

Resolución De Ecuaciones

Métodos Iterativos

Cristóbal López Silla - Licenciado en Matemáticas.

Índice general

Listado de Teoremas, Proposiciones, Corolarios Y Definiciones	7
Listado Ejercicios y Ejemplos	8
1. PREFACIO	11
1.1. Software A Utilizar	12
1.2. Software Utilizado Para Escribir Éste Documento	13
1.3. Reflexiones	14
2. MÉTODOS ITERATIVOS PARA RAÍCES REALES	15
2.1. Método De la Bisección	15
2.1.1. Introducción	15
2.1.2. Algoritmo Bisección	15
2.2. Método Regula Falsi	18
2.2.1. Introducción	18
2.2.2. Algoritmo Regula-Falsi	19
2.3. Método De Punto Fijo	22
2.3.1. Introducción	22
2.3.2. Convergencia Del Método De Punto Fijo	22
2.3.3. Algoritmo Del Punto Fijo	23
2.4. El Método De Wegstein	26
2.4.1. Introducción	26
2.4.2. Algoritmo De Wegstein	26
2.5. Método De Newton	27
2.5.1. Introducción	27
2.5.2. Ejemplo Gráfico Del Método De Newton	28
2.5.3. Algoritmo De Newton	28
2.6. Método De La Secante	31
2.6.1. Introducción	32
2.6.2. Ejemplo Gráfico Método Secante	32
2.6.3. Algoritmo Secante.	34
2.7. Métodos Illinois Y Pegasus	36
2.7.1. Algoritmo De Illinois	36
2.7.2. Algoritmo De Pegasus	38
2.8. Método De Steffensen	39
2.8.1. Introducción	39

2.8.2. Combinando el método de Δ^2 –Aitken y Punto Fijo Para Steffensen	40
2.8.3. Algoritmo De Steffensen	41
3. EJERCICIOS RESUELTOS MÉTODOS ITERATIVOS	45
4. MÉTODOS PARA POLINOMIOS	123
4.1. Propiedades De Los Polinomios	123
4.2. Método McLaurin De Acotación De Raíces	125
4.3. Separación De Raíces Reales	127
4.3.1. Regla De Los Signos De Descartes	127
4.3.2. Método De Sturm	129
4.3.3. Método de Sturm	131
4.4. Polinomios Con Coeficientes Racionales	132
5. EJERCICIOS RESUELTOS POLINOMIOS	143
6. CÁLCULO DE RAÍCES COMPLEJAS	167
6.1. Método De Müller	167
6.1.1. Introducción	167
6.1.2. Ejemplo Gráfico Del Método De Müller	169
6.1.3. Algoritmo De Müller	171
Bibliografía	173

Índice de figuras

1.	Método Newton $f(x) = x^2 - 2$ para aproximar $\sqrt{2}$	29
2.	Método Secante $f(x) = x^2 - 2$ para aproximar $\sqrt{2}$	33
1.	Gráfica polinomio $p(x) = 2x^3 - 3x - 1$	46
2.	Método Bisección $p(x) = 2x^3 - 3x - 1$ en $[1, 1.5]$	49
3.	Método Bisección $p(x) = 2x^3 - 3x - 1$ en $[-0.75, -0.25]$	50
4.	Función $f(x) = e^{-x^2}$	51
5.	Función $f(x) = 2\cos(x)$ recta $y = x$	54
6.	Función $f(x) = x - 2\cos(x)$	56
7.	Tasas De Convergencia	58
8.	Billar Circular	59
9.	Función Billar	60
10.	Tasas Convergencia Billar Newton	63
11.	Catenaria $x \cosh\left(\frac{1}{x}\right) - 0.1$	65
12.	Catenaria $\frac{1}{\operatorname{acosh}\left(\frac{x+0.1}{x}\right)}$	66
13.	TCL Catenaria $x \cosh\left(\frac{1}{x}\right) - 0.1$	67
14.	TCL Catenaria $\frac{1}{\operatorname{acosh}\left(\frac{x+0.1}{x}\right)}$	67
15.	Regula Falsi $p(x) = x^{10} - 1$ en $[0.5, 1.5]$	69
16.	TC Newton $p(x) = x^{10} - 1$ en $[0.5, 1.5]$	70
17.	Función Logística $f(x) = ax(1-x)$ $a \in [1, 4]$ <i>anatural</i>	73
18.	TCL Función Logística $f(x) = 0.5x(1-x)$	74
19.	TCL Función Logística $f(x) = \frac{3}{2}x(1-x)$ Estimación inicial 0.1	75
20.	TCL Función Logística $f(x) = \frac{3}{2}x(1-x)$ Estimación inicial 0.375	75
21.	Pto. Fijo Función Logística $f(x) = \frac{7}{2}x(1-x)$	76
22.	Errores Pto. Fijo Función Logística $f(x) = \frac{15}{4}x(1-x)$	77
23.	Fractal De Feigenbaum	78
24.	$f(x) = 8x - \cos x - 2x^2$	86
25.	Derivada de $g_1(x)$	90
26.	Derivada de $g_3(x)$	91
27.	Función De Raabe $f(a) = a \ln a - 2a + \ln(\sqrt{2\pi})$	93
28.	Problema de la caja	94
29.	Función $f(x) = x^3 - 13x^2 + 40x - 25$	95

30.	FReynolds	109
31.	FReynolds2	113
32.	Savitsky1	120
1.	Corona Circular Acotación McLaurin	127
2.	Polinomio $f(x) = x^4 - 4x^3 - x^2 + 12x - 6$	139
1.	Raíces Unidad De $x^{11} + 1 = 0$	145
1.	Müller para $f(x) = 16x^4 - 40x^3 + 5x^2 + 20x + 6$	170

Listado de Teoremas, Proposiciones, Corolarios Y Definiciones

2.1.1. Bolzano	15
2.3.1. Punto Fijo	22
2.5.1. Convergencia De Newton	28
2.8.1. Convergencia Cuadrática Steffensen	40
2.8.1. Diferencia Progresiva	40
3.0.1. Número Iteraciones	109
4.1.1. Ecuación Algebraica	123
4.1.1. Teorema Fundamental Del Álgebra	123
4.1.1. Galois	123
4.1.2. División De Polinomios	124
4.1.2. Polinomio Divisor	124
4.1.1. El Resto	124
4.1.3. Multiplicidad De Una Raíz	124
4.1.4. MCD Polinomios	124
4.2.1. Teorema Acotación McLaurin	125
4.2.1. Corolario Acotación McLaurin	125
4.2.1. Corona Circular	126
4.3.1. Separación Raíces En Intervalos	127
4.3.1. Cambios De Signo	128
4.3.1. Regla de los signos de Descartes	128
4.3.2. Secuencia De Sturm	129
4.3.1. Multiplicidad Polinomios	129
4.3.2. Teorema De Sturm	130
4.3.1. Número Raíces	130
4.4.1. Divisores Término Independiente	133
4.4.2. Estudio Raíces Racionales En Polinomios	133

Listado Ejercicios y Ejemplos

3.0.1. Polinomio Bisección	45
3.0.2. Coseno Regula-Falsi	50
3.0.3. Coseno Punto Fijo	53
3.0.4. Raíces Quintas	56
3.0.5. Función Billar	58
3.0.6. Problema Catenaria	64
3.0.7. Raíces Décimas	68
3.0.8. Ecuación Logística	70
3.0.9. Diodos	80
3.0.10. Economía	83
3.0.11. Coseno Diferentes Métodos	85
3.0.12. Identidad De Raabe	92
3.0.13. Problema De La Caja	94
3.0.14. Método De Halley	97
3.0.15. Polinomio Grado Dos	99
3.0.16. Sistema No Lineal	100
3.0.17. Esfera Sumergida	101
3.0.18. Problema Partícula	103
3.0.19. Problema Reynolds-Colebrook	106
3.0.20. Método Savitsky	117
4.2.1. Acotación De Raíces De Un Polinomio	126
4.3.1. Determinamos Cambios De Signo	128
4.3.2. Hallamos Raíces	128
4.3.3. Determinamos Raíces	130
4.3.4. Separamos En Intervalos	132
4.4.1. Resolvemos Polinomio De Grado Cuatro	133
4.4.2. Otro Más De Hallar Raíces	136
4.4.3. Ejemplo Deflación	139

5.0.1. Polinomio Grado Tres 143

5.0.2. Polinomio Grado Trece 144

5.0.3. Secuencia Sturm 147

5.0.4. Polinomio Grado Nueve 148

5.0.5. Polinomio Coeficientes PI 151

5.0.6. Polinomio Grado Cinco 153

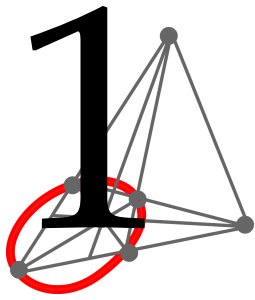
5.0.7. Raíces Complejas 155

5.0.8. Acotar Raíces 159

5.0.9. Separación De Raíces 160

5.0.10. Aproximamos Raíces 163

5.0.11. Estudiamos Raíces 164



PREFACIO

Aquí empieza un proyecto personal de hacer llegar al estudiante universitario la capacidad de aprender Métodos Numéricos utilizando software libre, en nuestro caso he optado por el uso del software Octave. Octave es una alternativa libre y bastante compatible con el software que la inmensa mayoría conocemos, y que me abstengo a mentar.

No sólo contamos con Octave, si quieres puedes utilizar como alternativa el software Scilab, de hecho; no creo que los códigos que presento en este trabajo sea muy difícil de poder portarlos a Scilab.

El presente texto no es una guía o tutorial de Octave, pienso que los tutoriales llegan a ser demasiado pesados, particularmente aprendo más realizando directamente mis tareas matemáticas. Hoy en día el San Google y algún foro que vaya sobre tus dudas te pueden solucionar la vida, evitando tener que leer eternos manuales con infinidad de comandos. Por descontado, lo principal es tener interés en lo que estes haciendo, con esa premisa se derriban multitud de barreras.

Este libro en formato PDF trata sobre diferentes formas y algoritmos para poder calcular las raíces (o ceros) de cualquier ecuación no lineal de una variable. Se divide principalmente en dos partes. Una primera parte contiene los principales métodos iterativos: bisección, regula, punto fijo, Newton-Raphson, etc. Una segunda parte esta dedicada a cómo calcular las raíces de las ecuaciones polinómicas: desde Ruffini hasta el método de Sturm. Terminamos con una dedicación especial a cómo hallar raíces complejas aproximadas con el método de Müller. He incluido una gran cantidad de problemas resueltos que he sacado de diferentes libros que tengo en mi posesión y que están debidamente nombrados al final del documento en el apartado de bibliografía.

Éste documento ha sido realizado en su integridad bajo software libre, utilizando \LaTeX como código para generar el documento, y distribuciones Gnu/Linux. Todos los códigos realizados aquí los puedes descargar de forma gratuita en el siguiente enlace de GitHub.

[ENLACE PARA DESCARGAR CÓDIGOS OCTAVE](#)

Los códigos son principalmente de guía, un modelo a seguir, el cual tú puedes cambiar a tu gusto o conveniencia. Lo que sí te recomiendo es que no te dediques a copiar y pegar los códigos, porque lo más seguro es que no obtengas experiencia ni con Octave ni realizando el estudio de las

raíces.

Otra cosa que te recomiendo es que seas crítico con las soluciones aquí expuestas y las que tú mismo/a obtengas, en la justa medida. Realiza las tareas que te dejo pendientes, si un ejercicio está realizado con determinados métodos, puedes probar con otros métodos.

1.1 Software A Utilizar

Definitivamente hay multitud de software que puedes utilizar para calcular las raíces de una ecuación con los métodos propuestos en éste documento. Los que te aconsejo:

1. [GNU Octave](#)-> Alternativa libre a Matlab, es el programa principal que se utiliza en este documento. Esta disponible para Mac-OS, Windows, Linux y Android.
2. [Octave Source Packages](#)-> Conjunto de paquetes que extienden las capacidades de Octave, altamente recomendable.
3. [EPSTK](#)-> Conjunto de comandos extra Octave para una mejor resolución en los gráficos. Viene muy bien para dibujar en coordenadas polares.
4. [QtOctave](#)-> Es un GUI basado en QT4 que te permite un uso más amigable de Octave. En Linux lo puedes encontrar en los repositorios de las principales distribuciones. Fue creado hace unos años por un español, que abandonó el proyecto, pero la comunidad Linux sigue dándole soporte porque es muy útil.
5. [WxMaxima](#)-> Es un GUI basado en wxWidgets el cual facilita el cálculo simbólico y no simbólico bajo el programa Maxima. Disponible en las 3 principales plataformas. Viene muy bien como apoyo para cálculos simbólicos de raíces y demás en éste documento. Muy recomendable su uso.
6. [Sublime Text + SublimeREPL](#)-> Aunque no es código libre sí es muy seguido este editor de textos no enriquecido. Viene muy bien para editar nuestros propios códigos de Octave. El plugin [SublimeREPL](#) es un muy buen plugin de Sublime Text que nos permite ejecutar nuestros códigos Octave desde el mismo editor. Viene a ser un muy buen sustituto de QtOctave.
7. [Cantor](#)-> Si eres linuxero y usas KDE te puede venir muy bien para trabajar con Maxima el programa Cantor.

1.2 Software Utilizado Para Escribir Éste Documento

Éste manual ha sido escrito en su totalidad con el lenguaje \LaTeX utilizando los siguientes editores:

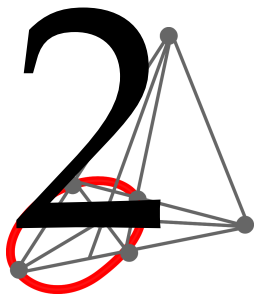
- [LyX](#)
- [TeXMaker](#)
- [TeXStudio](#)
- [Gummi](#)

Como documentos de apoyo he utilizado intensamente [Edición De Textos Científicos Con Latex](#), y me ha servido de apoyo el foro [TEX de Stackexchange](#). Para dibujar muchas de las gráficas de Octave en éste documento he utilizado [Matlab2TikZ](#).

1.3 Reflexiones

Antes de terminar el prefacio me gustaría que te quedase muy claro lo siguiente, es cierto que con cualquier programa matemático puedes calcular las raíces con tan sólo un comando, sin tener que escribir los algoritmos, pero el objetivo de esta materia en las universidades y en éste documento es que tú conozcas cómo funcionan los diferentes métodos y seas capaz de elegir cuál es el mejor para determinadas ecuaciones no lineales. Además, este tipo de estudio viene muy bien para afianzar conceptos matemáticos, como también a conocer nuevos conceptos. Otra cosa a tener muy en cuenta es que te ayudan a aprender a ser crítico con los resultados que puedas obtener con una calculadora o con un determinado software científico.

Sin más, te dejo con el contenido de esta especie de manualillo que espero sinceramente que te sea de verdadera ayuda para aprender este campo específico de las matemáticas aplicadas.



MÉTODOS ITERATIVOS PARA RAÍCES REALES

2.1 Método De la Bisección

2.1.1. Introducción

El método de la bisección consiste básicamente en encontrar la raíz de una función en un determinado intervalo. Dicho intervalo lo vamos dividiendo cada paso por la mitad, quedándonos con la mitad que tenga la raíz. Al hacer cada vez más pequeños los intervalos lo que conseguimos es una mejor aproximación a la raíz buscada. El proceso de ir partiendo los intervalos se apoya en el método de la sucesión de intervalos encajados de Cantor.

Para cerciorarnos que la raíz se encuentra en un determinado intervalo lo que hacemos en realidad es aplicar el Teorema de Bolzano.

Teorema 2.1.1 ► Bolzano

Sea una función cualquiera $f : [a, b] \rightarrow \mathbb{R}$, tal que $f \in C([a, b])$. Entonces:
Si $f(a) \cdot f(b) < 0 \Rightarrow \exists c \in]a, b[/ f(c) = 0$

El Teorema de Bolzano lo que nos dice es que si en un intervalo dado una función continua posee un cambio de signo entonces, podemos asegurar que la función posee al menos una raíz en dicho intervalo, independientemente de la multiplicidad.

2.1.2. Algoritmo Bisección

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```
1 % Metodo Biseccion
```

```
2 function [] = biseccion(f,a,b,tol,maxiter)
3     filename = 'biseccion.txt';
4     fid = fopen(filename,'w');
5     fprintf(fid,'k\t\ttx (k)\t\tt\t\tt\t\ttError\n');
6
7     t = a:0.01:b;
8
9     fx = inline(f);
10    func = fx(t);
11
12    fa = feval (fx,a);
13    fb = feval (fx,b);
14
15    if fa*fb > 0
16        disp ('La funcion debe tener signo distinto en los extremos del intervalo')
17        return
18    end
19    iter = 0;
20    x = [ ];
21    errores = [ ];
22    imagenes = [ ];
23    incr = b-a;
24    % warning('off', 'Octave:possible-matlab-short-circuit-operator');
25    while (incr > tol) && ( iter < maxiter)
26        c = (a+b)/2;
27        x = [x,c];
28        fc = feval (fx,c);
29        if fc == 0
30            a = c;
31            b = c;
32        elseif fa*fc < 0
33            b = c;
34            fb = fc;
35        else
36            a = c;
37            fa = fc;
38        end
39        errores =[errores,incr];
40        imagenes = [imagenes,fc];
41        incr = b-a;
42        iter++;
43    end
44    iter2 = 1:iter;
45    sol = c;
46    if incr > tol
```



```
47     disp('Insuficientes iteraciones.')
48 else
49     for k=1:iter
50         fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter2(k),x(k),errores(k));
51     end
52     fprintf (fid,'\nLa ósolucin es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
    error de %e',
53         sol,iter,errores(end));
54     fclose(fid);
55     printf ('La solucion es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un error de %
    e\n',
56         sol,iter,errores(end));
57 end
58 clf();
59 subplot(1,3,1);
60 fplot(fx,[a-5,a+5])
61 set (gca, 'axislocation', 'zero');
62 set (gca, 'axislocation', 'zero');
63 set(gca, 'box', 'off');
64 grid;
65 title('La Funcion');
66 legend({f},'location','southoutside');
67 legend boxon;
68 xlabel('y');
69 ylabel('x');
70 set (get (gca, 'ylabel'), 'rotation', 360);
71 subplot(1,3,2);
72 plot(t,func,x,imagenes,'*r')
73 set (gca, 'axislocation', 'zero');
74 set (gca, 'axislocation', 'zero');
75 set(gca, 'box', 'off');
76 grid;
77 title('Metodo Biseccion');
78 legend({f,'Aproximaciones'},'location','southoutside');
79 legend boxon;
80 xlabel('y');
81 ylabel('x');
82 set (get (gca, 'ylabel'), 'rotation', 360);
83 subplot(1,3,3);
84 plot(iter2,errores,iter2,errores,'*r')
85 set (gca, 'axislocation', 'bottom');
86 set (gca, 'axislocation', 'right');
87 set(gca, 'box', 'off');
88 grid;
89 title('Tendencia de errores en Biseccion');
```

