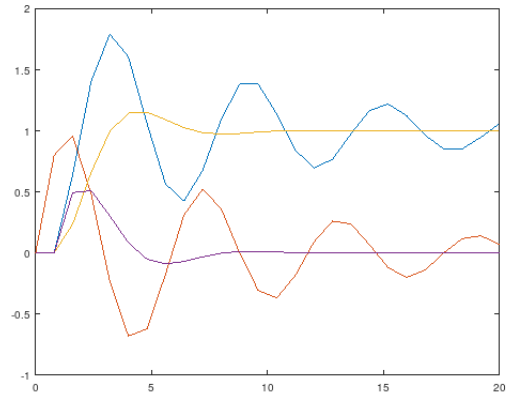


Figura 2: $h = 0.8$

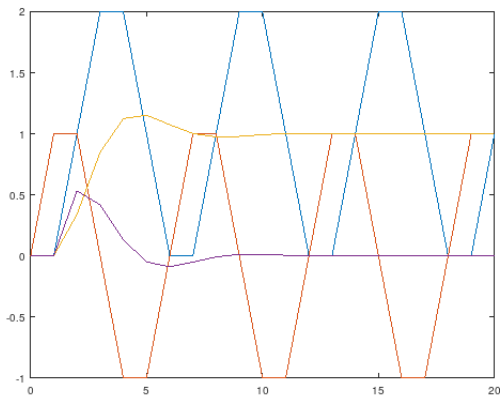


Figura 3: $h = 1$

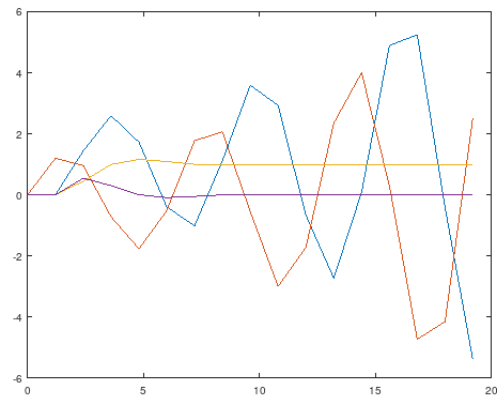


Figura 4: $h = 1.2$

2. Usando parámetros $m = k = F = 1$ y probando para distintos valores de h y b vimos que para $b = 0$, el método se mostraba inestable para la mayoría de tamaños probados y para $b = 10$ aquellos $h < 0.2$ el método de Forward Euler preserva la estabilidad numérica.

