Modelación y Simulación 2024

Lab 02

01.agosto.2024

1. Implementar en Python una función para hallar los ceros de una función diferenciable $F : \mathbb{R}^n \to \mathbb{R}^n$ usando el método de Newton multidimensional. Aquí

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \text{y} \qquad F(\mathbf{x}) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{pmatrix}.$$

Como parámetros su algoritmo debe recibir la función F, el vector inicial de búsqueda $\mathbf{x}_0 \in \mathbb{R}^n$, y criterios de paro $maxIter$ y $tol > 0$. Para la salida, su función debe devolver la lista de aproximaciones realizadas y el valor de punto $\mathbf{x}^*$ donde está el cero.

Con su implementación, resolver el siguiente sistema de ecuaciones con 7 cifras decimales de precisión

$$\begin{aligned} 3x - cos(yz) - \frac{1}{2} &= 0, \\ x^2 - 81(y + 0.1)^2 + \sin z + 1.06 &= 0, \\ e^{-xy} + 20z + \tfrac{10\pi - 3}{3} &= 0. \end{aligned}$$

A partir de punto inicial $\mathbf{x}_0 = (0.1, 0.1, 0.1)^T$. Imprima las iteraciones obtenidas por su algoritmo para mostrar evidencia de la convergencia.

2. Usar su implementación anterior para hallar todos los ceros de las siguientes funciones:

i) $F : \mathbb{R}^2 \to \mathbb{R}^2$, $F(x, y) = \begin{pmatrix} 3x^2 - y^2 \\ 3xy^2 - x^3 - 1 \end{pmatrix}$

ii) $G : \mathbb{R}^3 \to \mathbb{R}^3$, $G(x, y, z) = \begin{pmatrix} 12x - 3y^2 - 4z - 7.17 \\ x + 10y - z - 11.54 \\ y^3 - 7z^3 - 7.631 \end{pmatrix}$

3. Para cada una de las ecuaciones diferenciales a continuación, hallar todos los puntos críticos (puntos de equilibrio), y usar los autovalores de $DF(\mathbf{x})$ para clasificarlos de acuerdo a su tipo: (atractor, repulsor, nodo, twist, espilar, punto silla, centro). Luego, dibujar el campo vectorial asociado, marcando los puntos de equilibrio encontrados, y verificando que la visualización corresponde a la clasificación hallada.

i) $\dfrac{dy}{dx} = \dfrac{4x - 3y + 7xy}{4x + 2y + 2x^2 - 3y^2}$,

ii) $\dfrac{dy}{dx} = \dfrac{-18 + 6x + 2y - xy}{33 - 10x - 3y + x^2}$.

(la ecuación (i) posee 4 puntos de equilibrio; la ecuación (ii) tiene 3).

4. Implementar los algoritmos de Euler, de Heun, y de Taylor de orden 2 para resolver computacionalmente una EDO. Utilizar estos métodos para resolver las siguientes ecuaciones diferenciales:

a) $\frac{dy}{dx} = 22e^{x/5} - 5x - 25$, $y(0) = -3$, en el intervalo $[0, 5]$;

b) $\frac{dy}{dt} = -\sin t$, $y(0) = 1$, en el intervalo $[0, 6\pi]$.

Mostrar una gráfica de la solución numérica obtenida contra la solución exacta de la EDO, y comparar el error.
En otra gráfica mostrar el campo vectorial de la EDO, e incorporar un plot de la solución numérica obtenida por uno de los algoritmos (el más preciso).

5. Una población de animales $P(t)$ se modela por la ecuación diferencial

$$\frac{dP}{dt} = 0.0225\,P - 0.0003\,P^2; \qquad P(0) = 25. \tag{1}$$

(asuma una escala de tiempo t en meses).

a) Hallar la población límite de animales (el valor límite cuanto $t \to \infty$).

b) Utilizar algún método computacional para estimar la población de animales para 10 años a futuro, primero usando un tamaño de paso $h = 1$ año, y luego usando $h = 0.5$ años.

c) Derivado del cálculo en (b), responder ¿qué porcentaje de la población límite? se alcanza después de 5 años? ¿Y a los 10 años?