Métodos numéricos en ingeniería
Dr. Adolfo Centeno Téllez

Entrega Proyecto primer parcial

Luis Adrian Carmona Villalobos A01748395

Fecha de entrega: 14/09/2021

Objetivos

1. Aplicar métodos numéricos para el seguimiento en una trayectoria de un dron.
2. Aplicación de los métodos numéricos en ingeniería en Sistemas digitales y robótica

Propósito

El propósito de este proyecto es poder resolver cualquier funcion dada para la trayectoria de un dron utilizando los cuatro métodos vistos hasta ahora en clase que le permita al usuario continuar en el desarrollo del modelamiento matemático para el control de un cuadricóptero.

Para el modelamiento matemático y control de un cuadricóptero lleva varios procesos en los que hay diferentes ecuaciones, se tomara una ecuación a la que se le aplicara métodos numéricos para comprobar la trayectoria que llevará la funcion cerca de sus raíces.

Introducción.

En la industria de los drones con relación al control de sistemas de vuelo es la dificultad en su maniobra. Existen muchos sistemas de vuelo entre los que se tienen los cuadricópteros que viene ser helicóptero de cuatro motores, estos permiten mejor control de estabilidad, diseño, mantenimiento entre otras cosas debido a que presenta ángulos constantes entre sus hélices y los ejes de referencia, por lo tanto, su análisis matemático se facilita.

Descripción del problema a resolver.

El modelo matemático del cuadrimotor es realizado basando en las siguientes consideraciones: El Cuadrimotor es un cuerpo solido en tres dimensiones, sujeto a una fuerza principal y tres. Su centro de masa es localizado en el centro del vehículo, los efectos giroscópicos son cancelados debido a la disposición de sus hélices, y los efectos externos por rozamiento con el aire son despreciables. El modelo es obtenido a partir de las ecuaciones de Euler-Lagrange, en uno de estos procesos para describir el modelo matemático esta la ecuación para el seguimiento de una trayectoria en la que hace que se mantenga constante un tiempo a una variable de distancia.

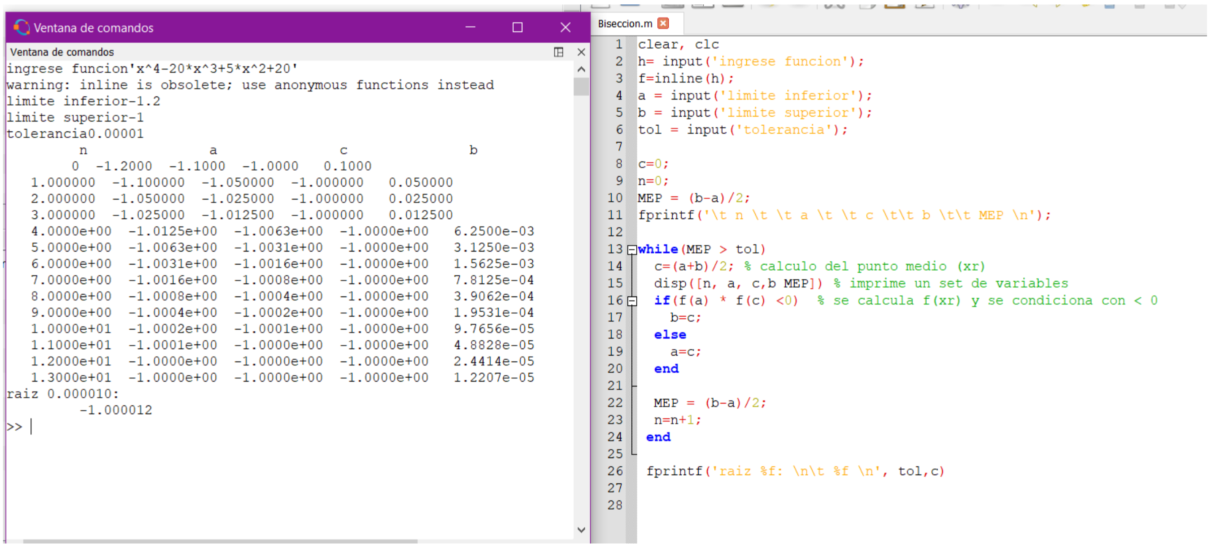
Por ejemplo, una ecuación con condiciones iniciales para que se mantenga constante en el valor de 1cm el resto del tiempo:

$$Z(t) = 0.0048vot\^4 - 0.064vot\^3 + 0.24vot\^2$$

Resultados.

