Análisis Numérico para Ingeniería

Clase Nro. 4

"Ver lo que está delante de nuestros ojos requiere un esfuerzo constante."

Eric Arthur Blair

Temas a tratar en esta clase:

- Sistemas de EDOs de 1er. Orden.
- Transformación de EDOs de orden n.
- Resolución de Sistemas de 1er. Orden.
- Estrategias de ajuste del paso h.

Sistemas de EDOs de 1er. orden

- Infinidad de problemas de diversas áreas del conocimiento como la *física*, la *química*, la *biología*, la *economía*, la *ingeniería*, etc. se expresan a menudo en términos de ecuaciones diferenciales ordinarias.
- Estas ecuaciones pueden estar relacionadas entre sí, por lo que conforman sistemas de ecuaciones diferenciales.
- Las ecuaciones diferenciales de **orden n**, pueden transformarse en sistemas de ecuaciones de **primer orden**, para su resolución.

Ejemplo de Sistemas de EDOs

El modelo SIR, desarrollado por el químico escocés William Ogilvy Kernack y el matemático A.G. McKendrick, modela la evolución de una enfermedad infecto-contagiosa.

El modelo divide a la población en individuos suceptibles de ser infectados, individuos infectados e individuos recuperados de la enfermedad.



MODELO SIR

$$\frac{dS}{dt} = -\beta \cdot S(t) \cdot I(t)$$

$$\frac{dI}{dt} = \beta \cdot S(t) \cdot I(t) - \gamma \cdot I(t)$$

$$\frac{dR}{dt} = \gamma \cdot I(t)$$

Ejemplo de Sistemas de EDOs

En el año 1978 en la revista científica British Medical Journal se publicaron los datos de un brote de gripe en un internado del norte de Inglaterra que se extendio del 22 de enero al 4 de febrero, infectando a 512 de las 763 personas que allí estudiaban.

Debido a que en este caso concreto se cuenta con datos reales y la cantidad de individuos permanece constante, es un escenario ideal para corroborar la exactitud del modelo SIR para predecir la evolución de la infección.

Precisamente para este caso en particular se han determinado los siguientes valores: $\gamma = 0.4477$ y $\beta = 0.0022$.

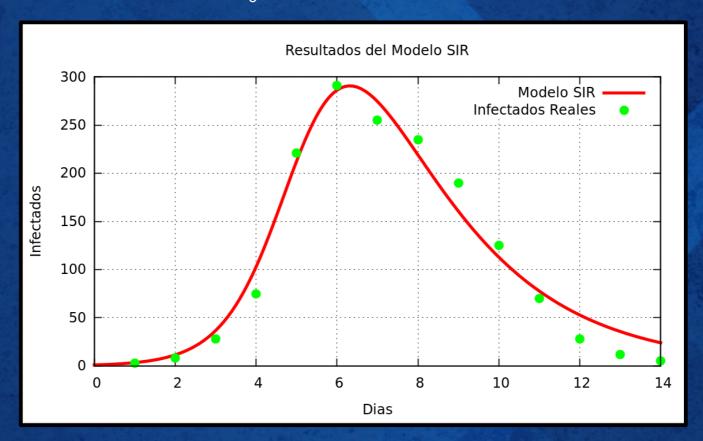
Ejemplo de Sistemas de EDOs

En este caso los valores iniciales son:

Individuos Suceptibles: $S_0 = 762$,

Individuos Infectados: $I_0 = 1$,

Individuos Recuperados: $R_0 = 0$



Modelo SEIR

El modelo **SEIR**, es una adaptación del modelo **SIR** explicado anteriormente

El modelo divide a la población en individuos suceptibles de ser infectados, individuos expuestos, individuos infectados e individuos recuperados de la enfermedad.



MODELO SEIR

$$\frac{dS}{dt} = -\beta \cdot S(t) \cdot I(t) / N$$

$$\frac{dE}{dt} = \beta \cdot S(t) \cdot I(t) / N - \sigma \cdot E(t)$$

$$\frac{dI}{dt} = \sigma \cdot E(t) - \gamma \cdot I(t)$$

$$\frac{dR}{dt} = \gamma \cdot I(t)$$

Modelo SEIR para el COVID-19

Para los que deseen saber más sobre el tema y quieran experimentar con estos modelos, les dejo el enlace a este Blog.

BLOG DEL INSTITUTO DE MATEMÁTICAS DE LA UNIVERSIDAD DE SEVILLA

EMERGENCIA COVID-19

Terra incognita

Divertimentos con delantal

Los monos de Kubrick

Fondo

Quiénes somos

Análisis del Covid-19 por medio de un modelo SEIR

② 20 marzo, 2020 🛔 José Manuel Gutiérrez y Juan Luis Varona 🗁 EMERGENCIA COVID-19 🔘 6

En matemáticas, modelizar es intentar extraer los aspectos significativamente importantes de una situación real y plasmarlos en forma de expresiones y ecuaciones matemáticas. Salvo en procesos muy sencillos, lo que se obtenga nunca va a ser exacto, porque el modelo, necesariamente, no puede captar toda la realidad ni todas las variables involucradas, ni incorporar todos los datos iniciales de los que se parte. Pero sí que puede servir para estudiar el proceso y prever, de alguna forma, cómo la introducción de medidas concretas puede hacer que su evolución futura varíe.

En particular, es lo que ocurre con el desarrollo de las epidemias, y es lo que vamos a intentar hacer aquí mediante el modelo epidemiológico SEIR, bien adaptado a la epidemia de coronavirus COVID-19. Para su modelización, incorporaremos variaciones para poder tener en cuenta las medidas de contención que se adoptan. No intentaremos hacer una predicción real con números concretos (en particular, para ello necesitaríamos más datos médicos y políticos de los que nosotros disponemos), sino mostrar la influencia de las medidas en el desarrollo de la epidemia. Sólo queremos mostrar las tendencias cualitativas —con una explicación más o menos técnica de las herramientas matemáticas que se usan en los cálculos y las incertidumbres inherentes al proceso, las mismas que se encuentran todos los epidemiólogos—, no dar previsiones cuantitativas de cómo va a ser el futuro de la epidemia. (Esta entrada es una extracto de un artículo más extenso que se puede consultar completo aquí).