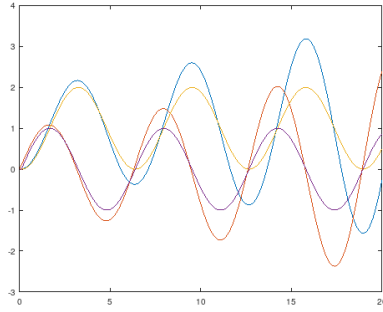


Figura 5: $h = 0.1, b = 0$

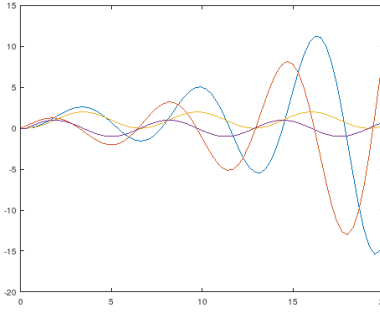


Figura 6: $h = 0.3, b = 0$

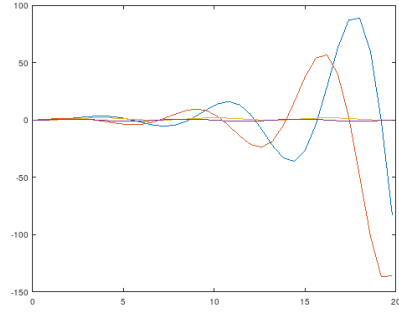


Figura 7: $h = 0.6, b = 0$

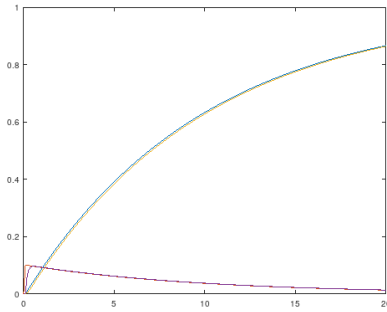


Figura 8: $h = 0.1, b = 10$

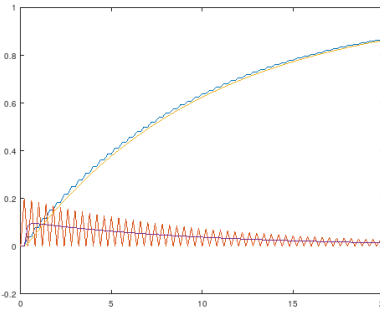


Figura 9: $h = 0.2, b = 10$

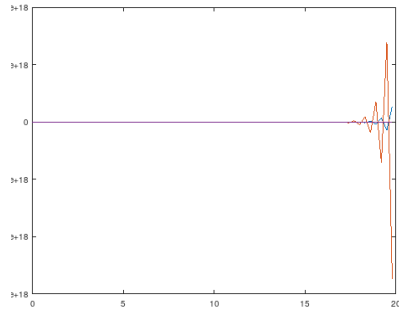


Figura 10: $h = 0.3, b = 10$

3. Para analizar de manera teórica los resultados obtenidos primero calculamos los autovalores de la matriz A y luego observamos la condición necesaria para la estabilidad numérica.

$$|\lambda h + 1| < 1$$

b	h	$ \lambda_1 h + 1 $	$ \lambda_2 h + 1 $
1	0.5	0.8660	0.8660
	0.8	0.9165	0.9165
	1.0	1.0000	1.0000
0	0.1	1.0050	1.0050
	0.3	1.0440	1.0440
	0.6	1.1662	1.1662
10	0.1	0.9899	0.0101
	0.2	0.9798	0.9798
	0.3	0.9697	1.9697

Problema 7. Método de Backward Euler

Análisis del Error

Para el error de primer paso (E_{pp}) y error máximo (E_m) de este método repetimos de manera análoga lo hecho con el método Forward Euler y obtuvimos:

h	E_{pp}	E_m
0.1	0.009	0.056
0.01	0.0001	0.00604

Se observa que en este caso, el error de primer paso se mantiene en el orden de h^2 y el error máximo se cercano a h .

Análisis de la estabilidad numérica

Para el análisis de la estabilidad numérica observamos que para todos los pasos h que tomamos, el método de Backward Euler se mantenía estable.

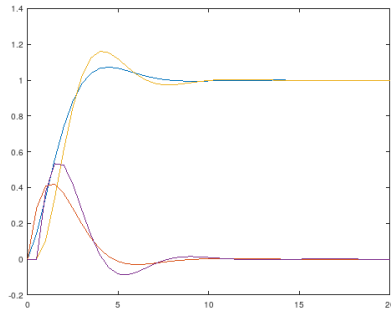


Figura 11: $h = 0.5, b = 1$

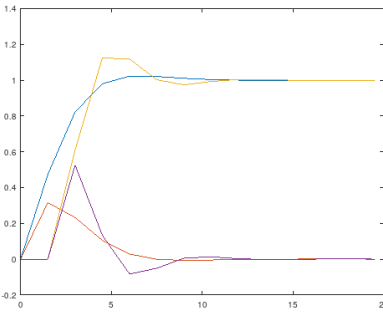


Figura 12: $h = 1.5, b = 1$

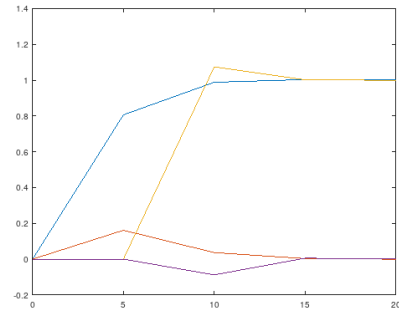


Figura 13: $h = 5, b = 1$

Y de la misma forma ocurre con $b = 0$ y $b = 10$.