Para este proyecto se tuvo problemas en demostrar la validación de los métodos numéricos ya que al principio se usó ejemplos de funciones reales que se usan para el seguimiento de los drones pero estas funciones en a,b,c y d los valores numéricos típicamente son menores a uno y mayores a cero, ya que el seguimiento los valores son constantes y rápidos, los métodos numéricos sirven para cualquier funcion pero para poder mostrarlo mejor se usaron variables mayores a uno para que se demuestre fácilmente la ecuación.

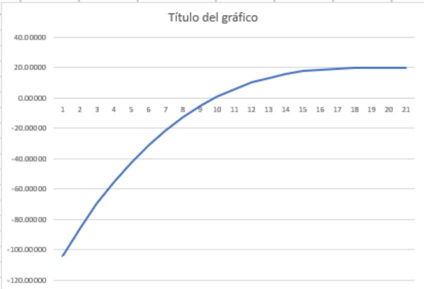
Método de Bisección



```
Ventana de comandos
ingrese funcion'x^4-20*x^3+5*x^2+20'
warning: inline is obsolete; use anonymous functions instead
limite inferior-1.2
limite superior-1
tolerancia0.00001
        n              a               c               b
       0   -1.2000   -1.1000   -1.0000    0.1000
   1.000000  -1.100000  -1.050000  -1.000000   0.050000
   2.000000  -1.050000  -1.025000  -1.000000   0.025000
   3.000000  -1.025000  -1.012500  -1.000000   0.012500
   4.0000e+00  -1.0125e+00  -1.0063e+00  -1.0000e+00   6.2500e-03
   5.0000e+00  -1.0063e+00  -1.0031e+00  -1.0000e+00   3.1250e-03
   6.0000e+00  -1.0031e+00  -1.0016e+00  -1.0000e+00   1.5625e-03
   7.0000e+00  -1.0016e+00  -1.0008e+00  -1.0000e+00   7.8125e-04
   8.0000e+00  -1.0008e+00  -1.0004e+00  -1.0000e+00   3.9062e-04
   9.0000e+00  -1.0004e+00  -1.0002e+00  -1.0000e+00   1.9531e-04
   1.0000e+01  -1.0002e+00  -1.0001e+00  -1.0000e+00   9.7656e-05
   1.1000e+01  -1.0001e+00  -1.0000e+00  -1.0000e+00   4.8828e-05
   1.2000e+01  -1.0000e+00  -1.0000e+00  -1.0000e+00   2.4414e-05
   1.3000e+01  -1.0000e+00  -1.0000e+00  -1.0000e+00   1.2207e-05
raiz 0.000010:
        -1.000012
>>
```

```
Biseccion.m
clear, clc
h= input('ingrese funcion');
f=inline(h);
a = input('limite inferior');
b = input('limite superior');
tol = input('tolerancia');

c=0;
n=0;
MEP = (b-a)/2;
fprintf('\t n \t \t a \t \t c \t\t b \t\t MEP \n');

while(MEP > tol)
  c=(a+b)/2; % calculo del punto medio (xr)
  disp([n, a, c,b MEP]) % imprime un set de variables
  if(f(a) * f(c) <0)  % se calcula f(xr) y se condiciona con < 0
    b=c;
  else
    a=c;
  end

  MEP = (b-a)/2;
  n=n+1;
 end

 fprintf('raiz %f: \n\t %f \n', tol,c)
```

