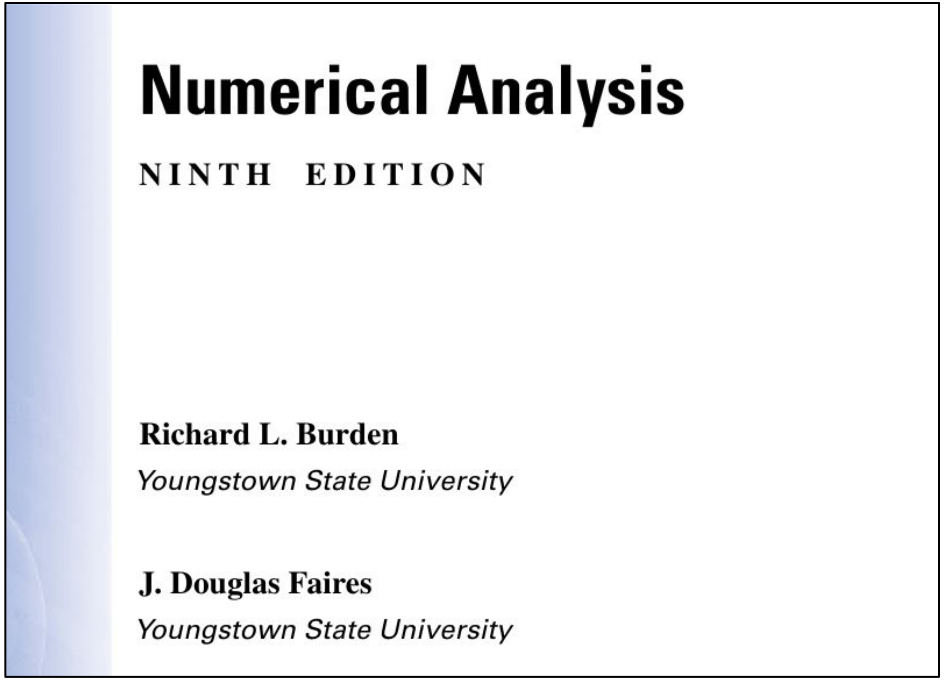
Método de diferencias finitas para problemas no lineales

Ana Cristina Molina
Andrés Arteaga

Física Computacional II
20202

Contenido

1. Método
2. Algoritmo
3. Código en C++*
4. Ejemplos


Numerical Analysis

NINTH EDITION

Richard L. Burden

Youngstown State University

J. Douglas Faires

Youngstown State University

*Repositorio de Github:
https://github.com/anacmolina/CompII-20202/tree/main/Parcial%20III

Método

La ecuación diferencial de segundo orden no lineal a resolver es:

$$\boxed{y'' = f(x, y, y'),} \quad \boxed{a \le x \le b,}$$

$$f(x, y, y') = \alpha \text{ and } y(b) = \beta$$

- f_y $f_{y'}$ ebe satisfacer que:

  y son continuas en D

$f_y(x, y, y') \ge \delta$ en D para algún $\delta > 0$;

- Existen las constante k y L con
- $$k = \max_{(x,y,y') \in D} |f_y(x, y, y')|$$

$$L = \max_{(x,y,y') \in D} |f_{y'}(x, y, y')|.$$

Método

Se divide el intervalo [a, b] en N+1 subintervalos y puntos asociados:

$$\boxed{x_i = a + ih,}\ \boxed{i = 0, 1, \ldots, N+1.}$$

$$\boxed{h = (b-a)/(N+1)}$$

Se usan series de Taylor para expandir la función y alrededor de *x*. Y reescribir *y*”.

$$\boxed{y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^+),}$$

$$\boxed{y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^-),}$$

Método

$$y(x_{i+1}) + y(x_{i-1}) = 2y(x_i) + h^2 y''(x_i) + \frac{h^4}{24}[y^{(4)}(\xi_i^+) + y^{(4)}(\xi_i^-)],$$

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{24}[y^{(4)}(\xi_i^+) + y^{(4)}(\xi_i^-)].$$

$$\boxed{y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{12}y^{(4)}(\xi_i),}$$

Método

A partir de los polinomios de Lagrange se puede aproximar la función y'.