Cuando analizamos esto teóricamente en base a la condición de estabilidad numérica para Backward Euler nos queda claro porqué ocurre esto.

$$\left| \frac{1}{1 - \lambda h} \right| < 1$$

b	h	$\left \frac{1}{1 - \lambda_1 h} \right $	$\left \frac{1}{1 - \lambda_2 h} \right $
1	0.5	0.7559	0.7559
	1.5	0.4588	0.4588
	5	0.1796	0.1796
0	0.5	0.8944	0.8944
	1.5	0.5547	0.5547
	5	0.1961	0.1961
10	0.5	0.9519	0.1681
	1.5	0.8684	0.0630
	5	0.6644	0.0198

Problema 8. Método de Heun

Análisis del Error

Para el error de primer paso (E_{pp}) y error máximo (E_m) de este método repetimos de manera análoga lo hecho con el método Backward Euler y obtuvimos:

h	E_{pp}	E_m
0.1	0.00017	0.0024
0.01	0.00000016	0.000023

Se observa que en este caso, el error es considerablemente inferior a los de los métodos de Euler. Esto ocurre debido a que el algoritmo de Heun realiza una evaluación adicional de la función f para calcular k_2 lo que deriva en una aproximación más precisa.

Análisis de la estabilidad numérica

Para el análisis de la estabilidad numérica observamos que no se mantiene la estabilidad para cualquier h como en Backward Euler para ningún $b \in \{0, 1, 10\}$.

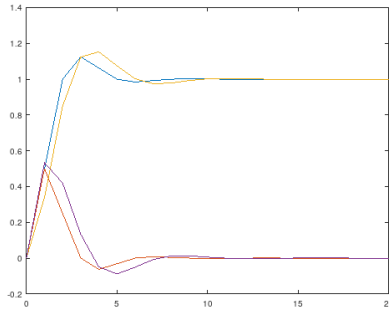


Figura 14: $h = 1, b = 1$

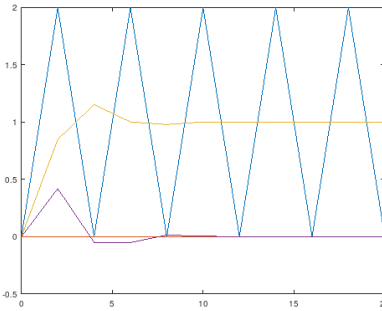


Figura 15: $h = 2, b = 1$

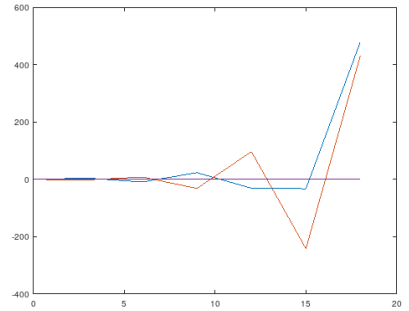


Figura 16: $h = 3, b = 1$

Ocorre algo similar para $b = 10$ y $b = 0$.

Cuando analizamos esto teóricamente en base a la condición de estabilidad numérica para Heun concluimos que lo observado es correcto para $b = 1$ y $b = 10$ pero para $b = 0$, no encontramos valores que validen la siguiente fórmula:

$$\text{cond}(\lambda) = |1 + \lambda h + \frac{h^2 \lambda^2}{2}| < 1$$

b	h	$\text{cond}(\lambda_1)$	$\text{cond}(\lambda_2)$
1	1	0.5	0.5
	2	1	1
	3	3.04	3.04
0	0.5	1.0078	1.0078
	0.1	1.0000	1.0000
	0.01	1.0000	1.0000
10	0.1	0.9899	0.5001
	0.2	0.9800	0.9800
	0.3	0.9702	2.4398

Problema 9. Regla Trapezoidal

Análisis del Error

Para el error de primer paso (E_{pp}) y error máximo (E_m) de este método repetimos de manera análoga lo hecho con el método Backward Euler y obtuvimos:

h	E_{pp}	E_m
0.1	0.00009	0.0014
0.01	0.000000084	0.000014

Se observa que en este caso, el error es inferior a los de los métodos de Euler y Heun. Esto ocurre debido a que el algoritmo de Regla Trapezoidal es un método implícito monopaso de orden mayor.

Análisis de la estabilidad numérica

Para el análisis de la estabilidad numérica observamos que ocurre algo muy similar a lo ocurrido con Backward Euler en donde el método se mantiene estable para cualquier h ingresado, salvo para $b = 0$, donde nuevamente no encontramos ningún h que verifique la condición teórica de estabilidad.