	Metodo de Biseccion									
	Formula	f(x) = at^4+bt^3+ct^2+dt+e		1			ERROR	0.0001		
	Ejemplo 1	f(x) = t^4-20t^3+5t^2+20		xi	-1.2					
				xu	-1					
Graficar		f = inline (str)								
	x	f(x)		x	f(x)					
	-2.0000	-104.00000		-5	-5.28640	-5.2864				
	-1.9000	-86.09790		6	-1.00000	6				
	-1.8000	-69.94240			-6.28640	< 0	xu = xr			
	-1.7000	-55.45790								
	-1.6000	-42.56640		2					< 0 xu=xr	
	-1.5000	-31.18750							> 0 xi =xr	
	-1.4000	-21.23840	s	xi	xu	xr (xi + xu) /2	f(xi)	f(xr)	f(xi)f(xr)	
	-1.3000	-12.63390	1	-1.2	-1	-1.1	-5.2864	0.8941	-4.7266	error
xi	-1.2000	-5.28640	2	-1.2	-1.1	-1.15	-104	-2.055994	213.8233	error
	-1.1000	0.89410	3	-1.15	-1.1	-1.125	-86.0979	-0.546631	47.0638	error
xu	-1.0000	6.00000	4	-1.125	-1.1	-1.1125	-69.9424	0.182223	-12.7451	error
	-0.9000	10.12610	5	-1.125	-1.1125	-1.11875	-55.4579	-0.180071	9.9863	error
	-0.8000	13.36960	6	-1.11875	-1.1125	-1.115625	-42.5664	0.001608	-0.0684	raiz encontrada
	-0.7000	15.83010	7	-1.11875	-1.115625	-1.1171875	-31.1875	-0.089098	2.7788	error
	-0.6000	17.60960	8							
	-0.5000	18.81250								
	-0.4000	19.54560			xr	-1.115625				
	-0.3000	19.91810				-4.157278229		ERROR	0.01	
	-0.2000	20.04160								
	-0.1000	20.03010								
	0.0000	20.00000								
	0.1000	20.07010								
	0.2000	20.36160								
	0.3000	20.99810								
	0.4000	22.10560								
	0.5000	23.81250								
	0.6000	26.24960								
	0.7000	29.55010								
	0.8000	33.84960								
	0.9000	39.28610								
	1.0000	46.00000								
	1.1000	54.13410								
	1.2000	63.83360								
	1.3000	75.24610								
	1.4000	88.52160								
	1.5000	103.81250								
	1.6000	121.27360								
	1.7000	141.06210								
	1.8000	163.33760								
	1.9000	188.26210								
	2.0000	216.00000								



Método de Secante

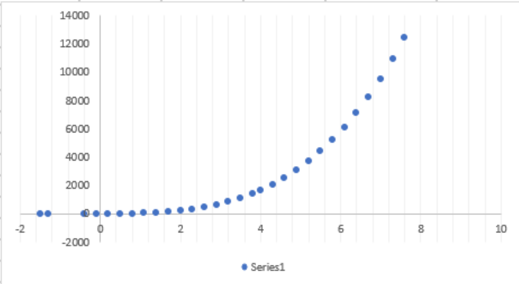
Ventana de comandos

```
ingrese funcion' x^4-20*x^3+5*x^2+20'
limite inferior0.5
limite superior1.1
tolerancia0.0001

 x0   x1       x2          error
 0 0.5000 1.1000 1.1299
 0.839338  1 1.1000 1.1299 1.1154
 0.0121885  2 1.1299 1.1154 1.1156
 0.000162155  3 1.1154 1.1156 1.1157
 3.20472e-08 raiz=  1.115653
>>
```

Secante.m

```
clear, clc

cf= input('ingrese funcion'); %captura la funcion como string '4*x**2-5*x'
f=inline(cf); % convierte un texto en una funcion
x0 = input('limite inferior'); %
x1 = input('limite superior');%
tol = input('tolerancia'); % margen de error

error= 100; % variable que almacena el error actual se inicializa


n=0; % contador de iteraciones
fprintf(' \n x0   x1      x2\t error  \n');  % imprime

while(error > tol)
x2= x1 -(x1-x0)*f(x1)/(f(x1) - f(x0));
  error = abs(f(x2));
  fprintf(' %i %4.4f %4.4f %4.4f  \n', n,x0,x1,x2,error);

  x0= x1;
  x1=x2;
  n=n+1;
  end

 fprintf('raiz=  %f  \n', x2);
```