$$f(x) = \sum_{k=0}^{n} f(x_k)L_k(x) + \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} f^{(n+1)}(\xi(x)),$$

$$f'(x) = \sum_{k=0}^{n} f(x_k)L_k'(x) + D_x\left[\frac{(x-x_0)\cdots(x-x_n)}{(n+1!)}\right] f^{(n+1)}(\xi(x))$$
$$+ \frac{(x-x_0)\cdots(x-x_n)}{(n+1)!} D_x[f^{(n+1)}(\xi(x))].$$

$$f'(x_j) = \sum_{k=0}^{n} f(x_k)L_k'(x_j) + \frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \prod_{\substack{k=0 \\ k\neq j}}^{n} (x_j - x_k),$$

Método

$$f'(x_j) = f(x_0)\left[\frac{2x_j - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)}\right] + f(x_1)\left[\frac{2x_j - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)}\right] + f(x_2)\left[\frac{2x_j - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}\right] + \frac{1}{6} f^{(3)}(\xi_j) \prod_{\substack{k=0 \\ k \neq j}}^{2} (x_j - x_k),$$

$$x_1 = x_0 + h \quad \text{and} \quad x_2 = x_0 + 2h, \quad \text{for some } h \neq 0.$$

$$x_j = x_1$$

$$f'(x_1) = \frac{1}{h}\left[-\frac{1}{2} f(x_0) + \frac{1}{2} f(x_2)\right] - \frac{h^2}{6} f^{(3)}(\xi_1),$$

$$y'(x_i) = \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{6} y'''(\eta_i),$$

Método

Reemplazando en la ecuación diferencial de segundo orden se encuentra que

$$\boxed{\frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} = f\left(x_i, y(x_i), \frac{y(x_{i+1}) - y(x_{i-1})}{2h} - \frac{h^2}{6}y'''(\eta_i)\right) + \frac{h^2}{12}y^{(4)}(\xi_i),}$$

Reescribiendo los valores de $y(x_i)$, cómo w_i; se encuentra que:

$$\boxed{w_0 = \alpha, \quad w_{N+1} = \beta}$$

$$\boxed{-\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} + f\left(x_i, w_i, \frac{w_{i+1} - w_{i-1}}{2h}\right) = 0,} \qquad \boxed{i = 1, 2, \ldots, N.}$$

Método

Sistemas de ecuaciones no lineales (1)

$$
\begin{aligned}
2w_1 - w_2 + h^2 f\left(x_1, w_1, \frac{w_2 - \alpha}{2h}\right) - \alpha &= 0,\\
-w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3 - w_1}{2h}\right) &= 0,\\
&\vdots\\
-w_{N-2} + 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N - w_{N-2}}{2h}\right) &= 0,\\
-w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta - w_{N-1}}{2h}\right) - \beta &= 0
\end{aligned}
$$

Método

Método de Newton para sistemas no lineales

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},$$

$$A(\mathbf{x}) = \begin{bmatrix} a_{11}(\mathbf{x}) & a_{12}(\mathbf{x}) & \cdots & a_{1n}(\mathbf{x}) \\ a_{21}(\mathbf{x}) & a_{22}(\mathbf{x}) & \cdots & a_{2n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ a_{n1}(\mathbf{x}) & a_{n2}(\mathbf{x}) & \cdots & a_{nn}(\mathbf{x}) \end{bmatrix},$$

$$\mathbf{G}(\mathbf{x}) = \mathbf{x} - A(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x})$$

$$g_i(\mathbf{x}) = x_i - \sum_{j=1}^{n} b_{ij}(\mathbf{x}) f_j(\mathbf{x}).$$

$$\frac{\partial g_i}{\partial x_k}(\mathbf{x}) = \begin{cases} 1 - \displaystyle\sum_{j=1}^{n} \left(b_{ij}(\mathbf{x}) \frac{\partial f_j}{\partial x_k}(\mathbf{x}) + \frac{\partial b_{ij}}{\partial x_k}(\mathbf{x}) f_j(\mathbf{x}) \right), & \text{if } i = k, \\ -\displaystyle\sum_{j=1}^{n} \left(b_{ij}(\mathbf{x}) \frac{\partial f_j}{\partial x_k}(\mathbf{x}) + \frac{\partial b_{ij}}{\partial x_k}(\mathbf{x}) f_j(\mathbf{x}) \right), & \text{if } i \neq k. \end{cases}$$

