Práctica 10

Métodos Numéricos y Computación

En esta práctica, estamos interesados en hallar un valor, p, tal que $f(p) = 0$ donde f es una función real de variable real. Si $f : [a, b] \to \mathbb{R}$ es una función continua tal que $f(a)f(b) < 0$, entonces por el Teorema de Bolzano, existe al menos un punto $p \in (a, b)$ tal que $f(p) = 0$. Sin pérdida de generalidad podemos suponer que $f(x)$ solo tiene una raíz en $[a, b]$. Los métodos iterativos para la resolución de ecuaciones no lineales son aquellos en los que, a partir de un punto inicial p_0, se construye una sucesión $(p_n)_{n\in\mathbb{N}}$ de forma que $\lim_{n\to\infty} p_n = p$. En las clases de teoría hemos presentado el método de la bisección y tres métodos de punto fijo: Newton, secante y Regula-Falsi.

En primer lugar, conviene disponer de intervalos $[a, b]$ en los que estemos seguros que hay una raíz de la función ($f(a)f(b) < 0$). Un procedimiento interesante es el de búsqueda incremental que consiste en dividir el intervalo original $[a, b]$ en intervalos de pequeña longitud, h, por medio de puntos $a = x_0, x_1 = x_0 + h, x_2 = x_0 + 2h, \ldots, x_n = b$ y, empezando por la izquierda, vamos comprobando si el signo cambia en cada uno de los subintervalos $[x_k, x_{k+1}]$.

Ejercicio 1 *Implementa una función* `busqueda_incremental` *que, dada una función* `f`*, los extremos de un intervalo* `a` *y* `b`*, y el número de subintervalos* `n` *en que dividiremos* $[a, b]$*, nos devuelva los subintervalos en que están las raíces de* f*. Aplica esta función para separar las raíces de* $f(x) = e^{-x/10}\cos(x)$*, en el intervalo* $[0, 15]$ *con 50 subintervalos.*

El método de la bisección permite obtener una sucesión de subintervalos "encajados" $[a_n, b_n]$ cuya sucesión de puntos medios $(p_n)_{n\in\mathbb{N}}$ aproxima la solución buscada. Aunque hay más criterios (que veremos más adelante), el cálculo de las sucesivas iteraciones parará cuando $|a_k - b_k| < \varepsilon$ (ε se conoce como la tolerancia). Es interesante destacar que el error cometido en cada paso n del método de la bisección es:

$$|p - p_n| \leq 2^{-n+1}|a_0 - b_0|.$$

Ejercicio 2 *Implementa una función* `biseccion` *que, dada una función* `f`*, los extremos de un intervalo* `a` *y* `b` *y la tolerancia* `tol`*, devuelva la solución de* $f(x) = 0$ *mediante el método de la bisección (junto al número de pasos que han sido necesarios). Aplica esta función para calcular las raíces de* $f(x) = e^{-x/10}\cos(x)$*, utilizando los intervalos obtenidos con la función* `busqueda_incremental` *y una tolerancia de* 10^{-4}*.*

Una buena parte de los métodos de resolución de ecuaciones no lineales son de punto fijo. En términos generales, estos métodos consisten en transformar la ecuación $f(x) = 0$ en una equivalente del tipo $g(x) = x$ y, a partir de un punto inicial p_0, se obtiene $(p_n)_{n\in\mathbb{N}}$ mediante la expresión $p_{n+1} = g(p_n)$ y de forma que, bajo ciertas condiciones, cumple $\lim_{n\to\infty} p_n = p$. Por ejemplo, encontrar la solución de $\cos(x) - xe^x = 0$ es equivalente a resolver $g(x) = x$ donde $g(x) = \cos(x)/e^x$.

El cálculo de las iteraciones comenzará en el punto inicial p_0 y se detendrá cuando o bien se alcance un número máximo de iteraciones (que llamaremos *maxiter*), o bien se satisfaga algún criterio de parada en términos de la tolerancia $tol > 0$. Algunos criterios son los siguientes:

$$\begin{aligned} |p_k - p_{k-1}| &< tol, \text{ ó} && (1)\\ |p_k - p_{k-1}|/|p_k| &< tol, \text{ ó} && (2)\\ |f(p_k)| < tol. & && (3) \end{aligned}$$

Ejercicio 3 *Implementa una función* `punto_fijo` *que, dada una función* `g`*, un punto inicial* `p0`*, la tolerancia* `tol` *y el número máximo de iteraciones* `maxiter`*, devuelva la solución de $g(x) = x$ mediante el método general de punto fijo (junto al número de pasos que han sido necesarios usando el criterio de parada* (1)*). Aplica esta función para aproximar la solución de $\cos(x) = xe^x$ (lee el texto anterior en azul) utilizando $p_0 = 2$ como punto inicial, una tolerancia de 10^{-4} y un máximo de 50 iteraciones.*

