

WUOLAH



Zukii

www.wuolah.com/student/Zukii



1542

Sesion 5 Resuelta Maxima

Sesion 5 Resuelta Maxima



1º Cálculo



Grado en Ingeniería Informática



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

SESION 5 CORREGIDA(ejercicios típicos de examen)

1. Calcula los siguientes valores con dos decimales: e , π , $\sqrt{123}$ y $4.3^{8.688}$. Calcula el error absoluto y el error relativo.

```
kill(all);
define(f(x),x-%e); /* función */
a:0;b:4; /* extremos a y b del intervalo */
err_a:10^(-3); /* error absoluto */
while abs(b-a)/2 > err_a
do(
  c:(a+b)/2,
  if f(c)=0 then return(c),
  if f(a)*f(c)<0 then b:c else a:c
);
c,numer;
```

(para hallar el valor del numero con un margen de error de 10^{-3})
(no lo hago con todos porque es muy pesado)

```
float(%e - 2.71);
```

Zukii

`float(100*(%e - 2.71)/%e);`

`float(%pi);`

`float(%pi-3.14);`

`float(100*(%pi - 3.14)/%pi);`

`float(sqrt(123));`

`float(sqrt(123) - 11.09);`

`float(100 * (sqrt(123) - 11.09) / sqrt(123));`

`float(4.3^8.688);`

`float(4.3^8.688 - 3188`

`float(100 * (4.3^8.688 - 318840.83) / 4.3^8.688);`

2 Usa el método de bisección para encontrar una solución de las siguientes ecuaciones con cinco decimales exactos:

$\exp(-x)\cos(x)=0, x\in[0,20]$.

$\cos(x)+\sin(3x)=0, x\in[0,10]$.

$\log(x^2+1)-\cos(x), x\in[0,3]$.

Resuelve las ecuaciones anteriores con un error absoluto inferior a una millonésima.

Resuelve las ecuaciones anteriores con un error aproximado relativo menor al cinco por ciento.

Calcula $12345\sqrt{}$ con cinco decimales exactos usando el método de bisección.

```
biseccion(expr,var,ext_inf,ext_sup):=
  block(
    [a,b,c, /* extremos y punto medio */
    fa,fb,fc, /* valores de la función en dichos puntos */
    contador:0, /* número de repeticiones */
    tolx:10^(-5),tolfx:10^(-5) /* error permitido */
    ],
    local(f),
    define(f(x),subst(x,var,expr)),
    a:float(min(ext_inf,ext_sup)),
    b:float(max(ext_inf,ext_sup)),
    c:(a+b)*0.5,
    fa:f(a),
    fb:f(b),
    fc:f(c),
    if abs(fc)<tolfx then return([c,1,(b-a)*0.5,fc]),
    if sign(fa)=sign(fb) then error("la función no cambia de signo en los
    extremos"),
    while ((b-a)>tolx and abs(fc)>tolfx)
```

```

do(
  contador:contador+1,
  c:float((a+b)/2),
  fc:f(c),
  if abs(fc)<tolfx then return(),
  if sign(fa) = sign(fc) then (a:c,fa:fc) else (b:c,fb:fc)
),
[c,contador,(b-a)*0.5,f(c)]
)$

```

```
wxplot2d([exp(-x)*cos(x)], [x,2,5])$
```

```
biseccion(exp(-x)*cos(x), x ,2,5);
```

biseccion(exp(-x)*cos(x), x ,0,2); /* Cogemos el 0,2 ya que vemos que la funcion tiene 2 puntos de corte y por eso no podemos analizarla con el método de bisección al tener dos c*/

```

biseccion(expr,var,ext_inf,ext_sup):=
  block(
    [a,b,c, /* extremos y punto medio */
    fa,fb,fc, /* valores de la función en dichos puntos */
    contador:0, /* número de repeticiones */
    tolx:10^(-5),tolfx:10^(-5) /* error permitido */
    ],
    local(f),
    define(f(x),subst(x,var,expr)),
    a:float(min(ext_inf,ext_sup)),
    b:float(max(ext_inf,ext_sup)),