$\mathbf{p}$ be a solution of $\mathbf{G}(\mathbf{x}) = \mathbf{x}$.

$$\partial g_i(\mathbf{p})/\partial x_k = 0,$$

for $i = k$,

$$0 = 1 - \sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_i}(\mathbf{p}),$$

$$\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_i}(\mathbf{p}) = 1.$$

Método

$$k \neq i,$$

$$0 = -\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_k}(\mathbf{p}),$$

$$\sum_{j=1}^{n} b_{ij}(\mathbf{p}) \frac{\partial f_j}{\partial x_k}(\mathbf{p}) = 0.$$

$$J(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \frac{\partial f_1}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}) \\ \frac{\partial f_2}{\partial x_1}(\mathbf{x}) & \frac{\partial f_2}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_2}{\partial x_n}(\mathbf{x}) \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \frac{\partial f_n}{\partial x_2}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_n}(\mathbf{x}) \end{bmatrix}.$$

$$A(\mathbf{p})^{-1}J(\mathbf{p}) = I, \text{the identity matrix,} \quad \text{so } A(\mathbf{p}) = J(\mathbf{p}).$$

$$\boxed{\mathbf{G}(\mathbf{x}) = \mathbf{x} - J(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x}),}$$

$$\boxed{\mathbf{x}^{(k)} = \mathbf{G}\left(\mathbf{x}^{(k-1)}\right) = \mathbf{x}^{(k-1)} - J\left(\mathbf{x}^{(k-1)}\right)^{-1}\mathbf{F}\left(\mathbf{x}^{(k-1)}\right).}$$

$$J(\mathbf{x}^{(k-1)})\mathbf{y} = -\mathbf{F}(\mathbf{x}^{(k-1)})$$

Método

Jacobiano asociado a sistema de ecuaciones (1)

$$J(w_1,\ldots,w_N)_{ij} = \begin{cases} -1 + \dfrac{h}{2} f_{y'}\left(x_i, w_i, \dfrac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i=j-1 \text{ and } j=2,\ldots,N, \\ 2 + h^2 f_y\left(x_i, w_i, \dfrac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i=j \text{ and } j=1,\ldots,N, \\ -1 - \dfrac{h}{2} f_{y'}\left(x_i, w_i, \dfrac{w_{i+1}-w_{i-1}}{2h}\right), & \text{for } i=j+1 \text{ and } j=1,\ldots,N-1, \end{cases}$$

Método

Sistemas de ecuaciones a resolver *Ax=b*

$$
\begin{aligned}
&J(w_1,\ldots,w_N)(v_1,\ldots,v_n)^t \\
&\quad= -\left(2w_1 - w_2 - \alpha + h^2 f\left(x_1, w_1, \frac{w_2-\alpha}{2h}\right),\right. \\
&\qquad -w_1 + 2w_2 - w_3 + h^2 f\left(x_2, w_2, \frac{w_3-w_1}{2h}\right), \ldots, \\
&\qquad -w_{N-2} + 2w_{N-1} - w_N + h^2 f\left(x_{N-1}, w_{N-1}, \frac{w_N-w_{N-2}}{2h}\right) \\
&\qquad \left. -w_{N-1} + 2w_N + h^2 f\left(x_N, w_N, \frac{\beta-w_{N-1}}{2h}\right) - \beta\right)^t
\end{aligned}
$$

Método

Factorización de Crout para matrices tridiagonales

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}.$$

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{bmatrix} \quad U = \begin{bmatrix} 1 & u_{12} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

$$Ax = b, A = LU \Rightarrow LUx = b$$

$$Lz = b \text{ y } Ux = z$$

$$a_{11} = l_{11};$$

$$a_{i,i-1} = l_{i,i-1}, \quad \text{for each } i = 2, 3, \ldots, n;$$

$$a_{ii} = l_{i,i-1}u_{i-1,i} + l_{ii}, \quad \text{for each } i = 2, 3, \ldots, n;$$

$$a_{i,i+1} = l_{ii}u_{i,i+1}, \quad \text{for each } i = 1, 2, \ldots, n-1.$$

Método

Algoritmo y código en C++

Step 1 Set $h = (b - a)/(N + 1)$;
$w_0 = \alpha$;
$w_{N+1} = \beta$.