EDOs de orden n

Una **EDO** de **orden n** puede **transformarse** en un **sistema** de **n** ecuaciones de **primer orden**.

$$\frac{d^{n} y}{dx^{n}} = f(x, y, \frac{dy}{dx}, \frac{d^{(2)} y}{dx^{2}}, \dots, \frac{d^{(n-1)} y}{dx^{n-1}})$$

CONDICIONES INICIALES

Algoritmo de transformación

Las **EDOs** de **orden n** se **transforman** en **sistemas** de **n** ecuaciones de **primer orden** para poder **resolverlas** con los **métodos** ya vistos.

$$\frac{d^{n} y}{dx^{n}} = f(x, y, \frac{dy}{dx}, \frac{d^{(2)} y}{dx^{2}}, \dots, \frac{d^{(n-1)} y}{dx^{n-1}}) = f(x, y, p, q, \dots, w)$$

$$\frac{dy}{dx} = y' = \mathbf{p}$$

$$\frac{d^{(2)}y}{dx^2} = y'' = \mathbf{p}' = \mathbf{q}$$

 $\frac{d^{(n-1)}y}{dx^{n-1}}=z'=w$

Sustituimos cada derivada por una nueva variable.

Ejemplo (I)

Ecuación diferencial de orden 3

$$y'''+x\cdot y''-x\cdot y'-2\cdot y=x$$

 $x_0=0$; $y_{x_0}=y''_{x_0}=0$; $y'_{x_0}=1$

Sistema de 3 ecuaciones diferenciales de 1er. orden

$$y' = p$$

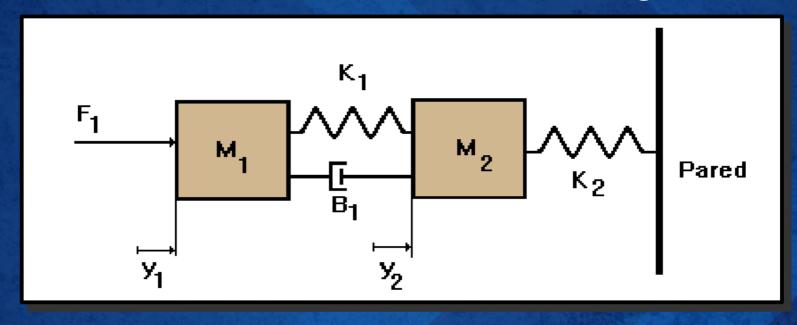
$$p' = q$$

$$q' = x - x \cdot q + x \cdot p + 2 \cdot y$$

Sustituimos cada derivada por una nueva variable.

Resolución completa de un problema

Sistema Masa – Resorte - Amortiguador



Sistema de 2 ecuaciones diferenciales de orden 2

$$\begin{aligned} &M_{1}\cdot y\,{'}\,{'}_{1}+B_{1}\cdot y\,{'}_{1}+k_{1}\cdot y_{1}-B_{1}\cdot y\,{'}_{2}-k_{2}\cdot y_{2}=F_{1}(t)\\ &-B_{1}\cdot y\,{'}_{1}-k_{1}\cdot y_{1}+M_{2}\cdot y\,{'}\,{'}_{2}+B_{1}\cdot y\,{'}_{2}+(k_{1}+k_{2})\cdot y_{2}=0 \end{aligned}$$

Transformación del Ejemplo (II)

Sistema de 2 ecuaciones diferenciales de orden 2

$$M_{1} \cdot y''_{1} + B_{1} \cdot y'_{1} + k_{1} \cdot y_{1} - B_{1} \cdot y'_{2} - k_{2} \cdot y_{2} = F_{1}(t)$$

$$-B_{1} \cdot y'_{1} - k_{1} \cdot y_{1} + M_{2} \cdot y''_{2} + B_{1} \cdot y'_{2} + (k_{1} + k_{2}) \cdot y_{2} = 0$$

Sistema de 4 ecuaciones diferenciales de 1er. orden

$$y_{1}' = v_{1}$$

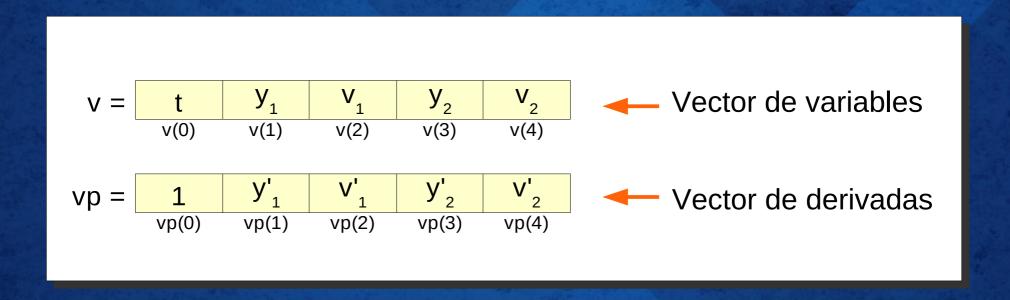
$$v_{1}' = \frac{F_{1}(t) - B_{1} \cdot v_{1} + B_{1} \cdot v_{2} - k_{1} \cdot y_{1} + k_{2} \cdot y_{2}}{M_{1}}$$

$$y_{2}' = v_{2}$$

$$v_{2}' = \frac{B_{1} \cdot v_{1} + k_{1} \cdot y_{1} - B_{1} \cdot v_{2} - (k_{1} + k_{2}) \cdot y_{2}}{M_{2}}$$

Implementación con vectores

La utilización de vectores simplifica notablemente la implementación de los métodos para resolver sistemas de ecuaciones diferenciales.