	Metodo Secante										
									Tolerancia	0.0000001	
	Formula	f(x) = at^4+bt^3+ct^2+dt+e									
	Ejemplo 1	f(x) = t^4-20t^3+5t^2+20									
				iter	Xi	X0	f(Xi)	f(X0)	X (i+1)	Error f(Xa)	
Graficar				1	1.1	0.5	54.1341	23.8125	0.0288	-0.3719	error
	x	f(x)		2	0.0288	1.1	20.0046	54.1341	-0.5991	0.0105	error
	-1.5	-31.1875		3	-0.5991	0.0288013	17.6233	20.0046	-5.2457	0.0089	error
	-1.3	-12.6339		4	-5.2457	-0.5990704	-1972.1173	17.6233	-0.6402	-0.0719	error
	-0.4	19.5456		5	-0.6402	-5.2456535	16.9690	-1972.1173	-0.6795	0.0006	error
	-0.1	20.0301		6	-0.6795	-0.6402256	16.2467	16.9690	-1.5632	0.0057	error
	0.2	20.3616		7	-1.5632	-0.6795148	-38.2117	16.2467	-0.9432	-0.0066	error
x0	0.5	23.8125		8	-0.9432	-1.5632373	8.4594	-38.2117	-1.0556	0.0011	error
	0.8	33.8496			-1.0556	-0.9431578	3.2907	8.4594	-1.1271	0.0006	error
x1	1.1	54.1341			-1.1271	-1.0555503	-0.6712	3.2907	-1.1150	-0.0001	error
	1.4	88.5216			-1.1150	-1.1271081	0.0387	-0.6712	-1.1156	0.0000	error
	1.7	141.0621			-1.1156	-1.1149848	0.0004	0.0387	-1.1157	0.0000	raiz encontrada
	2	216			-1.1157	-1.1156456	0.0000	0.0004	-1.1157	0.0000	raiz encontrada
	2.3	317.7741									
	2.6	451.0176									
	2.9	620.5581									
	3.2	831.4176									
	3.5	1088.8125									
	3.8	1398.1536									
	4	1636									
	4.3	2044.4701									
	4.6	2520.2656									
	4.9	3069.5101									
	5.2	3698.5216									
	5.5	4413.8125									
	5.8	5222.0896									
	6.1	6130.2541									
	6.4	7145.4016									
	6.7	8274.8221									
	7	9526									
	7.3	10906.6141									
	7.6	12424.5376									



Método de Newton-Raphson

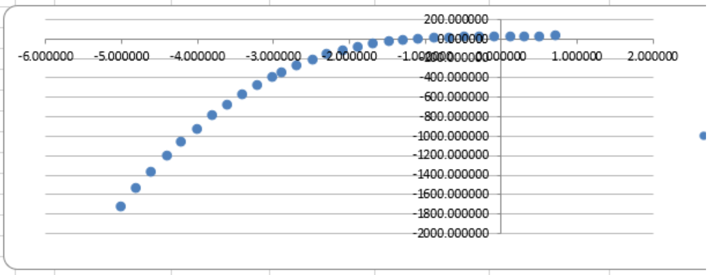
	Metodo Newthon Raphson			
	Formula	f(x) = at^4+bt^3+ct^2+dt+e		
	Ejemplo 1			
	Ejemplo 1	f(x) = t^4-20t^3+5t^2+20		
		f ' =4t^3-60t^2+10t		
Graficar				
	x	f(x)		
	-5.000000	-1730.000000		
	-4.800000	-1545.798400		
	-4.600000	-1373.174400		
	-4.400000	-1212.070400		
	-4.200000	-1062.390400		
	-4.000000	-924.000000		
	-3.800000	-796.726400		
	-3.600000	-680.358400		
	-3.400000	-574.646400		
	-3.200000	-479.302400		
	-3.000000	-394.000000		
	-2.879385	-347.258916		
	-2.679385	-277.276695		
	-2.479385	-216.306353		
	-2.279385	-163.883131		
	-2.079385	-119.503872		
	-1.879385	-82.627018		
	-1.679385	-52.672610		
	-1.479385	-29.022290		
raiz	-1.279385	-11.019301		
	-1.079385	2.031515		
	-0.879385	10.863718		
	-0.679385	16.249264		
	-0.479385	18.998511		
	-0.279385	19.960219		
	-0.079385	20.021544		
	0.120615	20.108045		
	0.320615	21.183680		
	0.520615	24.250807		

p_0	-1.279385

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Donde f' denota la derivada de f.