Step 2 For $i = 1, \ldots, N$ set $w_i = \alpha + i \left(\dfrac{\beta - \alpha}{b - a} \right) h$.

Step 3 Set $k = 1$.

Step 4 While $k \leq M$ do Steps 5–16.

Step 5 Set $x = a + h$;
$t = (w_2 - \alpha)/(2h)$;
$a_1 = 2 + h^2 f_y(x, w_1, t)$;
$b_1 = -1 + (h/2) f_{y'}(x, w_1, t)$;
$d_1 = -\left(2w_1 - w_2 - \alpha + h^2 f(x, w_1, t)\right)$.

Algoritmo

Step 6 For $i = 2, \ldots, N-1$

set $x = a + ih$;

$t = (w_{i+1} - w_{i-1})/(2h)$;

$a_i = 2 + h^2 f_y(x, w_i, t)$;

$b_i = -1 + (h/2) f_{y'}(x, w_i, t)$;

$c_i = -1 - (h/2) f_{y'}(x, w_i, t)$;

$d_i = -\left(2w_i - w_{i+1} - w_{i-1} + h^2 f(x, w_i, t)\right)$.

Step 7 Set $x = b - h$;

$t = (\beta - w_{N-1})/(2h)$;

$a_N = 2 + h^2 f_y(x, w_N, t)$;

$c_N = -1 - (h/2) f_{y'}(x, w_N, t)$;

$d_N = -\left(2w_N - w_{N-1} - \beta + h^2 f(x, w_N, t)\right)$.

Step 8 Set $l_1 = a_1$; *(Steps 8–12 solve a tridiagonal linear system using Algorithm 6.7.)*

$u_1 = b_1/a_1$;

$z_1 = d_1/l_1$.

Algoritmo

Step 9 For $i = 2, \ldots, N-1$ set $l_i = a_i - c_i u_{i-1}$;
$$u_i = b_i / l_i;$$
$$z_i = (d_i - c_i z_{i-1}) / l_i.$$

Step 10 Set $l_N = a_N - c_N u_{N-1}$;
$$z_N = (d_N - c_N z_{N-1}) / l_N.$$

Step 11 Set $v_N = z_N$;
$$w_N = w_N + v_N.$$

Step 12 For $i = N-1, \ldots, 1$ set $v_i = z_i - u_i v_{i+1}$;
$$w_i = w_i + v_i.$$

Algoritmo

Step 13 If $\|\mathbf{v}\| \leq TOL$ then do Steps 14 and 15.

Step 14 For $i = 0, \ldots, N+1$ set $x = a + ih$;
OUTPUT (x, w_i).

Step 15 STOP. (*The procedure was successful.*)

Step 16 Set $k = k + 1$.

Step 17 OUTPUT ('Maximum number of iterations exceeded');
(*The procedure was unsuccessful.*)
STOP.

Algoritmo

Ejemplos:

Ejercicio 1

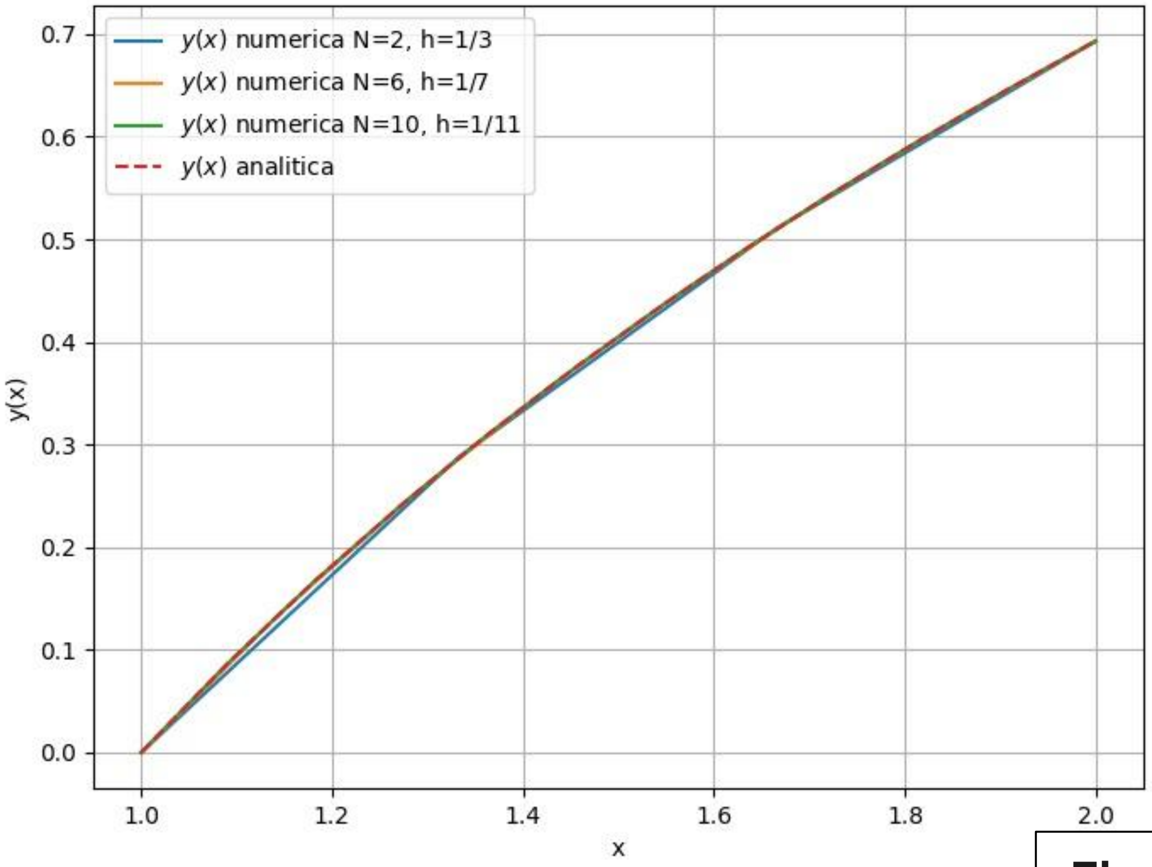
$$\boxed{y'' = -(y')^2 - y + \ln x, \quad 1 \le x \le 2, \quad y(1) = 0, \quad y(2) = \ln 2.}$$