Implementación con vectores (II)

$$y_{1}' = v_{1}$$

$$v_{1}' = \frac{F_{1}(t) - B_{1} \cdot v_{1} + B_{1} \cdot v_{2} - k_{1} \cdot y_{1} + k_{2} \cdot y_{2}}{M_{1}}$$

$$y_{2}' = v_{2}$$

$$v_{2}' = \frac{B_{1} \cdot v_{1} + k_{1} \cdot y_{1} - B_{1} \cdot v_{2} - (k_{1} + k_{2}) \cdot y_{2}}{M_{2}}$$

$$v = \underbrace{\begin{array}{c|ccccc} t & y_1 & v_1 & y_2 & v_2 \\ v(0) & v(1) & v(2) & v(3) & v(4) \end{array}}_{v(0)} \quad \text{Vector de variables}$$

$$vp = \underbrace{\begin{array}{c|ccccc} 1 & y'_1 & v'_1 & y'_2 & v'_2 \\ vp(0) & vp(1) & vp(2) & vp(3) & vp(4) \end{array}}_{vp(4)} \quad \text{Vector de derivadas}$$

Implementación con vectores (III)

$$vp(1) = v(2)$$

$$vp(2) = \frac{F_1(v(0)) - B_1 \cdot v(2) + B_1 \cdot v(4) - k_1 \cdot v(1) + k_2 \cdot v(3)}{M_1}$$

$$vp(3) = v(4)$$

$$vp(4) = \frac{B_1 \cdot v(2) + k_1 \cdot v(1) - B_1 \cdot v(4) - (k_1 + k_2) \cdot v(3)}{M_2}$$

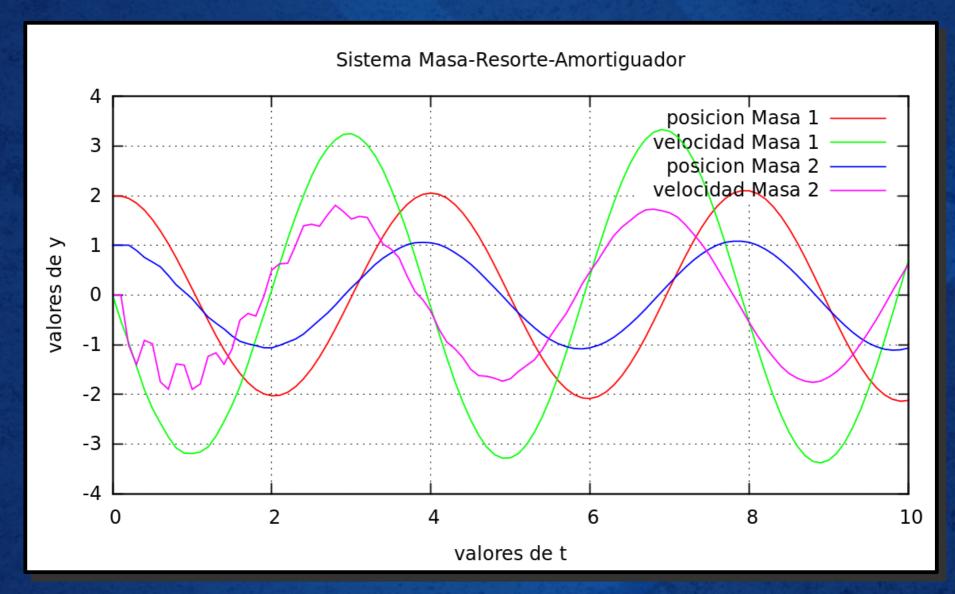
Definición de la Función Derivada

```
MODULE sistema MRA
! Cantidad de Ecuaciones del Sistema
INTEGER, PARAMETER :: cant ec = 4
CONTAINS
FUNCTION v prima(v)
! Definición del Sistema de Ecuaciones Diferenciales
REAL(8), PARAMETER :: m1=1.0, m2=0.05, k1=5.0, k2=5.0, b1=1.0
REAL(8), DIMENSION(0:cant_ec) :: v, v_prima
v_{prima}(0) = 1.0
v_prima(1) = v(2)
v_prima(2) = (-b1*v(2)+b1*v(4)-k1*v(1)+k2*v(3))/m1
v prima(3) = v(4)
v_{prima}(4) = (-b1*v(4)+b1*v(2)-k1*v(3)+k1*v(1)-k2*v(3))/m2
END FUNCTION v prima
END MODULE
```

Euler Simple (Implementación)

```
SUBROUTINE EulerSimple(vi, cant ec, max iter, h, formato)
!Metodo de Euler Simple
INTEGER, INTENT(IN) :: cant_ec, max_iter
REAL(8), INTENT(IN), DIMENSION(0:cant_ec) :: vi
REAL(8), INTENT(IN) :: h
CHAR*15, INTENT(IN) :: formato
INTEGER iter
REAL(8), DIMENSION(0:cant ec) :: v
 WRITE (*, formato) 0, vi
 v = vi
 DO iter = 1, max iter
   v = v + h*v prima(v)
   WRITE (*, formato) iter, v
 END DO
END SUBROUTINE EulerSimple
```

