

WUOLAH



esthergargal

www.wuolah.com/student/esthergargal



369

maxima-para-noobs-hasta-biseccion.pdf

MÁXIMA PARA NOOBS



1º Cálculo



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

MÁXIMA PARA NOOBS

operación, `!` imprime la salida por pantalla

operación `$` elimina la salida del resultado por pantalla

`numer: true/false` modo numérico o exacto

`float (número)` expresión decimal de un número

`bfloat (número)` número con precisión real grande \rightarrow `fpprec: 24,` cambia la precisión de `bfloat`

funciones usuales

`sqrt()` `sin()` `diff()` derivada

`exp()` `cos()` `integrate()`

`log()` `tan()` `atan()` `arctg`

DEFINIR FUNCIONES

`define (func, (expr, var))`

\rightarrow `define (f(x), (x+3, x)),`

`func := expresión`

\rightarrow `f(x,y) := x^2 + y,`

DIBUJAR FUNCIONES

menú \rightarrow Tramado \rightarrow Tramado 2D

`wplot2d (f(x), [var, min, max]),`

\rightarrow `wxplot (f(x), [x, -5, 5]),`

\rightarrow `wxplot ([x^2, x-3], [x, 0, 5]),`

`draw2d (explicit (func, var, min, max)),` dibuja gráficos en forma implícita

constantes

`%e` `e`

`%pi` `π`

`%i` unidad imaginaria

opciones

`color = color`

`line_width = número`

`varrange = [min, max]` acotar una variable `var`

dibujar puntos $\left\{ \begin{array}{l} \text{points} ([x, y], [a, b]) \\ \text{point_type} = \text{num} \quad \text{tipo} \\ \text{point_size} = \text{num} \quad \text{tamaño} \\ \text{points} ([x_1, x_2], [y_1, y_2]) \end{array} \right.$

VARIABLES

`var: num` asigna a `var` el valor de `num`

`var :: [x, y, z]` lo mismo pero evalúa ambos lados de la expresión

`kill (all)` $\left\{ \begin{array}{l} \text{elimina el valor} \\ \text{de una o todas} \\ \text{las variables} \end{array} \right.$

`remvalue (var)`

`remvalue (all)`

LISTAS

`nombrelista: [a, b, c]` `a, b, c` pueden ser puntos, números, funciones...

`nombrelista [posición]` podemos referirnos a una posición concreta de la lista `nombrelista`

`makelist (regla, contador, valor inicial, valor final)`

\rightarrow `makelist (n^2, n, 1, 10),` cuadrados de los naturales del 1 al 10.

BLOQUES

(comando1, comando2) puedes separar comandos por comas

block (
 [a, b], variables locales
 local (f), funciones locales
 f(x):= expr, definir función
); produce salida de la solución del bloque

CONDICIONALES Y BUCLES

if condición then expresión1 else expresión2

for contador:min thru max do expresión

while condición do expresión se repite mientras se cumpla condición

unless condición do expresión se repite a menos que se cumpla condición

ejemplo

```
valores: [1, 3, 6, 9],  
suma: 0,  
for i in valores do (suma.suma+i),  
suma,
```

print ()
disp ()
display ()

} permiten sacar resultados intermedios por pantalla

ECUACIONES

nombre_ecuación: expresión1 = expresión2

lhs (nombre_ecuación) se queda con el lado izquierdo de la ecuación nombre_ecuación

rhs (nombre_ecuación) se queda con el lado derecho de la ecuación nombre_ecuación

$$\underbrace{x}_{\text{lhs}} = \underbrace{3+5}_{\text{rhs}}$$

solve (expresión, var) resuelve ecuaciones de grado 1-4 de forma exacta

var: solve (expresión, var) hace una lista con las soluciones
sol: map (rhs, var)

solve ([expresión1, expresión2], var) resuelve sist. de ecuaciones para dibujarlas usamos implicit con wxdraw2d

implicit (expresión1, x, $\underbrace{\text{min}, \text{max}}_{\text{rango } x}$, y, $\underbrace{\text{min}, \text{max}}_{\text{rango } y}$)

to_poly_solve (expresión, var) resuelve ecuaciones de grado ≥ 5 de forma aproximada

find_root ($\underbrace{\text{función}}_{f(x)}$, var, $\underbrace{\text{min}, \text{max}}_{\substack{\text{rango en el que} \\ \text{queremos buscar}}}$) aplicación del Teorema de los ceros de Bolzano
↳ busca puntos donde la función = 0

