$\begin{cases} f_s' = [\left(\frac{-3}{2h}\right) \cdot f_s + \left(\frac{+4}{2h}\right) \cdot f_{s+1} + \left(\frac{-1}{2h}\right) \cdot f_{s+2}] \\ Er = -\frac{h^2 \ f_s'''}{3} \end{cases}$	$\begin{cases} f_s' = [\left(\frac{-1}{h}\right) \cdot f_s + \left(\frac{1}{h}\right) \cdot f_{s+1}] \\ Er = -\frac{h}{2} f_s'' \end{cases}$
$\begin{cases} f_s' = [\left(\frac{-1}{2h}\right) \cdot f_{s-1} + 0 \cdot f_s + \left(\frac{1}{2h}\right) \cdot f_{s+1}] \\ Er = \frac{h^2}{6} f_s''' \end{cases}$	$\begin{cases} f_s' = [\left(\frac{-1}{h}\right) \cdot f_{s-1} + \left(\frac{1}{h}\right) \cdot f_s] \\ Er = \pm\frac{h}{2} f_s'' \end{cases}$
$\begin{cases} f_s' = [\left(\frac{3}{2h}\right) \cdot f_s + \left(\frac{-4}{2h}\right) \cdot f_{s-1} + \left(\frac{1}{2h}\right) \cdot f_{s-2}] \\ Er = -\frac{h^2 \ f_s'''}{3} \end{cases}$	

La derivada primera calculada con algoritmos con 3 Puntos son tales que:

a) su velocidad de convergencia es igual a [2] ✔

ya que ese es el valor [del exponente de h] ✔ que aparece en el término del error

b) se pueden aplicar en forma exacta hasta funciones polinómicas de grado [2] ✔

ya que ese es el valor [de un grado menos que la derivada] ✔ que aparece en el término del error

Los algoritmos con 2 Puntos tienen [menor] ✔ velocidad de convergencia que los algoritmos con 3 Puntos, para igual incremento de abscisas. Para una reducción del incremento de abscisas de 10, el error de los algoritmos de 2 Puntos se reduce en [10] ✔ mientras que el error de los algoritmos de 3 Puntos se reduce en [100] ✔

TRANSFORMADA DISCRETA DE FOURIER y Mínimos Cuadrados

Si f(tn) está dada en forma discreta mediante NP cantidad de puntos, la TDF de f(tn) es la función F_tdf [de variable compleja] ✔

Cada valor de la función F_tdf, TDF de f(tn), está asociada a [valores de frecuencias discretas] ✔

La TDF inversa permite obtener la función discreta f(tn) conocidos los NP [valores complejos] ✖ de la TDF de f(tn).

Valores complejos y sus conjugados

La TDF está asociada con el Método de Mínimos Cuadrados (Min2) cuando se usan como Base [funciones senos y cosenos] ✔

Entonces, el sistema de ecuaciones lineales de Min2 ($\Phi^T * \Phi * alfa = \Phi^T * g$),

[tiene la matriz diagonal con valores conocidos en la diagonal, y su solución es conocida] ✔

MÉTODO DE EULER
El algoritmo denominado método de EULER Adelante o Explícito, permite resolver tanto una ecuación como un sistema de ecuaciones diferenciales ordinarias de orden

1

Se puede asegurar que su Error Local ($Er=C\ \Delta t^p$) es de orden p= 2 ✔, ya que la forma de cálculo algorítmica coincide con los

✖, ya que la forma de cálculo algorítmica coincide con los resultados de la Serie de Taylor de la solución exacta de la EDO hasta los 2 ✔ primeros sumandos de la mencionada serie.

La solución discreta que se obtiene es con puntos equidistantes, y su Error Local es tanto menor

La solución discreta que se obtiene es con puntos equidistantes, y su Error Local es tanto menor

a) cuanto menor ✔ es la separación de las abscisas donde se calculan las soluciones aproximadas;

b) cuanto menor ✔ es la curvatura de la solución exacta.

El método de EULER Adelante o Explícito tiene menor ✔ velocidad de convergencia que los métodos de Runge Kutta de Segundo Orden, para igual incremento de abscisas. Para una reducción del incremento de abscisas de 10, el error del método de EULER Adelante o Explícito se reduce en 100 ✔, mientras que el error de los métodos de Runge Kutta de Segundo Orden se reduce en 1000 ✔.

Pregunta **2**
Parcialmente correcta
Se puntúa 9,00 sobre 20,00
Marcar pregunta

METODOS DE RUNGE KUTTA

Los algoritmos de los métodos de Runge Kutta de Segundo Orden se pueden resumir como:

Dado $(t_n, u(t_n) = (t_n, u_n)$,
para n entero mayor o igual a 1;

Se calcula $k_1 = \Delta t\, f(t_n, u_n)$

$$u_G = u_n + k_1/(2\omega)$$
$$t_G = t_n + \Delta t/(2\omega)$$

Se calcula $k_2 = \Delta t\, f(t_G, u_G)$

$$u_{n+1} = u_n + (1-\omega)\, k_1 + \omega\, k_2$$
$$t_{n+1} = t_n + \Delta t$$

MÉTODO DE EULER

El algoritmo denominado método de EULER Adelante o Explícito, permite resolver tanto una ecuación como un sistema de ecuaciones diferenciales ordinarias de orden [uno] ✖, obteniendo una función solución discreta y aproximada a la solución exacta.