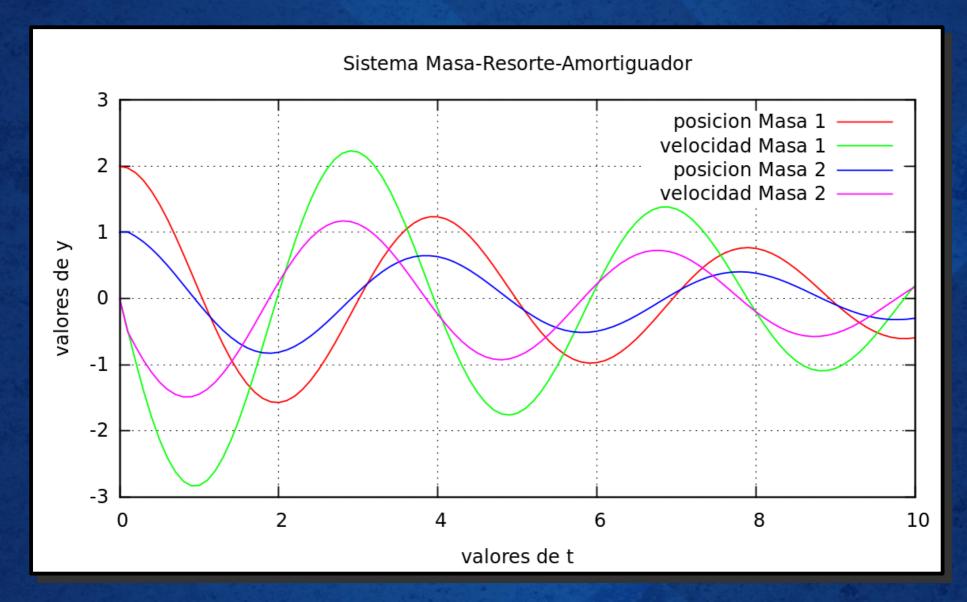
Solución Euler Simple (h=0.1)



Euler Modificado (Implementación)

```
SUBROUTINE EulerModificado(vi, cant ec, max iter, h, formato)
!Metodo de Euler Modificado
INTEGER, INTENT(IN) :: cant_ec, max_iter
REAL(8), INTENT(IN), DIMENSION(0:cant ec) :: vi
REAL(8), INTENT(IN) :: h
CHAR*15, INTENT(IN) :: formato
INTEGER iter
REAL(8), DIMENSION(0:cant ec) :: v
WRITE (*, formato) 0, vi
v = vi
 DO iter = 1, max iter
   vp = v prima(v)
   v = v + h*(vp + v prima(v + h*vp))/2.0
   WRITE (*, '(15, formato) iter, v
 END DO
END SUBROUTINE EulerModificado
```

Solución Euler Modificado (h=0.1)

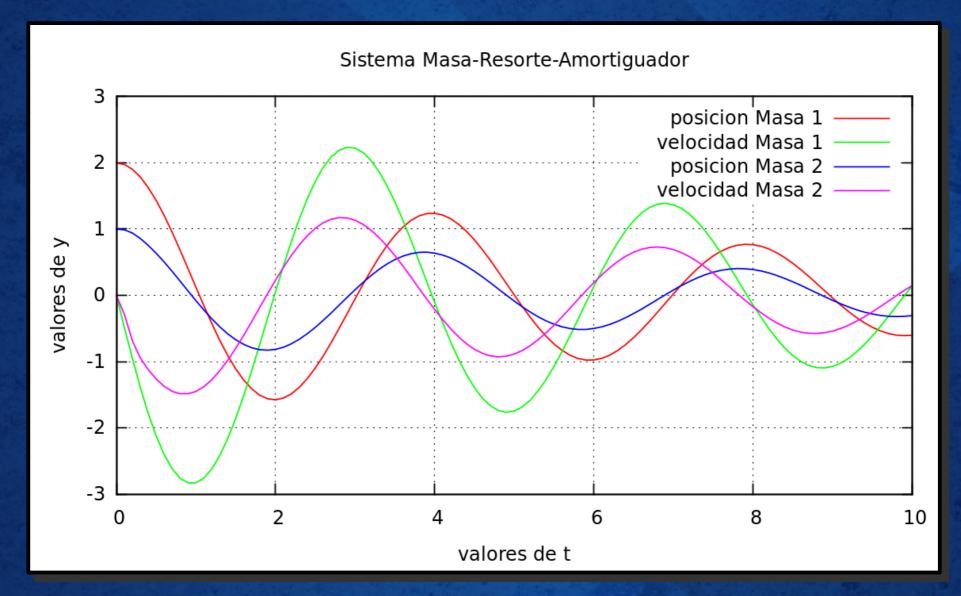


Runge-Kutta

(4to. Orden)

```
SUBROUTINE RungeKutta(vi, cant_ec, max_iter, h, formato)
!Metodo de Runge-Kutta (de 4to. orden)
REAL(8), INTENT(IN), DIMENSION(0:cant_ec) :: vi
REAL(8), INTENT(IN) :: h
INTEGER, INTENT(IN) :: cant ec, max iter
CHAR*15, INTENT(IN) :: formato
INTEGER iter
REAL(8), DIMENSION(0:cant_ec) :: v, k1, k2, k3, k4
 WRITE (*, formato) 0, vi
 v = vi
 DO iter = 1, max iter
  k1 = h*v prima(v)
  k2 = h*v prima(v+k1/2.0)
  k3 = h*v prima(v+k2/2.0)
  k4 = h*v prima(v+k3)
  v = v + (k1 + 2.0*k2 + 2.0*k3 + k4)/6.0
  WRITE (*, formato) iter, v
 END DO
END SUBROUTINE RungeKutta
```

Solución Runge-Kutta 4to. Orden (h=0.1)

