Lanzamiento de cohete de agua

Isabel Gonzalez Lezama

November 3, 2021

1 Introduction

¿Para qué sirven las ecuaciones? En realidad la aplicación de los polinomios en la vida diaria es de suma importancia. Se puede aplicar en el área de construcción, para el pronóstico del clima, para el cálculo en finanzas, al realizar alguna compra, etc. Emplearemos un ejemplo que se ocupa en la vida diaria: la gravedad. Siendo más específicos, la aceleración de un cohete de agua.

2 Descripción del problema a resolver

2.1 Lanzamiento de un cohete de agua

Imagina que eres un astronauta en la Estación Espacial Internacional. Estás arreglando unos paneles solares, cuando de pronto al presionar tu alguna herramienta que estés ocupando sale disparada de tus manos. Si no lo atrapas a tiempo, ésta estará viajando por el espacio en línea recta y a velocidad constante, a menos que la dirección sea la tierra y entre a la fuerza gravitatoria de esta y comience a acelerarse en su camino. Esto sucede porque la herramienta se mueve con movimiento rectilíneo uniforme y cuando entra en gravedad en movimiento acelerado, así construyendo una parábola como trayectoria. Esto es uno de los pocos ejemplos que podemos encontrar, así que resolveremos uno que sí pase en la tierra y a mortales como nosotros.

Este problema consistirá en una ecuación trazada por la trayectoria de un cohete de agua, formando una parábola. Se resolverá la ecuación con los métodos vistos: Gauss-Jordan, Jacobi, Gauss-Seidel, cramer. Se obtendrá una una aproximación de la raíz de la ecuación.

Se realiza un proyecto de un cohete de agua. Se trazan los puntos que recorre tomando como inicio el origen que partió y se obtiene unas coordenadas agora solo de tiene que hallar la ecuación de la parábola de eje vertical y que pasa por los puntos:

A(-1, 1), B (1, 9) ,C (-2, 0).

La ecuación estándar de una parábola es:

y = ax 2 + bx + c

Figure 1: Richard M. 2009. Lanzamiento de un cohete de agua. Ilustración.

Si pasa por los puntos A, B y C, sus coordenadas cumplen la ecuación anterior.
Procedemos así a resolver el ejemplo propuesto inicialmente:
y= a.x2+bx+c
A(-1, 1), B (1, 9) ,C (-2, 0)
En primer lugar, sustituimos el valor de nuestros puntos en la función general:
El punto A:
X= -1 e y = 1
1 = a.(-1)2+b(-1)+c
El punto B:
X= 1 e y = 9
9 = a.(1)2+b(1)+c
El punto C:
X= -2 e y = 0
0 = a.(-2)2+b(-2)+c
De este modo, obtenemos el siguiente sistema de ecuaciones:
a-b+c=1 ; a+b+c=9 ; 4a-2b+c=0 ;
1-1+1=1; 1+1+1=9; 4-2+1=0;

3 Resultados

3.1 Método de Gauss-Jordan

	Eliminacion Gaussiana									
	x	y	z	b						
F1	1	-1	1	1						
F2	1	1	1	9						
F3	4	-2	1	0						
	Intercambiar F1 y F2									
F1	1	1	1	9			F1 = F2			
F2	1	-1	1	1			F2 = F1			
F3	4	-2	1	0						
	Iter 1						F2 - 2F1			
	x	y	z							
F1	1	1	1	9		F2	1	-1	1	1
F2	-1	-3	-1	-17		-2*F1	-2	-2	-2	-18
F3	4	-2	1	0			-1	-3	-1	-17
	Iter 2						F3 - 5F1			
	x	y	z							
F1	1	1	1	9		F3	5	7	-4	9
F2	0	2	3	9		-5*F1	-5	-5	-5	-45
F3	0	2	-9	-36			0	2	-9	-36
	Iter 3						3F2 + 2F3			
	x	y	z							
F1	1	1	1	9		3F2	0	6	9	27
F2	0	2	3	9		2F3	0	-6	2	28
F3	0	0	11	55			0	0	11	55
			11z =	55						
			z=	55/11						
F3			z=	5						
F2		2y +	3z =	9						
		2y +	3(5) =	9						
		2y +	15 =	9						
			2y=	9-15						
			y=	=-6/2						
			y=	-3						
F1	x +	2(-3) -	5=	-1						
	x+	-6	-5 =	-1						
			x-11 =	-1						
			x=	-1 + 11						
			x=	10						

```
%%
while normVal>tol
    xold=x;

    for i=1:n
        sigma=0;

        for j=1:n

            if j~=i
                sigma=sigma+A(i,j)*x(j);
            end

        end

        x(i)=(1/A(i,i))*(b(i)-sigma);
    end

    itr=itr+1;
    normVal=abs(xold-x);
end
%%
fprintf('Solution of the system is : \n%f\n%f\n%f\n%f in %d iterations',x,it
```