Se puede asegurar que su Error Local (Er=C Δt^p) es de orden p= [2] ✔ ya que la forma de cálculo algorítmica coincide con los resultados de la Serie de Taylor de la solución exacta de la EDO hasta los [2] ✔ primeros sumandos de la mencionada serie.

La solución discreta que se obtiene es con puntos equidistantes, y su Error Local es tanto menor

a) cuanto [menor] ✔ es la separación de las abscisas donde se calculan las soluciones aproximadas;

b) cuanto [menor] ✔ es la curvatura de la solución exacta.

El método de EULER Adelante o Explícito tiene [igual] ✖ velocidad de convergencia que los métodos de Runge Kutta de Segundo Orden, para igual incremento de abscisas. Para una reducción del incremento de abscisas de 10, el error del método de EULER Adelante o Explícito se reduce en [10] ✖, mientras que el error de los métodos de Runge Kutta de Segundo Orden se reduce en [100] ✖.

DERIVACIÓN NUMÉRICA

Si f(x) está dada en forma discreta es posible considerar los siguientes algoritmos numéricos para calcular la derivada primera

1, menor, 100 y 1000

Para una función $f(x)$ dada en forma discreta, por puntos de abscisas equidistantes separadas en Δx; es posible calcular las derivadas primeras en forma aproximada, mediante reglas de derivación numérica cuyos errores se pueden expresar en la forma:

$$Er = (\Delta x)^p \cdot \left.\frac{d^r[f(x)]}{dx^r}\right|_\xi \cdot C_{te}$$

Donde p y r son números enteros que caracterizan a cada regla de derivación numérica, e indican: el Orden del Error y Grado del Polinomio hasta el cual se puede aplicar la regla sin cometer error.

Para las siguientes Reglas de Derivada Primera indicar los valores de p y r.

$\left.\frac{d[f(x)]}{dx}\right|_s = [\left(\frac{-1}{\Delta x}\right)\cdot f_{s-1} + \left(\frac{1}{\Delta x}\right)\cdot f_s]$ tiene Orden de Error: [1] ✔

$\left.\frac{d[f(x)]}{dx}\right|_s = [\left(\frac{-1}{\Delta x}\right)\cdot f_s + \left(\frac{1}{\Delta x}\right)\cdot f_{s+1}]$ es exacta hasta polinomios de grado: [1] ✔

$\left.\frac{d[f(x)]}{dx}\right|_s = [\left(\frac{3}{2\,\Delta x}\right)\cdot f_s + \left(\frac{-4}{2\Delta x}\right)\cdot f_{s-1} + \left(\frac{1}{2\Delta x}\right)\cdot f_{s-2}]$ tiene Orden de Error: [2] ✔

$\left.\frac{d[f(x)]}{dx}\right|_s = [\left(\frac{-3}{2\,\Delta x}\right)\cdot f_s + \left(\frac{+4}{2\Delta x}\right)\cdot f_{s+1} + \left(\frac{-1}{2\Delta x}\right)\cdot f_{s+2}]$ tiene Orden de Error: [2] ✔

$\left.\frac{d[f(x)]}{dx}\right|_s = [\left(\frac{-1}{2\,\Delta x}\right)\cdot f_{s-1} + 0\cdot f_s + \left(\frac{1}{2\,\Delta x}\right)\cdot f_{s+1}]$ tiene Orden de Error: [2] ✔

Para una función $f(x)$ dada en forma discreta, por puntos de abscisas equidistantes separadas en Δx; es posible calcular la integral definida mediante la regla de Trapecios Simple

$$I_{TS} = \int_{x_i}^{x_{i+1}} f(x)dx = \frac{\Delta x}{2}\left(f(x_i) + f(x_{i+1})\right) + E_{TS}$$

estando el error dado por

$$E_{TS} = (\Delta x)^p \cdot \left.\frac{d^r[f(x)]}{dx^r}\right|_{\xi} \cdot C_{te}$$

Donde p y r son números enteros que caracterizan a cada regla de integración numérica, e indican: el Orden del Error y Grado del Polinomio hasta el cual se puede aplicar la regla sin cometer error. En el caso de la regla de Trapecios Simple:

el orden del error es p= 2 ✗; y la regla de integración es exacta hasta cuando se integran polinomios de grado 1 ✓

