



Análisis Numérico



Métodos De Regula Falsi & Newton - Raphson

Índice :




1. [Corriendo el Programa](#)
2. [Ejemplo de un Problema de Regula Falsi](#)
3. [Estructura del Programa 'Sorces Codes'](#)
4. [Testing](#)
5. [Autor](#)
6. [Garantía](#)



A CONTINUACIÓN HAREMOS CORRER EL PROGRAMA DE ANÁLISIS NUMÉRICO EN NUESTRA CALCULADORA

1. Ingrese a las Bibliotecas instaladas en nuestra calculadora, presionando  y luego 
2. La pantalla cambiará a la siguiente:




3. Se muestran dos bibliotecas una es un juego [MARIO] y la otra es nuestro programa : [ANALISIS]
4. Dé un clic sobre  ó  ó sobre la tecla de Función asociada con esta pestaña ()
5. Ya se encuentra dentro del programa, la nueva pantalla es la siguiente:



6. Para ejecutar el programa, haga clic sobre  ó  ó simplemente presiona la tecla [A] en el teclado de su PC...
7. Obtendrá:

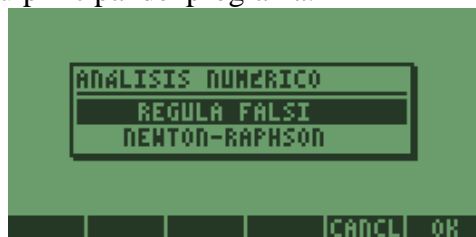




8. Presione sobre  simplemente presiona la tecla [ENTER] en el teclado de su PC, para ejecutar el programa de Análisis Numérico.
9. Saldrá una ventana de Bienvenida:




10. Presione cualquier tecla de la calculadora ó del teclado de su PC para continuar....

11. Estará ahora dentro del Menú principal del programa:



12. Con los cursores,  ó , elija que programa quiere ejecutar, el de Regula Falsi ó el de Newton Raphson.

13. A continuación haga clic sobre:  ó  ó simplemente presiona la tecla [ENTER] en el teclado de su PC

EJEMPLO PARA LA SOLUCIÓN DE UN PROBLEMA MEDIANTE EL MÉTODO DE LA REGULA FALSI

Ejm.

- Aproximar con el método de la Regula Falsi, las raíces de la ecuación:

$$X^2 - 10 \text{Log}x - 3 = 0$$

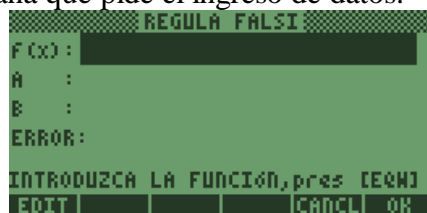
Que se encuentra en el intervalo $(2,3)$; con un error menor o igual a $e = 10^{-2}$ à $e \leq 0.01$




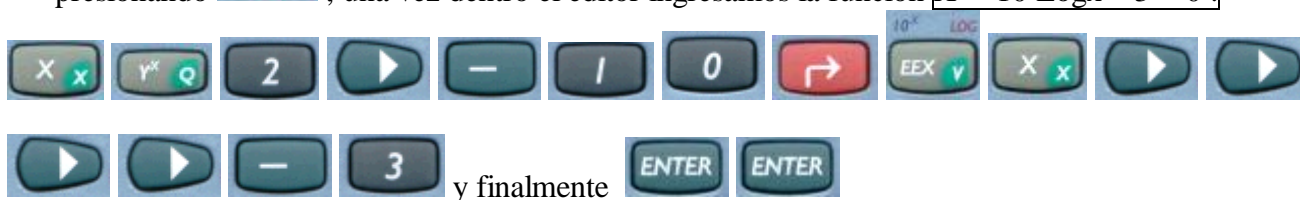
1. Ingrese al programa Regula Falsi



2. Saldrá inmediatamente la ventana que pide el ingreso de datos:



3. Introducimos la ecuación, para que sea + fácil nos vamos al editor de ecuaciones de la calculadora, presionando , una vez dentro el editor ingresamos la función $X^2 - 10 \text{Log}x - 3 = 0$:



4. Ingresamos ahora el límite inferior 'A' de nuestro intervalo $(a,b) = (2,3)$



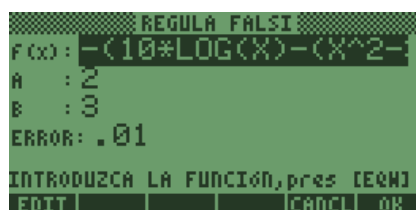
5. Ingresamos ahora el límite Superior 'B' de nuestro intervalo $(a,b) = (2,3)$



6. Introduzcamos el error admisible: $\epsilon \leq 0.01$



7. ya estan todos los datos necesarios:



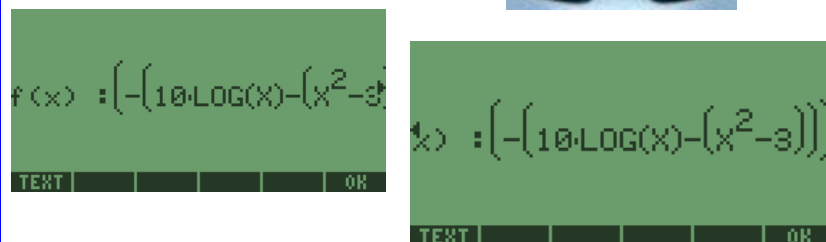
8. hagámoslo correr!



9. podríamos trabajar con 5 decimales

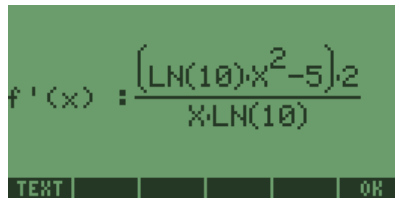


10. A continuación nos presenta la funcion $f(x)$, para verla nos desplazamos por la pantalla mediante



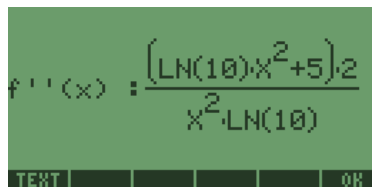
11. Espera que presionemos la tecla .

12. Nos muestra la derivada de $f(x)$, que es $f'(x)$:

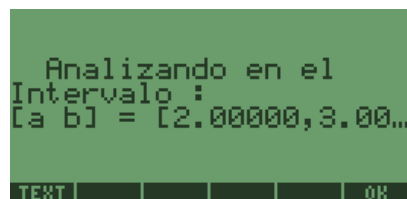

$$f'(x) = \frac{(LN(10) \cdot x^2 - 5) \cdot 2}{x \cdot LN(10)}$$

13. Espera que presionemos la tecla .

14. Nos muestra la segunda derivada de $f(x)$, que es $f''(x)$:


$$f''(x) = \frac{(LN(10) \cdot x^2 + 5) \cdot 2}{x^2 \cdot LN(10)}$$

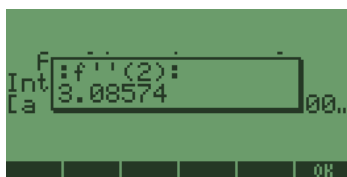
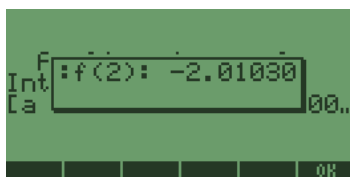
15. Espera que presionemos la tecla .