```

90 legend({'Errores', 'Puntos Errores'}, 'location', 'southoutside');
91 legend boxon;
92 xlabel('Numero Iteraciones');
93 ylabel('Errores');
94 set (get (gca, 'ylabel'), 'rotation', 360);
95 end

```

biseccion.m

En los ejercicios se ve con todo detalle los pasos que se realizan en el método de la bisección, al igual que en los demás métodos.

2.2 Método Regula Falsi

2.2.1. Introducción

Éste método trata de mejorar el de la bisección en su velocidad de convergencia, para mejorarlo no tomamos cada vez el punto medio del intervalo $[a, b]$ sino que se calcula en cada iteración la intersección con el eje de abscisas la recta que pasa por los puntos $(a, f(a))$ y $(b, f(b))$, siendo f la curva de partida del problema.

Consideremos el punto $P(c, 0)$ y calculemos la expresión para la recta. Tenemos que el vector director de la recta es:

$$\vec{v}(b - a, f(b) - f(a))$$

De esta forma podemos expresar la ecuación de la recta en su forma continua como:

$$\frac{x - a}{b - a} = \frac{y - f(a)}{f(b) - f(a)}$$

Sustituimos el punto P y despejamos c :

$$\begin{aligned} \frac{c - a}{b - a} &= \frac{-f(a)}{f(b) - f(a)} \Rightarrow c - a = \frac{-f(a) \cdot (b - a)}{f(b) - f(a)} \Rightarrow c = \frac{-f(a) \cdot (b - a)}{f(b) - f(a)} + a \Rightarrow \\ &\Rightarrow c = \frac{-b \cdot f(a) + a \cdot f(a) + a \cdot f(b) - a \cdot f(a)}{f(b) - f(a)} \\ c &= \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)} \end{aligned}$$

Éste c es el que se irá iterando cada vez en el método de la Regula Falsi, es decir, son nuestras aproximaciones para obtener la raíz que se busca de una función o curva f en un intervalo $[a,b]$. Para poder iterarlo hemos de darnos cuenta que el valor c es el que hace el papel del punto medio de la bisección, por lo que obtenemos a partir de $[a,b]$ 2 nuevos subintervalos: $[a,c]$ y $[c,b]$. A cada subintervalo le aplicamos el Teorema de Bolzano, y en el subintervalo que se cumpla el teorema le volvemos a aplicar el método de la Regula Falsi, es decir; volvemos a iterar. De esa forma lo que obtenemos es una sucesión finita de aproximaciones a la raíz que buscamos en un intervalo $[a,b]$ y que podemos definirla como $\{c_i\}_{i=0}^n$. Esto es en esencia los pasos que realizamos en éste método, al que hay que añadir los diferentes criterios de parada.

2.2.2. Algoritmo Regula-Falsi

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```
1 function [] = regulafalsi (f,a,b,tol,maxiter)
2     filename = 'regulafalsi.txt';
3     fid = fopen(filename,'w');
4     fprintf(fid,'k\t\ttx (k)\t\t\t\t\tError\n');
5
6     t = a:0.01:b;
7
8     fx = inline(f);
9     func = fx(t);
10
11     fa = feval(fx,a);
12     fb = feval(fx,b);
13     if fa*fb > 0
14         disp('La funcion debe tener signo distinto en los extremos del intervalo.')
15         return
16     end
17     iter = 0;
18     x = a;
19     errores = [ ];
20     imagenes = [ ];
21     incr = b-a;
22     while incr > tol && iter < maxiter
23         c = (a*fb-b*fa)/(fb-fa);
24         x = [x,c];
25         fc = feval(fx,c);
26         if fc == 0
27             a = c; b = c;
28         elseif fa*fc < 0
29             b = c; fb = fc;
30         else
```

```

31         a=c; fa=fc;
32     end
33     incr = abs(x(end)-x(end-1));
34     errores =[errores,incr];
35     iter++;
36 end
37 iter2 = 1:1:iter;
38 fc1 = feval(fx,x);
39 imagenes = [imagenes,fc1];
40 sol = c;
41 if incr > tol
42     disp('Insuficientes iteraciones.')
43 else
44     for k=1:iter
45         fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter2(k),x(k),errores(k));
46     end
47     fprintf (fid,'\nLa ósolucin es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
48     error de %e',sol,iter,errores(end));
49     fclose(fid);
50     printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\nCon un error de
51     %e\n',
52         sol, iter,errores(end));
53     end
54
55 clf();
56 subplot(1,3,1);
57 fplot(fx,[a-5,b+5])
58 set (gca, 'axislocation', 'zero');
59 set (gca, 'yaxislocation', 'zero');
60 set(gca, 'box', 'off');
61 grid;
62 title('La Funcion');
63 legend({f},'location', 'southoutside');
64 legend boxon;
65 xlabel('y');
66 ylabel('x');
67 set (get (gca, 'ylabel'), 'rotation', 360);
68 subplot(1,3,2);
69 plot(t,func,x,imagenes,'*r')
70 set (gca, 'axislocation', 'zero');
71 set (gca, 'yaxislocation', 'zero');
72 set(gca, 'box', 'off');
73 grid;
74 title('Metodo Regula Falsi');
75 legend({f,'Aproximaciones'},'location','southoutside');

```

```
74     legend boxon;
75     xlabel('y');
76     ylabel('x');
77     set (get (gca, 'ylabel'), 'rotation', 360);
78     subplot(1,3,3);
79     plot(iter2,errores,iter2,errores,'*r')
80     set (gca, 'axislocation', 'bottom');
81     set (gca, 'yaxislocation', 'right');
82     set(gca, 'box', 'off');
83     grid;
84     title('Tendencia de errores en Regula Falsi');
85     legend({'Errores','Puntos Errores'},'location','southoutside');
86     legend boxon;
87     xlabel('Numero Iteraciones');
88     ylabel('Errores');
89     set (get (gca, 'ylabel'), 'rotation', 360);
90 end
```

regulafalsi.m

2.3 Método De Punto Fijo

Definición 2.3.1 ► Punto Fijo

Dada una función $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ y un punto $x^* \in D$, decimos que x^* es un punto fijo si $f(x^*) = x^*$, es decir; x^* es solución de la ecuación $f(x) = x$; $\forall x \in D$.

2.3.1. Introducción

En éste método consideramos una función $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ cualquiera a la que queremos calcular las soluciones de la ecuación $f(x) = 0$. El método consiste en transformar dicha ecuación en una de punto fijo $g(x) = x$, de la cual al ir iterándola obtenemos la sucesión de iterados: $x_{k+1} = g(x_k) \forall k \in \mathbb{N}$.

En el método ya no partimos de un intervalo inicial $[a, b]$ para encontrar aproximaciones a la raíz, lo que hacemos es partir de un valor inicial x_0 que esté lo más próximo posible a la raíz a calcular. Para determinar éste valor inicial la forma más sencilla es dibujar con Octave la gráfica de la función, aunque también podemos utilizar el Teorema de Bolzano para comprobar su existencia, esto lo mejor es hacerlo a mano; o utilizando previamente el método de la Bisección.

Una cosa muy importante a tener en cuenta en el método de punto fijo es que la función g no tiene porqué ser necesariamente única, basta tomar como ejemplo la sencilla función $f(x) = x^2 - 2x$. Si en $f \equiv 0$ despejamos x lo podemos hacer de 3 formas diferentes:

$$\text{Forma 1} \Rightarrow x^2 - 2x = 0 \Rightarrow x = \frac{x^2}{2} \Rightarrow g_1(x) = \frac{x^2}{2}$$

$$\text{Forma 2} \Rightarrow x^2 - 2x = 0 \Rightarrow x^2 = 2x \Rightarrow g_2(x) = \sqrt{2x}$$

$$\text{Forma 3} \Rightarrow x^2 - 2x = 0 \Rightarrow x^2 = 2x \Rightarrow g_3(x) = -\sqrt{2x}$$

A partir de las diferentes funciones de punto fijo nos queda determinar cuál es la mejor para la raíz que deseamos aproximar, tanto en velocidad de convergencia como en cuál es la que está más cerca a la raíz a aproximar.

2.3.2. Convergencia Del Método De Punto Fijo

Consideremos la función $g : [a, b] \rightarrow \mathbb{R}$ de clase $\mathcal{C}^1([a, b])$, tal que $g(x) \in [a, b] \forall x \in [a, b]$. Sea la sucesión de punto fijo $x_{k+1} = g(x_k) \forall k \in \mathbb{N}$.

Supongamos también que $\exists k \in \mathbb{R} / |g'(x)| < K < 1$

Se cumple que $x_{k+1} \xrightarrow[k \rightarrow \infty]{} x^*$ para cualquier $x_0 \in [a, b]$. Además, el error de la iteración n-ésima por:

$$|x_n - x^*| \leq \frac{K^n}{1-K} |x_0 - x_1| \quad n \geq 1$$

También se tiene que si g' es continua entonces:

- Si $|g'(x^*)| > 1 \Rightarrow$ Los iterados NO convergen a x^*
- Si $|g'(x^*)| > 1 \Rightarrow$ Los iterados Convergen Linealmente a x^* con tasa de convergencia $c_L = |g'(x^*)|$
- Si $g'(x^*) = 0 \Rightarrow$ Los iterados convergen cuadráticamente a x^*

2.3.3. Algoritmo Del Punto Fijo

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function [] = pfijo(g,f,a,b,x1,tol,maxiter)
2     filename = 'puntofijo.txt';
3     fid = fopen(filename,'w');
4     fprintf(fid,'k\t\ttx (k)\t\t\t\t\tError\n');
5
6     t = a:0.01:b;
7
8     fx = inline(f);
9     gx = inline(g);
10    func = fx(t);
11
12    iter = 1;
13    x = x1;
14    incr = tol+1;
15    errores = incr;
16    imagenes = [ ];
17    while incr > tol && iter <= maxiter
18        x(iter+1) = feval(gx,x(iter));
19        incr = abs(x(iter+1)-x(iter));
20        errores = [errores,incr];
21        iter++;
22    end
23
24    iter2 = 1:iter;
25    fc1 = feval(fx,x);
26    imagenes = [imagenes,fc1];

```

```

27
28 e = abs(diff(x));
29 tc1 = e(2:end) ./ e(1:end-1); %Tasa de convergencia lineal
30 tcc = e(2:end) ./ e(1:end-1).^2; %Tasa de convergencia cuadratica
31 %tccubica = e(3:end)./e(1:end-1).^3; %Tasa de convergencia cubica
32
33 XR = x(2:end);
34 XL = x(1:end-1);
35 format free;
36 dg = diff(XR)./diff(XL);
37
38 if incr > tol
39     sol = [ ];
40     disp('Insuficientes iteraciones');
41 else
42     sol = x(end);
43     for k=1:iter
44         fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter2(k),x(k),errores(k));
45     end
46     fprintf (fid,'\nLa ósolucin es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
47     error de %e',sol,iter,errores(end));
48     %fprintf(fid,'\n\nTasa De Convergencia Cubica\n%.15f',tccubica');
49     fclose(fid);
50     printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',
51             sol, iter);
52 end
53 clf();
54 figure(1),
55 subplot(1,3,1);
56 fplot(fx,[a-5,b+5])
57 set (gca, 'axislocation', 'zero');
58 set (gca, 'yaxislocation', 'zero');
59 set(gca, 'box', 'off');
60 grid;
61 title('La Funcion');
62 legend({f},'location', 'southoutside');
63 legend boxon;
64 xlabel('y');
65 ylabel('x');
66 set (get (gca, 'ylabel'), 'rotation', 360);
67 subplot(1,3,2);
68 plot(t,func,x,imagenes,'*r')
69 set (gca, 'axislocation', 'zero');
70 set (gca, 'yaxislocation', 'zero');
71 set(gca, 'box', 'off');

```



```
71 grid;
72 title('Metodo Punto Fijo');
73 legend({f, 'Aproximaciones'}, 'location', 'southoutside');
74 legend boxon;
75 xlabel('y');
76 ylabel('x');
77 set (gca, 'ylabel', 'rotation', 360);
78 subplot(1,3,3);
79 plot(iter2,errores,iter2,errores,'*r')
80 set (gca, 'axislocation', 'bottom');
81 set (gca, 'axislocation', 'right');
82 set(gca, 'box', 'off');
83 grid;
84 title('Tendencia de errores en el Punto Fijo');
85 legend({'Errores', 'Puntos Errores'}, 'location', 'southoutside');
86 legend boxon;
87 xlabel('Numero Iteraciones');
88 ylabel('Errores');
89 set (gca, 'ylabel', 'rotation', 360);
90
91 figure(2);
92     subplot(1,2,1);
93     plot(iter2(1:length(tcl)),tcl)
94     set (gca, 'axislocation', 'bottom');
95     set (gca, 'axislocation', 'right');
96     set(gca, 'box', 'off');
97     grid;
98     title('Tasa Convergencia Lineal Punto Fijo');
99     legend({'T.C.L.'}, 'location', 'southoutside');
100    legend boxon;
101    xlabel('Numero Iteraciones');
102    ylabel('tcl');
103    subplot(1,2,2);
104    plot(iter2(1:length(tcc)),tcc)
105    set (gca, 'axislocation', 'bottom');
106    set (gca, 'axislocation', 'right');
107    set(gca, 'box', 'off');
108    grid;
109    title('Tasa Convergencia Cuadratica Newton');
110    legend({'T.C.C.'}, 'location', 'southoutside');
111    legend boxon;
112    xlabel('Numero Iteraciones');
113    ylabel('tcc');
114 end
```

pfijo.m

2.4 El Método De Wegstein

2.4.1. Introducción

También conocido como el método del **promotor de convergencia de Wegstein**. Éste método es una mejora sustancial del método del punto fijo. Consideremos la ecuación a resolver $f(x) = 0$, y su respectiva función de punto fijo $g(x)$; siendo x_0 el valor o estimador inicial. Los pasos a seguir en el método son los siguientes:

1. Al comienzo del paso $k-1$ -ésimo se conoce x_{k-1}
2. Se actualiza x_{k-1} con el punto fijo: $x_k = g(x_{k-1})$
3. Se calculan:

$$A = \frac{g(x_k) - x_k}{x_k - x_{k-1}}$$

$$\alpha = \frac{1}{1 - A}$$

4. Se corrige x_k con la sucesión de iterados:

$$x_{k+1} = x_k + \alpha \cdot (g(x_k) - x_k); \forall k \in \mathbb{N}$$

2.4.2. Algoritmo De Wegstein

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function [] = wegstein(f, x0, epsilon, max1)
2 fx = inline(f);
3 for k=1:max1
4     x1 = feval(fx, x0);
5     y1 = feval(fx, x1);
6     delta = (y1 - x1) / (x1 - x0);
7     alpha = 1 / (1 - delta);
8     x0 = x1 + alpha * (y1 - x1);
9     if abs(x1 - x0) < epsilon

```

```
10     break;
11   endif
12 endfor
13 err=abs(x1-x0)/2;
14 printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',
15         x0, k);
16 printf('Con un error de %.15e\n',err);
17 end
```

wegstein.m

2.5 Método De Newton

2.5.1. Introducción

El método de Newton consiste en aproximar las soluciones de la ecuación $f(x) = 0$ mediante la recta tangente, por lo que como primera premisa se debe cumplir que la función f sea derivable de orden 1 en un entorno, lo suficientemente grande, de la raíz a aproximar. Partimos siempre de una aproximación inicial de la raíz, x_0 , la cual podemos estimar fácilmente dibujando la gráfica de la función f .

Una vez determinado x_0 , en la primera iteración debemos calcular la recta tangente al punto $(x_0, f(x_0))$. Dicha recta tangente se puede calcular a partir de la ecuación de una recta en la forma punto-pendiente:

$$y - y_0 = m \cdot (x - x_0)$$

Siendo $y_0 = f(x_0)$. Como se sabe de Bachiller, la pendiente es $m = f'(x_0)$. Tomamos un punto x_1 que es la intersección de la recta tangente con el eje OX, luego $y = 0$, sustituimos y despejamos x_1 :

$$y = f(x_0) + f'(x_0) \cdot (x - x_0) \Rightarrow 0 = f(x_0) + f'(x_0) \cdot (x_1 - x_0) \Rightarrow -f(x_0) = f'(x_0) \cdot (x_1 - x_0) \Rightarrow x_1 - x_0 = \frac{-f(x_0)}{f'(x_0)} \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Si iteramos éste procedimiento obtenemos la sucesión de iterados del método de Newton:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}; \forall k \in \mathbb{N}$$

Podemos garantizar la convergencia cuadrática del método mediante el uso del siguiente teorema:

Sea $f \in \mathcal{C}^2([a, b])$, consideremos $x^* \in [a, b]$ una raíz de la ecuación $f(x) = 0$, tal que $f'(x^*) \neq 0$ (i.e. ; es una raíz simple)
 $\Rightarrow x_k \xrightarrow[k \rightarrow \infty]{} x^*$ para cualquier aproximación inicial x_0 , con velocidad de convergencia cuadrática, $\mathcal{O}(n^2)$.
 Si $f'(x^*) = 0 \Rightarrow$ El método de Newton converge linealmente, $\mathcal{O}(n)$.

Consideremos la función $f(x) = x^2 - 2$ con ecuación $x^2 - 2 = 0$, sus soluciones son $\pm\sqrt{2}$. Aproximemos de las dos raíces $x^* = \sqrt{2}$ con el método de Newton. Obviamente $f \in \mathcal{C}^2([1, 1.5])$, con $f'(\sqrt{2}) = 2\sqrt{2} \neq 0$, luego se cumplen las condiciones del teorema anterior, así que la convergencia es cuadrática. Tomemos como valor inicial $x_0 = 1.25$.

En la figura adjunta puedes ver las primeras 2 iteraciones para éste ejemplo, cabe destacar la rapidez del método, con tan sólo 2 iteraciones obtenemos una buena aproximación de la raíz, ya que las 4 primeras cifras decimales son exactas.

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function [] = newton(f,df,x1,a,b,tol,maxiter)
2
3     filename = 'newtonraphson.txt';
4     fid = fopen(filename,'w');
5     fprintf(fid,'k\t\ttx (k)\t\t\t\t\t\t\tError\n');
6
7     tic;
8     fx = inline(f);
9     dfx = inline(df);
10    t = x1-5:0.01:x1+5;

```

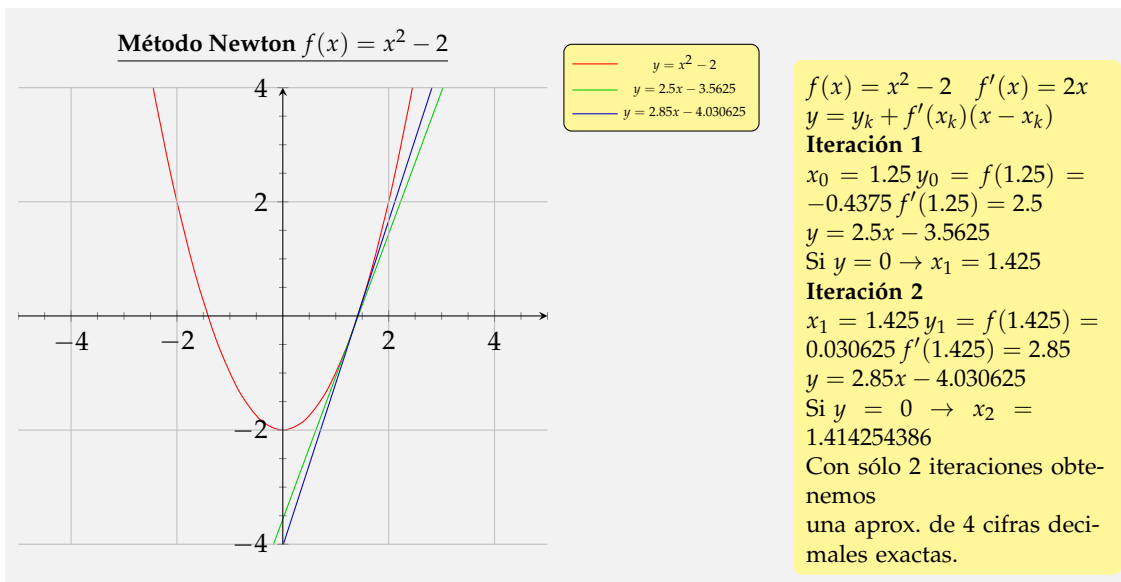


Figura 1: Método Newton $f(x) = x^2 - 2$ para aproximar $\sqrt{2}$

```

11
12     func = fx(t);
13
14     k = 1;
15     x = x1;
16     incr = tol+1;
17     sol = [ ];
18     errores = incr;
19     imagenes = [ ];
20     while incr > tol && k <= maxiter
21         delta = -feval(fx,x(k))/feval(dfx,x(k));
22         x(k+1) = x(k) + delta;
23         incr = abs(delta);
24         errores =[errores,incr];
25         sol = [sol,x];
26         k = k+1;
27     end
28
29     display('El algoritmo tiene un coste en tiempo de: ')
30     toc;
31     iter2 = 1:k;
32     fc1 = feval(fx,sol);
33     imagenes = [imagenes,fc1];
34
35     e = abs(diff(x));
  
```

```

36     tcl = e(2:end) ./ e(1:end-1); %Tasa de convergencia lineal
37     tcc = e(2:end) ./ e(1:end-1).^2; %Tasa de convergencia cuadratica
38
39     if incr > tol
40         sol = [ ];
41         disp('Diverge o insuficientes iteraciones');
42     else
43         sol = x(end);
44         for m=1:k
45             fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter2(m),x(m),errores(m));
46         end
47         fprintf(fid,'\nLa solución es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
error de %e\n\n',sol(end),k,errores(end));
48         printf ('La solución es %.15f, y se ha alcanzado en %d iteraciones.\n',
sol,k);
49     end
50
51     fprintf(fid,'Tasa de convergencia lineal = %f \n',tcl);
52     fprintf(fid,'Tasa de convergencia cuadratica = %f \n',tcc);
53     fclose(fid);
54
55     clf();
56     figure(1);
57     subplot(1,3,1);
58     fplot(fx,[a,b]);
59     set (gca, 'axislocation', 'zero');
60     set (gca, 'axislocation', 'zero');
61     set(gca, 'box', 'off');
62     grid;
63     title('La Funcion');
64     legend({f},'location', 'southoutside');
65     legend boxon;
66     xlabel('y');
67     ylabel('x');
68     set (get (gca, 'ylabel'), 'rotation', 360);
69     subplot(1,3,2);
70     plot(t,func,sol,imagenes,'*r')
71     set (gca, 'axislocation', 'zero');
72     set (gca, 'axislocation', 'zero');
73     set(gca, 'box', 'off');
74     grid;
75     title('Metodo Newton Raphson');
76     legend({f,'Aproximaciones'},'location','southoutside');
77     legend boxon;
78     xlabel('y');
79     ylabel('x');

```