La regla de Trapecios Múltiple o Compuesta, permite calcular la integral definida en un rango amplio de las abscisas entre x_1 y x_{N+1}, que se dividen en N intervalos iguales de longitud Δx, mediante la sucesiva aplicación de la regla de Trapecios Simple. Así la regla de Trapecios Múltiple o Compuesta resulta

$$I_{TM} = \int_{x_1}^{x_{N+1}} f(x)\,dx = \sum_{k=1}^{N}\left\{\int_{x_k}^{x_{k+1}} f(x)\,dx\right\} = \sum_{k=1}^{N} I_{TS_k}$$

y su error tiene orden p= 2 ✓; y la regla de integración es exacta hasta cuando se integran polinomios de grado 1 ✓

- Riemann menor

Orden de error lineal(interpretamos 1) y nunca exacta

- Trapecios simple

Orden error 3 y exacta hasta polinomio 1

- Trapecios múltiples

Orden error 2 y exacta hasta polinomio 1

- Simpson simple

Orden error 5 y exacta hasta grado 3

Deriva primera 2 puntos

- Orden error 1

Deriva primera y segunda con 3 puntos

- orden de error 2

Derivada segunda con 4 puntos

- orden de error 2

INTEGRACIÓN NUMÉRICA

Si f(x) está dada en forma discreta mediante NP cantidad de puntos, es posible calcular una integral definida mediante los algoritmos dados por las Sumas de Riemann, la Regla de Integración de Trapecios o la Regla de Integración de Simpson.

Las Sumas de Riemann se pueden aplicar para [cualquier cantidad] ✔ de NP puntos. En estos algoritmos se asume válido que la función discreta a integrar [permanece constante] ✔ en cada uno de los [(NP+1)] ✘ intervalos.

Las Regla de Integración de Trapecios Compuesta se pueden aplicar para [cualquier cantidad] ✔ de NP puntos. En este algoritmo se asume válido que la función discreta a integrar [varía linealmente] ✔ en cada uno de los [(NP)] ✘ intervalos.

El algoritmo de integración de Trapecios Simple, que se puede resumir como

$$I_{TS} = \int_{x_k}^{x_{k+1}} f(x)dx = \frac{\Delta x}{2}\left(f(x_k) + f(x_{k+1})\right) + E_{TS}$$

$$E_{TS} = C\left.\frac{d^r(f(x))}{dx^r}\right|_{\xi}(\Delta x)^p$$

donde p y r son enteros que caracterizan la regla de integración numérica.
El orden del error es p= [2] ✘; y la regla de integración de Trapecios Simple es exacta hasta cuando se integran polinomios de grado [1] ✔

En el caso de aplicar acumulativamente la regla de Trapecios Simple, el algoritmo se transforma en el denominado Trapecios Múltiple o Compuesto; resulta

$$I_{TM} = \int_{x_1}^{x_{N+1}} f(x)dx = \sum_{k=1}^{N}\left\{\int_{x_k}^{x_{k+1}} f(x)dx\right\} = \sum_{k=1}^{N} I_{TS}$$

su error tiene orden p= [2] ✔ y la regla de integración de Trapecios Múltiple o Compuesto es exacta hasta cuando se integran polinomios de grado [1] ✔

NP - 1 y **NP - 1** y orden de error es p=3

METODOS DE RUNGE KUTTA

Los algoritmos de los métodos de Runge Kutta de Segundo Orden permiten resolver tanto una ecuación como un sistema de ecuaciones diferenciales ordinarias de orden [1] ✔ obteniendo una función solución discreta y aproximada a la solución exacta. Se pueden resumir como:

Dado $(t_n, u(t_n)) = (t_n, u_n)$,
para n entero mayor o igual a 1;

Se calcula $k_1 = \Delta t\, f(t_n, u_n)$

$$u_G = u_n + k_1/(2\omega)$$

$$t_G = t_n + \Delta t/(2\omega)$$

Se calcula $k_2 = \Delta t\, f(t_G, u_G)$

$$u_{n+1} = u_n + (1-\omega)\,k_1 + \omega\, k_2$$

$$t_{n+1} = t_n + \Delta t$$

Se trata de algoritmos que para cualquier valor de ω tal que [0] ✔ <ω<= [1] ✔ aseguran que el Error Local del método (Er=C Δt^p) es de orden p= [3] ✔ Ello está asegurado porque los resultados del algoritmo coinciden con los resultados de la Serie de Taylor de la solución exacta de la EDO hasta los [2] ✘ primeros sumandos de la mencionada serie.

Cuando la entrada (o termino independiente) de la EDO es una función discreta definida en valores predeterminados de la variable t, para usar los algoritmos de Runge Kutta de Segundo Orden, es necesario adoptar ω = [1] ✘

Para igual incremento de abscisas, los métodos de Runge Kutta de Segundo Orden tienen [mayor] ✔ velocidad de convergencia que el método de EULER Adelante o Explícito. Para una reducción del incremento de abscisas de 10, el error de los métodos de Runge Kutta de Segundo Orden se reduce en [1000] ✔ mientras que el error del método de EULER Adelante o Explícito se reduce en [100] ✔

---

3 y 0.5