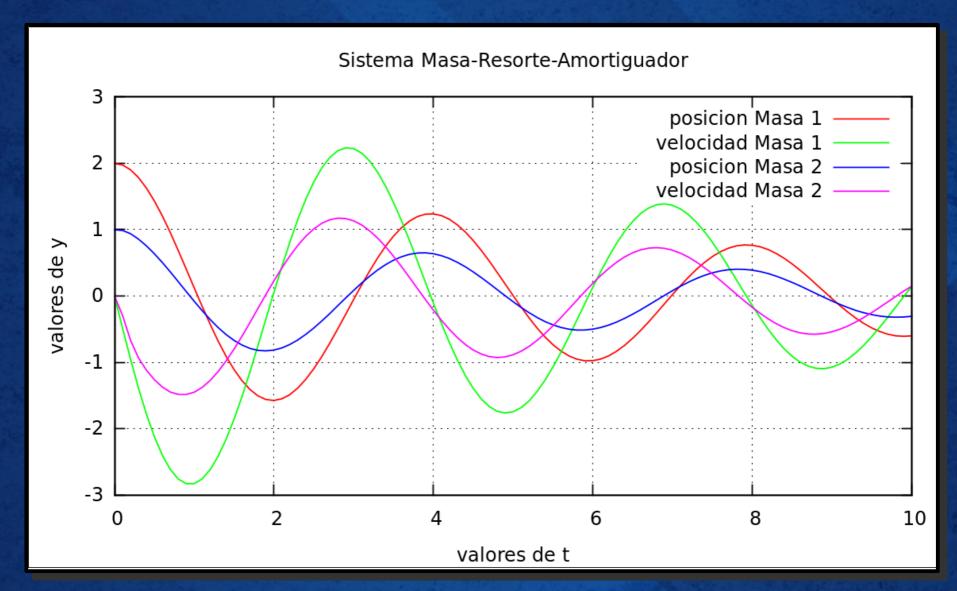


Runge-Kutta-Fehlberg

```
SUBROUTINE RungeKuttaFehlberg(vi, cant ec, max iter, h, formato)
!Metodo de Runge-Kutta-Fehlberg (de 6to. orden)
REAL(8), INTENT(IN), DIMENSION(0:cant_ec) :: vi
REAL(8), INTENT(IN) :: h
INTEGER, INTENT(IN) :: cant_ec, max_iter
CHAR*15, INTENT(IN) :: formato
INTEGER iter
REAL(8), DIMENSION(0:cant ec) :: v, k1, k2, k3, k4, k5, k6, e
 WRITE (*, formato) 0, vi, 0
 v = vi
 DO iter = 1, max iter
  k1 = h^*v prima(v)
  k2 = h*v prima(v + k1/4.0)
  k3 = h*v prima(v + (3.0*k1 + 9.0*k2)/32.0)
  k4 = h*v prima(v + (1932.0*k1 - 7200.0*k2 + 7296.0*k3)/2197.0)
  k5 = h^*v \text{ prima}(v + 439.0*k1/216.0 - 8.0*k2 + 3680.0*k3/513.0 - 845.0*k4/4104.0)
  k6 = h^*v \text{ prima}(v - 8.0*k1/27.0 + 2.0*k2 - 3544.0*k3/2565.0 + 1859.0*k4/4104.0 - 11.0*k5/40.0)
  v = v + (25.0 k1/216.0 + 1408.0 k3/2565.0 + 2197.0 k4/4104.0 - k5/5.0)
  e = k1/360.0 - 128.0*k3/4275.0 - 2197.0*k4/75240.0 + k5/50.0 + 2.0*k6/55.0
  WRITE (*, formato) iter, v, e
 END DO
END SUBROUTINE RungeKuttaFehlberg
```

Runge-Kutta-Fehlberg

$$(h=0.1)$$



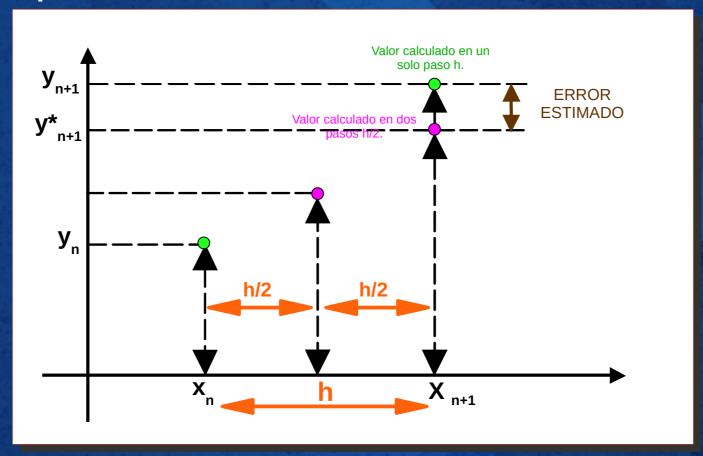
Estrategias de ajuste del paso h

La exactitud de los métodos numéricos de resolución de ecuaciones diferenciales depende fuertemente del **paso h** elegido.

Existen diferentes estrategias para ajustar automáticamente el **paso h** y de esta forma mantener el error de la solución por debajo de una cierta tolerancia establecida.

Estimación del Error

En aquellos métodos que **no tienen** una fórmula específica para la **estimación del error**, es posible estimarlo por medio de la diferencia del valor calculados con **un paso de avance h** y del obtenido al avanzar **dos pasos h/2**.



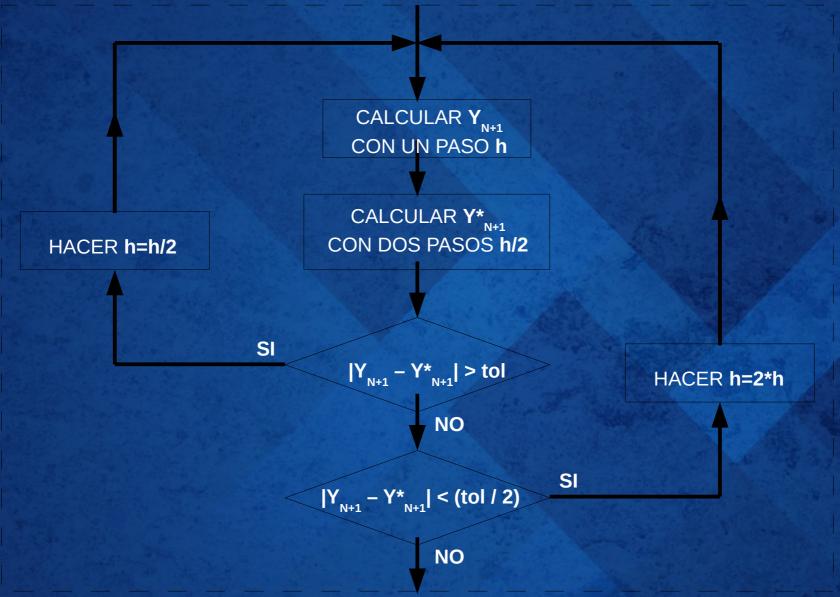
Estrategia de ajuste 1

• Se calcula el valor \mathbf{y}_{n+1} utilizando un paso de avance \mathbf{h} .