```
80     set (get (gca, 'ylabel'), 'rotation', 360);
81     subplot(1,3,3);
82     plot(iter2,errores,iter2,errores,'*r')
83     set (gca, 'axislocation', 'bottom');
84     set (gca, 'axislocation', 'right');
85     set(gca, 'box', 'off');
86     grid;
87     title('Tendencia de errores en Newton');
88     legend({'Errores', 'Puntos Errores'},'location','southoutside');
89     legend boxon;
90     xlabel('Numero Iteraciones');
91     ylabel('Errores');
92
93     figure(2);
94     subplot(1,2,1);
95     plot(iter2(1:length(tcl)),tcl)
96     set (gca, 'axislocation', 'bottom');
97     set (gca, 'axislocation', 'right');
98     set(gca, 'box', 'off');
99     grid;
100    title('Tasa Convergencia Lineal Newton');
101    legend({'T.C.L.'},'location','southoutside');
102    legend boxon;
103    xlabel('Numero Iteraciones');
104    ylabel('tcl');
105    subplot(1,2,2);
106    plot(iter2(1:length(tcc)),tcc)
107    set (gca, 'axislocation', 'bottom');
108    set (gca, 'axislocation', 'right');
109    set(gca, 'box', 'off');
110    grid;
111    title('Tasa Convergencia Cuadratica Newton');
112    legend({'T.C.C.'},'location','southoutside');
113    legend boxon;
114    xlabel('Numero Iteraciones');
115    ylabel('tcc');
116 end
```

newton.m

2.6 Método De La Secante

2.6.1. Introducción

El método de la secante tiene una velocidad de convergencia mayor que el de la bisección o de la regla, pero menor que el de Newton. Es un buen método para cuando el de Newton o el del punto fijo fallan, o cuando el cálculo de la primera derivada es muy costoso. Para aplicarlo se toman dos estimaciones iniciales x_1 y x_2 de la raíz o solución de la ecuación $f(x) = 0$, siendo f una curva cualquiera, a poder ser continua en un entorno de la solución o soluciones de la ecuación. En el método lo primero es calcular una recta secante a la curva/función f , es decir; una recta que pase por los puntos $(x_1, f(x_1))$ y $(x_2, f(x_2))$, luego se toma como siguiente estimación un x_3 que es la intersección de la secante con el eje OX. Si iteramos el proceso obtenemos como sucesión de iterados del método la expresión:

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}} \quad \forall k \in \mathbb{N}$$

También puedes encontrar la sucesión expresada de forma equivalente como:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \quad \forall k \in \mathbb{N}$$

La velocidad de convergencia del método es $\phi = \frac{1 + \sqrt{5}}{2}$, es decir; su velocidad de convergencia es el número áureo, es una convergencia superlineal.

2.6.2. Ejemplo Gráfico Método Secante

Consideremos la función $f(x) = x^2 - 2$ con ecuación $x^2 - 2 = 0$, sus soluciones son $\pm\sqrt{2}$. Aproximemos de las dos raíces $x^* = \sqrt{2}$ con el método de la Secante. Tomamos como puntos de partida $x_1 = 1$ y $x_2 = 2$. Para calcular las expresiones de las rectas secantes en cada iteración vamos a utilizar la expresión:

$$y = f(x_1) + \frac{x - x_1}{x_2 - x_1} [f(x_2) - f(x_1)]$$

Las sucesivas aproximaciones a la raíz las obtendremos sustituyendo en la expresión anterior $y = 0$ y despejando x .

$$x_1 = 1 \quad x_2 = 2 \implies y = -1 + \frac{x - 1}{1} 3 \implies y = 3x - 4 \implies x_3 = \frac{4}{3} = 1.33333333 \dots$$

$$x_1 = \frac{4}{3} \quad x_2 = 2 \implies y = -\frac{2}{9} + \frac{x - \frac{4}{3}}{\frac{2}{3}} \frac{20}{9} \implies y = \frac{10}{3}x - \frac{14}{3} \implies x_4 = \frac{7}{5} = 1.4$$

$$x_1 = \frac{7}{5} x_2 = 2 \Rightarrow y = -\frac{1}{25} + \frac{x - \frac{7}{5}}{3} \frac{51}{25} \Rightarrow y = \frac{17}{5} x - \frac{24}{5} \Rightarrow x_5 = \frac{24}{17} = 1.411764706$$

$$x_1 = \frac{24}{17} x_2 = 2 \Rightarrow y = -\frac{2}{289} + \frac{x - \frac{24}{17}}{10} \frac{580}{2899} \Rightarrow y = \frac{58}{17} x - \frac{82}{17} \Rightarrow x_6 = \frac{41}{29} = 1.413793103$$

$$x_1 = \frac{41}{29} x_2 = 2 \Rightarrow y = -\frac{1}{841} + \frac{x - \frac{41}{29}}{17} \frac{1683}{841} \Rightarrow y = \frac{99}{29} x - \frac{140}{29} \Rightarrow x_7 = \frac{140}{99} = 1.41414141 \dots$$

Luego después de 5 iteraciones obtenemos la aproximación $x^* = 1.4141414141 \dots$, es decir, obtenemos 3 cifras decimales exactas.

En la figura adjunta puedes ver las primeras 5 iteraciones para éste ejemplo.

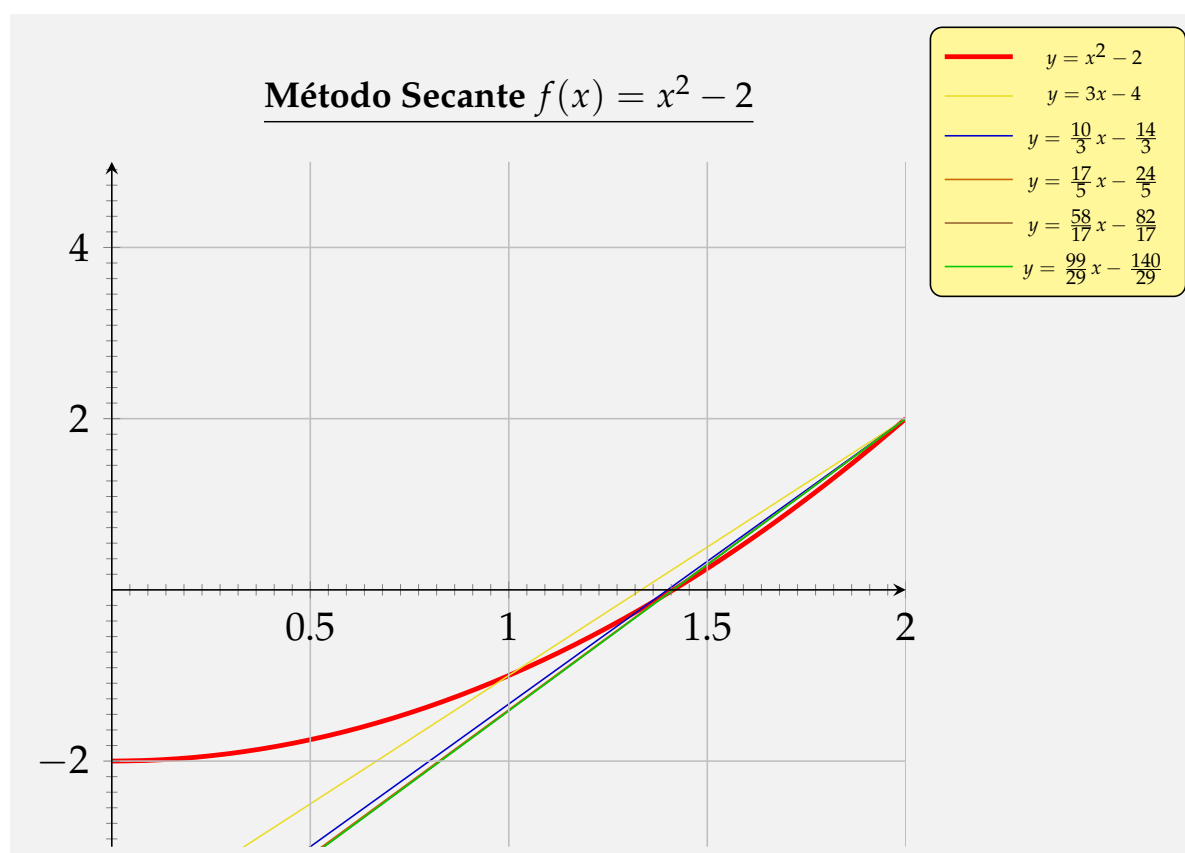


Figura 2: Método Secante $f(x) = x^2 - 2$ para aproximar $\sqrt{2}$

2.6.3. Algoritmo Secante.

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function [] = secante(f,x1,x2,tol,maxiter)
2     filename = 'secante.txt';
3     fid = fopen(filename,'w');
4     fprintf(fid,'k\t\ttx (k)\t\t\t\t\tError\n');
5
6     t = x1:0.01:x2;
7     fx = inline(f);
8     x = [x1,x2];
9     y = feval(fx,x);
10    func = fx(t);
11
12    k = 2;
13
14    sol = [ ];
15    errores = [ ];
16    incr = tol + 1;
17    delta = x2 - x1;
18    while incr > tol && k < maxiter
19        m = (y(k) - y(k-1))/delta;
20        delta = -y(k)/m;
21        x(k+1) = x(k) + delta;
22        y(k+1) = feval(fx,x(k+1));
23        sol = [sol,x];
24        incr = abs(delta);
25        errores=[errores,incr];
26        k++;
27    end
28
29    iter = 1:k-2;
30
31    if incr > tol
32        sol = [ ];
33        disp('Diverge o insuficientes iteraciones.')
34    else
35        sol = x(end);
36        for m=1:k-2
37            fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter(m),x(m),errores(m));
38        end
39        fprintf (fid,'\nLa ósolucin es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
40        error de %e',sol(end),k-2,errores(end));
41        fclose(fid);

```

```
41     printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',
42             sol,k-2);
43 end
44 clf();
45 subplot(1,3,1);
46 fplot(fx,[x1-5,x1+5])
47 set (gca, 'axislocation', 'zero');
48 set (gca, 'axislocation', 'zero');
49 set(gca, 'box', 'off');
50 grid;
51 title('La Funcion');
52 legend({f},'location', 'southoutside');
53 legend boxon;
54 xlabel('y');
55 ylabel('x');
56 set (get (gca, 'ylabel'), 'rotation', 360);
57 subplot(1,3,2);
58 plot(t,func,x,y,'*r')
59 set (gca, 'axislocation', 'zero');
60 set (gca, 'axislocation', 'zero');
61 set(gca, 'box', 'off');
62 grid;
63 title('Metodo Secante');
64 legend({f,'Aproximaciones'},'location','southoutside');
65 legend boxon;
66 xlabel('y');
67 ylabel('x');
68 set (get (gca, 'ylabel'), 'rotation', 360);
69 subplot(1,3,3);
70 plot(iter,errores,iter,errores,'*r')
71 set (gca, 'axislocation', 'bottom');
72 set (gca, 'axislocation', 'right');
73 set(gca, 'box', 'off');
74 grid;
75 title('Tendencia de errores en Secante');
76 legend('Errores','location','southoutside');
77 legend boxon;
78 xlabel('Numero Iteraciones');
79 ylabel('Errores');
80 set (get (gca, 'ylabel'), 'rotation', 360);
81 end
```

secante.m

2.7 Métodos Illinois Y Pegasus

Ambos métodos son generalizaciones del método de la Regula Falsi. Utilizan un paso de secante modificado para acelerar la convergencia lineal de dicho método.

Su algoritmo asociado comparte los dos primeros pasos con el de la Regula Falsi.

- 1) Se parte de un intervalo $]a, b[$ que contiene a la raíz x^* .
- 2) Se calculan $y_b = f(b)$ y $y_i = f(x_i)$, tal que $y_b \cdot y_i < 0$
- 3) Se define $x_k = \frac{x_i \cdot y_b - b \cdot y_i}{y_b - y_i}$ abscisa del punto de intersección con el eje OX de la recta secante que pasa por los puntos (x_i, y_i) y (b, y_b) .
- 4) Se evalúa $P := y_b \cdot y_k$

$$\text{Si } P > 0 \implies \begin{cases} b = x_k, & x_i = x_i \\ y_b = y_k, & y_i = \alpha \cdot y_i \end{cases}$$

$$\text{Si } P < 0 \implies \begin{cases} x_i = x_k, & b = b \\ y_i = y_k, & y_b = \alpha \cdot y_b \end{cases}$$

En donde se tiene:

$$\text{Método Illinois} \implies \alpha = \frac{1}{2}$$

$$\text{Método Pegasus} \implies \begin{cases} \alpha = \frac{y_b}{y_b + y_k}, & P > 0 \\ \alpha = \frac{y_i}{y_i + y_k}, & P < 0 \end{cases}$$

Ambos métodos tienen convergencia superlineal $\alpha \approx 1.642$.

2.7.1. Algoritmo De Illinois

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function []=illinois(f,a,b,delta,epsilon,max1)
2 %a,b extremos del intervalo que enmarca la raíz
3 %a=
4 %b=
5 %delta es la tolerancia para la raíz y epsilon para el valor de la funcion
6 %delta=

```

```
7 %epsilon=
8 %max1=
9 % max1 es el numero maximo de tolerancias
10 fx = inline(f);
11 ya=feval(fx,a);
12 yb=feval(fx,b);
13 %Evalua la funcion en los extremos del intervalo donde presuntamente se
14 %encuentra la raiz.
15 if ya*yb>0
16     disp('¡Ojo!: f(a)*f(b)>0')
17     break,
18 end
19 %Aqui no hay raiz en el intervalo en estudio.
20 for k=1:max1
21     dx=yb*(b-a)/(yb-ya);
22     c=b-dx;
23 %Se define la abcisa del punto c interseccion de la secante por
24 %(a,ya),(b,yb) con el eje x
25     ac=c-a;
26     yc=feval(fx,c);
27 %y se evalua su ordenada sobre la curva.
28     if yc==0
29         break;
30 %ya tenemos la solucion
31 elseif yb*yc>0
32 %una vez que sabemos que no es la raiz debemos saber en que intervalo (a,c)
33 %o (c,b) esta la raiz buscada, luego estudiamos los signos de yb*yc y de
34 %ya*yc
35     b=c;
36     ya=(0.5)*ya;
37     yb=yc;
38 %en este caso el nuevo intervalo (a,c) y multiplicamos la ya por 0.5
39     else
40         a=c;
41         ya=yc;
42         yb=(0.5)*yb;
43     end
44 %aqui el nuevo intervalo es (c,b)
45 dx=min(abs(dx),ac);
46 if abs(dx)<delta
47     break;
48 end
49 if abs(yc)<epsilon
50     break;
51 end
```

```

52 end
53 err=abs(b-a)/2;
54 yc=feval(fx,c);
55 printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',c, k);
56 printf ('Con un error de %e\n',err);
57 printf('Con f(raiz)=%e\n',yc);
58 clear;
59 end

```

illinois.m

2.7.2. Algoritmo De Pegasus

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function []=pegasus(f,a,b,delta,epsilon,max1)
2 fx = inline(f);
3 ya=feval(fx,a);
4 yb=feval(fx,b);
5 %Evalua la funcion en los extremos del intervalo donde presuntamente se
6 %encuentra la raiz.
7 if ya*yb>0
8     disp('¡Ojo!: f(a)*f(b)>0')
9     break,
10 end
11 %Aqui no hay raiz en el intervalo en estudio.
12 for k=1:max1
13     dx=yb*(b-a)/(yb-ya);
14     c=b-dx;
15     alpha=b/(b+c);
16     printf ('alpha = %.15f\n',alpha);
17 %Se define la abcisa del punto c interseccion de la secante por
18 %(a,ya),(b,yb) con el eje x
19     ac=c-a;
20     yc=feval(fx,c);
21 %y se evalua su ordenada sobre la curva.
22     if yc==0, break;
23 %ya tenemos la solucion
24 elseif yb*yc>0
25 %una vez que sabemos que no es la raiz debemos saber en que intervalo (a,c)
26 %o (c,b) esta la raiz buscada, luego estudiamos los signos de yb*yc y de
27 %ya*yc
28     b=c;
29     ya=(alpha)*ya;

```

```
30     yb=yc;
31     %en este caso el nuevo intervalo (a,c) y multiplicamos la ya por 0.5
32     else
33         a=c;
34         ya=yc;
35         yb=(alpha)*yb;
36     end
37     %aqui el nuevo intervalo es (c,b)
38     dx=min(abs(dx),ac);
39     if abs(dx)<delta,break,end
40     if abs(yc)<epsilon,break,end
41     end
42
43     err=abs(b-a)/2;
44     yc=feval(fx,c);
45     printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',c, k);
46     printf ('Con un error de %e\n',err);
47     printf ('Con f(raiz)=%e\n',yc);
48     clear;
49     end
```

pegasus.m

2.8 Método De Steffensen

2.8.1. Introducción

El método de Steffensen es un método con velocidad de convergencia cuadrática $\mathcal{O}(n^2)$, al igual que en Newton. Se basa en la aceleración de Aitken y el método de punto fijo. El método lo iniciamos a partir de un valor inicial x_0 que esté lo suficientemente cerca de la raíz a aproximar, si no es así lo que puede suceder es que encontremos la aproximación de la raíz muy lentamente, perdiendo su velocidad de convergencia cuadrática; o que el método diverja. Tiene la ventaja de que la función sólo debe ser continua en un entorno de la raíz, ya que no precisa de calcular ninguna derivada de la función. Por contra, tiene el inconveniente de que en cada iteración debemos calcular la imagen de la función 2 veces, lo cual puede aumentar su costo si la expresión algebraica de la función es complicada.

La sucesión de iteraciones del método es:

$$x_{n+1} = x_n - \frac{(f(x_n) - x_n)^2}{f(f(x_n)) - 2f(x_n) + x_n} \quad \forall n \in \mathbb{N}$$

Cuando decimos que debemos calcular dos veces el valor de la función nos referimos cuando calculamos $f(f(x_n)) = (f \circ f)(x_n) = f^2(x_n) \quad \forall n \in \mathbb{N}$

Para asegurar que el método tiene convergencia cuadrática podemos hacerlo comprobando que la raíz a aproximar tenga multiplicidad simple, es decir, que no se anule en su primera derivada de la función. Esto se puede traducir en el siguiente teorema:

Teorema 2.8.1 ► Convergencia Cuadrática Steffensen

Sea $x = g(x) / g(p) = 0$, con $g'(p) \neq 1$. Si $\exists \delta > 0 / g \in \mathcal{C}^3([p - \delta, p + \delta]) \Rightarrow$ El método de Steffensen es de orden $\mathcal{O}(n^2) \quad \forall p_0 \in [p - \delta, p + \delta]$

2.8.2. Combinando el método de Δ^2 –Aitken y Punto Fijo Para Steffensen

Para obtener Steffensen combinamos ambos métodos. El de Δ^2 –Aitken nos dice que si $\{\hat{x}_n\}_{n=1}^{\infty}$ es una sucesión que converge más rápidamente que una sucesión $\{x_n\}_{n=1}^{\infty}$, se tiene que:

$$\hat{x}_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} \quad \forall n \in \mathbb{N}$$

Definición 2.8.1 ► Diferencia Progresiva

Dadada la sucesión $\{x_n\}_{n=1}^{\infty}$ se define la Diferencia Progresiva Δx_n , mediante:

$$\Delta x_n = x_{n+1} - x_n \quad \forall n \geq 0$$

Las potencias de orden k se definen recursivamente como:

$$\Delta^k x_n = \Delta^{k-1}(\Delta x_n) \quad \forall n \geq 0 \quad \forall k \geq 2$$

Así pues, por la definición anterior tenemos lo siguiente:

$$\begin{aligned}
 \Delta^2 x_n &= \Delta(x_{n+1} - x_n) \\
 &= \Delta x_{n+1} - \Delta x_n \\
 &= (x_{n+2} - x_{n+1}) - (x_{n+1} - x_n) \\
 &= x_{n+2} - 2x_{n+1} + x_n
 \end{aligned}$$

Luego la sucesión del método Δ^2 -Aitken se puede reescribir como:

$$\hat{x}_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n} \quad \forall n \in \mathbb{N}$$

Sobre el método Δ^2 -Aitken aplicamos el método de punto fijo con función de punto fijo g del siguiente modo:

$$x_0, x_1 = g(x_0), x_2 = g(x_1), \hat{x}_0 = \{\Delta^2\}x_0, x_3 = g(x_2), \hat{x}_1 = \{\Delta^2\}x_1, \dots,$$

De esa forma es como actúa el algoritmo de Steffensen, cada 3 pasos en vez de aplicar el punto fijo aplica la aceleración de Aitken para acelerar la convergencia.