El método de Newton es uno de los métodos más utilizados y tiene una ilustración geométrica muy visual. Dado un valor inicial p_0 (que debe ser establecido previamente), hallamos la recta tangente a la curva en el punto $(p_0, f(p_0))$ y consideramos como nuevo valor de la sucesión, p_1, la abscisa donde la citada recta tangente corta al eje de abscisas. La solución se obtiene como el límite de la sucesión:

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}, \quad n = 0, 1, \ldots$$

Observa que, para el cálculo de p_{n+1}, necesitamos disponer de la derivada de la función f en el punto p_n.

Ejercicio 4 *Implementa una función* `newton` *que, dada una función* `f`*, su derivada* `df`*, un punto inicial* `p0`*, la tolerancia* `tol` *y el número máximo de iteraciones* `maxiter`*, devuelva la solución de $f(x) = 0$ mediante el método de Newton (junto al número de pasos que han sido necesarios usando el criterio de parada* (1)*). Aplica esta función para calcular las raíces de $f(x) = e^{-x/10}\cos(x)$, utilizando los puntos medios de los intervalos obtenidos con la función* `busqueda_incremental` *como puntos iniciales, una tolerancia de 10^{-4} y un máximo de 50 iteraciones.*

El método de la secante trata de evitar el cálculo de la derivada sustituyendo la recta tangente construida en el punto $(p_n, f(p_n))$ por la recta secante que pasa por los puntos $(p_{n-1}, f(p_{n-1}))$ y $(p_n, f(p_n))$. Para inicializar este método se requieren, por tanto, dos puntos p_0 y p_1 y la sucesión obtenida viene dada por:

$$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p_n) - f(p_{n-1})}, \quad n = 1, 2, \ldots$$

El método de la regula-falsi introduce una ligera modificación en el método de la secante para asegurar que la siguiente iteración siempre se encontrará entre las dos anteriores.

Ejercicio 5 *Implementa funciones* `secante` *y* `regula_falsi` *que, dada una función* `f`*, puntos iniciales* `p0` *y* `p1`*, la tolerancia* `tol` *y el número máximo de iteraciones* `maxiter`*, devuelvan la solución de $f(x) = 0$ mediante el método de la secante y regula-falsi (junto al número de pasos que han sido necesarios usando el criterio de parada* (1)*), respectivamente. Aplica estas funciones para calcular las raíces de $f(x) = e^{-x/10}\cos(x)$, utilizando los intervalos obtenidos con la función* `busqueda_incremental`*, una tolerancia de 10^{-4} y un máximo de 50 iteraciones.*

Finalmente, veamos un ejemplo de aplicación de todos estos métodos.

Ejercicio 6 *Se requiere conocer el tiempo (en segundos) en que una partícula que se mueve en el espacio según el vector de posición:*

$$R(t) = (2\cos(t), \sin(t), 0),$$

se encuentra más cerca del punto $P(2, 1, 0)$ en el intervalo de tiempo $[0, 2]$. Aplica el método de la bisección, el de Newton, el de la secante y el de regula-falsi con los argumentos que consideres oportunos (haz un gráfico si es necesario).

Ejercicios para entregar

Crea un script con nombre `PrimerApellido_SegundoApellido_Nombre.py` donde resuelvas los siguientes ejercicios teniendo en cuenta las soluciones de los anteriores y los contenidos de las presentaciones de la teoría. Incluye todos aquellos comentarios que consideres oportunos. La entrega deberá realizarse de 16:00 a 20:00 horas del jueves 30 de abril desde la tarea habilitada en Moodle dentro del Tema 6 con nombre `Práctica 10 - Ejercicios para entregar`.

Ejercicio 1 *La función `busqueda_incremental` podría ser utilizada para encontrar las raíces de una función. Supongamos, por ejemplo, la ecuación $\sin(x) = \log(x)$ que tiene una raíz en el intervalo $[2, 2.5]$. ¿Serías capaz de aproximar dicha solución utilizando la función `busqueda_incremental` exclusivamente? Piensa, razona, implementa una función si es necesario (no hay una solución única) y describe tu razonamiento y/o tu función sobre el script mediante todos los comentarios que consideres oportunos. Creatividad al poder.*