Problema 10. Método Explícito de Paso Variable

Simulamos el sistema masa resorte con el algoritmo rk23 usando $b = 1$.

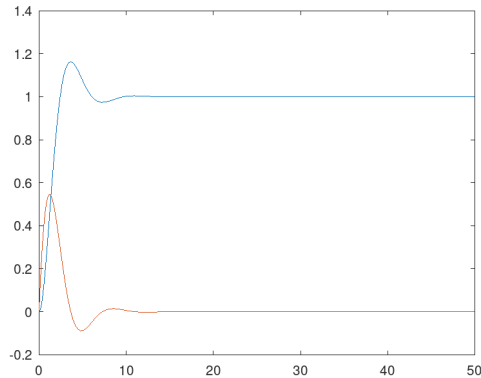


Figura 17: Simulación usando $b = 1$

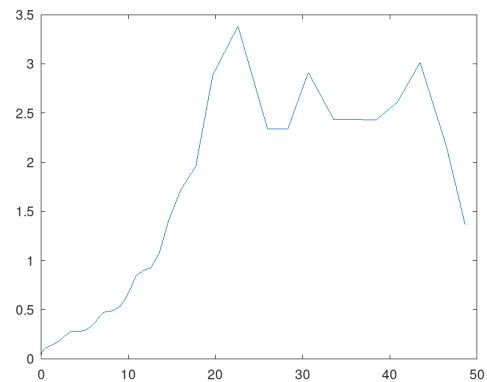


Figura 18: Evolución del tamaño del paso de integración h

Ahora, se corre la simulación con $b = 100$.

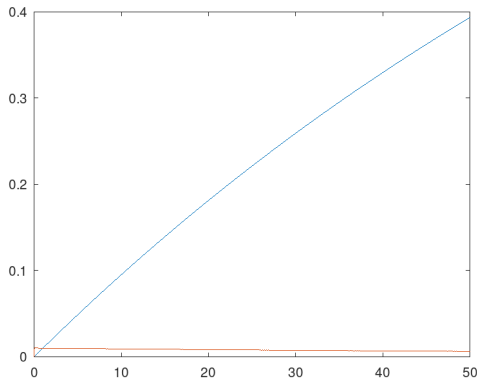


Figura 19: Simulación usando $b = 100$

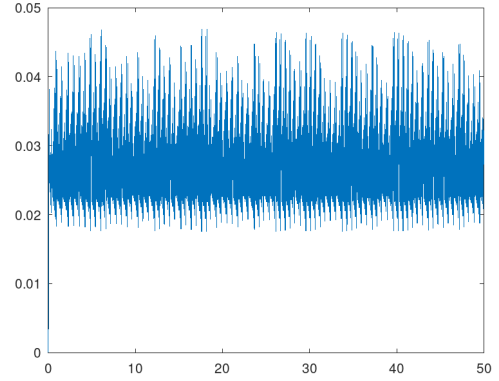


Figura 20: Evolución del tamaño del paso de integración h

Se puede observar que para $b = 1$, el sistema presenta oscilaciones amortiguadas suaves, y el paso de integración puede crecer y adaptarse sin dificultad. Por otro lado, para $b = 100$, el sistema tiene cambios muy rápidos al principio y luego se estabiliza. Esto hace que el algoritmo use pasos de integración más pequeños para mantener el error bajo control.

Problema 11. Método Implícito de Paso Variable

Simulamos el sistema masa resorte con el algoritmo dado usando $b = 0$.