2.8.3. Algoritmo De Steffensen

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```

1 function []=steffensen(fx,x0,a,b,tolx,nmax)
2     filename = 'steffensen.txt';
3     fid = fopen(filename,'w');
4     fprintf(fid,'k\t\ttx (k)\t\t\t\t\t\t\tError\n');
5
6     f = inline(fx);
7     t = a:0.01:b;
8     func = f(t);
9     err=tolx+1;
10    x=x0;
11    phi=0;
12    iter = 0;
13    errores = [ ];
14    sol = [ ];
15    imagenes = [ ];
16
17    while iter < nmax && err > tolx

```

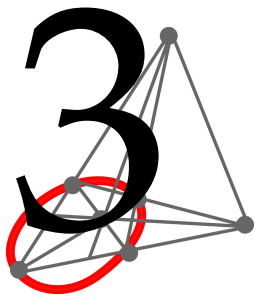
```

18         xx=x;
19         fxk=feval(f,x);
20         tolf=tolx*abs(phi);
21         if abs(fxk)<=tolf
22             break
23         end
24         fxk2=feval(f,x+fxk);
25         phi=(fxk2-fxk)/fxk;
26         x=xx-fxk/phi;
27         err=abs(x-xx);
28         errores =[errores,err];
29         sol = [sol,x];
30         iter++;
31     end
32
33     iter2 = 1:1:iter;
34     fc1 = feval(f,sol);
35     imagenes = [imagenes,fc1];
36
37     for k=1:iter
38         fprintf(fid,'%0f\t\t%.15f\t\t%.15f\n',iter2(k),sol(k),errores(k));
39     end
40     fprintf(fid,'\nLa ósolucin es %.15f.\nSe ha alcanzado en %d iteraciones.\nCon un
error de %e',sol(end),iter,errores(end));
41     fclose(fid);
42     printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',
sol(end),iter);
43
44
45     clf();
46     subplot(1,3,1);
47     fplot(f,[a,b])
48     set (gca, 'axislocation', 'zero');
49     set (gca, 'yaxislocation', 'zero');
50     set(gca, 'box', 'off');
51     grid;
52     title('La Funcion');
53     legend({fx},'location', 'southoutside');
54     legend boxon;
55     xlabel('y');
56     ylabel('x');
57     set (get (gca, 'ylabel'), 'rotation', 360);
58     subplot(1,3,2);
59     plot(t,func,x,imagenes,'*r')
60     set (gca, 'axislocation', 'zero');
61     set (gca, 'yaxislocation', 'zero');

```

```
62     set(gca, 'box', 'off');
63     grid;
64     title('Metodo Steffensen');
65     legend({fx,'Aproximaciones'},'location','southoutside');
66     legend boxon;
67     xlabel('y');
68     ylabel('x');
69     set (get (gca, 'ylabel'), 'rotation', 360);
70     subplot(1,3,3);
71     plot(iter2,errores,iter2,errores,'*r')
72     set (gca, 'axislocation', 'bottom');
73     set (gca, 'axislocation', 'right');
74     set(gca, 'box', 'off');
75     grid;
76     title('Tendencia de errores en Steffensen');
77     legend({'Errores', 'Puntos Errores'},'location','southoutside');
78     legend boxon;
79     xlabel('Numero Iteraciones');
80     ylabel('Errores');
81     set (get (gca, 'ylabel'), 'rotation', 360);
82 end
```

steffensen.m



EJERCICIOS RESUELTOS

MÉTODOS ITERATIVOS

Ejercicio 3.0.1 ► Polinomio Bisección

Determinar las raíces del polinomio $p(x) = 2x^3 - 3x - 1$ mediante el método de bisección mostrando el proceso paso a paso.

Resolución

Lo primero es determinar aproximadamente qué puntos son los que cortan al eje de abscisas. Una forma sencilla es dibujando la gráfica del polinomio. En Octave lo podemos hacer con los siguientes comandos:

```
1 x = linspace(-2,2);  
2 p = [2 0 -3 -1];  
3 y = polyval(p,x);  
4 plot(x,y)  
5 grid on
```

La gráfica de la función es la siguiente:

En dicha gráfica podemos apreciar que el polinomio posee 3 raíces simples, una es claramente -1, otra esta cerca de -0.5, y la tercera se encuentra cerca de 1.5. De hecho podemos determinar algebraicamente dichas soluciones. Aplicando Ruffini con raíz -1 obtenemos el polinomio de grado 2, $q(x) = 2x^2 - 2x - 1$, si lo resolvemos por la fórmula general de segundo grado obtenemos las raíces: $x = \frac{1 \pm \sqrt{3}}{2}$

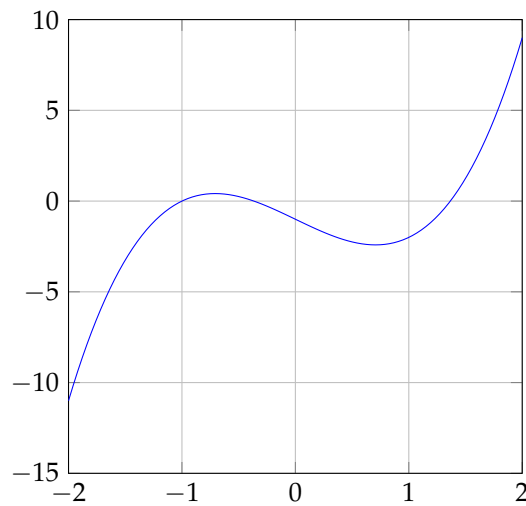


Figura 1: Gráfica polinomio $p(x) = 2x^3 - 3x - 1$

$$\begin{array}{r|rrrr} & 2 & 0 & -3 & -1 \\ -1 & & -2 & 2 & 1 \\ \hline & 2 & -2 & -1 & 0 \end{array}$$

Ambas raíces son números irracionales (infinitas cifras decimales), eso lo sabemos porque 3 es primo y la raíz cuadrada de un número primo es siempre irracional. Luego hemos obtenido de forma algebraica las tres raíces del polinomio, con una simple calculadora podemos obtener una aproximación numérica de las dos soluciones irracionales.

$$x_1 = \frac{1-\sqrt{3}}{2} = -0.3660254038 \quad x_2 = \frac{1+\sqrt{3}}{2} = 1.366025404$$

Vamos a aplicar el método de la bisección¹ de forma desglosada para la obtención de la raíz -1. Lo primero es establecer el intervalo de partida, en nuestro caso va a ser $[a, b] = [-1.5, -0.5]$. Calculamos $p(a) \cdot p(b)$ y comprobamos si es negativo (i.e.; si hay cambio de signo, Teorema de Bolzano) como primer paso. Después calculamos el punto medio de $[a, b]$.

$p(a) \cdot p(b) = p(-1.5) \cdot p(-0.5) = -3.25 \times 0.25 = -0.8125 < 0 \rightarrow$ Hay cambio de signo, podemos proseguir.

$$m_1 = \frac{a+b}{2} = \frac{-1.5-0.5}{2} = -1$$

Con esto establecemos 2 intervalos que son:

$$[a, m_1] = [-1.5, -1] \quad [m_1, b] = [-1, -0.5]$$

El siguiente paso es ver en cuál de dichos subintervalos hay un cambio de signo (la raíz está en el intervalo susodicho), nos quedaremos con el que tenga el cambio de signo:

¹El método se compone de la aplicación de 2 teoremas, el de Bolzano para comprobar si hay cambio de signo en cada subintervalo. Y el teorema de los Intervalos Encajados de Cantor, con el cual vamos dividiendo cada subintervalo en su mitad, creando así una sucesión de intervalos encajados monótona que converge.

$$p(a) \cdot p(m_1) = p(-1.5) \cdot p(-1) = -3.25 \times 0 = 0$$

Pero aquí nos detenemos ya porque hemos encontrado la raíz buscada, nos da como solución -1; exacta y con una iteración solamente. Volvamos a hacer lo mismo pero para la raíz $x_2 = \frac{1+\sqrt{3}}{2}$. Tomamos el intervalo de partida $[a, b] = [1, 1.5]$ y procedemos con el algoritmo de la bisección hasta que obtengamos 2 cifras decimales exactas respecto a nuestra solución exacta:

$$p(a) \cdot p(b) = p(1) \cdot p(1.5) = -2 \times 1.25 = -2.5 < 0 \rightarrow \text{Hay cambio de signo.}$$

$$m_1 = \frac{a+b}{2} = \frac{1+1.5}{2} = 1.25 \rightarrow$$

$$\rightarrow [a, m_1] = [1, 1.25] \quad [m_1, b] = [1.25, 1.5]$$

$$p(a) \cdot p(m_1) = p(1) \cdot p(1.25) = -2 \times (-0.84375) = 1.6875 > 0 \rightarrow \text{Aquí no se encuentra la raíz.}$$

$$p(m_1) \cdot p(b) = p(1.25) \cdot p(1.5) = -0.84375 \times 1.25 = -1.0546875 < 0 \rightarrow \text{Hay cambio de signo}$$

$$m_2 = \frac{m_1+b}{2} = \frac{1.25+1.5}{2} = 1.375 \rightarrow$$

$$\rightarrow [m_1, m_2] = [1.25, 1.375] \quad [m_2, b] = [1.375, 1.5]$$

$$p(m_1) \cdot p(m_2) = p(1.25) \cdot p(1.375) = -0.84375 \times 0.07421875 = -0.06262207031 < 0 \rightarrow \text{Hay cambio de signo.}$$

$$m_3 = \frac{m_1+m_2}{2} = \frac{1.25+1.375}{2} = 1.3125 \rightarrow$$

$$\rightarrow [m_1, m_3] = [1.25, 1.3125] \quad [m_3, m_2] = [1.3125, 1.375]$$

$$p(m_1) \cdot p(m_3) = p(1.25) \cdot p(1.3125) = -0.84375 \times (-0.4155273438) > 0 \rightarrow \text{Aquí no se encuentra la raíz.}$$

$$p(m_3) \cdot p(m_2) = p(1.3125) \cdot p(1.375) = -0.4155273438 \times 0.07421875 < 0 \rightarrow \text{Hay cambio de signo.}$$

$$m_4 = \frac{m_3+m_2}{2} = \frac{1.3125+1.375}{2} = 1.34375 \rightarrow$$

$$\rightarrow [m_3, m_4] = [1.3125, 1.34375] \quad [m_4, m_2] = [1.34375, 1.375]$$

$p(m_3) \cdot p(m_4) = p(1.3125) \cdot p(1.34375) = -0.4155273438 \times (-0.178527832) > 0 \rightarrow$ Aquí no se encuentra la raíz.

$p(m_4) \cdot p(m_2) = p(1.34375) \cdot p(1.375) = -0.178527832 \times 0.07421875 < 0 \rightarrow$ Hay cambio de signo.

$$m_5 = \frac{m_4 + m_2}{2} = \frac{1.34375 + 1.375}{2} = 1.359375 \rightarrow$$

$$\rightarrow [m_4, m_5] = [1.34375, 1.359375] \quad [m_5, m_2] = [1.359375, 1.375]$$

$p(m_4) \cdot p(m_5) = p(1.34375) \cdot p(1.359375) = -0.178527832 \times (-0.05414581302) > 0 \rightarrow$ Aquí no hay raíz.

$p(m_5) \cdot p(m_2) = p(1.359375) \cdot p(1.375) = -0.05414581302 \times 0.07421875 < 0 \rightarrow$ Hay cambio de signo.

$$m_6 = \frac{m_5 + m_2}{2} = \frac{1.359375 + 1.375}{2} = 1.3671875$$

Aquí nos detenemos porque si comparamos $m_6 = 1.3671875$ con $x_2 = \frac{1+\sqrt{3}}{2} = 1.366025404$ observamos que las 2 primeras cifras decimales coinciden, como queríamos. Lo que hemos de tener en cuenta es que aunque el método de la bisección siempre converge ², lo hace generalmente muy lento. Hemos necesitado 6 iteraciones para tan sólo conseguir 2 cifras decimales exactas respecto a la solución algebraica exacta. Luego nuestra solución con el método de la bisección después de 6 iteraciones es $x_2 \approx m_6 = 1.3671875$. Si queremos una mejor aproximación deberíamos seguir iterando el método de la bisección.

De la misma forma que se ha procedido con x_2 lo puedes hacer con $x_1 = \frac{1-\sqrt{3}}{2}$, te aconsejo que lo hagas porque es una buena forma de entender el método de la bisección.

Vamos a realizarlo seguidamente con nuestro algoritmo de la bisección realizado con Octave y veamos qué soluciones obtenemos para los diferentes casos. Veamos qué pasa para la raíz -1 con intervalo inicial $[-1.5, -0.5]$, con una tolerancia de 0.001 y un número máximo de iteraciones de 20. Vemos que hemos obtenido el mismo resultado que cuando lo hemos hecho a mano, cosa que era de esperar.

```
biseccion('2*power(x,3)-3*x-1',-1.5,-0.5,0.001,20)
```

```
La solucion es -1.0000000000000000.
```

```
Se ha alcanzado en 1 iteraciones.
```

²El método siempre converge porque el intervalo de partida siempre tiene un cambio de signo

Ahora lo vamos a realizar para $x_2 = \frac{1+\sqrt{3}}{2}$ en el intervalo $[1,1.5]$, una tolerancia de 0.001 y 20 iteraciones como máximo:

```
biseccion('2*power(x,3)-3*x-1',1,1.5,0.001,20)
```

La solución es 1.366210937500000.

Se ha alcanzado en 9 iteraciones.

Podemos apreciar que hemos necesitado 9 iteraciones para obtener 3 cifras decimales exactas. Si cambiamos la tolerancia a 0.000001 apreciamos que con 19 iteraciones obtenemos que las 5 primeras cifras decimales son exactas.

```
biseccion('2*power(x,3)-3*x-1',1,1.5,0.000001,20)
```

La solución es 1.366024971008301.

Se ha alcanzado en 19 iteraciones.

Además lo podemos apreciar de forma gráfica con la siguiente figura:

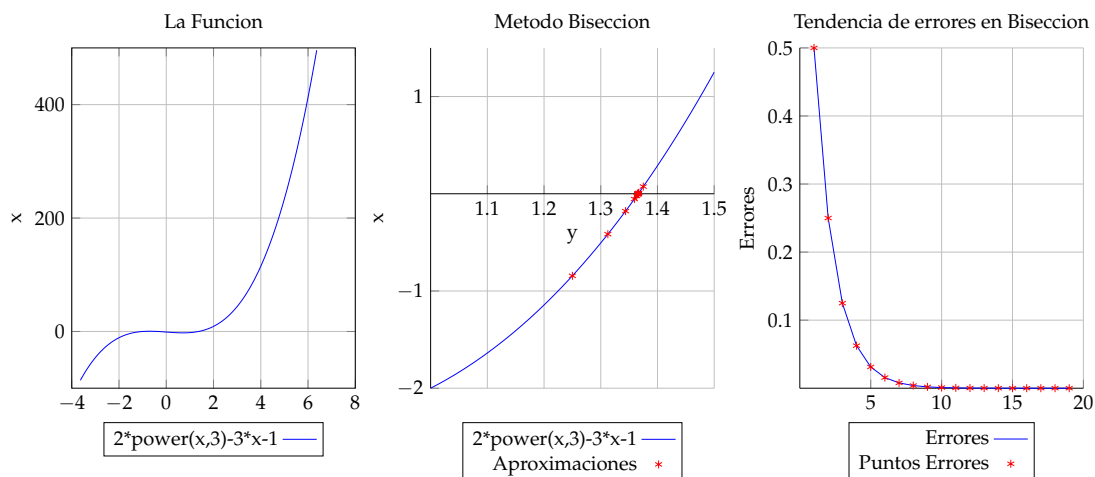


Figura 2: Método Bisección $p(x) = 2x^3 - 3x - 1$ en $[1,1.5]$

Los asteriscos rojos nos indican las diferentes aproximaciones que vamos obteniendo con el método en cada paso, en nuestro caso las aproximaciones son los puntos medios de cada subintervalo en el cual hay un cambio de signo. En la gráfica de la izquierda podemos apreciar la rapidez en la que converge nuestro método en cada caso particular.

Para terminar vamos a ver qué ocurre con $x_1 = \frac{1-\sqrt{3}}{2}$, le vamos a aplicar una tolerancia de 0.000001 con un máximo de 20 iteraciones, y todo ello sobre el intervalo $[-0.75, -0.25]$

```
biseccion('2*power(x,3)-3*x-1',-0.75,-0.25,0.000001,20)
```

La solución es -0.366024971008301.

Se ha alcanzado en 19 iteraciones.

Podemos apreciar que para alcanzar las 5 cifras decimales exactas hemos necesitado como antes 19 iteraciones. Las gráficas de resolución son muy similares también a las anteriores.

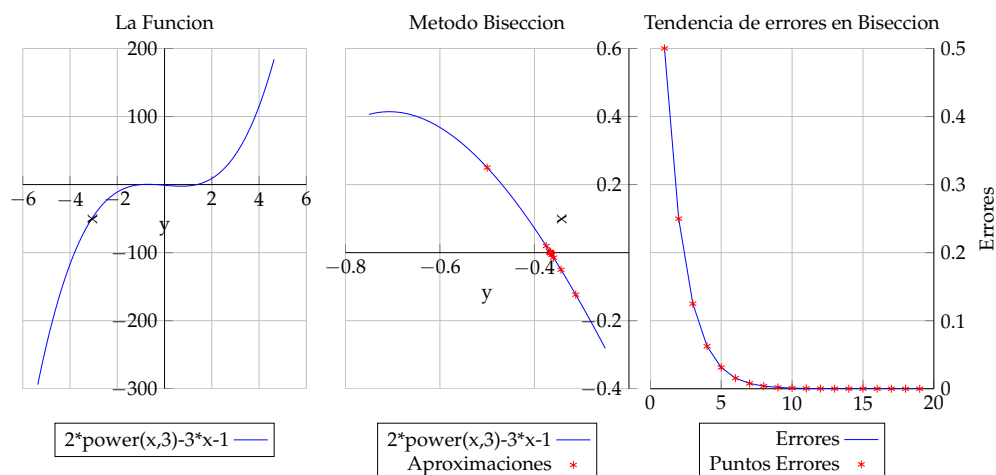


Figura 3: Método Bisección $p(x) = 2x^3 - 3x - 1$ en $[-0.75, -0.25]$

Ejercicio 3.0.2 ► Coseno Regula-Falsi

Halla la menor raíz positiva de la ecuación $e^{-x^2} - \cos x = 0$ con una tolerancia de 10^{-5} , utilizando el método de regula falsi y mostrando los pasos en las primeras iteraciones.

Resolución

En ésta ecuación hay una solución obvia que es $x = 0$. Tomamos como función $f(x) = e^{-x^2} - \cos x$ y dibujamos su gráfica para hacernos una idea de dónde se encuentran las raíces/soluciones.

