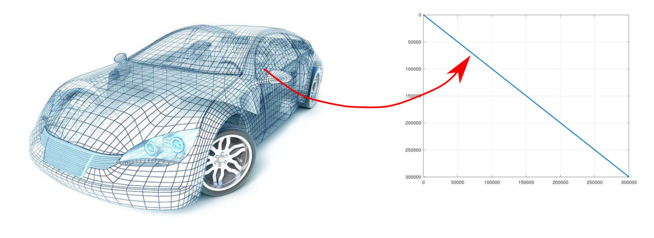
Universidad Politécnica de Cartagena

industriales
etsii UPCT

Resolución de Problemas de Valor Inicial (EDO's)



Rogelio Ortigosa Martínez
Silvestre Paredes Hernández
Departamento de Matemática Aplicada y Estadística, UPCT

Índice

1. Bloque 1: Métodos Numéricos para EDO
2. Bloque 2: Método de Euler
3. Bloque 3: Métodos de Runge-Kutta
4. Bloque 4: Método de Crank-Nicholson
5. Bloque 5: Funciones de Octave

Índice

1. Bloque 1: Métodos Numéricos para EDO
2. Bloque 2: Método de Euler
3. Bloque 3: Métodos de Runge-Kutta
4. Bloque 4: Método de Crank-Nicholson
5. Bloque 5: Funciones de Octave

Métodos Numéricos para EDO

- **Dado** el Problema de Valor Inicial (P.V.I.)

$$\boxed{\begin{cases} x'(t) = f(t, x(t)) \\ x(t_0) = x_0 \end{cases}}$$

- **Objetivo:** Conocer el valor de $x(t)$ para $0 < t_0 < t_1 < \cdots < t_N$, siendo $t_{k+1} = t_k + \Delta t$.

- Definimos

$$\boxed{\begin{aligned} x_k &= x(t_k)\,; \\ f_k &= f(t_k, x_k) \end{aligned}}$$

Métodos Numéricos para Sistemas de EDO

- **Dado** el Problema de Valor Inicial (P.V.I.)

$$\boxed{\begin{cases} \dfrac{dx_1}{dt} = f_1\left(t, x_1, \ldots, x_n\right) \\ \vdots \\ \dfrac{dx_n}{dt} = f_n\left(t, x_1, \ldots, x_n\right) \\ x(t_0) = x_0^0 \\ \vdots \\ x(t_n) = x_n^0 \end{cases}}$$

- **Objetivo:** Conocer el valor de $x\left(t\right) = \left(x_1(t), \ldots, x_n(t)\right)$ para $0 < t_0 < t_1 < \cdots < t_N$, siendo $t_{k+1} = t_k + \Delta t$.

Índice

1. Bloque 1: Métodos Numéricos para EDO
2. Bloque 2: Método de Euler
3. Bloque 3: Métodos de Runge-Kutta
4. Bloque 4: Método de Crank-Nicholson
5. Bloque 5: Funciones de Octave

Euler Explícito para EDO

- El método de Euler Explícito consiste en construir la sucesión

$$\boxed{x_{k+1} = x_k + \Delta t \cdot f_k;}$$

- Si $f\,(x(t), t) = a \cdot x(t) + g(t)$ es lineal, entonces el método es estable si

$$\boxed{\Delta t < \frac{2}{|a|}}$$

- Si $f\,(x(t), t)$ es no lineal, entonces el método puede ser estable si

$$\boxed{\Delta t < \frac{2}{|\frac{\partial f}{\partial x}(t_n, x_n)|}}$$

- El error global del método es de tipo $O(t)$.