Se calcula el valor $\mathbf{y^*}_{n+1}$ utilizando dos pasos de avance h/2.

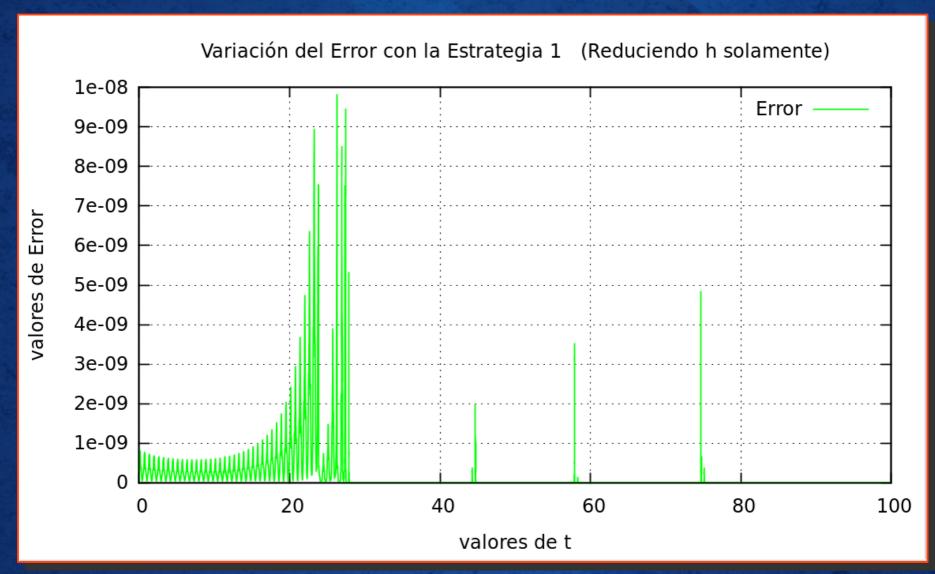
- Si el valor absoluto de la diferencia entre ambos valores es **mayor** a la tolerancia establecida, entonces se establece **h/2** como nuevo valor de **h** y se vuelve al **paso 1**.
- Si el valor absoluto de la diferencia entre ambos valores es menor a la mitad de la tolerancia establecida, entonces se establece 2*h como nuevo valor de h y se vuelve al paso 1.

Diagrama de Flujo de la estrategia 1



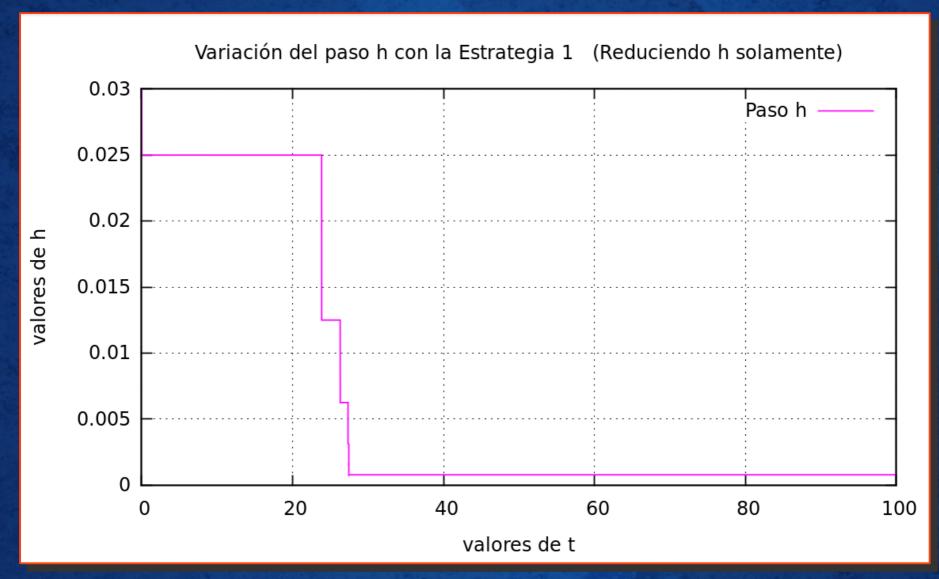
Variación del error con la Estrategia 1

El paso se reduce de ser necesario, pero nunca se aumenta.