```
1 x=-3:0.1:3;
2 y=exp(-x.^2)-cos(x);
3 plot(x,y), grid on
```

Viendo la figura apreciamos que tenemos 2 raíces más cerca de -1 y 1. Vamos a calcular sólo la que está cerca de 1 porque la función es simétrica respecto al eje OY, cumpliéndose pues que la otra raíz es la misma que la que está cerca de 1 pero cambiada de signo (su raíz opuesta).

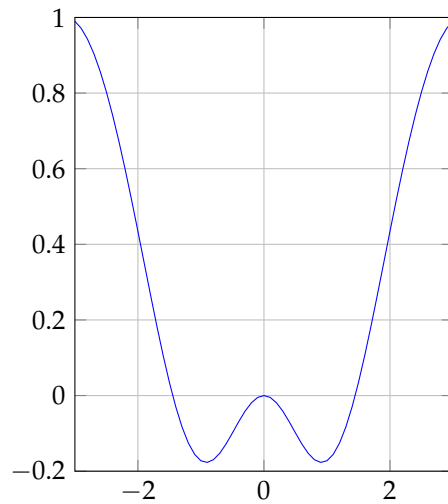


Figura 4: Función $f(x) = e^{-x^2}$

$$f(-x) = e^{-(-x)^2} - \cos(-x) = e^{-x^2} - \cos x = f(x); \forall x \in \mathbb{R}$$

Para aplicar el método de la Regula Falsi vamos a considerar el intervalo de partida $[a_1, b_1] = [1, 2]$

$$f(a_1) = f(1) = -0.172423$$

$$f(b_1) = f(2) = 0.434462$$

Siguiendo el método debemos calcular el punto c_1 donde la recta que une los puntos $(a_1, f(a_1))$ y $(b_1, f(b_1))$ corta al eje OX ($y = 0$):

$$c_1 = \frac{a_1 \cdot f(b_1) - b_1 \cdot f(a_1)}{f(b_1) - f(a_1)} = \frac{1 \cdot f(2) - 2 \cdot f(1)}{f(2) - f(1)} = \frac{0.434462 + 2 \cdot 0.172423}{0.434462 + 0.172423} = 1.284111$$

$$f(c_1) = f(1.284111) = -0.090521$$

Ahora hay que ver en qué subintervalo se encuentra nuestra raíz aplicando Bolzano, los subintervalos son $[a_1, c_1]$ y $[c_1, b_1]$. Tenemos que

$$f(c_1) \cdot f(b_1) = -0.090521 \times 0.434462 = -0.00393279347 < 0$$

Luego la raíz se encuentra en éste subintervalo. Si seguimos nuestro algoritmo de la Regula Falsi nombramos $a_2 = c_1, b_2 = b_1$, teniendo pues el subintervalo $[a_2, b_2] = [1.284111, 2]$. Calculamos c_2 :

$$c_2 = \frac{a_2 \cdot f(b_2) - b_2 \cdot f(a_2)}{f(b_2) - f(a_2)} = \frac{1.284111 \cdot f(2) - 2 \cdot f(1.284111)}{f(2) - f(1.284111)} = \frac{0.5578974333 + 0.181042}{0.434462 + 0.090521} = 1.40754926$$

$$f(c_2) = f(1.40754926) = -0.02461859979$$

Realizamos un paso más:

$$[a_2, c_2] = [1.284111, 1.40754926] \quad [c_2, b_2] = [1.40754926, 2]$$

$$f(b_2) \cdot f(c_2) = f(2) \cdot f(1.40754926) = 0.434462 \times (-0.02461859979) = -0.0106958461 < 0$$

Luego la raíz se encuentra en $[c_2, b_2] := [a_3, b_3] = [1.40754926, 2]$.

Calculamos c_3

$$c_3 = \frac{a_3 \cdot f(b_3) - b_3 \cdot f(a_3)}{f(b_3) - f(a_3)} = \frac{1.40754926 \cdot f(2) - 2 \cdot f(1.40754926)}{f(2) - f(1.40754926)} = \frac{0.6115266666 + 0.04923719958}{0.434462 + 0.02461859979} = 1.439319951$$

$$f(c_3) = -0.005119$$

Así pues, con 3 iteraciones del Método de la Regula Falsi obtenemos como raíz aproximada 1.439319951 con 2 cifras decimales exactas.

Vamos a resolverlo con nuestro algoritmo, tomamos como intervalo de partida el $[1, 2]$, con una tolerancia de 0.0000001 y un número máximo de iteraciones de 20:

```
regulafalsi('exp(-x.^2)-cos(x)',1,2,0.000001,20)
```

La solución es 1.447414195298149, y se ha alcanzado en 10 iteraciones.

Con un error de 3.220810e-07

Como vemos hemos obtenido 6 cifras decimales exactas con 10 iteraciones, lo cual está muy bien. En la siguiente tabla se puede apreciar iteración tras iteración el cálculo de la raíz con su respectivo error cometido.

Iteración k	Raíz x_k	Error e_k
1	1.000000000000000	0.284111105170264
2	1.284111105170264	0.123438085370137
3	1.407549190540402	0.031770712121548
4	1.439319902661950	0.006529446475057
5	1.445849349137007	0.001265429310186
6	1.447114778447193	0.000242289353378
7	1.447357067800570	0.000046281690852
8	1.447403349491422	0.000008836666484
9	1.447412186157907	0.000001687059263
10	1.447413873217170	0.000000322080979

Esta tabla se genera cada vez que ejecutamos el método y se almacena en el fichero *regulafalsi.txt*.

Como sabemos que la función es simétrica respecto al eje OY podemos concluir que la otra raíz que nos falta por calcular es la opuesta a la que hemos calculado, es decir; la raíz es aproximadamente -1.447413873217170 , sin tener que aplicar ningún método.

Ejercicio 3.0.3 ► Coseno Punto Fijo

- Halla las raíces de la ecuación $x = 2 \cdot \cos x$ por el método del punto fijo, ilustrando el método paso a paso.
- Estudia el valor de la derivada de la función elegida $g(x)$ cerca de la raíz y justifica el comportamiento del método.

Resolución

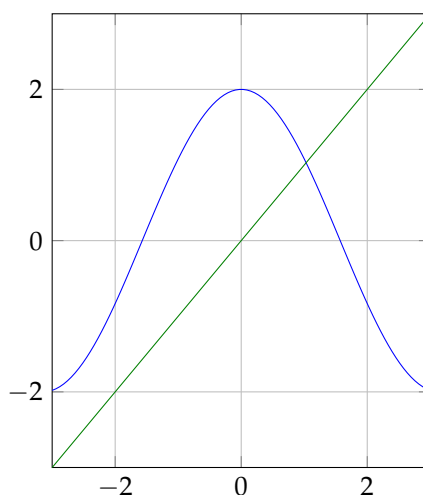
a) La función a la que le queremos hallar sus raíces es $f(x) = x - 2 \cdot \cos x$. Tomamos como función de punto fijo la obvia, $g_1(x) = 2 \cdot \cos x$. Vamos a ver que esta función de punto fijo es inestable, por lo que deberemos buscar otra como alternativa. Representamos la función $g_1(x) = 2 \cdot \cos x$ junto a la bisectriz del primer cuadrante para obtener la estimación inicial. Para ello podemos introducir en Octave las siguientes órdenes:

```

1 x = linspace(-3,3);
2 y = 2*cos(x);
3 plot(x,y,x,x), grid on

```

En el dibujo de ambas gráficas podemos ver que la bisectriz corta a la función en el punto con abscisa 1, por lo que vamos a tomar como estimación inicial $x_1 = 1$, y aplicamos el algoritmo de

Figura 5: Función $f(x) = 2 \cos(x)$ recta $y = x$

punto fijo:

$$x_2 = g_1(x_1) = g_1(1) = 2 \cdot \cos(1) = 1.080604612$$

$$x_3 = g_1(x_2) = g_1(1.080604612) = 2 \cdot \cos(1.080604612) = 0.941590072$$

$$x_4 = g_1(x_3) = g_1(0.941590072) = 2 \cdot \cos(0.941590072) = 1.177006409$$

$$x_5 = g_1(x_4) = g_1(1.177006409) = 2 \cdot \cos(1.177006409) = 0.7673820112$$

$$x_6 = g_1(x_5) = g_1(0.7673820112) = 2 \cdot \cos(0.7673820112) = 1.439461363$$

Y no continuamos porque se aprecia claramente que los resultados obtenidos no se estabilizan, por el contrario, son oscilantes. Teóricamente hay una forma de comprobarlo mediante la condición de convergencia, se trata de ver si $|g'(x)| < 1$. Veámoslo en nuestro caso:

$$g'_1(x) = -2 \cdot \sin x \rightarrow |g'_1(x)| = |-2 \cdot \sin x| < 2 \cdot |\sin x| < 2 > 1; \forall x \in \mathbb{R}$$

Por tanto, es claro que no se cumple la condición de convergencia para la función de punto fijo $g_1(x) = 2 \cdot \cos x$ que hemos considerado³. Así pues, debemos buscar otra función de punto fijo,

³Dicha función no cumple el criterio de convergencia ni en un entorno alrededor de un punto, ya que el módulo de su derivada es mayor que 1 en todo el conjunto de números reales, por lo que nos vemos obligados a buscar otra función de punto fijo. Si no pudiésemos encontrar otra deberíamos aplicar otro método iterativo como alternativa a buscar la solución.

para ello basta hacer unos cuantos pasos algebraicos a partir de nuestra ecuación de partida:

$$x = 2 \cdot \cos x \rightarrow \frac{x}{2} = \cos x \rightarrow \arccos\left(\frac{x}{2}\right) = x \rightarrow g_2(x) = \arccos\left(\frac{x}{2}\right)$$

Veamos si dicha función cumple el criterio de convergencia para un entorno de 1, $|x| < 1$ ⁴:

$$g_2'(x) = \frac{-1}{2\sqrt{1-(\frac{x}{2})^2}} \rightarrow |g_2'(x)| = \left| \frac{-1}{2\sqrt{1-(\frac{x}{2})^2}} \right| \leq \frac{1}{2} \left| \frac{1}{\sqrt{\frac{4-x^2}{4}}} \right| \leq \frac{1}{4} \left| \frac{1}{\sqrt{4-x^2}} \right| < 1; |x| < 1$$

Luego el criterio de convergencia se cumple en un entorno de 1, lo cual es suficiente para poder tomar como valor inicial $x_1 = 1$, ya que pertenece a dicho entorno. Aplicamos pues el método de punto fijo:

$$x_2 = g_2(x_1) = g_2(1) = \arccos\left(\frac{1}{2}\right) = 1.047197551$$

$$x_3 = g_2(x_2) = g_2(1.047197551) = \arccos\left(\frac{1.047197551}{2}\right) = 1.019726744$$

$$x_4 = g_2(x_3) = g_2(1.019726744) = \arccos\left(\frac{1.019726744}{2}\right) = 1.035770367$$

$$x_5 = g_2(x_4) = g_2(1.035770367) = \arccos\left(\frac{1.035770367}{2}\right) = 1.026419399$$

$$x_6 = g_2(x_5) = g_2(1.026419399) = \arccos\left(\frac{1.026419399}{2}\right) = 1.031875939$$

Como vemos, tras 6 iteraciones el método se mantiene estable para esta función de punto fijo. Apliquemos nuestro algoritmo de punto fijo para una tolerancia de 0.0001 y 20 iteraciones como tope:

```
pfijo('acos(x/2)','x-2*cos(x)',0,1.5,1,0.0001,20)
```

La solución es 1.029893453594281, y se ha alcanzado en 14 iteraciones.

Con un error de 7.308525e-05

Vamos a dibujar la función $f(x) = x - 2 \cdot \cos x$ en el intervalo $[-5,5]$ con las órdenes:

```
1 clf()
2 fplot('x-2*cos(x)', [-5,5])
3 grid on
```

⁴Notar que fuera de dicho entorno esta segunda función hará que nuestro método sea inestable.

Podemos apreciar que dicha función sólo corta al eje OX en la solución que hemos obtenido, por lo que el problema queda terminado. Para asegurarnos deberíamos hacer un razonamiento más teórico sobre su crecimiento y decrecimiento de la función, pero eso te lo dejo a ti lector como deberes por ser un sencillo ejercicio de segundo bachiller.

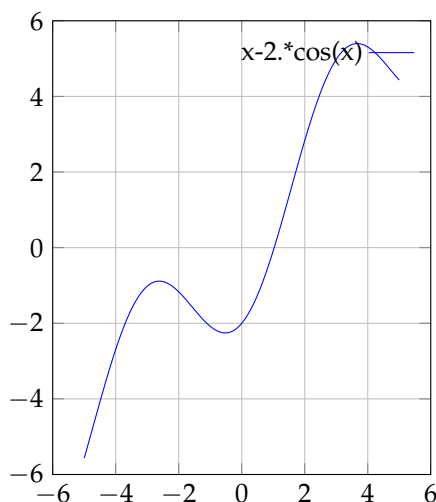


Figura 6: Función $f(x) = x - 2 \cos(x)$

Ejercicio 3.0.4 ► Raíces Quintas

- Utiliza el método de Newton para aproximar la raíz quinta de 2 con tres cifras decimales exactas, ilustrando el método paso a paso.
- Comprueba la convergencia cuadrática del método de Newton.

Resolución

a) Muy bien, lo primero es establecer la función a la que le vamos a aplicar el método. Nos dicen de aproximar la raíz quinta de 2, lo cual es lo mismo que decir $x = \sqrt[5]{2}$, vamos a manipularlo algebraicamente hasta obtener el polinomio de quinto grado con solución la raíz quinta de 2:

$$x = \sqrt[5]{2} \rightarrow x^5 = 2 \rightarrow x^5 - 2 = 0$$

Ya tenemos el polinomio o función a la que aplicar el método de Newton, $f(x) = x^5 - 2$. Seguidamente debemos aplicar el método, que de forma general es: $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$, partiendo de

un valor inicial de partida x_0 . Pero primero calculemos la derivada, $f'(x) = 5x^4$.

En nuestro caso ya sabemos que la solución exacta es $\sqrt[5]{2}$, pero lo que queremos es aproximarla con 3 cifras decimales exactas (esta solución es irracional), así que va a ser fácil determinar el valor inicial, vamos a tomar $x_0 = 1$, porque si calculamos con una calculadora tenemos que $\sqrt[5]{2} = 1.148698355$ vemos que 1 está cerca de la raíz.⁵ Perfecto, iteremos:

$$x_1 = 1 - \frac{f(1)}{f'(1)} = 1 + \frac{1}{5} = 1.2$$

$$x_2 = 1.2 - \frac{f(1.2)}{f'(1.2)} = 1.2 - \frac{0.48832}{10.368} = 1.152901235$$

$$x_3 = 1.152901235 - \frac{f(1.152901235)}{f'(1.152901235)} = 1.152901235 - \frac{0.036856916}{8.83361408} = 1.148728887$$

$$x_4 = 1.148728887 - \frac{f(1.148728887)}{f'(1.148728887)} = 1.148728887 - \frac{0.000265811}{8.706431225} = 1.148698357$$

Lo que debemos apreciar es que de todos los métodos que llevamos aplicados hasta ahora en este documento, y salvando la diferencia importante de que todas las funciones han sido diferentes; es que éste método ha sido el más rápido con diferencia. Si lo realizamos con nuestro algoritmo de Octave con 10^{-7} obtenemos lo siguiente:

```
newton('x.^5-2','5*x.^4',1,-1,2,1e-7,20)
```

La solución es 1.148698354997035, y se ha alcanzado en 6 iteraciones.

b) Y ahora veamos que la convergencia es cuadrática. En nuestro algoritmo de Newton ya lo hemos añadido todo para ver bajo gráficas qué tipo de convergencia alcanzamos. Viendo en la figura adjunta apreciamos que no se cumple la lineal porque la gráfica tiende a 0 en vez de tender a 1. En la gráfica de la cuadrática apreciamos que tiende cerca de 2, por lo que concluimos que la convergencia es 2. Además, el algoritmo también nos crea el fichero *newtonraphson.txt* en el cual podemos ver los valores de ambas tasas de convergencia. En nuestro caso:

Tasa de convergencia lineal = 0.235494

Tasa de convergencia lineal = 0.088587

Tasa de convergencia lineal = 0.007317

Tasa de convergencia lineal = 0.000053

⁵No podemos tomar como valor inicial 0 o en un entorno pequeño de 0 porque $f'(0) = 0$, y ya sabemos que el método de Newton falla en valores que hacen que su primera derivada se anule.

Tasa de convergencia cuadrática = 1.177469

Tasa de convergencia cuadrática = 1.880882

Tasa de convergencia cuadrática = 1.753737

Tasa de convergencia cuadrática = 1.741193

De ésta segunda forma podemos apreciar muy bien que la Tasa de Convergencia Lineal tiende claramente a 0 y la cuadrática aproximadamente a 1.7, que es cerca de 2.

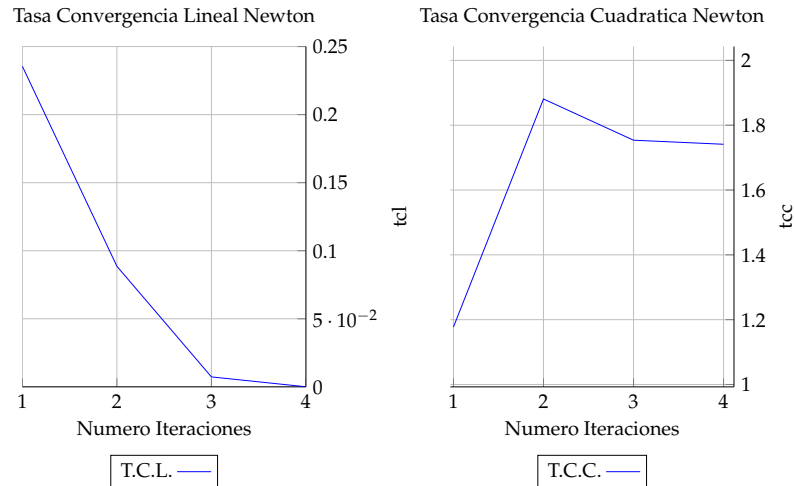


Figura 7: Tasas De Convergencia

Ejercicio 3.0.5 ► Función Billar

Para jugar al billar en una mesa circular, hemos de golpear la bola Q con la bola P, tras un impacto I en la banda. Conocido el radio R de la mesa y las posiciones en coordenadas cartesianas (cuyo origen es el centro de la mesa) de los puntos $P(x_p, y_p)$ y $Q(x_q, y_q)$, el punto de impacto viene definido por el ángulo central t.

Tras un análisis geométrico del problema, se prueba que los valores de t que proporcionan los puntos de impacto posibles son los ceros de la función:

$$f(t) = \frac{x_p \sin t - y_p \cos t}{\sqrt{(R \cos t - x_p)^2 + (R \sin t - y_p)^2}} + \frac{x_q \sin t - y_q \cos t}{\sqrt{(R \cos t - x_q)^2 + (R \sin t - y_q)^2}}$$

Consideramos un billar de radio unidad, $P = (0.6, -0.3)$ y $Q = (-0.6, -0.3)$. Determina los puntos de impacto utilizando los métodos de bisección, regula falsi, Newton y secante. Analiza la convergencia de los distintos métodos y compara los resultados obtenidos.

ESQUEMA

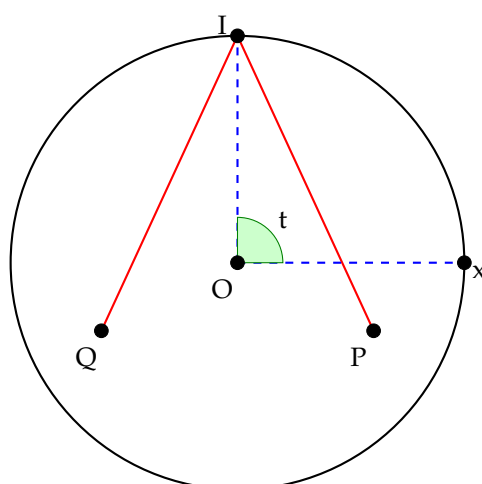


Figura 8: Billar Circular

Resolución

a) Método de bisección:

En los códigos verás un fichero llamado fbillar.m en el cual está introducida la función impacto $I(t)$ para poder llamarla cada vez que ejecutamos uno de nuestros algoritmos iterativos. Hagámonos una idea de la función impacto visualizando su gráfica, que puedes apreciar en la figura.

