Programación: Método del punto fijo (método de iteración simple)

Objetivos. Programar el método del punto fijo.

Requisitos. Entender bien la idea del ciclo `while`. Tener preparados varios ejemplos "buenos" y "malos" que muestren la convergencia y divergencia del método. Se recomienda usar las funciones g_1, g_2, g_3, g_4, g_5 de la clase anterior.

1. Idea del método. Buscamos el punto fijo de una función dada g. A partir de un punto dado x_0 construimos $x_1, x_2, \ldots$ mediante la siguiente fórmula recursiva:

$$x_p = g(x_{p-1}).$$

2. Guardar solamente el punto actual y el anterior. En el programa no es necesario guardar todos los puntos $x_0, x_1, x_2, \ldots$. En cada momento vamos a guardar solamente el punto actual x_p denotándolo por x y el punto anterior x_{p-1} que denotamos por $xprev$.

3. Condición de terminación y condición de continuación. Denotemos por d a la distancia entre el punto actual x y el punto anterior $xprev$. El método de punto fijo debe pararse cuando d es pequeño (menor que un número dado $xtol$) o cuando el número de los pasos hechos es mayor o igual al número de pasos máximo permitido:

$$d < xtol \qquad \vee \qquad p \geq pmax. \tag{1}$$

La condición de continuación será la negación de la condición de paro (1). Recuerde la ley de De Morgan sobre la negación de la disyunción:

$$\overline{a \vee b} =$$

Escriba la condición de continuación en el método de punto fijo. Hay que continuar mientras se cumpla la siguiente condición:

$$\underbrace{\qquad\qquad\qquad}_{?} \quad \underbrace{\quad}_{?} \quad \underbrace{\qquad\qquad\qquad}_{?}$$

4. Algoritmo FixPoint (pseudocódigo).
Sustituya los signos ??? por expresiones correctas:

```
Entrada: g, x0, xtol, pmax.
Variables locales: ???.
p := ???;
d := 2 * xtol;
x := ???;
Mientras ...:
    xprev := x;
    x := ???;
    d := abs(x - xprev);
    p := p + 1;
Salida: x, p.
```

5. Problema FixPoint. Escriba una función con argumentos g, $x0$, $xtol$, $pmax$ que aplique el método de iteración de punto fijo a la función g con el punto inicial $x0$. La función debe regresar la última aproximación al punto fijo y el número de las iteraciones hechas.

6. Pruebas. Pruebe la función FixedPoint con varios ejemplos "buenos" y "malos"; se recomienda usar las funciones g_1, g_2, g_3, g_4, g_5 de la clase anterior. Ponga $pmax = 30$, $xtol = 10^{-8}$. Como la aproximación inicial $x0$ puede usar el extremo izquierdo a_k (escriba este número con el punto decimal).

- $g_1(x) = \sqrt[3]{x+1}, \quad [a_1, b_1] = [1, 2], \quad x0 = 1.0.$
- $g_2(x) = \log(2 - x), \quad [a_2, b_2] = [0, 0.8], \quad x0 = 0.0.$
- $g_3(x) = x - \dfrac{x^3 - x - 1}{3x^2 - 1}, \quad [a_3, b_3] = [1.2, 2], \quad x0 = 1.2.$
- $g_4(x) = x^3 - 1, \quad [a_4, b_4] = [1, 2], \quad x0 = 1.0.$
- $g_5(x) = 2 - e^x, \quad [a_5, b_5] = [0, 0.8], \quad x0 = 0.0.$

Por ejemplo, en Wolfram Mathematica puede escribir el programa de la siguiente manera:

```
FixPoint[g_, x0_, xtol_, pmax_] := ...
g1[x_] := (x + 1) ^ (1/3);
FixPoint[g1, 1.0, 10^(-8), 30]
```

7. Problema adicional FixedPointRecur. Escriba una función recursiva con argumentos g, $x0$, $xtol$, $pmax$ que realice el método de iteración de punto fijo. La función tiene que regresar el punto fijo y el número de las iteraciones hechas.