

# Programación: método babilónico para calcular raíces cuadradas

**Objetivos.** Programar el algoritmo babilónico para calcular raíces cuadradas.

**Requisitos.** Concepto de función contractiva, idea del método del punto fijo, ciclos while.

Se recomienda resolver estos ejercicios antes de la clase práctica.

**1. Verificación que la función es contractiva.** Sea  $a > 1$ . Definimos la función  $f_a: [\sqrt{a}, +\infty) \rightarrow [\sqrt{a}, +\infty)$  mediante la regla

$$f_a(x) = \frac{1}{2} \left( x + \frac{a}{x} \right).$$

Demuestre que para cada  $x \geq \sqrt{a}$

$$0 \leq f'(x) \leq \frac{1}{2}.$$

Simplifique la siguiente expresión:

$$f(x) - \sqrt{a} = \underbrace{\hspace{2cm}}_{?} - \sqrt{a} = \frac{1}{2x} \left( \hspace{2cm} \right) = \frac{1}{2x} \left( \hspace{2cm} \right)^2.$$

Muestre que para cada  $x \geq \sqrt{a}$

$$f(x) \geq \sqrt{a}.$$

**2. Estructura de la función.**

```
function [x, niter] = babel(a, tol, maxiter),
    x = a;
    er = tol + 1;
    niter = 0;
    while ???,
        xprev = x;
        x = 0.5 * (x + a / x);
        er = abs(x - xprev);
        niter = niter + 1;
    end
end
```

Aquí  $a$  es el número dado (se supone que  $a > 1$ ),  $x$  es una aproximación de  $\sqrt{a}$ ,  $\text{tol}$  es el error aproximado permitido,  $\text{maxiter}$  es el número máximo de iteraciones permitidas y  $\text{niter}$  es el número de iteraciones realizadas. Todavía nos falta escribir la condición en el ciclo `while`.

**3. Condición de terminación y condición de continuación.** El ciclo debe **terminarse** si se cumple **alguna** de las siguientes dos condiciones:

- el número de las iteraciones realizadas es mayor o igual que el número máximo de iteraciones permitidas:

$$\text{niter} \geq \text{maxiter}$$

- la diferencia entre las dos aproximaciones sucesivas es menor que la “tolerancia” dada:

$$\text{er} < \text{tol}$$

Escriba la condición de terminación como una expresión lógica compuesta. Elija la operación lógica adecuada (la conjunción  $\wedge$  o la disyunción  $\vee$ ):

$$(\text{niter} \geq \text{maxiter}) \quad \underbrace{\quad}_{\text{¿}\wedge \text{ o } \vee?} \quad (\text{er} < \text{tol})$$

Escriba la condición de **continuación** del ciclo. Niegue la condición anterior y aplique la ley de De Morgan:

$$(\text{niter} < \text{maxiter}) \quad \underbrace{\quad}_{\text{¿}\wedge \text{ o } \vee?} \quad \underbrace{\quad}_{?}$$

Complete la función `babel`. Recordamos que las operaciones lógicas  $\wedge$  y  $\vee$  se denotan como `&&` y `||` en muchos lenguajes de programación.

**4. Pruebas.** Complete la función `babel` y haga pruebas:

```
[x, niter] = babel(9, 0.01, 100)
```

```
[x, niter] = babel(9, 0.000001, 2)
```