Ejemplo con Euler Explícito

Para resolver el problema

$$\begin{cases} x'(t) = x(t) - t^2 + 1; \\ x(0) = \dfrac{1}{2} \qquad\qquad\qquad t \in [0, 10] \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x)  x(1) - t.**2 + 1;
>> DfDy = @(t,x)  1;
>> [x,t] = ForwardEuler(0,0.5,10,100,f,DfDy,0.5);
```

Euler Explícito para Sistemas de EDO

- El método de Euler Explícito en forma matricial es

$$\boxed{\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_n^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^{k} \\ \vdots \\ x_n^{k} \end{pmatrix} + \Delta t \begin{pmatrix} f_1^{k} \\ \vdots \\ f_n^{k} \end{pmatrix} \Rightarrow \begin{cases} x_j^k = x_j(t_k) \\ f_j^k = f_j(t_k, x_1^k, \ldots, x_n^k) \end{cases}}$$

- Si $f\left(x(t), t\right) = A \cdot x(t) + G(t)$ es lineal, entonces el método es estable si

$$\boxed{\Delta t < \frac{2}{\max\left|\mathrm{eig}(A)\right|}}$$

- Si $f\left(x(t), t\right)$ es no lineal, entonces el método puede ser estable si

$$\boxed{\Delta t < \frac{2}{\max\left|\mathrm{eig}(Jf(t_n, x_n))\right|}}$$

- El error global del método es de tipo $O(t)$.

Ejemplo 2D con Euler Explícito

Para resolver el problema

$$\begin{cases} x''(t) = \left(1 - x^2(t)\right) x'(t) - x(t); \\ x(0) = 2 \\ x'(0) = 0 \end{cases}$$

Transformamos la ecuación de grado 2 en un sistema de orden 1 mediante el cambio

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = x'(t) \end{cases} \Longrightarrow \begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = \left(1 - x_1^2(t)\right) x_2(t) - x_1(t); \\ x_1(0) = 2 \\ x_2(0) = 0 \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x)  [x(2);(1 - x(1)^2) * x(2) - x(1)];
>> DfDy = @(t,x)  [0,1;-2 * x(1) * x(2) - 1,(1 - x(1) * *2)];
>> [x,t] = ForwardEuler(0,10,100,f,DfDy),[2;0];
```

Euler Implícito para EDO

- El método de Euler Implícito consiste en construir la sucesión

$$\boxed{x_{k+1} = x_k + \Delta t \cdot f_{k+1};}$$

- El método es incondicionalmente estable.
- El error global del método es de tipo $O(t)$.

Ejemplo con Euler Implícito

Para resolver el problema

$$\begin{cases} x'(t) = x(t) - t^2 + 1; \\ x(0) = \dfrac{1}{2} \qquad\qquad t \in [0, 10] \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x)   x(1) - t.**2 + 1;
>> DfDy = @(t,x)   1;
>> [x,t] = BackwardEuler(0,10,100,f,DfDy,0.5);
```

Euler Implícito para Sistemas de EDO

- El método de Euler Implícito en forma matricial será

$$\boxed{\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_n^{k+1} \end{pmatrix} = \begin{pmatrix} x_1^{k} \\ \vdots \\ x_n^{k} \end{pmatrix} + \Delta t \begin{pmatrix} f_1^{k+1} \\ \vdots \\ f_n^{k+1} \end{pmatrix}}$$

Siendo

$$\boxed{\begin{aligned} x_j^k &= x_j(t_k) \\ f_j^k &= f_j(t_k, x_1^k, \ldots, x_n^k) \end{aligned}}$$

Euler Implícito para Sistemas de EDO (II)

- Podemos expresar la ecuación vectorial anterior como

$$\boxed{\begin{pmatrix} x_1^{k+1} \\ \vdots \\ x_n^{k+1} \end{pmatrix} - \begin{pmatrix} x_1^{k} \\ \vdots \\ x_n^{k} \end{pmatrix} - \Delta t \begin{pmatrix} f_1^{k+1} \\ \vdots \\ f_n^{k+1} \end{pmatrix} = 0}$$

Lo que nos conduce a una sistema de n ecuaciones con n incógnitas que, en general, no será lineal y habrá que resolverlo con alguno de los métodos del tema 3, por ejemplo, el método de Newton-Raphson para sistemas.

Ejemplo 2D con Euler Explícito

Para resolver el problema

$$\begin{cases} x''(t) = \left(1 - x^2(t)\right) x'(t) - x(t); \\ x(0) = 2 \\ x'(0) = 0 \end{cases}$$

Transformamos la ecuación de grado 2 en un sistema de orden 1 mediante el cambio

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = x'(t) \end{cases} \Longrightarrow \begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = \left(1 - x_1^2(t)\right) x_2(t) - x_1(t); \\ x_1(0) = 2 \\ x_2(0) = 0 \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x) [x(2);(1 - x(1)^2) * x(2) - x(1)];
>> DfDy = @(t,x) @(t,x)[0,1;-2 * x(1) * x(2) - 1,(1 - x(1) **2)];
>> [x,t] = BackwardEuler(0,10,100,f,DfDy,[2;0]);
```