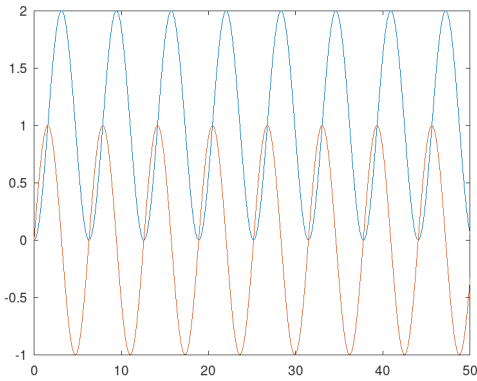


Figura 21: Simulación usando $b = 0$

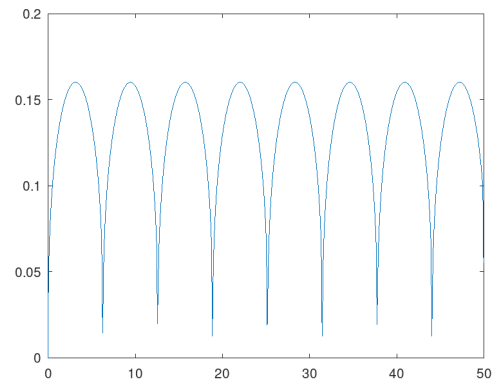


Figura 22: Evolución del tamaño del paso de integración h

Ahora, se corre la simulación con $b = 1$.

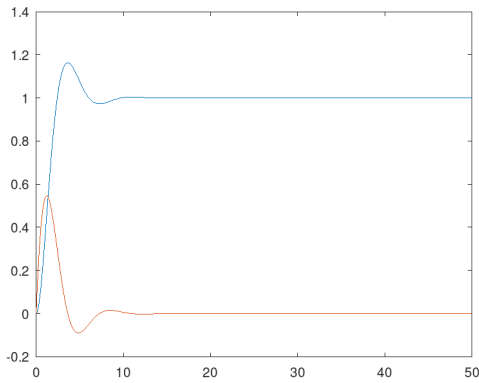


Figura 23: Simulación usando $b = 1$

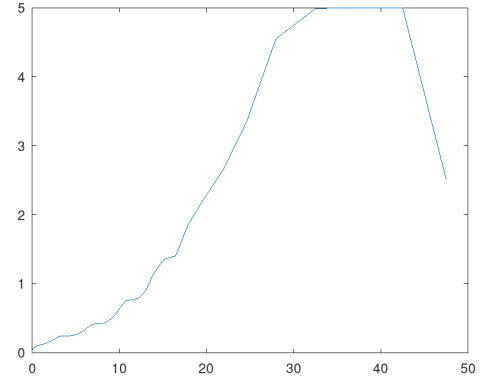


Figura 24: Evolución del tamaño del paso de integración h

Finalmente, realizamos la simulación con $b = 100$.

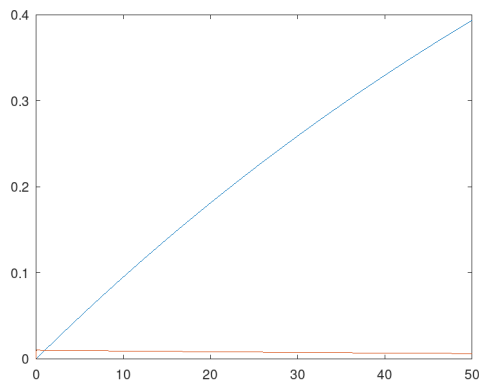


Figura 25: Simulación usando $b = 100$

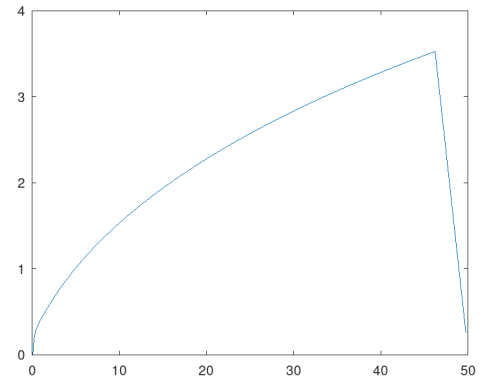


Figura 26: Evolución del tamaño del paso de integración h

En primer lugar, el sistema Masa Resorte con $b = 0$ es marginalmente estable, por lo que el sistema oscila indefinidamente. Debido a la alta frecuencia de las oscilaciones, el método debe usar pasos pequeños para mantener la precisión, especialmente cerca de máximos y mínimos.

En el caso de $b = 1$, las oscilaciones disminuyen y el método puede utilizar pasos de mayor tamaño. La estabilidad del método implícito permite una integración más eficiente que en el caso anterior.

Finalmente, con $b = 100$, el sistema se vuelve rígido. Sin embargo, el método implícito maneja bien esta rigidez y permite usar pasos grandes.