ERRORES absolutos y relativos

- **ABSOLUTO** $|\text{valor exacto} - \text{valor aproximado}|$
- **RELATIVO** $|\text{valor exacto} - \text{valor aproximado}| / \text{valor exacto}$

SUSTITUIR VARIABLES

$\text{subst}(a, b, c)$ sustituye en c , b por a ej: $\frac{\sin a}{c} \rightarrow \frac{\sin b}{c}$

$\text{subst}([b=a], c)$ sustituye en c , b por a

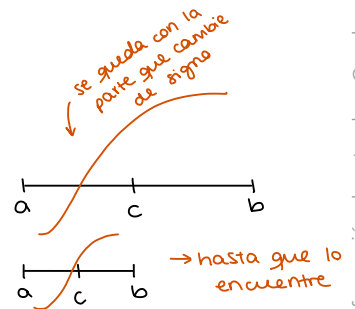
$\text{at}(\text{función}(x, y), [x=\text{num1}, y=\text{num2}])$ evalúa la función en x e y

BISECCIÓN

En cada paso se reduce a la mitad la longitud del intervalo

Calcular un número definido de decimales

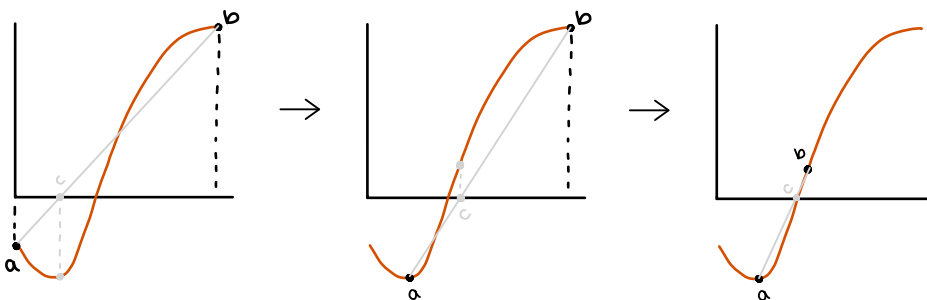
```
 $\sqrt[3]{5} = x \rightarrow 0 = x^3 - 5$   
define(f(x), x^3-5); /* función */  
a:0; /* extremos a y b del intervalo */  
b:4;  
err_a:10^(-3); /* error absoluto */  
while abs(b-a)/2 > err_a do(  
  c:(a+b)/2, /* calculamos el punto medio */  
  if f(c)=0 then return(c), /* si encontramos la solución, paramos */  
  if f(a)*f(c)<0 then b:c else a:c /* elegimos el intervalo donde cambie el signo */  
);  
c;
```



MÉTODO REGULA-FALSI

$$c = \frac{f(b) \cdot a - f(a) \cdot b}{f(b) - f(a)}$$

Utiliza los valores de la función en los extremos para calcular un pivote c .
Usa el punto de corte del segmento que une los extremos



```
kill(all);  
define(f(x), x^3-5); /* función */  
a:0.0; b:4.0; /* extremos a y b del intervalo */  
err_a:10^(-3); /* error absoluto */  
contador:0;  
maxiter:30;  
while abs(b-a)/2 > err_a and contador < maxiter  
do(  
  contador:contador+1,  
  c:(f(b)*a-f(a)*b)/(f(b)-f(a)),  
  if f(c)=0 then return(c),  
  if f(a)*f(c)<0 then b:c else a:c  
);  
[c, contador];
```