Índice

1. Bloque 1: Métodos Numéricos para EDO
2. Bloque 2: Método de Euler
3. Bloque 3: Métodos de Runge-Kutta
4. Bloque 4: Método de Crank-Nicholson
5. Bloque 5: Funciones de Octave

Método de Runge-Kutta Explícito de orden 2

- Esquema

$$\begin{cases} K_1 = \Delta t \cdot f(t_n, x_n) \\ K_2 = \Delta t \cdot f(t_{n+1}, x_n + K_1) \\ x_{n+1} = x_n + \dfrac{1}{2}(K_1 + K_2) \end{cases}$$

- Si $f\left(x(t), t\right) = a \cdot x(t) + g(t)$ es lineal, entonces el método es estable si

$$\Delta t < \frac{2}{|a|}$$

- Si $f\left(x(t), t\right)$ es no lineal, entonces el método puede ser estable si

$$\Delta t < \frac{2}{\left|\frac{\partial f}{\partial x}(t_n, x_n)\right|}$$

- El error global del método es de tipo $O((\Delta t)^2)$.

Método de Runge-Kutta Explícito de orden 4

- Esquema

$$\begin{cases} K_1 = \dfrac{\Delta t}{2} \cdot f(t_n, x_n) \\ K_2 = \dfrac{\Delta t}{2} \cdot f(t_n + \dfrac{\Delta t}{2}, x_n + K_1) \\ K_3 = \dfrac{\Delta t}{2} \cdot f(t_n + \dfrac{\Delta t}{2}, x_n + K_2) \\ K_4 = \dfrac{\Delta t}{2} \cdot f(t_{n+1}, x_n + 2K_3) \\ x_{n+1} = x_n + \dfrac{1}{3}(K_1 + 2K_2 + 2K_3 + K_4) \end{cases}$$

- Método condicionalmente estable, Δt debe ser pequeño.
- El error global del método es de tipo $O((\Delta t)^4)$.

Ejemplo con Runge-Kutta de Orden 2

Para resolver el problema

$$\begin{cases} x'(t) = x(t) - t^2 + 1; & \\ x(0) = \dfrac{1}{2} & t \in [0, 10] \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x)   x(1) - t.**2 + 1;
>> DfDy = @(t,x)   1;
>> [x,t] = RungeKutta(0,10,100,f,DfDy,0.5);
```

Ejemplo 2D con Runge-Kutta de Orden 2

Para resolver el problema

$$\begin{cases} x''(t) = \left(1 - x^2(t)\right) x'(t) - x(t); \\ x(0) = 2 \\ x'(0) = 0 \end{cases}$$

Transformamos la ecuación de grado 2 en un sistema de orden 1 mediante el cambio

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = x'(t) \end{cases} \Longrightarrow \begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = \left(1 - x_1^2(t)\right) x_2(t) - x_1(t); \\ x_1(0) = 2 \\ x_2(0) = 0 \end{cases}$$

Utilizaremos el siguiente código

$$\begin{aligned} &>> \mathtt{f} = @(\mathtt{t},\mathtt{x}) \quad [\mathtt{x}(2); (1 - \mathtt{x}(1)^2) * \mathtt{x}(2) - \mathtt{x}(1)]; \\ &>> \mathtt{DfDy} = @(\mathtt{t},\mathtt{x}) \quad @(\mathtt{t},\mathtt{x})[0,1; -2 * \mathtt{x}(1) * \mathtt{x}(2) - 1, (1 - \mathtt{x}(1) * *2)]; \\ &>> [\mathtt{x},\mathtt{t}] = \mathtt{RungeKutta}(0,10,100,\mathtt{f},[2,0],\mathtt{DfDy}); \end{aligned}$$

Índice

1 Bloque 1: Métodos Numéricos para EDO

2 Bloque 2: Método de Euler

3 Bloque 3: Métodos de Runge-Kutta

4 Bloque 4: Método de Crank-Nicholson

5 Bloque 5: Funciones de Octave

Método de Crank-Nicholson Implícito de orden 2

- Esquema

$$\left\{ x_{n+1} = x_n + \frac{1}{2}\left(f_n + f_{n+1}\right) \right.$$

- Método incondicionalmente estable.
- El error global del método es de tipo $O((\Delta t)^2)$.

Ejemplo con Crank-Nicholson

Para resolver el problema

$$\begin{cases} x'(t) = x(t) - t^2 + 1; \\ x(0) = \dfrac{1}{2} & t \in [0, 10] \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x) x(1) - t.**2 + 1;
>> DfDy = @(t,x) 1;
>> [x,t] = CrankNicholson(0,10,100,f,DfDy,0.5);
```

