

Computació Numèrica

Sessió 12. Equacions Diferencials amb Matlab

M. Àngela Grau Gotés

Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya · BarcelonaTech.

17 de maig de 2020

“Donat el caràcter i la finalitat exclusivament docent i eminentment il·lustrativa de les explicacions a classe d'aquesta presentació, l'autor s'acull a l'article 32 de la Llei de propietat intel·lectual vigent respecte de l'ús parcial d'obres alienes com ara imatges, gràfics o altre material contingudes en les diferents diapositives”



© 2019 by M. Àngela Grau Gotés.

Licència Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional.

- 1 Mètodes d'aproximació numèrica
 - Mètode d'Euler
 - Mètodes de Runge-Kutta
 - Mètodes múltiples
- 2 Solvers de Matlab
 - Equacions diferencials de primer ordre
 - Equacions diferencials de segon ordre
- 3 Sistemes d'equacions diferencials de primer ordre
- 4 Referències

Del problema de valors inicials:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(a) = \alpha \end{cases} \quad (1)$$

calcular $y(t)$ per a $t \in [a, b]$.

Discretització

Donat N prenem $t_i = a + ih$, per $i = 0, 1, 2, \dots, N$, amb
$$h = t_i - t_{i-1} = (b - a)/N.$$

Mètode d'Euler

Mètode d'Euler

El mètode d'Euler construeix

$$\omega_i \approx y(t_i)$$

tal que

$$\begin{cases} \omega_0 = \alpha \\ \omega_{i+1} = \omega_i + h f(t_i, \omega_i) \end{cases} \quad (2)$$

Error local $\mathcal{O}(h^2)$

$$y(t_{i+1}) = y(t_i) + h f(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i)$$

Mètode d'Euler Modificat

Mètode del punt mig explícit

Per $\omega_0 = \alpha$ el mètode és:

$$\begin{aligned}k_1 &= \omega_i + \frac{h}{2} f(t_i, \omega_i) \\ \omega_{i+1} &= \omega_i + h f\left(t_i + \frac{h}{2}, k_1\right)\end{aligned}\tag{3}$$

Tal que $\omega_i \approx y(t_i)$, i per tant, $\omega_N \approx y(b)$.

Aquest mètode és un mètode de Runge-Kutta de segon orde. L'error local és $\mathcal{O}(h^3)$ i l'error global queda $\mathcal{O}(h^2)$

Mètode d'Euler millorat

Mètode de Heun de segon ordre

Per $\omega_0 = \alpha$ el mètode és:

$$\begin{aligned}k_1 &= h f(t_i, \omega_i) \\k_2 &= h f(t_{i+1}, \omega_i + k_1) \\ \omega_{i+1} &= \omega_i + \frac{1}{2} (k_1 + k_2)\end{aligned}\tag{4}$$

Tal que $\omega_i \approx y(t_i)$, i per tant, $\omega_N \approx y(b)$

Aquest mètode és conegut per mètode de Heun de segon orde.
També és un RK2. L'error local és $\mathcal{O}(h^3)$ i l'error global queda $\mathcal{O}(h^2)$

Mètodes de Runge-Kutta

Mètode de Runge-Kutta de quart ordre

El mètode construeix $\omega_i \approx y(t_i)$ tal que

$$\begin{aligned}\omega_0 &= \alpha \\ k_1 &= h f(t_i, \omega_i) \\ k_2 &= h f\left(t_i + \frac{h}{2}, \omega_i + \frac{k_1}{2}\right) \\ k_3 &= h f\left(t_i + \frac{h}{2}, \omega_i + \frac{k_2}{2}\right) \\ k_4 &= h f(t_{i+1}, \omega_i + k_3) \\ \omega_{i+1} &= \omega_i + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}\tag{5}$$

Mètodes múltiples

Adams-Bashforth d'ordre 4

Les equacions del mètode són:

$$\omega_0 = \alpha, \quad \omega_1 = \alpha_1, \quad \omega_2 = \alpha_2, \quad \omega_3 = \alpha_3,$$

$$\omega_{i+1} = \omega_i + \frac{h}{24} \left(55f(t_i, \omega_i) - 59f(t_{i-1}, \omega_{i-1}) + \right. \\ \left. + 37f(t_{i-2}, \omega_{i-2}) - 9f(t_{i-3}, \omega_{i-3}) \right) \\ i = 0, 1, 2, \dots, N-1$$

És un mètode explícit.

Adams-Moulton d'ordre 4

Les equacions del mètode són:

$$\omega_0 = \alpha, \quad \omega_1 = \alpha_1, \quad \omega_2 = \alpha_2, \quad \omega_3 = \alpha_3,$$

$$\omega_{i+1} = \omega_i + \frac{h}{24} \left(9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - \right. \\ \left. - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2}) \right) \\ i = 0, 1, 2, \dots, N-1$$

És un mètode implícit.

MATLAB®

Solvers de Matlab

Matlab has several different functions (built-ins) for the numerical solution of ODEs. These solvers can be used with the following syntax:

```
[outputs] = function_handle(inputs)  
[t,state] = solver(@dstate,tspan,ICs,options)
```

An array. The solution of the ODE (the values of the state at every time).