n	Pn-1	f(Pn-1)	f ' (Pn-1)	Pn	f(Pn)	E
1	-1.27938524	-11.01930140	-119.37997507	-1.37168968	-18.66974841	0.00010000
2	-1.37168968	-18.66974841	-136.93236620	-1.50803252	-32.04758416	0.00010000
3	-1.50803252	-32.04758416	-165.24809208	-1.70196870	-55.72742530	0.00010000
4	-1.70196870	-55.72742530	-210.54188918	-1.96665440	-97.83176572	0.00010000
5	-1.96665440	-97.83176572	-282.15626356	-2.31338342	-172.21279739	0.00010000
6	-2.31338342	###########	-393.76093733	-2.75073710	-301.18652908	0.00010000
	-2.75073710	###########	-564.75505846	-3.28404181	-518.12387880	0.00010000
	-3.28404181	###########	-821.60890606	-3.91466287	-868.34586268	0.00010000
	-3.91466287	###########	############	-4.63913925	-1406.04729708	0.00010000
	-4.63913925	###########	############	-5.44858232	-2185.29049135	0.00010000



Área del gráfico

New_rap.m

```matlab
clear, clc %limpia
f = input('f(x)= ', 's'); % x^3+3*x^2-1
sf = str2sym(f); % convierte string en funcion
tol = input('tolerancia del metodo = '); % tolerancia
x0 = input('valor inicial = '); % valor inicial
v = symvar(sf); % extrae las variables de la funcion
f1 = diff(sf); % calcula la derivada de una funcion
sf
v
f1
sw = 0;
while (sw==0)
        % subs = funcion que reemplaza valores constantes en las variables
        % de una funcion
    x1 = x0 - (subs (sf, v, x0) / subs (f1, v, x0));
    fprintf ('\t x0 %f \t x1 %f \t tol %f \n', x0, x1, tol)
    if abs(x0 - x1) > tol % aun no se encuentra la raiz
        x0 = x1;
        sw=0;

    else %raiz encontrda
        sw =1;
    end
end

vpa(x1)
fplot(sf);
```

Command Window

```
f1 =

4*x^3 - 60*x^2 + 10*x

         x0 -1.270000    x1 -0.661220    tol
0.000100
         x0 -0.661220    x1 0.166956     tol
0.000100

         x0 0.166956     x1 -1275.512594
tol 0.000100
         x0 -1275.512594         x1
-955.398492      tol 0.000100
         x0 -955.398492          x1
-715.317550      tol 0.000100
         x0 -715.317550          x1
-535.262986      tol 0.000100

         x0 -535.262986          x1
-400.230189      tol 0.000100

         x0 -400.230189          x1
-298.966310      tol 0.000100

         x0 -298.966310          x1
-223.032493      tol 0.000100
```

Conclusiones.

Con este proyecto se puede concluir que no solo se puede solucionar con un método numérico problemas de ingeniera, algunos son mas eficientes que otros dependiendo lo que se busque, pero los métodos numéricos en la rama de la ingeniería en robótica proporcionan mucha utilidad sobre todo en problemas como los drones donde el margen de error tiene que ser mínimo por todos los procesos que conllevaría tener un margen alto de error.

También se vio en las graficas el comportamiento constante del tiempo decreciendo al acercarse a las raíces de la funcion, con lo que se este proyecto comprueba la funcion y comportamiento de la funcion para el seguimiento en la trayectoria de un dron.

Bibliografía

Parra Muñoz, M., Feitosa Fortaleza, E., & Alves da Silva, J. (2013). Modelamiento matemático y control de un helicóptero de cuatro motores. *Scientia Et Technica*, *18*(4), 672-681. https://doi.org/10.22517/23447214.8195