3.2 Método de Jacobi

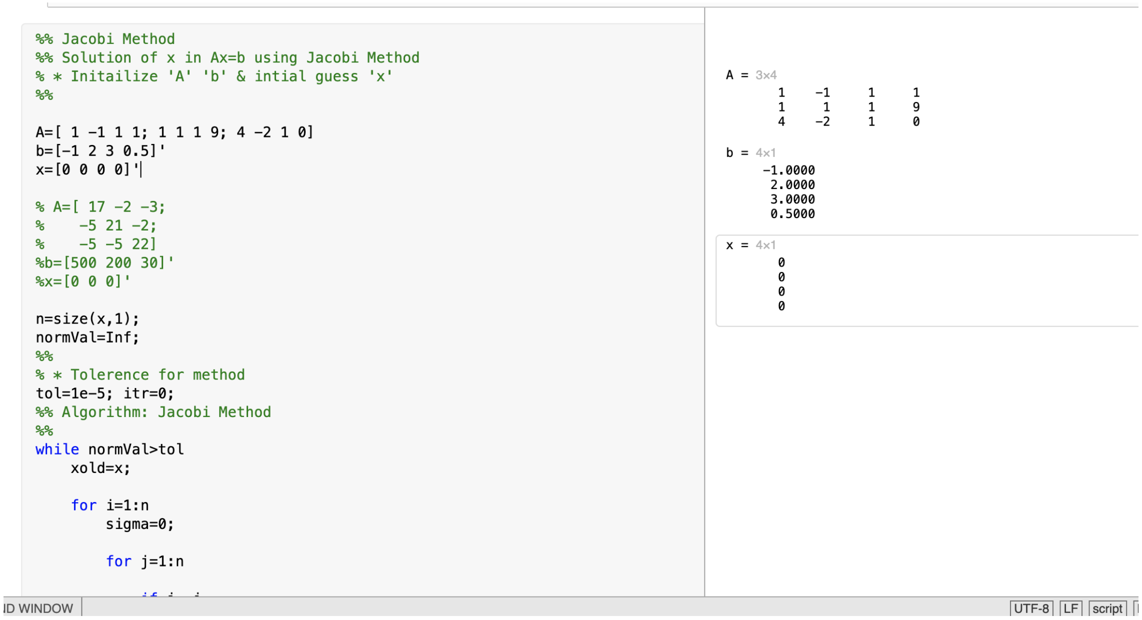
Sistema a solucionar								b
1	x	-1	y	1	z	=		1
1	x	1	y	1	z	=		9
4	x	-2	y	1	z	=		0
Verificamos que la matriz sea								
diagonal dominante								
Fila 1	Valor inicial							1
Fila 2	Valor inicial							1
Fila 3	Valor inicial							1
Fila 1	suma valores restantes							#ERROR!
Fila 2	suma valores restantes							#ERROR!
Fila 3	suma valores restantes							#ERROR!
Situacior	#ERROR!							

No. Iter	x	y	z	err x	err y	err z
0	0	0	0	x inicial		
1	1	9	0			
2	10	8	14	0,900000000	0,125000000	1,000000000
3	-5	-15	-24	-3,000000000	-1,533333333	-1,583333333
4	10	38	-10	1,500000000	1,394736842	-1,400000000
5	49	9	36	0,795918367	3,222222222	1,277777778
6	-26	-76	-178	-2,884615385	-1,118421053	-1,202247191
7	103	213	-48	1,252427184	1,356807512	-2,708333333
8	262	-46	14	0,606870229	-5,630434783	4,428571429
9	-59	-267	-1140	-5,440677966	-0,827715356	-1,012280702
10	874	1208	-298	1,067505721	1,221026490	-2,825503356
11	1507	-567	-1080	0,420039814	-3,130511464	-0,724074074
COMP EC1	17	1507	-2	-567	-3	-1080
COMP EC2	-5	1507	21	-567	-2	-1080
COMP EC3	-5	1507	-5	-567	22	-1080

3.3 Cramer

4 Conclusión

Con este proyecto se puede concluir que estos métodos son de gran utilidad para casi cualquier área de la ingeniería. Los tres métodos pueden tener varias aplicaciones en diferentes campos, como en este caso para la ciencia, para problemas sencillos o muy complejos. Los problemas resueltos tiene la ventaja de tener un margen de error mínimo ya que gracias a la programación nosotros mismos



determinamos la tolerancia que queramos y eso es sumamente importante, ya que si lo hiciéramos a mano todos estos cálculos tardaremos mucho tiempo y muy posiblemente cometamos muchos errores en su cálculo.

