

**Laboratorio I**  
**Cálculo Numérico**  
**Bisección, Regla Falsa, Secante y Newton**

## 1 Introducción

La cantidad para pagar una hipoteca en un periodo fijo de tiempo está dada por **la ecuación de anualidad ordinaria** :

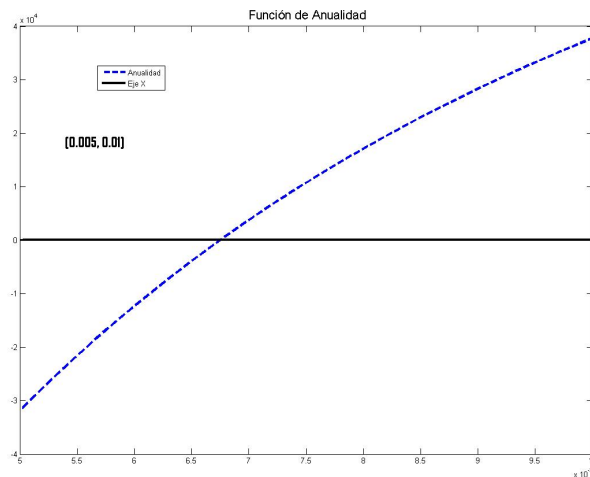
$$A = \frac{P}{i}[1 - (1 + i)^{-n}],$$

donde  $A$  es la cantidad de la hipoteca,  $P$  es la cantidad en cada pago periódico,  $i$  es la tasa de interés por periodo y  $n$  es el número de pagos. Supongamos que la hipoteca es a 30 años por la cantidad de \$135, 000 con pagos mensuales de \$1, 000 al mes. ¿Cuál es la tasa de interés  $i$  que necesita el cliente?

Sea  $x = i$  y definamos la función:

$$f(x) = 135000 - \frac{1000}{x}[1 - (1 + x)^{-360}],$$

se necesita  $x^*$  tal que  $f(x^*) = 0$ .



## 2 Bisección

Sea  $f : [a, b] \rightarrow \mathbb{R}$  continua tal que  $f(a) * f(b) < 0$  y  $f(x)$  tiene una única raíz en  $[a, b]$ .

---

Fijar:  $tol$  tolerancia en la longitud del intervalo.

Fijar  $maxkiter$  número máximo de iteraciones.

Evaluar:  $fa \leftarrow f(a)$ ,  $fb \leftarrow f(b)$ .

Mientras ( $fa * fb < 0$ , y,  $(b - a) < tol$ , y,  $kiter < maxkiter$ )

$c \leftarrow \frac{a+b}{2}$ ,  $fc \leftarrow f(c)$

Si  $f(a)f(c) < 0$

$b \leftarrow c$ ,  $fb \leftarrow fc$

de otro modo

$a \leftarrow c$ ,  $fa \leftarrow fc$

Fin de Si

$kiter \leftarrow kiter + 1$

Fin de Mientras

Si ( $fa * fc * fb > 0$ )

$x^* \leftarrow \frac{a+b}{2}$

de otro modo

$x^* \leftarrow c$

Fin de Si

Variables de salida:  $x^*$ ,  $kiter$

---

### 3 Regla Falsa

Sea  $f : [a, b] \rightarrow \mathbb{R}$  continua tal que  $f(a) * f(b) < 0$  y  $f(x)$  tiene una única raíz en  $[a, b]$ .

---

Fijar:  $tol$  tolerancia en la longitud del intervalo.

Fijar  $maxkiter$  número máximo de iteraciones.

Evaluar:  $fa \leftarrow f(a)$ ,  $fb \leftarrow f(b)$ .