```

1  % Para que los metodos iterativos funcionen es biseccion('fbillar(t)',1,2,1e-4,100)
2
3  function [y] = fbillar(t)
4      P=[0.6,-0.3]; Q = [-0.6,-0.3];
5      y1 = P(1)*sin(t)-P(2)*cos(t);
6      y2 = Q(1)*sin(t)-Q(2)*cos(t);
7      den1 = sqrt((cos(t)-P(1)).^2+(sin(t)-P(2)).^2);
8      den2 = sqrt((cos(t)-Q(1)).^2+(sin(t)-Q(2)).^2);
9      y1 = y1./den1;
10     y2 = y2./den2;
11     y = y1 + y2;
12 end

```

fbillar.m

Al ser una función trigonométrica sabemos que es una función periódica ⁶ de periodo 2π , por

⁶Los trozos de función se repiten cada cierto periodo T.

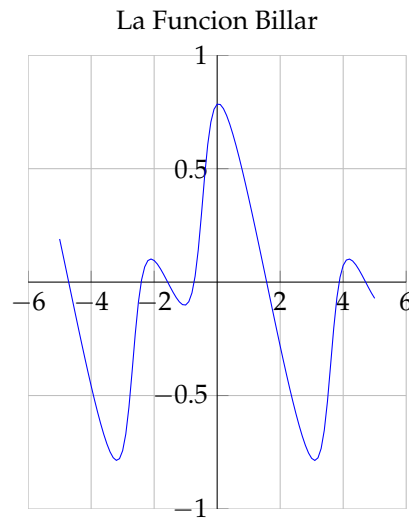


Figura 9: Función Billar

lo que nos vamos a fijar en el intervalo $[0, 2\pi]$. Sabiendo que las funciones seno y coseno son de periodo 2π con lo que no es difícil de comprobar que nuestra función impacto es de periodo 2π , $I(t) = I(t + 2\pi); \forall t \in \mathbb{R}$.

Como el método de bisección siempre converge queda ver cuáles son los intervalos a utilizar, en $[0, 2\pi]$ vemos que hay 4 raíces reales en las que podemos situarlas en los 4 respectivos intervalos: $[1, 2]$, $[3, 4]$, $[4, 5]$ $[5, 6]$. Ejecutamos el algoritmo de bisección para el intervalo $[1, 2]$ con el comando:

```
biseccion('fbillar(t)', 1, 2, 1e-4, 100)
```

Y obtenemos la solución:

```
La solucion es 1.570739746093750.
Se ha alcanzado en 14 iteraciones.
Con un error de 1.220703e-04
```

Antes que nada te aconsejo que mires con detenimiento la gráfica de la función en este intervalo y la solución que hemos obtenido, igual así no lo recuerdas pero la solución viene en radianes y es una aproximación de $\frac{\pi}{2}$ que son 90° . Luego aquí hemos hecho un poco el primo ya que no es necesario aplicar ningún método iterativo para saber que la solución exacta es un ángulo de 90° , no te desesperes si no has caído, es normal y como mal menor piensa que el juego del billar nos da una forma de aproximar $\frac{\pi}{2}$, aunque sea un poco burda.

Por tanto tenemos que las raíces distintas en $[0, 2\pi]$ son:

Intervalo	Solución En Radianes	Solución En Grados	Iteraciones
[1,2]	1.570739746093750	89.996758164622406	14
[3,4]	3.871276855468750	221.807825145036162	14
[4,5]	4.712341308593750	269.997268607577325	14
[5,6]	5.553405761718750	318.186712070118517	14

En la tabla observamos que al tener los 4 intervalos la misma longitud nos da que el método de la bisección con una tolerancia de $1e-4$ siempre nos hace el mismo número de iteraciones, 14 en éste caso.

Quizá te estés preguntando por qué no calculamos las raíces negativas, pues no las calculamos porque son las opuestas que las positivas por ser nuestra función periódica, por lo que ya sabemos cuáles son. Además, al ser negativas lo que debemos saber es que los ángulos son negativos, por lo que las trayectorias de las bolas se hacen en sentido positivo; mientras que las raíces positivas nos indican que las trayectorias son en sentido negativo.⁷

b) Método Regula Falsi:

Lo aplicamos con una tolerancia de $1e-12$ y un máximo de 50 iteraciones, obteniendo los siguientes resultados:

Intervalo	Solución En Radianes	Solución En Grados	Errores	Iteraciones
[1,2]	1.570796326794897	90.000000000000000	2.220446e-16	5
[3,4]	3.871320309816781	221.810314895779868	2.304823e-13	13
[4,5]	4.712388980384690	270.000000000000000	8.881784e-16	6
[5,6]	5.553457650952138	318.189685104193813	6.670220e-13	35

Se aprecia que aun habiendo aumentado considerablemente la tolerancia el método es mucho más rápido que el de la bisección, incluso obtenemos en algunos una tolerancia mayor que la de $1e-12$. Además, observamos que la primera solución ya nos la marca como 90 grados, al igual que la tercera que nos dice que son 270°⁸

c) Método de Newton:

Lo primero va a ser crear un fichero con código para implementar la primera derivada de la función del problema, al fichero le he llamado *fbillardderivada.m* y el código es el siguiente:

```

1 function [dy] = fbillardderivada(t)
2     P=[0.6,-0.3]; Q = [-0.6,-0.3];
3     x = P(1);
4     y = P(2);

```

⁷El sentido positivo es el sentido contrario a las agujas del reloj, y el negativo es en el sentido de las agujas del reloj.

⁸Esto depende del ordenador, de la cantidad de cifras decimales que indiquemos en el programa que aparezcan por pantalla y del tipo de redondeo de Octave, entre otros muchos factores.

```

5  x1 = Q(1);
6  y1 = Q(2);
7  den1 = sqrt(1-2*(x*cos(t)+y*sin(t))+x^2+y^2);
8  den2 = sqrt((1-2*(x*cos(t)+y*sin(t))+x^2+y^2).^3);
9
10 num1 = x*cos(t)+y*sin(t);
11 num2 = (x*sin(t)-y*cos(t)).^2;
12
13 dy1 = num1/den1;
14 dy2 = num2/den2;
15
16 den11 = sqrt(1-2*(x1*cos(t)+y1.*sin(t))+x^2+y^2);
17 den21 = sqrt((1-2*(x1*cos(t)+y1*sin(t))+x^2+y^2).^3);
18
19 num11 = x1*cos(t)+y1*sin(t);
20 num21 = (x1*sin(t)-y1*cos(t)).^2;
21
22 dy3 = num11/den11;
23 dy4 = num21/den21;
24
25 dy = dy1 - dy2 + dy3 - dy4;
26 end

```

fbillarderivada.m

Para calcular la derivada de la función lo puedes hacer a mano con mucha paciencia o usar el programa de matemáticas simbólicas *wxMaxima* de código libre. Una vez que lo tenemos todo, y para no hacer el problema muy largo vamos a calcular con Newton sólo la primera raíz y comparar con los métodos anteriores, así que en Octave introducimos y obtenemos:

```
newton('fbillar(t)', 'fbillarderivada(t)', 1, 0, 1.5, 1e-6, 20)
```

La solución es 1.570796326794897, y se ha alcanzado en 5 iteraciones.

Si comparamos con el método de la bisección es claro que obtenemos una mayor precisión en Newton y en un número sensiblemente menor de iteraciones, tan sólo 5 iteraciones. También podemos comparar si el método de Newton tiene convergencia lineal o cuadrática mediante las siguientes gráficas.

Pero éste problema tiene un handicap muy a tener en cuenta, y es que calcular la derivada de nuestra función y sustituir los valores en cada iteración tiene un coste muy elevado, aunque hoy en día con los ordenadores que tenemos casi no se nota, a no ser que tengamos grandes cálculos. En nuestro caso le he aplicado las funciones *tic* y *toc* a nuestro algoritmo de newton con ésta función y no tarda más allá de 0.11 segundos. Aunque sí es cierto que lo mejor es no aplicar

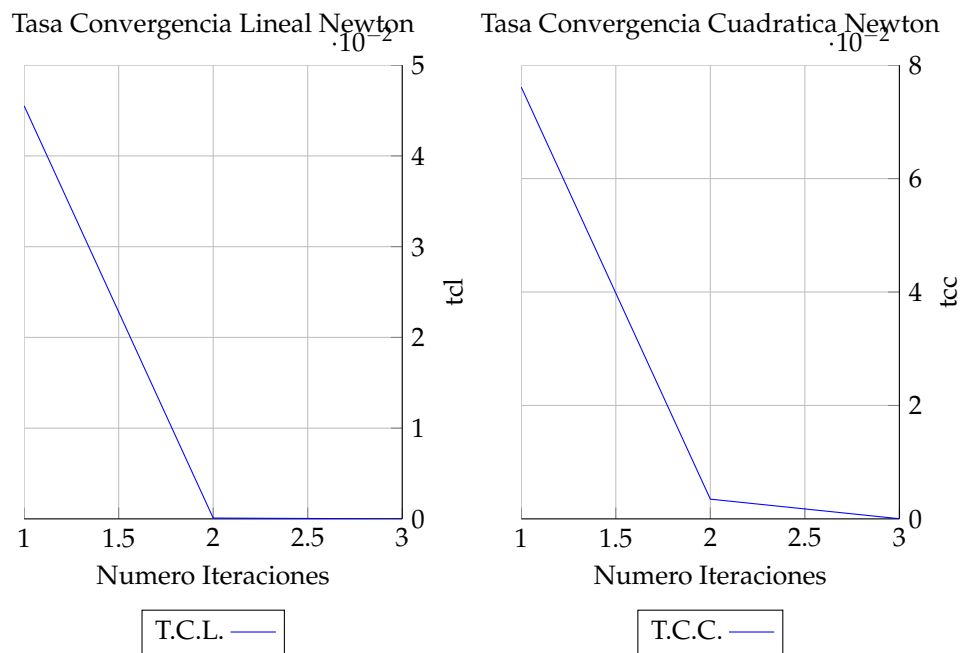


Figura 10: Tasas Convergencia Billar Newton

Newton y tirar por otro lado, porque el tiempo que invertimos es bastante calculando la primera derivada de nuestra función a mano o con ordenador; ése es el verdadero motivo por el que en éste prolema el método de Newton no es recomendable.

d) Método de la Secante:

En éste método no tenemos que calcular ninguna derivada, por lo que si obtenemos unos resultados similares a los obtenidos en Newton sería nuestra verdadera alternativa para calcular las raíces:

```
secante('fbillar(t)',1,2,1e-6,10)
```

La solucion es 1.570796326794897, y se ha alcanzado en 4 iteraciones.

Efectivamente, incluso con una iteración menos que con Newton obtenemos idéntico resultado y sin el costo adicional de tener que calcular la primera derivada. Por tanto, el método de la secante es la mejor alternativa de las 4 vistas aquí para nuestro problema.⁹

⁹Aunque no lo hemos visto aquí, es obvio que el método del punto fijo en éste problema es prácticamente inviable, no sólo por el elevado costo que supone encontrar la función de punto fijo, sino también comprobar si el módulo de su primera derivada cumple o no el criterio de convergencia. Otra alternativa muy interesante es aplicar el algoritmo de Steffensen.

Ejercicio 3.0.6 ► Problema Catenaria

Se denomina catenaria a la forma que adopta una cadena o un cable suspendido de sus 2 extremos. Eligiendo adecuadamente el sistema de referencia, la ecuación de la catenaria es: $y = \lambda \cosh\left(\frac{x}{\lambda}\right)$, donde el origen de coordenadas está situado en la vertical del punto más bajo de la cadena, a distancia λ del mismo. Sea d la diferencia de altura entre el punto de la cadena de abscisa igual a 1 y el de altura mínima: $d = y(1) - y(0)$. Conociendo d , podemos determinar λ resolviendo la ecuación $\lambda + d = \lambda \cosh\left(\frac{1}{\lambda}\right)$. Supongamos $d = 0.1$.

- a) Determina con precisión de milésimas el valor de λ utilizando la ecuación de punto fijo

$$\lambda = \lambda \cosh\left(\frac{1}{\lambda}\right) - d$$

- b) Comprueba que la función de punto fijo

$$\lambda = \frac{1}{\arg \cosh\left(\frac{\lambda+d}{\lambda}\right)}$$

es también equivalente a la ecuación

$$\lambda + d = \lambda \cosh\left(\frac{1}{\lambda}\right)$$

. Itera con esta ecuación y la misma estimación inicial para hallar λ .

- c) ¿Qué relación existe entre la derivada respecto a de las funciones de iteración consideradas en los apartados anteriores y el proceso de convergencia?

Resolución

- a) Tenemos que nuestra función de punto fijo es $g(\lambda) = \lambda \cosh\left(\frac{1}{\lambda}\right) - d$. Vamos a dibujar la función junto con la bisectriz ppal. del plano para ver dónde cortan y qué valor inicial podemos tomar. ¹⁰

```
1 x=1:0.1:6;
2 plot(x,x.*cosh(1./x)-0.1,x,x), grid on;
3 legend({'g(x)', 'Bisectriz'}, 'location', 'southoutside');
```

¹⁰Ten cuidado, no dibujes la función incluyendo valores de λ en $]0,1[$ porque sus imágenes se disparan y no verás nada. El que sus imágenes se disparen es debido a que la función \cosh es una combinación de la función exponencial, $\cosh x = \frac{1}{2}(e^x + e^{-x})$

En la figura se aprecia que ambas funciones se cortan alrededor de 5, por lo que tomaremos como valor inicial 4 en nuestro método de punto fijo.

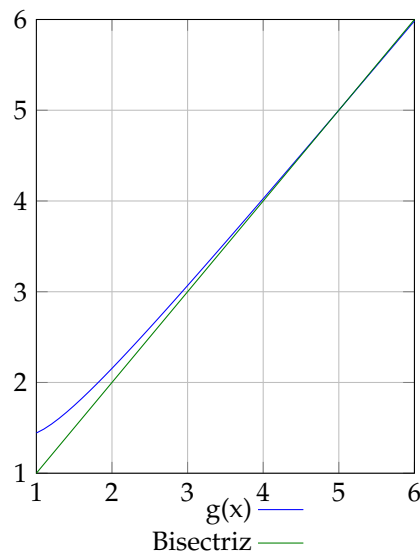


Figura 11: Catenaria $x \cosh\left(\frac{1}{x}\right) - 0.1$

Lo vamos a aplicar con una tolerancia de $1e-4$:

```
pfijo('x.*cosh(1./x)-0.1','x-x.*cosh(1./x)-0.1',1,5,4,1e-4,300)
```

La solución es 5.011784191306107, y se ha alcanzado en 255 iteraciones.

Y vemos que sí obtenemos la solución pero el número de iteraciones necesarias es excesivo, así que debemos encontrar otra función de punto fijo, la que vamos a usar en b).

b) Vamos a demostrar cómo llegamos a esa función de punto fijo.

$$\lambda + d = \lambda \cosh\left(\frac{1}{\lambda}\right) \rightarrow \frac{\lambda + d}{\lambda} = \cosh\left(\frac{1}{\lambda}\right) \rightarrow \frac{1}{\lambda} = \arg \cosh\left(\frac{\lambda + d}{\lambda}\right) \rightarrow \lambda = \frac{1}{\arg \cosh\left(\frac{\lambda + d}{\lambda}\right)}$$

Y ahora que ya lo hemos demostrado lo que vamos a hacer es dibujarla junto con la bisectriz:

```
1 x=4:0.1:6;
2 plot(x,1./acosh((x+0.1)./x),x,x), grid on;
3 legend({'g(x)', 'Bisectriz'}, 'location', 'southoutside');
```

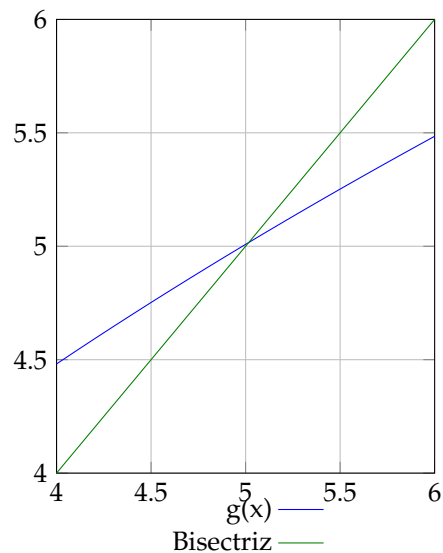


Figura 12: Catenaria $\frac{1}{\operatorname{acosh}\left(\frac{x+0.1}{x}\right)}$

En la figura observamos que obtenemos la misma solución que con la anterior función de punto fijo, apliquemos el método y a ver qué pasa:

```
pfixo('1./(\operatorname{acosh}((x+0.1)/x))','x-1./(\operatorname{acosh}((x+0.1)/x))',1,5,4,1e-4,20)
```

La solución es 5.016512483824275, y se ha alcanzado en 15 iteraciones.

Con un error de 6.658831e-05

¡Bingo!, con esta función de punto fijo obtenemos la aproximación deseada y además mucho más rápidamente, sólo con 15 iteraciones, luego esta segunda función de punto fijo es la óptima para nuestro problema de la catenaria.¹¹

c) Para realizar el apartado lo que haremos es ver la tendencia para ambas funciones su Tasa de Convergencia Lineal y compararlas. En la primera función tiende a 1, por eso la aproximación es lenta y mala. En la segunda función la tasa tiende a $0.5 < 1$, por esa razón el método converge rápidamente y con una aproximación muy buena.

¹¹ Atención, ésta es la óptima para $d = 0.1$, pero puede que no lo sea para otros valores diferentes de d , te animo a que experimentes por tu cuenta, es un buen ejercicio de aprendizajes, ¡ánimooo! ¡No muerde!

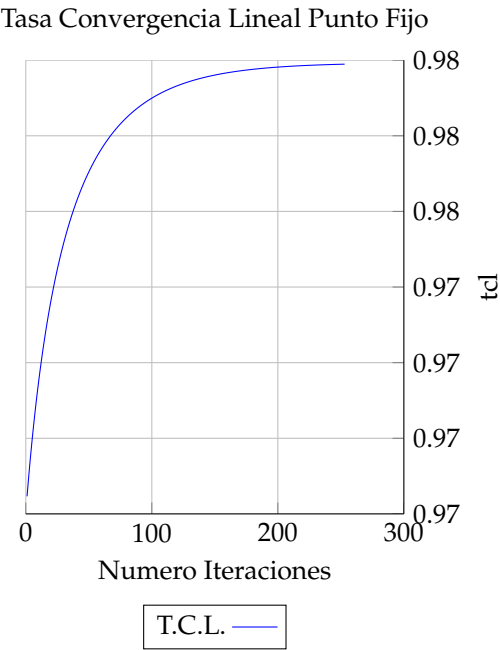


Figura 13: TCL Catenaria $x \cosh(\frac{1}{x}) - 0.1$

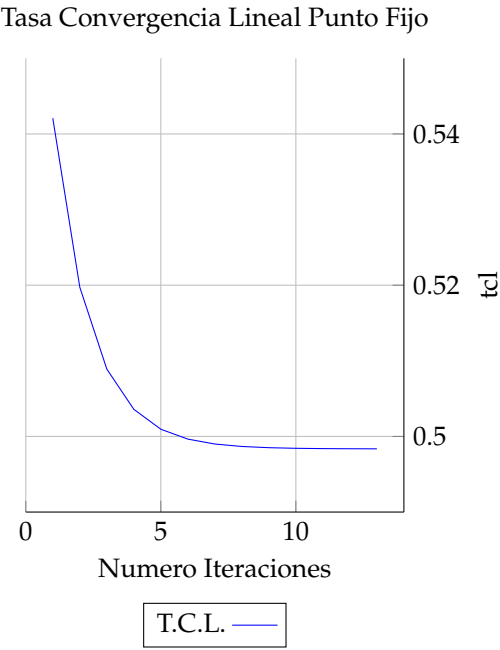


Figura 14: TCL Catenaria $\frac{1}{\operatorname{acosh}(\frac{x+0.1}{x})}$

Ejercicio 3.0.7 ► Raíces Décimas

Compara diferentes métodos para resolver la ecuación $p(x) = x^{10} - 1 = 0$, con una tolerancia de 10^{-6} .

Resolución

Bueno la ecuación tiene 10 raíces entre complejas y reales, de las reales tenemos 2, que son 1 y -1. Nos vamos a centrar en 1. Apliquemos el método de la bisección:

```
biseccion('x.^10-1',0.5,1.5,1e-6,30)
La solucion es 1.000000000000000.
Se ha alcanzado en 1 iteraciones.
biseccion('x.^10-1',0,1.5,1e-6,30)
La solucion es 0.999999761581421.
Se ha alcanzado en 21 iteraciones.
```

Bien, lo primero que observamos es que hay que tener en cuenta qué intervalo escogemos, en el primer caso obtenemos la solución exacta y en sólo 1 iteración, eso se debe a que el punto medio del intervalo coincide exactamente con la solución. Sin embargo, en el segundo caso su punto medio no es la solución, además no obtenemos la solución exacta y le cuesta hacerlo en 21 iteraciones. Por tanto, en éste caso no hace falta que sigamos buscando métodos porque hemos obtenido la solución ideal. Obviamente esto no ocurrirá en la mayoría de los casos, porque los métodos están hechos para aproximar raíces o para buscar raíces que no podemos (o es difícil) calcularlas mediante operaciones algebraicas. Veamos qué pasa con el método de la Regula Falsi:

```
regulafalsi('x.^10-1',0.5,1.5,1e-6,200)
La solucion es 0.999989952440931, y se ha alcanzado en 135 iteraciones.
Con un error de 9.723436e-07
regulafalsi('x.^10-1',0,1.5,1e-6,200)
La solucion es 0.999989784169578, y se ha alcanzado en 158 iteraciones.
Con un error de 9.886274e-07
```

Con éste método apreciamos que no obtenemos la solución exacta y además es muy lento, esto se debe a que en los intervalos elegidos la función casi no varía y esta cerca de cero, como puedes apreciar en la figura adjunta, puedes ver en rojo que las iteraciones se acumulan todas en el intervalo elegido con una distancia entre ellas muy pequeña.

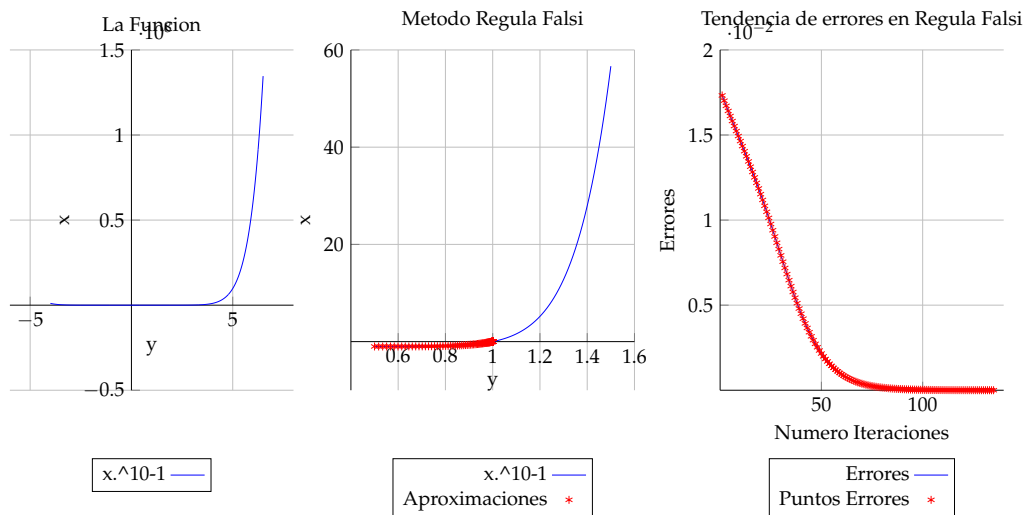


Figura 15: Regula Falsi $p(x) = x^{10} - 1$ en $[0.5, 1.5]$

Pasemos al método de Newton a ver qué pasa. Lo primero que debes tener en cuenta es que no podemos tomar como punto inicial el cero o puntos alrededor de un entorno pequeño del 0, porque su derivada se anula. Apliquemos el método:

```
newton('x.^10-1','10*x.^9',0.5,0.1,1e-6,50)
```

El algoritmo tiene un coste en tiempo de:

Elapsed time is 0.030031 seconds.

La solución es 1.000000000000000, y se ha alcanzado en 44 iteraciones.

Aquí obtenemos la solución exacta pero con muchas iteraciones, con lo cual perdemos la convergencia cuadrática del método, debido a que las rectas tangentes que se calculan en cada iteración varían muy poco. Eso se puede apreciar muy bien en las gráficas de tasas de convergencia en la figura adjunta.

Y ahora veamos qué ocurre con el método de la secante.