Solución analítica

$$y = \ln x.$$

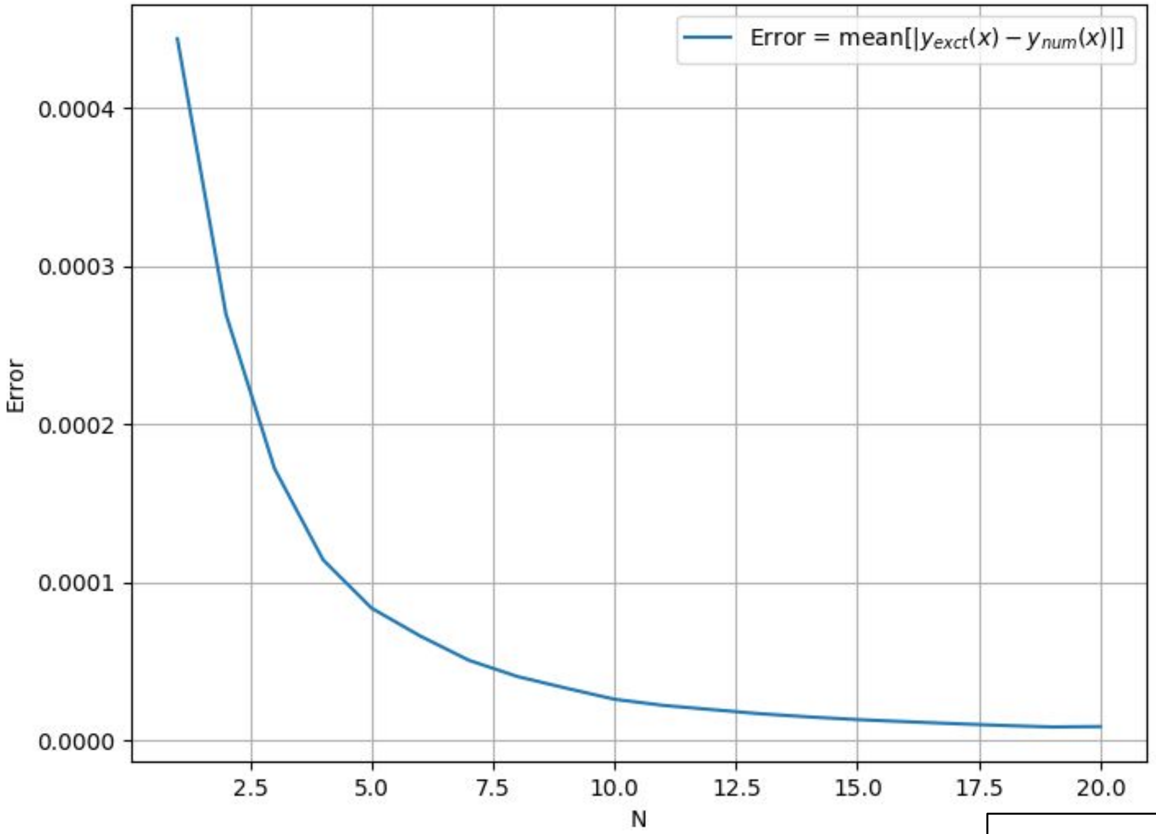
Ejercicio 1

Solución numérica



Ejercicio 1

Convergencia



Ejercicio 1

Oscilador forzado

$$M\frac{d^2x}{dt^2} = -Sx - r\frac{dx}{dt} + F(t)$$

$$F(t) = 2(1 - sint),\ M = 2kg,\ S = 1N/m,\ r = 0.3Ns/m$$

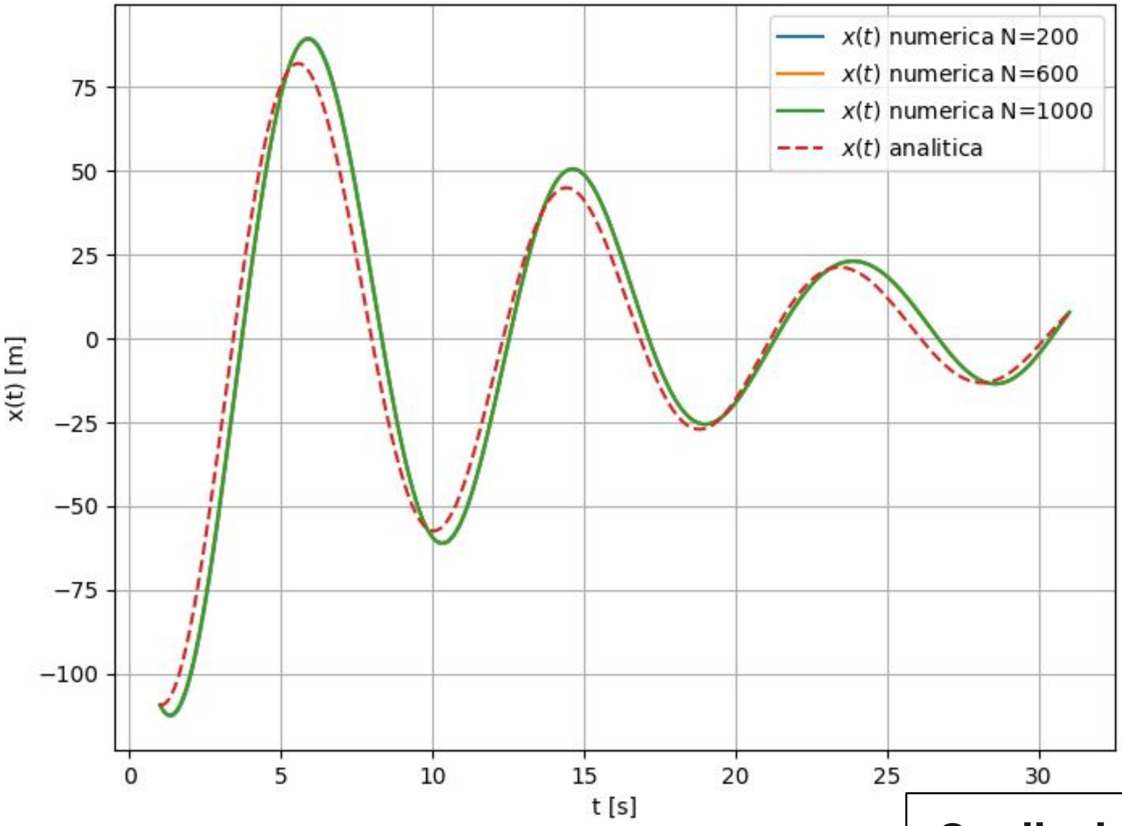
Solución analítica

$$x(t) = 32\ e^{-0.075t}(C_1 cos(0.703118t) + C_2 sin(0.703118t)) + 2 + \frac{200}{109} sin(t) + \frac{60}{109} cos(t)$$