Ejemplo 2D con Crank-Nicholson de Orden 2

Para resolver el problema

$$\begin{cases} x''(t) = \left(1 - x^2(t)\right) x'(t) - x(t); \\ x(0) = 2 \\ x'(0) = 0 \end{cases}$$

Transformamos la ecuación de grado 2 en un sistema de orden 1 mediante el cambio

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = x'(t) \end{cases} \Longrightarrow \begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = \left(1 - x_1^2(t)\right) x_2(t) - x_1(t); \\ x_1(0) = 2 \\ x_2(0) = 0 \end{cases}$$

Utilizaremos el siguiente código

```
>> f = @(t,x)  [x(2);(1 - x(1)^2) * x(2) - x(1)];
>> DfDy = @(t,x)  @(t,x)[0,1;-2 * x(1) * x(2) - 1,(1 - x(1) **2)];
>> [x,t] = CrankNicholson(0,10,100,f,DfDy,[2,0]);
```

Índice

1. Bloque 1: Métodos Numéricos para EDO
2. Bloque 2: Método de Euler
3. Bloque 3: Métodos de Runge-Kutta
4. Bloque 4: Método de Crank-Nicholson
5. Bloque 5: Funciones de Octave

Resolución de Ecuaciones Diferenciales Ordinarias con Octave

- **lsode**: La función *lsode*, se utiliza para resolver problemas de valor inicial de la forma

$$\begin{cases} x'(t) = f\,(t, x(t)) \\ \qquad x(t_0) = x_0 \end{cases}$$

Uso:

- $[x, istate, msg] = lsode(fcn, x_0, t)$

donde fcn puede contener tanto el nombre de la función, como el de su Jacobiano.

Ejemplo con lsode

Para resolver el problema

$$\begin{cases} x'(t) = x(t) - t^2 + 1; \\ x(0) = \dfrac{1}{2} \end{cases}$$

Utilizaremos el siguiente código

$$\begin{aligned} &>> \mathrm{f} = @(\mathrm{x}, \mathrm{t}) \quad \mathrm{x}(1) - \mathrm{t}. **2 + 1; \\ &>> \mathrm{t} = \mathrm{linspace}(0, 4, 100); \\ &>> \mathrm{x} = \mathrm{lsode}(\mathrm{f}, [0.5], \mathrm{t}); \end{aligned}$$

Ejemplo 2D con lsode

Para resolver el problema

$$\begin{cases} x''(t) = \left(1 - x^2(t)\right) x'(t) - x(t); \\ x(0) = 2 \\ x'(0) = 0 \end{cases}$$

Primero transformamos la ecuación de segundo grado en un sistema de primer orden mediante el cambio

$$\begin{cases} x_1(t) = x(t) \\ x_2(t) = x'(t) \end{cases} \Longrightarrow \begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = \left(1 - x_1^2(t)\right) x_2(t) - x_1(t); \\ x_1(0) = 2 \\ x_2(0) = 0 \end{cases}$$

Utilizaremos el siguiente código

$$>> \mathrm{f} = @(\mathrm{x}, \mathrm{t}) \quad [\mathrm{x}(2), (1 - \mathrm{x}(1)^2) * \mathrm{x}(2) - \mathrm{x}(1)];$$
$$>> \mathrm{t} = \mathrm{linspace}(0, 10, 100);$$
$$>> \mathrm{x} = \mathrm{lsode}(\mathrm{f}, [2, 0], \mathrm{t});$$