Matlab algorithm (e.g., ode45, ode23)

Handle for function containing the derivatives

Vector that specifies the interval of the solution (e.g., [t0:tF])

A vector of the initial conditions for the system (row or column)

Solvers de Matlab

- Nonstiff problems:
 - ode45: medium accuracy. Use most of the time.
 - ode23: low accuracy. Use large error tolerances or moderately stiff problems.
 - ode113: low to high accuracy.
- Stiff problems:
 - ode15s: low to medium accuracy. Use if ode45 is slow.
 - ode23s: low accuracy. Use large error tolerances.
 - ode23t: low accuracy. Use for moderately stiff problems where you need a solution without numerical damping.
 - ode23ts: low accuracy.

Ordinary differential equations

Equacions diferencials ordinàries de primer ordre

Problema de valors inicials:

$$\left. \begin{aligned} y'(t) &= f(t, y(t)) \\ y(a) &= \alpha \end{aligned} \right\} \quad (6)$$

calcular $y(b)$ i $y(t)$ per a $t \in [a, b]$.

Exercici

Resoldre $\left\{ \begin{aligned} t^2 y'(t) &= y(t) + 3t, \\ y(1) &= -2, \end{aligned} \right.$ a l'interval $1 \leq t \leq 4$.

Sintàxi bàsica

- A. Creeu un fitxer `yp.m` per a la funció $f(t, y)$.

```
function dydt = yp(t,y)
dydt = (y + 3t)./t^2;
end
```

- B. Resol

```
[t,y]=ode45(@yp,[t0,tf],y0);
```

- C. Grafica el resultat:

```
plot(t,y,'+')
```

- D. Tabula el resultat

```
disp([t,y])
```

Exercicis

2.Problema de valors inicials:

$$\begin{aligned}y'(t) &= y + t^2 \\ y(0) &= 1\end{aligned}\tag{7}$$

calcular $y(1)$ i $y(t)$ per a $t \in [0, 1]$.

3.Problema de valors inicials:

$$\begin{aligned}y'(t) &= \alpha y(t) - \gamma y(t)^2 \\ y(0) &= 10\end{aligned}\tag{8}$$

calcular $\lim_{t \rightarrow \infty} y(t) = \frac{\alpha}{\gamma} \quad t \in [0, \infty)$. $\alpha = 2, \gamma = 0.0001$

Equacions diferencials ordinàries

Sistemes de primer ordre

Equacions diferencials ordinàries de segon ordre

Problema de valors inicials:

$$\left. \begin{aligned} y''(t) + p(t)y'(t) + q(t)y(t) &= g(t) \\ y(a) &= y_0 \\ y'(a) &= y_1 \end{aligned} \right\} \quad (9)$$

calcular $y(b)$ i $y(t)$ per a $t \in [a, b]$.

Exercici

Resoldre

$$y''(t) + ty(t) = 0,$$

amb $y(0) = 2$, $y'(0) = 8$ a l'interval $0 \leq t \leq 4$.

Sintàxi bàsica

Primer de tot cal convertir l'equació diferencial de segon ordre en un sistema d'equacions diferencials de primer ordre.

Sigui $x_1(t) = y(t)$, $x_2(t) = y'(t)$, llavors l'equació (9) és

$$\left. \begin{aligned} x_1'(t) &= x_2(t) \\ x_2'(t) &= -q(t)x_1(t) - p(t)x_2(t) + g(t) \\ x_1(a) &= y_0 \\ x_2(a) &= y_1 \end{aligned} \right\} \quad (10)$$

Sintàxi bàsica

- A. Crea un fitxer `yp.m` per al sistema $f(t, y)$.

```
function dxdt = yp(t,y)
    dxdt(1)=x(2);
    dxdt(2)=-t*x(1);
end
```

- B. Resol

```
[t,x]=ode45(@yp,[t0,tf],[y0,y1]);
```

- C. Grafica el resultat:

```
plot(t,x(1),'+'),
```

- D. Tabula el resultat

```
disp([t,x(1)])
```

Exercici

Non-linear pendulum

Simple pendulum

```
function [t,z] = call_pend(T_out)
tspan=[0 T_out]; % set time interval
z0=[pi/3,0]; % set initial conditions
[t,z]=ode23(@pend,tspan,z0);
plot(t,z(:,1))

    function dzdt = pend(t,z)
        G=9.8; L=2; % set constants
        z1=z(1); % get z1
        z2=z(2); % get z2
        dzdt = [z2 ; -G/L*sin(z1)];
    end
end
```

Exercici

Stiff system

L'oscil·lador de van der Pol és un oscil·lador amb amortiment no lineal.





$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0 \quad (11)$$

en què x és la posició, funció del temps t , i μ és un paràmetre escalar que incorpora la no linealitat i l'amortiment.

Resoleu els problema de valors inicials:

- 3. $m = 1$, $x(0) = 0$, $x'(0) = 2$,
- 4. $m = 1000$, $x(0) = 0$, $x'(0) = 1$.

Guies de MATLAB

-  MathWorks Documentation Center, Matlab Users's Guide online
-  MathWorks Documentation Center, Matlab Functions's Guide online
-  MathWorks Documentation Center, Matlab Users's Guide in pdf
-  MathWorks Documentation Center, Tutorials