Mientras ( $fa * fb < 0$ , y,  $(b - a) < tol$ , y,  $kiter < maxkiter$ )

$$m \leftarrow \frac{fb-fa}{b-a}$$

$$c \leftarrow a - \frac{fa}{m}, \quad fc \leftarrow f(c)$$

Si  $f(a)f(c) < 0$

$$b \leftarrow c, \quad fb \leftarrow fc$$

de otro modo

$$a \leftarrow c, \quad fa \leftarrow fc$$

Fin de Si

$$kiter \leftarrow kiter + 1$$

Fin de Mientras

Si ( $fa * fc * fb > 0$ )

$$x^* \leftarrow \frac{a+b}{2}$$

de otro modo

$$x^* \leftarrow c$$

Fin de Si

Variables de salida  $x^*$ ,  $kiter$

---

## 4 Secante

Sea  $f : [a, b] \rightarrow \mathbb{R}$  continua tal que  $f(x)$  tiene una única raíz en  $[a, b]$ .

---

Fijar:  $tol$  tolerancia en la longitud del intervalo.

Fijar  $maxkiter$  número máximo de iteraciones.

Escoger dos puntos iniciales  $a1, a2 \in [a, b]$ .

Evaluar:  $f1 \leftarrow f(a1), f2 \leftarrow f(a2)$ .

Mientras ( $|f(a2)| > 0$ , y,  $kiter < maxkiter$ )

$$m \leftarrow \frac{f2-f1}{a2-a1}$$

Si  $m \neq 0$ .

$$c \leftarrow a1 - \frac{f1}{m}$$

$$a1 \leftarrow a2, a2 \leftarrow c.$$

$$f1 \leftarrow f2, f2 \leftarrow f(c).$$

$$kiter \leftarrow kiter + 1.$$

de otro modo

Parar pendiente igual a cero

Fin de Si

Fin de Mientras

$$x^* \leftarrow a$$

Variables de salida  $a_2, kiter$

---

## 5 Newton

Sea  $f : [a, b] \rightarrow \mathbb{R}$  continuamente diferenciable tal que  $f(x)$  tiene una única raíz en  $[a, b]$ .

---

Fijar:  $tol$  tolerancia en la longitud del intervalo.

Fijar  $maxkiter$  número máximo de iteraciones.

Escoger un punto inicial  $a1 \in [a, b]$ .

Escoger  $h = 10^{-5}$ .

Evaluar:  $f1 \leftarrow f(a1)$ .

Mientras ( $|f(a1)| > 0$ , y,  $kiter < maxiter$ )

$a2 \leftarrow a1 + h$ ,  $f2 \leftarrow f(a2)$

$m \leftarrow \frac{f2-f1}{a2-a1}$  aproximación a la derivada.

Si  $m \neq 0$ .

$c \leftarrow a1 - \frac{f1}{m}$

$a1 \leftarrow c$ .

$f1 \leftarrow f(c)$ .

$kiter \leftarrow kiter + 1$ .

de otro modo

Parar pendiente igual a cero

Fin de Si

Fin de Mientras

$x^* \leftarrow a$

Variables de salida  $x^*$ ,  $kiter$

---

## 6 Resultados Numéricos

Programar en Matlab:

1. `function [fx] = anualidad(x)`  
La función de anualidad ordinaria.
2. `function [x, kiter] = biseccion(fname, a, b)`  
Método de bisección. Tolerancia interna de  $10^{-12}$ .
3. `function [x, kiter] = reglafalsa(fname, a, b)`  
Método de regla falsa. Tolerancia interna de  $10^{-12}$ .
4. `function [x, kiter] = secante(fname, a1, a2)`  
Método de la secante. Tolerancia interna de  $10^{-12}$ .
5. `function [x, kiter] = newton(fname, a1)`  
Método de Newton. Tolerancia interna de  $10^{-12}$ .

### Comandos:

```
[x, kiter] = biseccion('Anualidad', 0.006, 0.008);  
[x, kiter] = reglafalsa('Anualidad', 0.006, 0.008);  
[x, kiter] = secante('Anualidad', 0.007, 0.008);  
[x, kiter] = newton('Anualidad', 0.008);
```

### Resultados Numéricos

Método	$x$	$kiter$
Bisección	0.00674991715979	31
Regla Falsa	0.00674991715907	13
Secante	0.006749917159071	8
Newton	0.006749917159072	5