

Tarea 3: Métodos Numéricos

Ejemplo 1

```
%Condiciones iniciales del problema
x0 = 0;
x1 = 2;
n = 100;
es = 1e-6;
i = 0;

l = 5;
r = 1;
V = 8.5;

%Declaramos la función que vamos a ocupar
syms h
f = ( (r^2 * acos((r - h) / r)) - ((r - h) * sqrt(2*r*h - h^2))) * l - V;
ff = matlabFunction(f);

while (i < n) && (abs(x1 - x0) > es)

    % Método de la secante
    x2 = x1 - ff(x1)*(x1 - x0) / (ff(x1) - ff(x0)); % Fórmula de la secante

    % Actualizamos valores
    x0 = x1;
    x1 = x2;

    ea = abs((x1 - x0) / x1) * 100;
    i = i + 1;

    fprintf('%i\t %f\t %f\n', i, x1, ea);
end
```

1	1.082254	84.799568
2	1.059339	2.163137
3	1.064650	0.498892
4	1.064647	0.000303
5	1.064647	0.000000

Comparación con el Método de bisección:

```
l = 5;
r = 1;
V = 8.5;

xrold = 0;
xr = 0;
xl = 0;
xu = 2;
```

```

n = 100;
ea = 1.0;
es = 0.000001;

i = 0;

while (i<n) && (ea>es)
    xrold = xr;
    xr = (xl + xu) / 2;
    i = i+1;

    if(xr ~= 0)
        ea = (abs((xr - xrold) / xr)) * 100;
    end

    prueba = g(xl,l,r,V) * g(xr,l,r,V);

    if(prueba<0)
        xu = xr;
    else
        xl = xr;
    end

    fprintf('%i\t%f\t%f\n', i, xr, ea)
end

```

1	1.000000	100.000000
2	1.500000	33.333333
3	1.250000	20.000000
4	1.125000	11.111111
5	1.062500	5.882353
6	1.093750	2.857143
7	1.078125	1.449275
8	1.070312	0.729927
9	1.066406	0.366300
10	1.064453	0.183486
11	1.065430	0.091659
12	1.064941	0.045851
13	1.064697	0.022931
14	1.064575	0.011467
15	1.064636	0.005733
16	1.064667	0.002866
17	1.064651	0.001433
18	1.064644	0.000717
19	1.064648	0.000358
20	1.064646	0.000179
21	1.064647	0.000090
22	1.064647	0.000045
23	1.064647	0.000022
24	1.064647	0.000011
25	1.064647	0.000006
26	1.064647	0.000003
27	1.064647	0.000001
28	1.064647	0.000001

```
function vol = g(h,l,r,V)
    vol = ( ((r^2 * acos((r-h) / r)) - ((r-h) * sqrt(2*r*h - h^2))) * l ) -
V;
end
```

Ejemplo 2:

```
%Declaración de los vectores con los coeficientes de la función
```

```
a = [0.7 -4 6.2 -2];
b = [-3.704 16.3 -21.97 9.34];
c = [1 -3 5 -1 -10];
```

```
%Método con la función roots() de Matlab
```

```
r_a = 3x1
    3.2786
    2.0000
    0.4357
```

```
r_a = roots(a)
r_b = roots(b)
```

```
r_b = 3x1
    2.2947
    1.1525
    0.9535
```

```
r_c = roots(c)
```

```
r_c = 4x1 complex
    1.0000 + 2.0000i
    1.0000 - 2.0000i
    2.0000 + 0.0000i
   -1.0000 + 0.0000i
```

Ejemplo 3: parte 1

```
%Datos iniciales del problema
```

```
x0 = 6;
x1 = 10;
n = 100;
es = 1e-5;
i = 0;
```

```
%Concentración requerida primero
```

```
C1 = 15;
```

```
%Declaración de la función a ocupar
```

```
syms t
f1 = 75 * exp(-1.5 * t) + 20 * exp(-0.075 * t) - C1;
ff1 = matlabFunction(f1);
```

```

while (i < n) && (abs(x1 - x0) > es)

    %Ocupamos el método de la secante

    x2 = x1 - ff1(x1) * (x1 - x0) / (ff1(x1) - ff1(x0));

    x0 = x1;
    x1 = x2;

    ea = abs((x1 - x0) / x1) * 100;

    i = i + 1;

    fprintf('%i\t %f\t %f\n',i,x1,ea)
end

```

1	3.298893	203.132016
2	4.447336	25.823180
3	4.063090	9.456995
4	3.997262	1.646825
5	4.001682	0.110443
6	4.001634	0.001192
7	4.001634	0.000001

Ejemplo 3: parte 2

```

%Condiciones iniciales para la parte 2 del problema
x0 = 0;
x1 = 5;
n = 1000;
es = 1e-5;
i = 0;

%Calculamos el nuevo número de partículas
%Evaluamos la función original en t=0
%Para luego el valor de c que nos diera lo multiplicamos por 0.01

C2 = 0.95;

%Declaración de la función a ocupar
syms t
f1 = 75 * exp(-1.5 * t) + 20 * exp(-0.075 * t) - C2;
ff1 = matlabFunction(f1);

while (i < n) && (abs(x1 - x0) > es)

    %Volvimos a ocupar el método de la secante
    x2 = x1 - ff1(x1) * (x1 - x0) / (ff1(x1) - ff1(x0));

    x0 = x1;
    x1 = x2;

```

```
ea = abs((x1 - x0) / x1) * 100;  
  
i = i + 1;  
  
fprintf('%i\t %f\t %f\n',i,x1,ea)  
end
```

1	5.790348	13.649407
2	17.374222	66.672763
3	24.269048	28.409958
4	31.465329	22.870509
5	36.463608	13.707581
6	39.411174	7.479012
7	40.449504	2.566977
8	40.619055	0.417417
9	40.626955	0.019446
10	40.627008	0.000130
11	40.627008	0.000000