```
secante('x.^10-1',0,1.5,1e-6,200)
```

La solución es 0.051574707031250, y se ha alcanzado en 5 iteraciones.

Vaya, aquí el método claramente falla de forma estrepitosa, obtenemos una solución incorrecta aun no teniendo problemas de divergencia. Para solucionarlo debemos tener en cuenta el método de la bisección, ya que además de converger siempre, se tiene en consideración en cada iteración un subintervalo que contiene a la raíz mediante la partición en mitades iguales los subintervalos. Si en el método de la bisección elegimos el intervalo $[0.9375, 1.125]$ vemos que el método

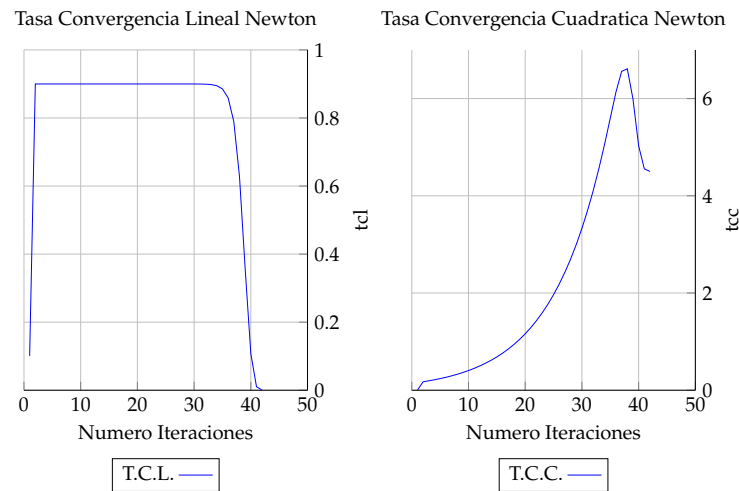


Figura 16: TC Newton $p(x) = x^{10} - 1$ en $[0.5, 1.5]$

sólo utiliza 3 iteraciones. Si lo aplicamos ahora en el método de la secante se obtienen resultados coherentes.

```
secante('x.^10-1',0.9375,1.1255,1e-6,20)
```

La solución es 0.999999999999997, y se ha alcanzado en 7 iteraciones.

Puedes probar éste mismo intervalo en el de la Regula Falsi o en el de Newton eligiendo un valor inicial contenido en dicho intervalo.

```
newton('x.^10-1','10*x.^9',1.1255,0,1,1e-6,50)
```

El algoritmo tiene un coste en tiempo de:

Elapsed time is 0.002492 seconds.

La solución es 1.0000000000000979, y se ha alcanzado en 6 iteraciones.

Ejercicio 3.0.8 ► Ecuación Logística

La ecuación logística $x = ax(1 - x)$, $a > 0$ es una ecuación de punto fijo que modela una población, cuyo crecimiento está limitado. Es muy conocida porque los iterados pueden presentar un comportamiento caótico.

- a) Determina analíticamente los puntos fijos de la ecuación logística.
- b) ¿Para qué valores de a es atrayente cada uno de los puntos fijos?
- c) ¿Para qué valores de a las iteraciones convergen cuadráticamente?
- d) Demuestra que, para valores del parámetro a entre 0 y 4, la función de punto fijo $g(x) = ax(1-x)$ aplica el intervalo $[0, 1]$ en si mismo. Consecuentemente, si la sucesión de iterados entra en este intervalo, ya no sale nunca de él.
- e) Comprueba que para $a = 0.5$, con cualquier estimación inicial en $[0, 1]$, los iterados convergen a 0. Comprueba que la convergencia es lineal y que la tasa de convergencia coincide con la derivada en 0.
- f) Comprueba que para $a = 3/2$, con cualquier estimación inicial en $[0, 1]$, los iterados convergen a $1 - 2/3 = 1/3$. Comprueba que la convergencia es lineal y que la tasa de convergencia coincide con la derivada en $1/3$.
- g) Comprueba que para $a = 2$ la convergencia es cuadrática.
- h) Representa gráficamente las iteraciones para algún valor entre 3 y 4. Comenta el comportamiento de los iterados.
- i) El fractal de Feigenbaum representa el parámetro en el eje de abscisas, y los valores que toman los iterados correspondientes a cada valor de a , cuando las iteraciones se estabilizan en el eje de ordenadas. Escribe un programa para dibujar éste fractal.

Resolución

a) Hemos de resolver la ecuación $x = ax(1-x) \rightarrow x - ax(1-x) = 0 \rightarrow x - ax + ax^2 = 0$

$$x = 0$$

$$\rightarrow x(ax - a + 1) = 0 \nearrow$$

$$\searrow ax - a + 1 = 0 \rightarrow x = \frac{a-1}{a} \rightarrow x = 1 - \frac{1}{a}$$

b) Un punto fijo x^* es Atractor o atrayente si $|g'(x^*)| < 1$. Tenemos que $g(x) = ax(1-x) = ax - ax^2$

$$g'(x) = a - 2ax = a(1-2x)$$

$$\rightarrow |g'(0)| = |a| < 1 \rightarrow \text{El punto } x^* = 0 \text{ es atractor si } |a| < 1 \rightarrow a \in]-1, 1[$$

$$\rightarrow \left| g'\left(1 - \frac{1}{a}\right) \right| = \left| a\left(1 - 2\left(1 - \frac{1}{a}\right)\right) \right| = \left| a - 2a + \frac{2a}{a} \right| = |2 - a| < 1 \leftrightarrow -1 < 2 - a < 1 \leftrightarrow -3 < -a < -1 \leftrightarrow 1 < a < 3$$

Luego el punto $x^* = 1 - \frac{1}{a}$ es atractor si $a \in]1, 3[$

Aquí lo que hemos hecho es saber dónde nuestros puntos fijos hacen que el método de punto fijo converja.

c) Las iteraciones convergen cuadráticamente cuando la derivada se anula en el punto fijo. Luego:

$$g'(x) = a - 2ax = 0$$

$$g'(0) = a = 0 \rightarrow \text{Para el punto fijo } x^* = 0 \text{ la convergencia es cuadrática si } a = 0.$$

$$g'\left(1 - \frac{1}{a}\right) = 2 - a = 0 \rightarrow a = 2 \rightarrow \text{Para el punto fijo } x^* = 1 - \frac{1}{a} \text{ la convergencia es cuadrática si } a = 2.^{12}$$

Podemos apreciar que para sólo 2 valores de a obtenemos una convergencia cuadrática del punto fijo, para los demás valores hay convergencia, pero desgraciadamente no es cuadrática.

d) Considerando $g(x) = ax(1 - x) = -ax^2 + ax$ en $[0,1]$ como dominio, tenemos que la función es continua en un compacto, por lo que alcanza sus máximos y mínimos en $[0,1]$. Hemos de darnos cuenta que nuestra función es un conjunto de parábolas en las que su coeficiente director es $-a < 0$ porque $a > 0$, por tanto; nuestras parábolas son \cap . Así pues su vértice es un máximo. Además $g(x) = ax(1 - x) = 0 \leftrightarrow x = 0 \vee x = 1$ son los mínimos, es decir, los puntos $(0,0)$ y $(1,0) \in [0,1] \times [0,1]$.

$$\text{Calculemos los vértices: } x = \frac{-b}{2a} = \frac{-a}{2(-a)} = \frac{1}{2}$$

$$g\left(\frac{1}{2}\right) = \frac{a}{2}\left(1 - \frac{1}{2}\right) = \frac{a}{4} \rightarrow 0 < \frac{a}{4} < 1 \leftrightarrow 0 < a < 4$$

Así pues, para $a \in]0,4[$ tenemos que $g(x) \in [0,1]$, como queríamos demostrar. Lo podemos dibujar con el siguiente código en Octave, con fichero llamado *logisticadibujo.m*:

¹²Observa que ambos valores de a que hemos obtenido pertenecen a sus respectivas regiones (intervalos) de convergencia.


```

1 function [] = logisticadibujo()
2     x=linspace(0,1,100);
3     clf();
4     for a=1:1:4
5         plot(x,a*x.*(1-x))
6         text(0.4 , 0.125, 'a=1');
7         text(0.4 , 0.4, 'a=2');
8         text(0.4 , 0.6, 'a=3');
9         text(0.4 , 0.8, 'a=4');
10        hold on
11    end
12 end

```

logisticadibujo.m

Puedes ver el resultado en la figura adjunta.

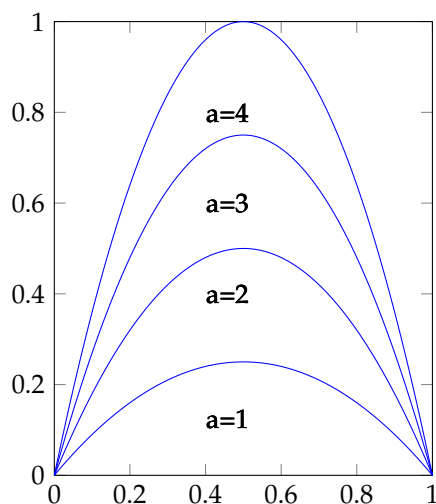


Figura 17: Función Logística $f(x) = ax(1-x)$ $a \in [1,4]$ *anatural*

e) Tomemos $a = 0.5$ y como estimación inicial 0.1:

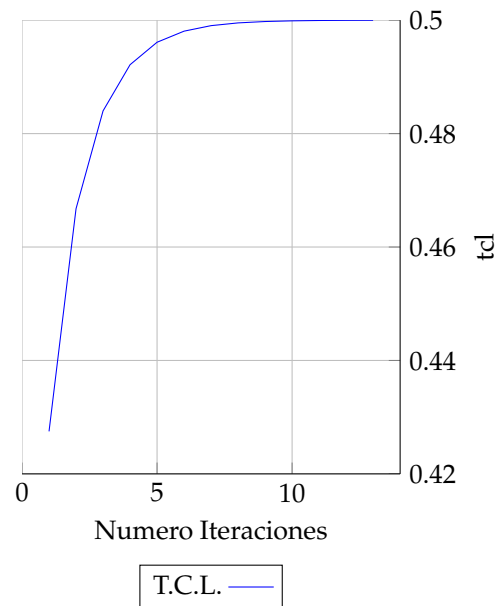
```
pfijs('0.5*x.*(1-x)', 'x-0.5*x.*(1-x)', 0, 1, 0.1, 1e-5, 20)
```

La solución es 0.000005026861483, y se ha alcanzado en 15 iteraciones.

En la figura adjunta se ve que la Tasa de Convergencia Lineal tiende a 0.5 que es lo que vale $g'(0) = a = 0.5$, como se esperaba, luego la convergencia es lineal.

f) Tomemos $a = 3/2 = 1.5$ y dos estimaciones iniciales 0.1 y 0.375 tales que cumplen:

Tasa Convergencia Lineal Punto Fijo

Figura 18: TCL Función Logística $f(x) = 0.5x(1-x)$

$0.1 < 1 - 1/a = 1 - 2/3 = 1/3$ $0.375 > 1 - 1/a = 1 - 2/3 = 1/3$. Apliquemos el método para ambas:

```
pfijo('1.5*x.*(1-x)', 'x-1.5*x.*(1-x)', 0, 1, 0.1, 1e-6, 25)
```

La solución es 0.333332828068138, y se ha alcanzado en 23 iteraciones.

```
pfijo('1.5*x.*(1-x)', 'x-1.5*x.*(1-x)', 0, 1, 0.375, 1e-6, 25)
```

La solución es 0.333334332553122, y se ha alcanzado en 16 iteraciones.

En los resultados obtenidos se ve que para 0.375 se converge más rápido, aunque se obtienen el mismo número de cifras decimales exactas y la misma velocidad lineal de convergencia, aunque una un poco más rápida que otra.

g) Tan sólo hemos de aplicar el método y ver que se cumple lo que nos piden, sin más.

```
pfijo('2*x.*(1-x)', 'x-2*x.*(1-x)', 0, 1, 0.375, 1e-6, 30)
```

tcc =

3.3333 2.2667 2.0157 2.0001

La solución es 0.500000000000000, y se ha alcanzado en 6 iteraciones.

h) Tomemos primero $a = 3.5 = 7/2$, entonces en $[0,1]$ tenemos 2 posibles puntos fijos (solución), uno es 0 y el otro es $1 - 1/a = 1 - 2/7 = 5/7$, pero no es así. Apliquemos el método:

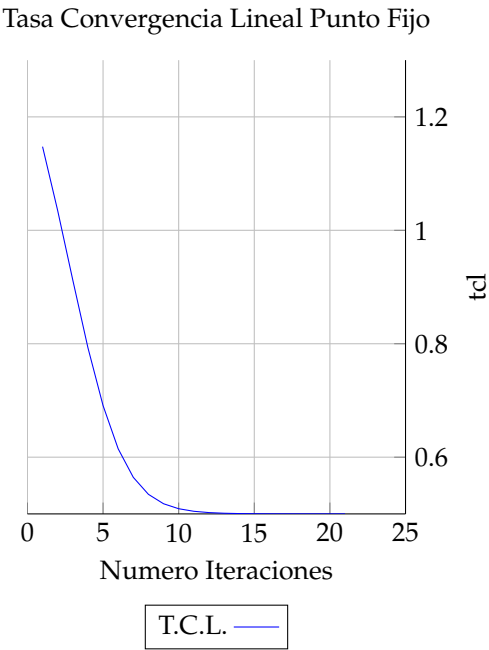


Figura 19: TCL Función Logística $f(x) = \frac{3}{2}x(1 - x)$ Estimación inicial 0.1

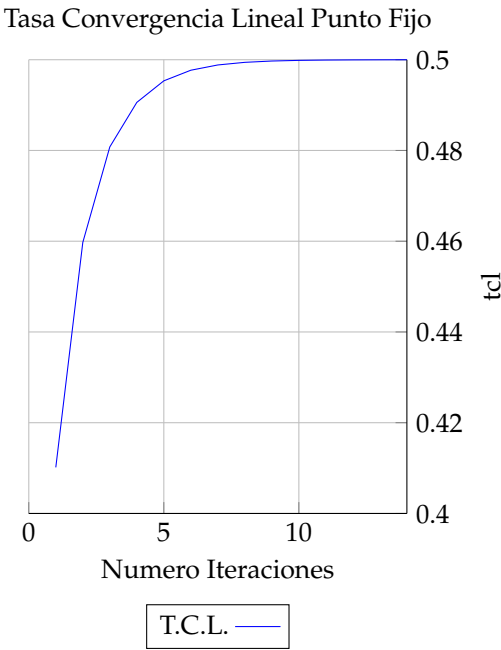


Figura 20: TCL Función Logística $f(x) = \frac{3}{2}x(1 - x)$ Estimación inicial 0.375

`pfijo('3.5*x.*(1-x)', 'x-3.5*x.*(1-x)', 0.1, 0.1, 1e-6, 500)`

Insuficientes iteraciones

Aun aplicando el método con un tope de 500 iteraciones vemos que el método diverge, si visualizamos la gráfica de la función junto con sus iteraciones lo que vemos es que hay 2 valores diferentes que se repiten periódicamente, ver figura adjunta.

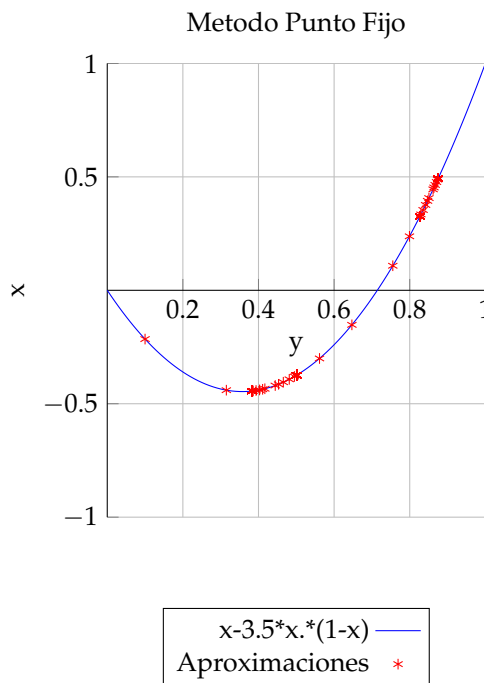


Figura 21: Pto. Fijo Función Logística $f(x) = \frac{7}{2}x(1-x)$

Si tomamos ahora $a = 3.75$ con valor inicial 0.5 el método no sólo diverge, sino que además sus iteraciones oscilan fuertemente, creando una especie de caos, lo que hace pensar en que escondido haya algún tipo de fractal caótico, el fractal de Feigenbaum.

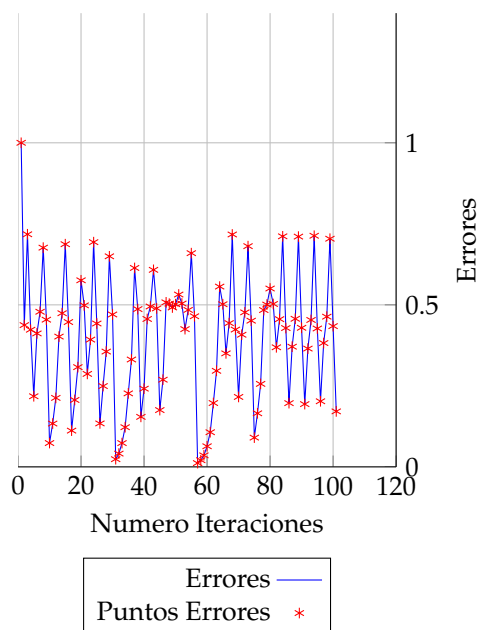
```
pfijo('3.75*x.*(1-x)', 'x-3.75*x.*(1-x)', 0, 1, 0.5, 1e-6, 100)
```

Insuficientes iteraciones

Este caos lo puedes ver muy bien en la gráfica sobre la Tendencia de Errores.

i) Esto no tiene mas que realizar el algoritmo en un fichero que le he llamado **feigenbaum.m** y cuyo código es:

Tendencia de errores en el Punto Fijo

Figura 22: Errores Pto. Fijo Función Logística $f(x) = \frac{15}{4}x(1-x)$