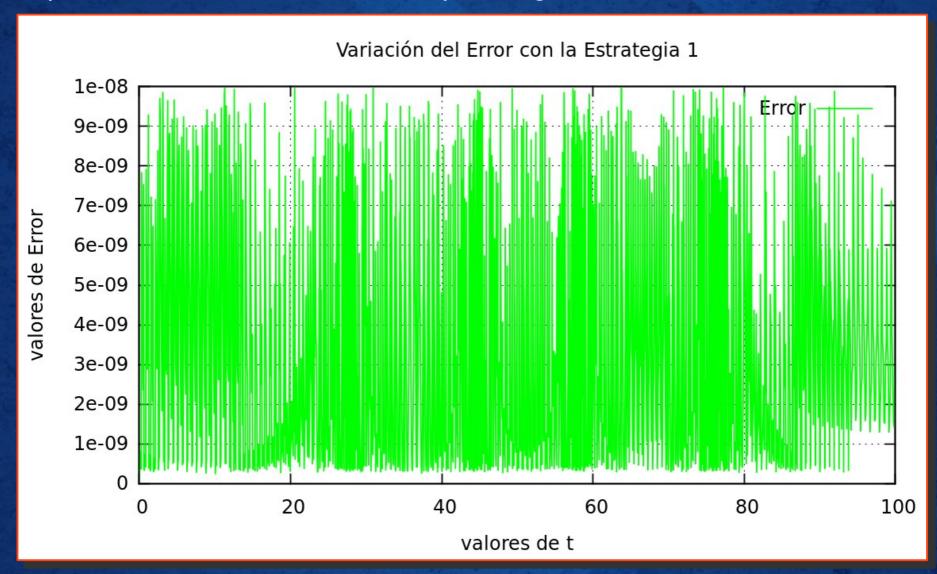
Variación del paso h con la Estrategia 1

El paso se reduce de ser necesario, pero nunca se aumenta.



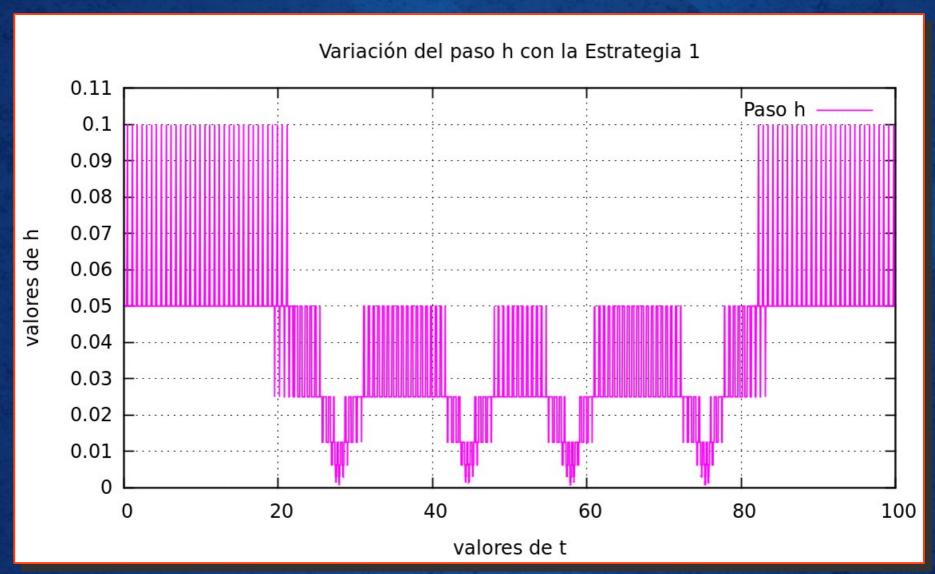
Variación del error con la Estrategia 1

El paso se reduce o se aumenta para lograr la cota de error deseada.



Variación del paso h con la Estrategia 1

El paso se reduce o se aumenta para lograr la cota de error deseada.



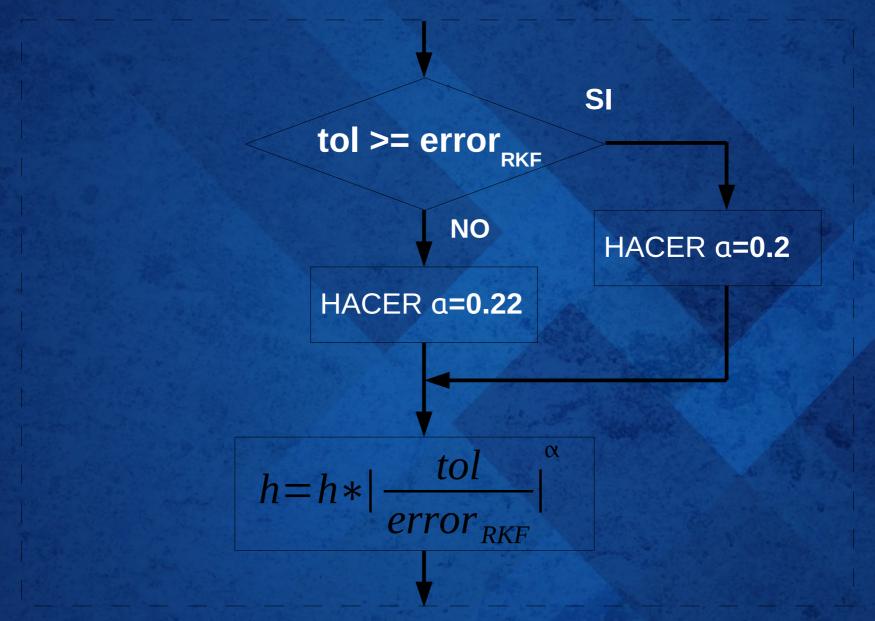
Estrategia de ajuste 2

A partir de una estimación del error obtenida por medio de un método como por ejemplo Runge-Kutta-Fehlberg, podemos ajustar el valor de h, en cada paso.

$$h_{nuevo} = h_{actual} \cdot \left| \frac{tol_{problema}}{\varepsilon_{RKF}} \right|^{\alpha}$$