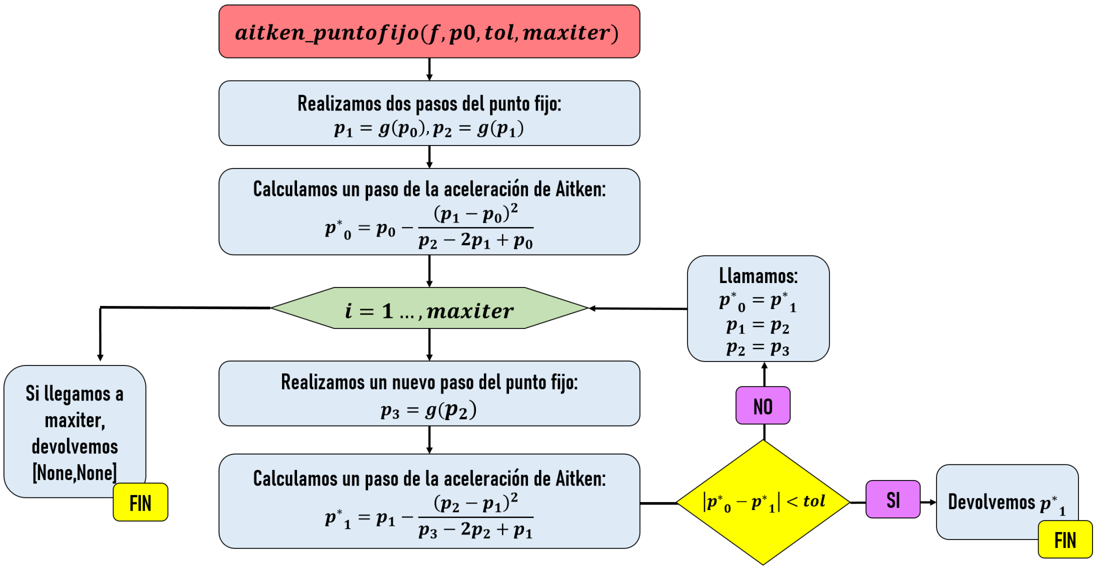
Ejercicio 2 *Una propiedad general es que si una sucesión $(p_n)_{n\in\mathbb{N}}$ cumple que $\lim_{n\to\infty} p_n = p$, entonces la sucesión $(p_n^*)_{n\in\mathbb{N}}$ donde:*

$$p_n^* = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}, \quad n = 0, 1, \ldots$$

también cumple que $\lim_{n\to\infty} p_n^* = p$, pero sabemos que lo hace más rápidamente. Esto se conoce como el método de aceleración de la convergencia de Aitken. Supongamos que $(p_n)_{n\in\mathbb{N}}$ es la sucesión de aproximaciones que proporciona el método general del punto fijo. Implementa una función `aitken_puntofijo` que, dada una función `f`, un punto inicial `p0`, la tolerancia `tol` y el número máximo de iteraciones `maxiter`, devuelva la sucesión $(p_n^*)_{n\in\mathbb{N}}$ hasta que se cumpla el criterio de parada (1).

En el ejercicio 3 vimos cómo eran necesarios 46 pasos del método de punto fijo para aproximar la solución de $\cos(x) = xe^x$ con una tolerancia de 10^{-4}. Aplica la función `aitken_puntofijo` usando la misma tolerancia, el mismo criterio de parada (tal y como estaba definido) y un máximo de 50 iteraciones. Comprueba que la convergencia es ahora más rápida.

Una posibilidad para crear esta función (y no la única) es seguir los siguientes pasos:



Ejercicio 3 *Modifica la función* `newton` *e implementa* `newton_5ptos` *de forma que la derivada se aproxime según la fórmula de los cinco puntos con* $h = 10^{-2}$ *(ya no necesitaremos* `df` *como input). Aplica esta función para calcular la solución de* $\cos(x) = xe^x$ *en el intervalo* $[0, 2]$*, utilizando el punto medio del intervalo obtenido con la función* `busqueda_incremental` *como punto inicial (escoge tú mismo/a un número de subintervalos que consideres oportuno), una tolerancia de* 10^{-4} *y un máximo de 50 iteraciones.*

Ejercicio 4 *Modifica la función* `secante` *e implementa* `secante_parada` *de forma que el usuario pueda seleccionar un criterio de parada entre* (1)*,* (2) *ó* (3)*. Aplica esta función para calcular, usando los tres criterios de parada, la solución de* $\cos(x) = xe^x$ *en el intervalo* $[0, 2]$*, utilizando como puntos iniciales los extremos del intervalo obtenido con la función* `busqueda_incremental` *(escoge tú mismo/a un número de subintervalos que consideres oportuno), una tolerancia de* 10^{-4} *y un máximo de 50 iteraciones.*

Ejercicio 5 *Aplica los métodos de la bisección, Newton, secante (con el criterio de parada* (1)*) y regula-falsi para aproximar* $\sqrt[5]{2}$ *con una tolerancia de* 10^{-4}*. Para cada función, escoge los argumentos que consideres adecuados.*