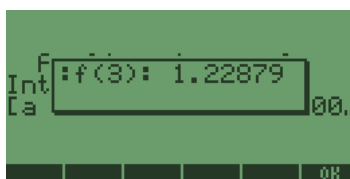
Analizando en el
Intervalo :
[a b] = [2.000000, 3.000000]

16. trabajara con el intervalo indicado: 

17. Evalúa los limites inferior y superior en la función y en su segunda derivada:



$f(2) = -2.01030$ $f''(2) = 3.08574$

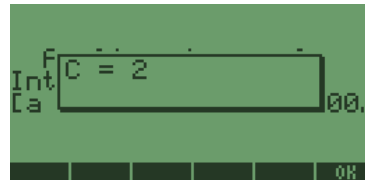


$f(3) = 1.22879$ $f''(3) = 2.48255$


18. claramente se ve que el método converge por el límite superior 3, donde se tienen signos = (+)
veamos que dice el programa:

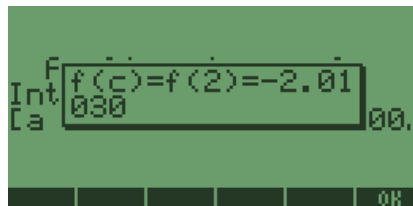


Correcto!

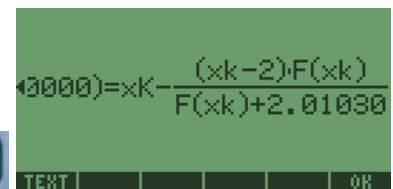
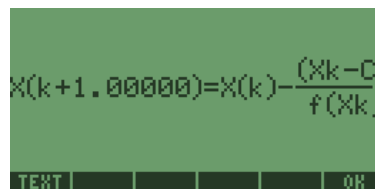
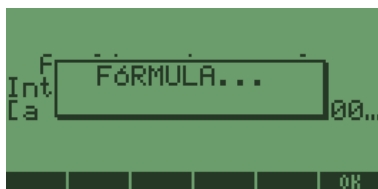


19. Presionamos  y ahora nos indica :

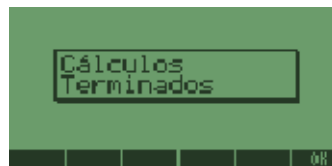
20.  à evalua $C = 2$ en $f(x)$



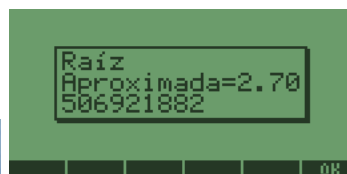
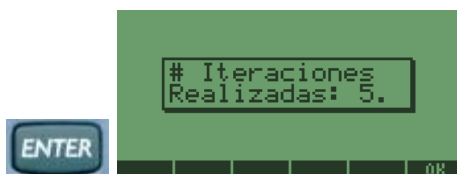
21.  à Nos indica la Fórmula de la Regula Falsi :




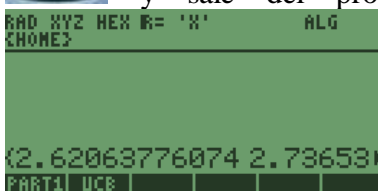
22.  y listo,



23. veamos que se obtiene:



24.  y sale del programa, dejando en pantalla las iteraciones parciales realizadas



pueden visualizarse à



Cuando en la parte superior de la pantalla salga , significa que el programa está trabajando, esperar.

- Proceder de manera similar para utilizar el método de Newton Raphson

Estructura del Programa:

El programa consta de 5 subprogramas detallados a continuación:

1. Menú inicial:

Llamado con la variable **PART1**



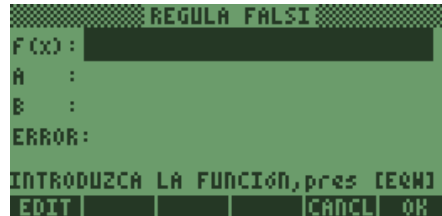
nos permite elegir que método de solución utilizar,

source – code:

```
« UCB
"ANÁLISIS NUMÉRICO"
{ {
"  REGULA FALSI"
1. } {
"  NEWTON-RAPHSON"
2. } } 1. CHOOSE
IF 1. ==
THEN à J
«
CASE J 1. ==
THEN SECA
END J 2. ==
THEN NEWRAP
END
END
»
ELSE
"ELIGE UNA OPCIÓN"
DOERR
END
»
```

2. SubMenú para el método de regula Falsi (prog llamado **SECA** **β** Secante = regula falsi)

En el cual se pide el ingreso de datos:



Source Code:

```
« "REGULA FALSI" {  
  { "f(x):"  
  "INTRODUZCA LA FUNCIÓN,pres [EQW]"  
  } { "A :"  
  "PUNTO a del Intervalo (a,b)"  
  } { "B :"  
  "PUNTO b del Intervalo (a,b)"  
  } { "ERROR:"  
  "ejm: 0.01" } } {  
  1. 0. } { } { }  
  INFORM  
  IF  
  THEN EVAL  
  "Número de decimales  
  para el Cálculo?"  
  "" INPUT STR$à FIX  
  SECAUNO  
  END  
  »
```


3. Programa SECAUNO, en el cual luego de ingresados los datos, empieza el programa...

Source Code:

```
« 'E' STO 0. 'Xo'
STO 103. NEG CF • F
A B
  « F "f(x) " • TAG
SCROLL F DERVX
"f(x) " • TAG
SCROLL F DERVX
DERVX "f"(x) "
• TAG SCROLL CLLCD
2000. .11 BEEP
" Analizando en el
Intervalo :
[a b] = "
[ 'A' 'B' ] • NUM +
3. DISP 1. WAIT F
'X' A = SUBST EVAL
• NUM DUP RE SIGN
'SIGNA' STO "f(" A
+ ")" + • TAG "" +
MSGBOX F DERVX
DERVX 'X' A = SUBST
EVAL • NUM DUP RE
SIGN 'SIGNAP' STO
"f(" A + ")" +
• TAG "" + MSGBOX F
'X' B = SUBST EVAL
• NUM DUP RE SIGN
'SIGNB' STO "f(" B
+ ")" + • TAG "" +
MSGBOX F DERVX
DERVX 'X' B = SUBST
EVAL • NUM DUP RE
SIGN 'SIGNBP' STO
"f(" B + ")" +
• TAG "" + MSGBOX
-103. CF
  IF SIGNAP SIGNA
SAME
  THEN A 'Xk' STO
B 'C' STO
  ELSE B 'Xk' STO
A 'C' STO
  END
"INICIO CON Xk=" Xk
+ MSGBOX "C = " C +
MSGBOX "f(c)=f(" C
```

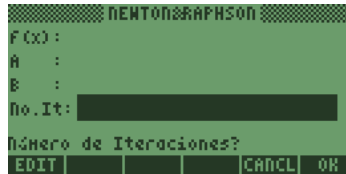
```

+ ")" = " + F 'X' C =
SUBST EVAL • NUM DUP
'Fc' STO + MSGBOX
" FÓRMULA..."
MSGBOX 'X(k+1.)=X(k
)-(Xk-C)*f(Xk)/(f(
Xk)-f(C))' SCROLL '
X(k+1.)' xK xk C -
'F(xk)' * 'F(xk)'
Fc - / - = SCROLL
CLEAR CLLCD 0. 'T
STO
DO
  « Xk DUP C -
  F 'X' Xk = SUBST
  EVAL • NUM * F 'X'
  Xk = SUBST EVAL
  • NUM Fc - / - EVAL
  'Xk' STO Xk Xo -
  ABS 'EE' STO Xk
  'Xo' STO Xk EVAL I
  1. + 'T' STO
    » EVAL
  UNTIL EE E <
  END STD 2500.
.1 BEEP
"Cálculos Terminados"
MSGBOX CLLCD
"# Iteraciones Realizadas: "
I + MSGBOX
"Raíz Aproximada="
Xk + MSGBOX { Xo Fc
C Xk SIGNBP SIGNB
SIGNAP SIGNA E EE I
} PURGE
»
»

```

4. SubMenú para el método de Newton Raphson En el cual se pide el ingreso de datos:

Llamado con la variable NEWRAP



Source Code:

```
« "NEWTON&RAPHSON"
{ { "f(x):"
"INTRODUZCA LA FUNCIÓN,pres [EQW]"
} { "A  : "
"PUNTO a del Intervalo (a,b)"
} { "B  : "
"PUNTO b del Intervalo (a,b)"
} { "No.It:"
"Número de Iteraciones?"
} } { 1. 0. } { } {
} INFORM
IF
THEN EVAL
"Número de decimales
para el Cálculo?"
"" INPUT STRaà FIX
NEWUNO
END
»
```