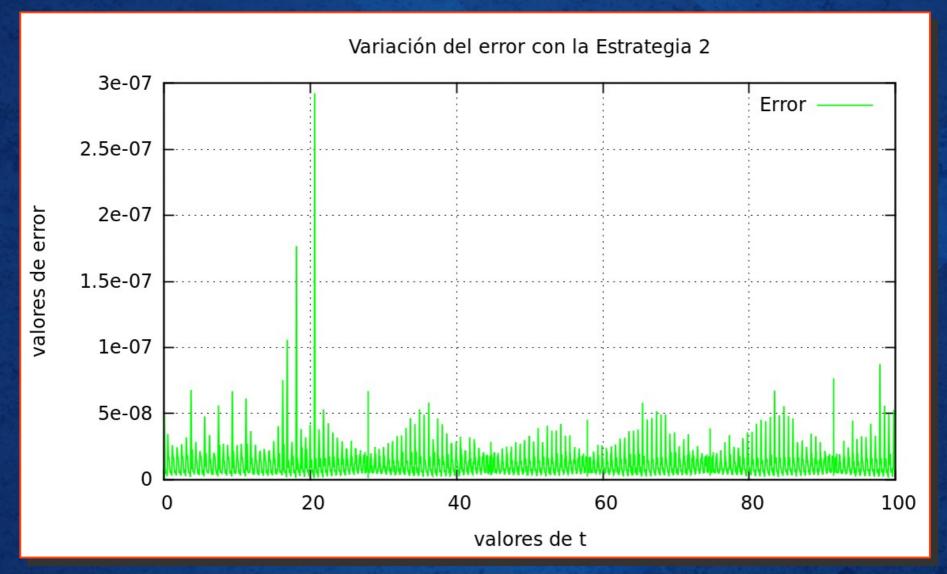
El valor de α se ajusta en base a la relación entre el valor estimado del error y la tolerancia establecida.

Diagrama de la estrategia 2



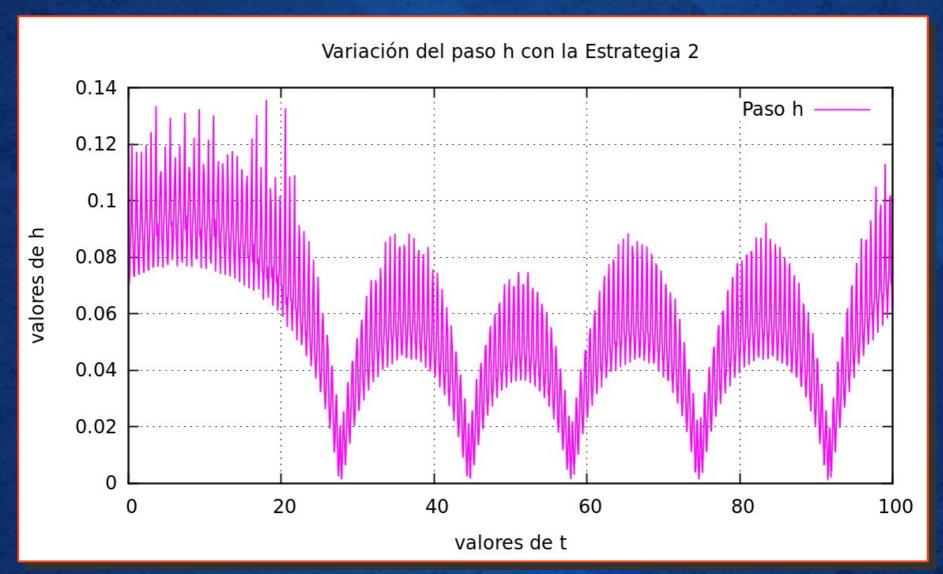
Variación del error con la Estrategia 2

El paso se ajusta de acuerdo al valor de error calculado.

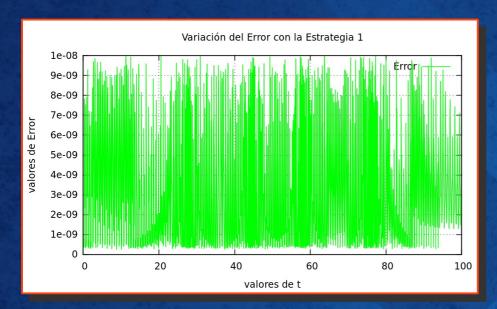


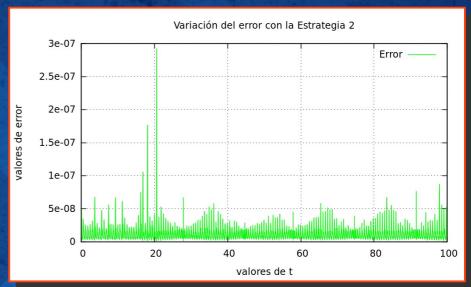
Variación del paso h con la Estrategia 2

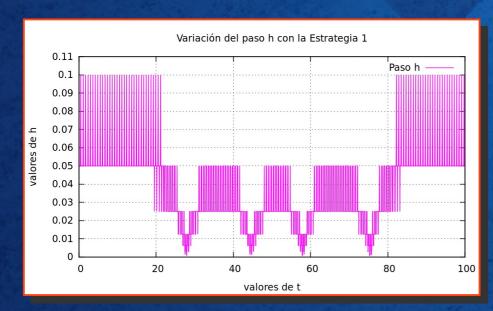
El paso se ajusta de acuerdo al valor de error calculado.

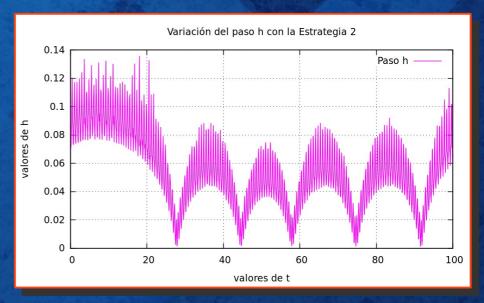


Comparación de ambas estrategias









Resumen de la Estrategia de ajuste 1

Ventajas

- No se necesita de un método que calcule explícitamente el error en cada paso.
- Los valores de h obtenidos son siempre múltiplos del h original.

Desventajas

La implementación es un poco más complicada que la de la estrategia (II).

Estrategia de ajuste 2

Ventajas

Su implementación es muy sencilla.

Desventajas

El valor de h que se obtiene generalmente no es un múltiplo del h original.

Preguntas . . .