También se pudo concluir que el método y la ingeniería más específicamente en ciencias como la física tiene una relación unida ya que si tomamos en cuenta que cada vez que se desarrolle un nuevo método de cálculo dentro de lo que son los métodos numéricos los cálculos en trayectorias, velocidades en cuestión de tiempo o incluso como el ejemplo de espacial, en alguna trayectoria de algún cohete que resulta interesante saber la relación de masa y velocidad respecto a su función se podrían volver más exactos, más veloces o en algunas ocasiones ambas.

	Sistema a solucionar							
	1	x	-1	y	1	z	=	1
	1	x	1	y	1	z	=	9
	4	x	-2	y	1	z	=	0
	Verificamos que la matriz sea							
	diagonal dominante							
	Fila 1	Valor inicial						1
	Fila 2	Valor inicial						1
	Fila 3	Valor inicial						$
	Fila 1	suma valores restantes						#ERROR!
	Fila 2	suma valores restantes						#ERROR!
	Fila 3	suma valores restantes						#ERROR!
	Situacion							
		DOMINANTE						

No. Ite	x	y	z	err x	err y	err z
0	0	0	0	x		
1	1	8	12			
2	-3	0	12	-1,333333333	#DIV/0!	0,000000000
3	-11	8	60	-0,727272727	1,000000000	0,800000000
4	-51	0	204	-0,784313725	#DIV/0!	0,705882353
5	-203	8	828	-0,748768473	1,000000000	0,753623188
6	-819	0	3276	-0,752136752	#DIV/0!	0,747252747
7	-3275	8	13116	-0,749923664	1,000000000	0,750228728
8	-13107	0	52428	-0,750133516	#DIV/0!	0,749828336
9	-52427	8	209724	-0,749995231	1,000000000	0,750014305
10	-209715	0	838860	-0,750008345	#DIV/0!	0,749989271
11	-838859	8	3355452	-0,749999702	1,000000000	0,750000894
12	-3355443	0	13421772	-0,750000522	#DIV/0!	0,749999329
13	-13421771	8	53687100	-0,749999981	1,000000000	0,750000056
14	-53687091	0	214748364	-0,750000033	#DIV/0!	0,749999958
15	-214748363	8	858993468	-0,749999999	1,000000000	0,750000003
16	-858993459	0	3435973836	-0,750000002	#DIV/0!	0,749999997
17	-3435973835	8	13743895356	-0,750000000	1,000000000	0,750000000
18	-13743895347	0	54975581388	-0,750000000	#DIV/0!	0,750000000
19	-54975581387	8	219902325564	-0,750000000	1,000000000	0,750000000
20	-219902325555	0	879609302220	-0,750000000	#DIV/0!	0,750000000
21	-879609302219	8	3518437208892	-0,750000000	1,000000000	0,750000000
22	-351843720888	0	140737488355	-0,750000000	#DIV/0!	0,750000000

		x		y		z		
Comp. ec1	8	-351843720888	-4	0	-3	140737488355	=	-703687441776
Comp. ec2	2	-351843720888	-5	0	3	140737488355	=	351843720888
Comp. ec3	-3	-351843720888	1	0	9	140737488355	=	137219051146

ewthonRap.m

```
clear, clc
f = input('f(x)= ', 's');%'x^2+ 64*x + 4'
sf = str2sym (f); %convierte un string en una funcion
tol = input('tolerancia del metodo = ');%tolerancia
x0 = input('valor inicial = ');%valor inicial, elemplo: -3
v = symvar(sf);%extrae las variables de la funcion
f1 = diff(sf); %calcula la derivada de una funcion
sf
v
f1

y=subs (sf,v,x0)
z=subs(f1,v,x0)
y
z


sw = 0;
while (sw==0)
    %subs es una 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 encontrada
        sw =1;
    end
end

vpa(x1)
```