5. Programa NEWUNO, en el cual luego de ingresados los datos, empieza el programa...

Source Code:

```
« T STO 0. 'Xo'
STO 103. NEG CF • F
A B
« F "f(x) " • TAG
SCROLL F DERVX
"f(x) " • TAG
SCROLL F DERVX
DERVX "f"(x) "
• TAG SCROLL CLLCD
1500. .1 BEEP
" Analizando en el
Intervalo :
[a b] = "
[ 'A' 'B' ] • NUM +
3. DISP 1. WAIT F
'X' A = SUBST EVAL
• NUM DUP RE SIGN
'SIGNA' STO "f(" A
+ ")" + • TAG "" +
MSGBOX F DERVX
DERVX 'X' A = SUBST
EVAL • NUM DUP RE
SIGN 'SIGNAP' STO
"f(" A + ")" +
• TAG "" + MSGBOX F
'X' B = SUBST EVAL
• NUM DUP RE SIGN
'SIGNB' STO "f(" B
+ ")" + • TAG "" +
MSGBOX F DERVX
DERVX 'X' B = SUBST
EVAL • NUM DUP RE
SIGN 'SIGNBP' STO
"f(" B + ")" +
• TAG "" + MSGBOX
-103. CF
IF SIGNAP SIGNA
SAME
THEN A 'Xk' STO
ELSE B 'Xk' STO
END
"INICIO CON Xk=" Xk
+ MSGBOX
"FORMULA..." MSGBOX
'X(k)=X(k-1.)-f(Xk-
1.) / f(Xk-1.)'
```

```

SCROLL 'X(k+1.)'
'Xk' F DUP DERVX /
- = SCROLL CLEAR
CLLCD 0. 'NIT' STO
DO
  « Xk EVAL F
'X' Xk = SUBST EVAL
• NUM F DERVX 'X' Xk
= SUBST EVAL • NUM /
- 'Xk' STO Xk Xo -
ABS 'EE' STO Xk
'Xo' STO Xk EVAL
NIT 1. + 'NIT' STO
  » EVAL
UNTIL I NIT ==
END STD 1500.
.1 BEEP
"Cálculos Terminados"
MSGBOX CLLCD
"# Iteraciones Realizadas: "
I + MSGBOX
"Raíz Aproximada="
Xk + MSGBOX { EE
NIT Xk SIGNBP SIGNB
SIGNAP SIGNA Xo I }
PURGE
»
»

```

5. **Ultimo Programa, que despliega un GROB** (objeto gráfico en pantalla) con información sobre nosotros, los autores:

Guardado como variable **UCB**



Source code:

« STD

GROB

131

64

```
FFFFFFFFFFFFFFFFFFFFFFFF70FFFFFFFFFFFFFFFFFFFFFFFF70FFFF00C
FFFFFFFFFFFFFFFFFFFFFFFF70FFFF1211EFFFBC578801FCAB610162C70FFFF0802CFFF9CE4
FEEDDFCAB4DDABBD70FFF709069FFFAA65FE8D1FAAB014ADDD70FFF180041EFF8664FE7D
FE6A59C1BDCD70FFFC6CF82CF77EA300800FE6490C022E70FF70030388FFFFFFFFFFFFFFFF
FFFF70FF38620910FFFFFFFFFFFFFFFFFFFFFFFF70FF11900423EFFFFFFFFFFFFFFFFFFFFFFFF70FF1A
149164EFFFFFFFFFFFFFFFFFFFFFFFF70FF00810402CFFFFFFFFFFFFFFFFFFFFFFFF70FF84297494D7CF
FFFFFFFFFFFFFFFF70F702E2091987BF5FFFFFFFFFFFFFFFF70F7450004A29744CFFFFFFFFF
FFFFFFFF70F30900110247757FFFFFFFFFFFFFFFF70F38010102417747FFFFFFFFFFFFFFFF70F
1852010C66EFFFFFFFFFFFFFFFF70F9480010086EFFFFFFFFFFFFFFFF70F14E0010092
EFFFFFFFFFFFFFFFF70FCA05110241CFFFFFFFFFFF3CFFFFFFFF70F8245290825DFFFFFFFFF
FFFFFFFF70F4202010805CFF0806FCE08DB1FFFFFFFF70F8602000081CFFE6DA7BEC9D9DF
FFFF70F400AF00200DFF02DC77E04DA1FFFFFFFF707618188040A8FFE8D8776DDC8FEFFFFFF70741
0000200ABFF06027811C570FFFFFFFF70731204044129FFFFFFFFFFFFFFFF705618040C42ABFF
FFFFFFFFFFFFFFFF70771000000028FFFFFFFFFFFFFFFF70731000100028FFFFFFFFFFFFFFFF
FFFFFFFF7070100000002BFFFFFFFFFFFFFFFF70731002000228FF1CD56F371D3C8F6F2A507
710020000A9FFBDD46FD3DCBDEF47A9605010090001A9FFBD555FE55CDD8F0BA070741241200
028FFAD643F611DDD7792226073122020002BFF83A37F1EC026879C9940761648F002A9FFFFFF
FFFFFFFF70F0044D81040CFFFFFFFFFFFFFFFF70F220498E105DFFFFFFFFFFFFFFFF
FFFF70F628CA30001CFFFFFFFFFFFFFFFF70F8202E1800DCFFFFFFFFFFFFFFFF70F03
00E04003CFF1103248740B081EFFF70F5400A01080EFFFDDDA77B7BEAE6DEFFF70F1409070080E
FF11444BB79EA02EEFFF70F1809020040EFFF70F196BB7CECE8EEFFF70F390E070042FFF01C024C
738C061FFF70FB310100027FFFFFFFFFFFFFFFF70F711003002AFFFFFFFFFFFFFFFF
F70F7020CC2018FFFFFFFFFFFFFFFF70FF040F0580CFFFFFFFFFFFFFFFF70FF000C
E000CFFFFFFFFFFFFFFFF70FF18100060EFFF81664881E110100FEF70FF10100020EFFF6D
A5BEEDDCDCBD6EF70FF30600810FFFF2EA3961EE1140A5AEF70FF70030308FFFF8EA3C6AE
EDD1DB13EF70FFF04CF00CFFF61343081FD1C00CCDF70FFF171063EFFFFFFFFFFFFFFFF
FF70FFF7A4321D4EFE4E6FE7E6CFF7E6745C60FFF8530457DF67D67D75DDFF755F6D650FFF
1CD0657EFF6E77C76CE7C765F6D640FFFF00C757DFF5F77D775FFF755FEE650FFFFFFFF7C4EF
76F74D775CFF7E64C6450
```

PICT STO { # 0h

0h } PVIEW KEY

WAIT DROP ERASE

CLEAR

»

Nota Importante .-

El programa fue elaborado y testado con la ROM 1.19-6 (2001), puede que no funcione con sistemas operativos anteriores, para ver la ROM que lleva su calculadora escriba VERSIÓN y presione [ENTER]:

```
RAD XYZ HEX R= 'X'  
[HOME]  
4:  
3:  
2: "Version HP49-B  
   Revision #1.19-6"  
1: "Copyright HP 2001"  
ARIO ANALI :0: :1: :2: |
```

Autor .-

Este programa fue elaborado como proyecto final de la materia Análisis Numérico, en la Universidad Católica Boliviana 'San Pablo' (La Paz - Bolivia) con el catedrático Ing. Jose Gil...

Juan Carlos Mark Quiroga FAQ'S :mailto:jcmq_@hotmail.com?subject=Frequently Asqued Questions?

Erik Dueñas

Rodrigo Pereira <mailto:thef1boy@hotmail.com?subject=ROD>

Sergio Rivero <mailto:srivero@hotmail.com?subject=SECO>

Garantía

Provided as Freeware for private non-commercial or educational use.

You are granted the right to use and to make an unlimited number of copies of this software

Provided "as-is"

No warranty of any kind is expressed or implied

The author:

JCM_ (2004)

©2003