```

```

c:(a+b)*0.5,
fa:f(a),
fb:f(b),
fc:f(c),
if abs(fc)<tolfx then return([c,1,(b-a)*0.5,fc]),
if sign(fa)=sign(fb) then error("la función no cambia de signo en los extremos"),
while ((b-a)>tolx and abs(fc)>tolfx)
do(
  contador:contador+1,
  c:float((a+b)/2),
  fc:f(c),
  if abs(fc)<tolfx then return(),
  if sign(fa) = sign(fc) then (a:c,fa:fc) else (b:c,fb:fc)
),
[c,contador,(b-a)*0.5,f(c)]
)$

```

```
bisseccion(cos(x)+sin(3*x), x ,0,2)
```

```
wxplot2d([cos(x)+sin(3*x)], [x,0,10])$
```

```

bisseccion(expr,var,ext_inf,ext_sup):=
block(
  [a,b,c, /* extremos y punto medio */
  fa,fb,fc, /* valores de la función en dichos puntos */
  contador:0, /* número de repeticiones */
  tolx:10^-5,tolfx:10^-5/* error permitido */
  ],
  local(f),
  define(f(x),subst(x,var,expr)),
  a:float(min(ext_inf,ext_sup)),

```

```

b:float(max(ext_inf,ext_sup)),
c:(a+b)*0.5,
fa:f(a),
fb:f(b),
fc:f(c),
if abs(fc)<tolfx then return([c,1,(b-a)*0.5,fc]),
if sign(fa)=sign(fb) then error("la función no cambia de signo en los extremos"),
while ((b-a)>tolx and abs(fc)>tolfx)
do(
  contador:contador+1,
  c:float((a+b)/2),
  fc:f(c),
  if abs(fc)<tolfx then return(),
  if sign(fa) = sign(fc) then (a:c,fa:fc) else (b:c,fb:fc)
),
[c,contador,(b-a)*0.5,f(c)]
)$

```

```
bisseccion(log(x^2+1)-cos(x), x ,0,3);
```

```
wxplot2d([log(x^2+1)-cos(x)], [x,0,3])$
```

iv) Habria que cambiar en todos tolx y tolfx por 10^{-6} que es una millonésima

v) Ahora tendríamos que cambiar tolx y tolfx por 5 y $100 \cdot (b-a)/b$ y fc no ya que haríamos $fc - 0/fc \cdot 100$

```
bisseccion(log(x^2+1)-cos(x), x ,0,3);
```

3. Calcula $\sqrt{12345}$ con cinco decimales exactos usando el método de bisección. Es decir, con un error absoluto de 10^{-5}

```

biseccion(expr,var,ext_inf,ext_sup):=
  block(
    [a,b,c, /* extremos y punto medio */
    fa,fb,fc, /* valores de la función en dichos puntos */
    contador:0, /* número de repeticiones */
    tol:10^-8/* error permitido */
    ],
    local(f),
    define(f(x),subst(x,var,expr)),
    a:float(min(ext_inf,ext_sup)),
    b:float(max(ext_inf,ext_sup)),
    c:(a+b)*0.5,
    fa:f(a),
    fb:f(b),
    fc:f(c),
    if abs(fc)<tol then return([c,1,(b-a)*0.5,fc]),
    if sign(fa)=sign(fb) then error("la función no cambia de signo en los extremos"),
    while (b-a)>tol
      do(
        contador:contador+1,
        c:float((a+b)/2),
        fc:f(c),
        if abs(fc)<tol then return(),
        if sign(fa) = sign(fc) then (a:c,fa:fc) else (b:c,fb:fc)
      ),
    [c,contador,(b-a)*0.5,f(c)]
  )

```


Zukii

)\$

Zukii

```
biseccion(x-sqrt(12345), x ,100,1000);
```

```
wxplot2d([x-sqrt(12345)], [x,110,112])$
```