$$C_1 = -\frac{278}{109} \text{ and } C_2 = -\frac{110425000}{38319931}.$$

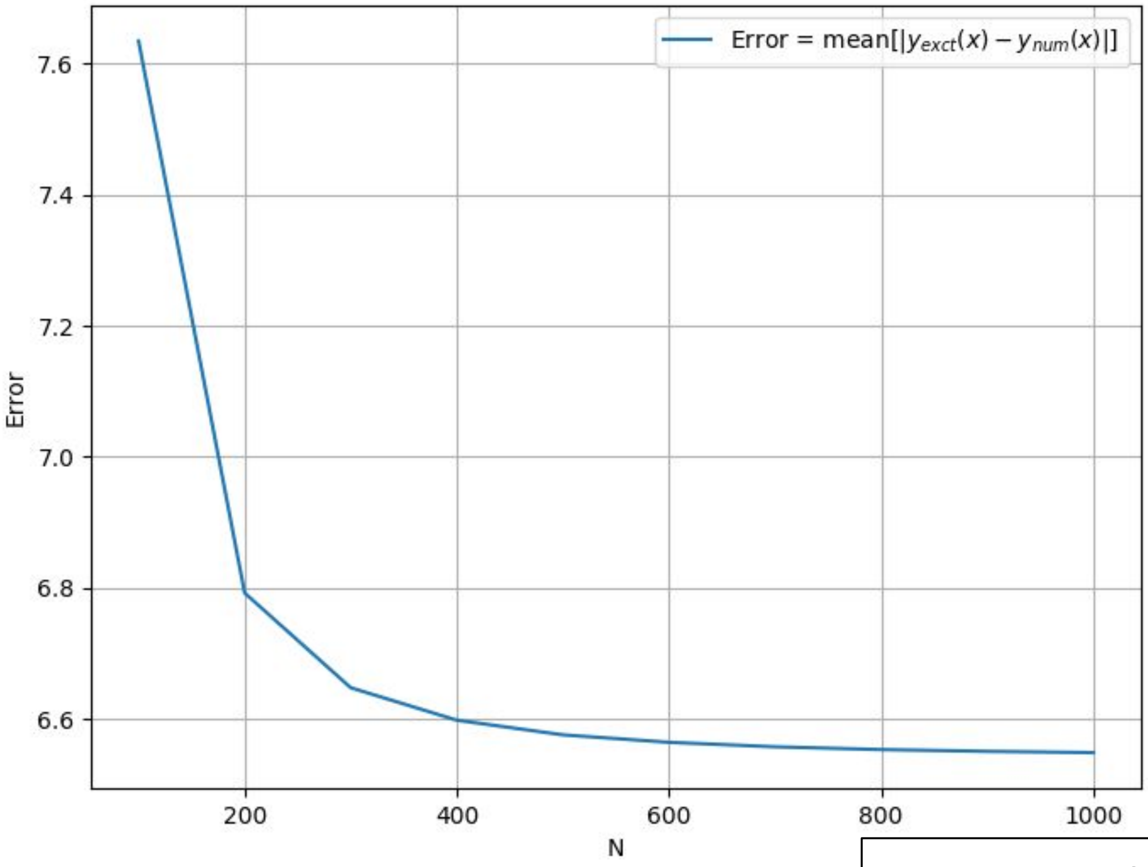
Oscilador forzado

Solución numérica



Oscilador forzado

Convergencia



Oscilador forzado

Referencias

- Burden, Richard L., and J. Douglas Faires. 2011. *Numerical analysis*. Pacific Grove, CA: Brooks/Cole Pub. Co.
- Murad, Muhammad Amin & Murad, S & Hussien, Ahmad. (2017). NUMERICAL SOLUTION OF SYSTEM OF DAMPED FORCED OSCILLATOR ORDINARY DIFFERENTIAL EQUATIONS, Cihan International Journal of Social Science, Issue,Vol. 1, No.1 , P. 22 (Jul. – Sep. 2017).. 1. 22.