```

1 function feigenbaum
2     p=0; q=4; % Rango de variacion del parametro
3     r=0; s=1; % Intervalo de valores de la iteracion
4     h = (q-p)/500;
5     X = zeros(1,100001); Y=X;
6     j = 1;
7     for a = p:h:q % áParmetro
8         x = 0.5; % Estimacion Inicial
9         for k=1:400 % Iterados
10            x = a*x*(1-x);
11            if k>200 % Dibuja el punto
12                X(j) = a;
13                Y(j) = x;
14                j++;
15            end
16        end
17        j-1;
18    end
19
20    clf();
21    plot(X,Y, '.r')

```

```
22 axis([p,q,r,s])
23 print -dps Feigenbaum.eps
24 end
```

feigenbaum.m

Y te dejo adjunta una figura del fractal.

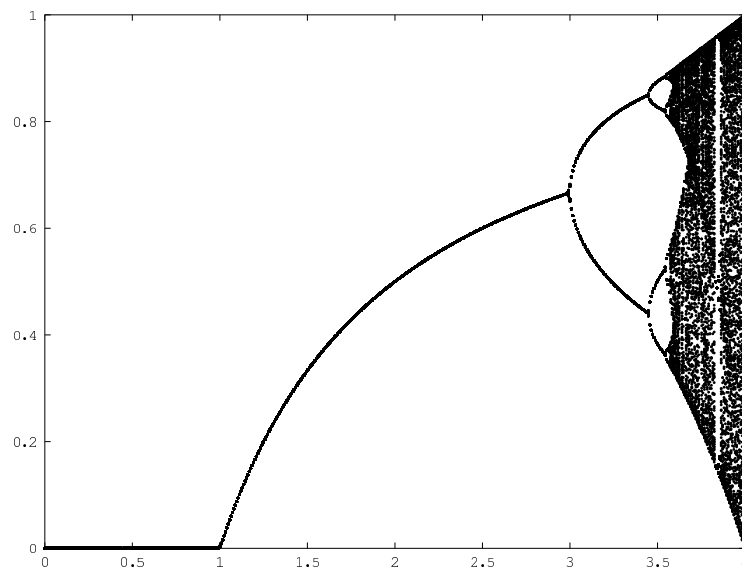


Figura 23: Fractal De Feigenbaum

Con esto se termina el ejercicio en sí, pero te aconsejo que experimentes con los demás métodos para ver qué es lo que ocurre. Debes caer que en los demás métodos la función no es $g(x) = ax(1-x)$, la función es $f(x) = x - ax(1-x) = ax^2 + (1-a)x$.

O por ejemplo, puedes probar cambiando la función de punto fijo por ésta otra:

$$ax^2 + (1 - a)x = 0 \rightarrow x^2 = \frac{1 - a}{a}x \rightarrow x = \pm \sqrt{\frac{1 - a}{a}}x \rightarrow g_2(x) = \pm \sqrt{\frac{1 - a}{a}}x$$

¿Con qué parte te quedas?, ¿la positiva?, ¿la negativa?, ¿ambas, redefiniéndola como una función a trozos?, ¿ninguna?. Ahí te dejo un bonito ejercicio para que investigues por tu cuenta, no es difícil pero sí algo laborioso. ¡Ánimo!

Ejercicio 3.0.9 ► Diodos

Cierto diodo se comporta como un condensador cuya capacidad C depende de la tensión V entre sus extremos, según la fórmula:

$$C(V) = \frac{C_0}{(1 + |V|/\varphi)^\gamma}$$

donde $C_0 = 81 \times 10^{-12} \text{ F}$ es la capacidad nominal del diodo, $\varphi = 0.6 \text{ V}$ y $\gamma = 0.44$ son parámetros característicos del mismo. A su vez, la tensión del diodo depende de la carga que almacena, según la Ley de Faraday: $V = Q/C$.

- Halla la carga Q del diodo, cuando la tensión entre sus extremos es de 1 V.
- Representa gráficamente la carga en función de V para valores de la tensión entre 0 y 5V. Para analizar el comportamiento del diodo se necesita conocer la tensión V en función de la carga Q , lo que lleva a la ecuación

$$V = \frac{Q}{C} = \frac{Q}{C_0} (1 + |V|/\varphi)^\gamma \rightarrow C_0 V - Q(1 + |V|/\varphi)^\gamma = 0$$

, en la que V es la incógnita.

- Para un valor $Q = 10^{-10} \text{ Coulomb}$ determina con precisión de millonésimas el valor de V correspondiente, utilizando el método de Newton.
- Resuelve el problema por el método del punto fijo.
- En el método del punto fijo, si $x_{k+1} = g(x_k)$, el cociente

$$\frac{g(x_{k+1}) - g(x_k)}{x_{k+1} - x_k} = \frac{x_{k+2} - x_{k+1}}{x_{k+1} - x_k}$$

aproxima el valor de la derivada de la función de iteración $g(x)$ en el punto fijo. Comprueba esta iteración para las iteraciones efectuadas en este caso, derivando la función de punto fijo $g(V)$ con respecto a V .

- Explica la relación entre el valor de g' y el comportamiento de las iteraciones del método de punto fijo.

Resolución

- Este primer apartado es una chorrada, observa: $V = Q/C \rightarrow Q = V \cdot C = 1 \cdot C \rightarrow Q(1) = C(1)$

$$C(1) = \frac{81 \times 10^{-12}}{\left(1 + \frac{1}{0.6}\right)^{0.44}} = \frac{81 \times 10^{-12}}{\left(\frac{1.6}{0.6}\right)^{0.44}} = 5.260884615 \times 10^{-11}$$

b) Tan sólo hay que implementar el siguiente código:

```

1 function diodo(V)
2     V = linspace(0,5);
3     C0 = 81e-12; fi = 0.6; gama = 0.44;
4     denom = (1+abs(V)/fi).^gama;
5     num = V*C0;
6     Q = num./denom;
7     clf();
8     grid;
9     plot(V,Q)
10 end

```

c) La ecuación a resolver es $C_0 V - Q \left(1 + \frac{|V|}{\varphi}\right)^\gamma = 0$, de ello definimos la función:

$$f(V) = C_0 V - Q \left(1 + \frac{|V|}{\varphi}\right)^\gamma$$

Vamos a obtener su primera derivada sin tener en cuenta el valor absoluto:

$$f'(V) = C_0 - \frac{\gamma Q}{\varphi} \left(1 + \frac{V}{\varphi}\right)^{\gamma-1}$$

Ejecutamos el método de Newton, sabiendo que la raíz está próxima a 2.5, por lo que elegimos 2.0 como aproximación inicial, con una tolerancia de 1e-6 y un máximo de iteraciones de 50. Obtenemos la aproximación a la raíz con tan sólo 10 iteraciones y coste de tiempo muy pequeño:

```
newton('diodo2(V)','diodo2derivada(V)',2.0,0.3,1e-6,50)
```

El algoritmo tiene un coste en tiempo de:

Elapsed time is 0.004084 seconds.

La solución es 2.566887136956549, y se ha alcanzado en 10 iteraciones.

d) Para aplicar el método del punto fijo hemos de calcular una función de punto fijo:

$$\begin{aligned}
 C_0 V - Q \left(1 + \frac{|V|}{\varphi}\right)^\gamma &= 0 \rightarrow C_0 V = Q \left(1 + \frac{|V|}{\varphi}\right)^\gamma \rightarrow V = \frac{Q}{C_0} \left(1 + \frac{|V|}{\varphi}\right)^\gamma \rightarrow \\
 &\rightarrow g(V) = \frac{Q}{C_0} \left(1 + \frac{|V|}{\varphi}\right)^\gamma
 \end{aligned}$$

Aplicamos el método con los siguientes resultados:

```
pfixo('diodoptofijo(V)','diodo2(V)',2.0,0,3,1e-6,50)
```

```
La solucion es 2.566887312786494, y se ha alcanzado en 15 iteraciones.
```

e) Si tenemos en cuenta en nuestro código del programa de punto fijo que: $x_{k+2} = x(2 : end)$ y que $x_{k+1} = x(1 : end)$, podemos añadir las siguientes tres líneas a nuestro código para calcular la derivada de g:

```
1 XR = x(2:end);
2 XL = x(1:end-1);
3 dg = diff(XR)./diff(XL)
```

Ahora al ejecutar nuestro programa de punto fijo obtenemos como resultado de nuestro cálculo de la derivada de la función de punto fijo:

```
dg =
Columns 1 through 8:
0.33959 0.35045 0.35442 0.35584 0.35635 0.35654 0.35660 0.35662
Columns 9 through 13:
0.35663 0.35664 0.35664 0.35664 0.35664
```

Aquí lo verdaderamente útil es que podemos calcular valores aproximados de la derivada de la función de punto fijo sin tener que calcular previamente su expresión algebraica, todo ello gracias a la definición de punto fijo que se cumple en nuestra función de punto fijo, valga la redundancia.

f) Si comparamos dg con tcl vemos que nos salen los mismos resultados, es decir, $tcl \approx dg = 0.35664$. Vemos que en éste problema en particular la convergencia es rápida pero sin llegar a ser cuadrática.

Ejercicio 3.0.10 ► Economía

Nos ofrecen un crédito de 6.000 a devolver en 50 mensualidades de 150. Llamando C al importe del préstamo, n al número de pagos, a al importe del plazo e i al tipo de interés por periodo, se cumple la ecuación siguiente:

$$Cr^n = a \frac{r^n - 1}{r - 1} / r = 1 + i$$

- Utiliza el método de Newton para obtener el interés del crédito partiendo de la estimación inicial $r = 1.1$ hasta alcanzar una precisión del orden de 10^{-6} .
- Resuelve el problema utilizando el método del punto fijo, tomando 3 funciones diferentes para hacer el estudio, analiza en cada caso la convergencia del método.

Resolución

a) Debemos determinar nuestra función f con variable independiente el interés r :

$$Cr^n = a \frac{r^n - 1}{r - 1} \rightarrow Cr^n(r - 1) = a(r^n - 1) \rightarrow Cr^n(r - 1) - a(r^n - 1) = 0 \rightarrow f(r) = Cr^n(r - 1) - a(r^n - 1)$$

Ahora debemos calcular su primera derivada ya que vamos a aplicar el método de Newton-Raphson:

$$f'(r) = Cnr^{n-1}(r - 1) + Cr^n - anr^{n-1} = r^{n-1}(Cn(r - 1) + Cr - an)$$

Para simplificar el trabajo sustituimos nuestros datos del problema en ambas funciones:

$$f(r) = 6000r^{50}(r - 1) - 150(r^{50} - 1)$$

$$f'(r) = r^{49}(300000(r - 1) + 6000r - 7500) \rightarrow f'(r) = 1500r^{49}(200(r - 1) + 4r - 5)$$

$$f'(r) = 1500r^{49}(204r - 205)$$

Y a estas funciones simplificadas les aplicamos el método de Newton con un valor inicial de $r = 1.1$ y tolerancia 10^{-6}

```
newton('6000*r.^50.*(r-1)-150.*(r.^50-1)', '1500*r.^49.*(204*r-205)',  
1.1,-1.5,1e-6,50)
```

La solución es 1.009128765779636, y se ha alcanzado en 13 iteraciones.

b) Veamos las tres formas diferentes de obtener la función de punto fijo:

$$\text{I) } Cr^n = a \frac{r^n - 1}{r - 1} \rightarrow r - 1 = a \frac{r^n - 1}{Cr^n} \rightarrow r = 1 + a \frac{r^n - 1}{Cr^n} \rightarrow g_1(r) = 1 + a \frac{r^n - 1}{Cr^n}$$

Sustituimos los datos del problema:

$$g_1(r) = 1 + 150 \frac{r^{50} - 1}{6000r^{50}} \rightarrow g_1(r) = 1 + \frac{r^{50} - 1}{40r^{50}}$$

Aplicamos el método de punto fijo:

```
pfijo('1+(r.^50-1)/(40*r.^50)', '6000*r.^50.*(r-1)-150.*(r.^50-1)', 1.1, 1.5, 1.1, 1e-6, 50)
```

La solución es 1.009131856951823, y se ha alcanzado en 33 iteraciones.

En éste caso la convergencia es muy lenta de tipo lineal y no cuadrática, y se estabiliza un poco por debajo de 0.8.

$$\text{II) } Cr^n = a \frac{r^n - 1}{r - 1} \rightarrow r^n = a \frac{r^n - 1}{C(r - 1)} \rightarrow r = \left(a \frac{r^n - 1}{C(r - 1)} \right)^{\frac{1}{n}} \rightarrow g_2(r) = \left(a \frac{r^n - 1}{C(r - 1)} \right)^{\frac{1}{n}}$$

Volvemos a sustituir los datos del problema:

$$g_2(r) = \left(150 \frac{r^{50} - 1}{6000(r - 1)} \right)^{\frac{1}{50}} \rightarrow g_2(r) = \left(\frac{r^{50} - 1}{40(r - 1)} \right)^{\frac{1}{50}}$$

Aplicamos el método de punto fijo:

```
pfijo('((r.^50-1)/(40*(r-1)))^(1/50)', '6000*r.^50.*(r-1)-150.*(r.^50-1)', 1.1, 1.5, 1.1, 1e-6, 50)
```

La solución es 1.009129747274095, y se ha alcanzado en 20 iteraciones.

$$\text{III) } Cr^n = a \frac{r^n - 1}{r - 1} \rightarrow \frac{Cr^n(r - 1)}{a} = r^n - 1 \rightarrow r^n = 1 + \frac{Cr^n(r - 1)}{a} \rightarrow g_3(r) = \left(1 + \frac{Cr^n(r - 1)}{a} \right)^{\frac{1}{n}}$$

Sustituimos los datos del problema:

$$g_3(r) = \left(1 + \frac{6000r^{50}(r-1)}{150}\right)^{\frac{1}{50}} \rightarrow g_3(r) = (1 + 40r^{50}(r-1))^{\frac{1}{50}}$$

Finalmente aplicamos el método:

```
pfigo('(1+40*r.^50)*(r-1)).^(1/50)', '6000*r.^50.*(r-1)-150.*(r.^50-1)', 1, 1.5, 1.1, 1e-6, 50)
Insuficientes iteraciones
```

En éste caso el método diverge.

Ejercicio 3.0.11 ► Coseno Diferentes Métodos

Dada la ecuación no lineal: $8x - \cos x - 2x^2 = 0$

- Haz una representación gráfica para averiguar el número de soluciones y obtener una estimación inicial de las mismas.
- Resuélvela por: bisección, regula falsi, secante y Newton con tolerancia 10^{-5} .
- Transfórmala en una ecuación de punto fijo de dos formas diferentes de modo que en una de ellas el proceso converja con un número de iteraciones similar al obtenido al método de Newton.
- Halla la tasa de convergencia para cada método utilizado y compara los resultados obtenidos.

Resolución

a) Lo primero que vamos a hacer es establecer nuestra función como:

$$f(x) = 8x - \cos x - 2x^2$$

En éste caso podemos saber cuántas raíces tiene nuestra ecuación sin necesidad de dibujarla, si despejamos el coseno en nuestra ecuación tenemos: $8x - 2x^2 = \cos x$. En la parte derecha tenemos una parábola y en el izquierdo, donde se corten van a ser nuestras raíces, y en éste caso como mucho van a ser 2 puntos, de hecho son 2 puntos. Dibujemos con Octave:

En la figura de la gráfica de la función vemos que efectivamente tenemos dos raíces, una primera en torno al cero y pico, la segunda cerca del 4.

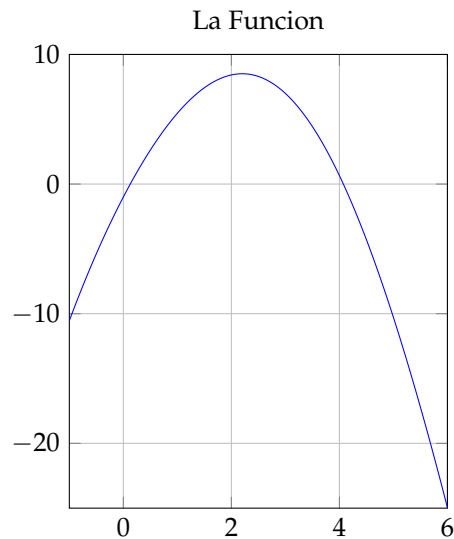


Figura 24: $f(x) = 8x - \cos x - 2x^2$

b) Método Bisección:

Vamos a tomar como intervalos iniciales $[-0.5, 0.5]$ y $[3.5, 4.5]$:

```
biseccion('8*x-cos(x)-2*x.^2',-0.5,0.5,1e-5,20)
```

La solucion es 0.128074645996094.

Se ha alcanzado en 17 iteraciones.

Con un error de 1.525879e-05

```
biseccion('8*x-cos(x)-2*x.^2',3.5,4.5,1e-5,20)
```

La solucion es 4.073219299316406.

Se ha alcanzado en 17 iteraciones.

Con un error de 1.525879e-05

En ambas raíces obtenemos buenas aproximaciones aunque con un número un poco elevado de iteraciones.

Método Regula Falsi:

Tomamos los mismos intervalos que hemos utilizado en la bisección:

```
regulafalsi('8*x-cos(x)-2*x.^2',-0.5,0.5,1e-5,20)
```

La solucion es 0.128077837733803, y se ha alcanzado en 6 iteraciones.

Con un error de 5.913176e-06

```
regulafalsi('8*x-cos(x)-2*x.^2',3.5,4.5,1e-5,20)
```

La solución es 4.073224583175699, y se ha alcanzado en 6 iteraciones.
Con un error de 4.875398e-06

Poniendo las mismas condiciones que en la bisección vemos que el método es mucho más rápido, menos del doble de iteraciones son las que han hecho falta para obtener las soluciones.

Método de la Secante:

También vamos a utilizar las mismas condiciones que hemos utilizado en los 2 métodos anteriores:

```
secante('8*x-cos(x)-2*x.^2',-0.5,0.5,1e-5,20)
```

La solución es 0.128077102753662, y se ha alcanzado en 5 iteraciones.

```
secante('8*x-cos(x)-2*x.^2',3.5,4.5,1e-5,20)
```

La solución es 4.073225094955231, y se ha alcanzado en 5 iteraciones.

Este método mejora ligeramente el número de iteraciones respecto a la Regula Falsi, tan sólo una iteración menos.

Método Newton-Raphson:

Primero es calcular la primera derivada de la función:

$$f'(x) = 8 + \sin x - 4x$$

Para la primera raíz vamos a considerar como valor inicial 0.25, y para la segunda raíz tomamos 4.25:

```
newton('8*x-cos(x)-2*x.^2','8+sin(x)-4*x',0.25,-0.5,0.5,1e-5,20)
```

La solución es 0.128077102753090, y se ha alcanzado en 4 iteraciones.

```
newton('8*x-cos(x)-2*x.^2','8+sin(x)-4*x',4.25,3.5,4.5,1e-5,20)
```

La solución es 4.073225094959651, y se ha alcanzado en 5 iteraciones.

Aquí se utilizan muy pocas iteraciones, es el mejor de todos los vistos hasta ahora, y todo hace pensar que su convergencia es cuadrática, pero esto lo analizaremos un poco después.

Hagamos un in-pass aquí. Hasta ahora las raíces de nuestra ecuación de partida sólo las hemos podido calcular de forma aproximada aplicando métodos iterativos o contemplando la gráfica de su función asociada. El problema de esta ecuación es que no podemos resolverla de forma exacta utilizando sólo simples cálculos algebraicos. Si contemplamos detenidamente la gráfica de la función vemos que tiene cierto parecido a una parábola a la que se le ha aplicado una

modificación del tipo coseno. Así pues, podríamos pensar que podríamos calcular un polinomio de segundo grado (parábola) que se aproxime lo suficientemente a la función de partida. Veamos qué pasa si aproximamos la función coseno mediante el desarrollo de McLaurin¹³.

$$h(x) = \cos x \quad h'(x) = -\sin x \quad h''(x) = -\cos x \quad h'''(0) = \sin x \quad h^{iv}(x) = \cos x$$

$$P(x) = h(0) + \frac{h'(0)}{1!}x + \frac{h''(0)}{2!}x^2 = 1 + 0 - \frac{1}{2}x^2 \rightarrow P(x) = 1 - \frac{1}{2}x^2$$

Sustituimos $P(x)$ en la ecuación y resolvemos:

$$8x - 1 + \frac{1}{2}x^2 - 2x^2 = 0 \rightarrow -16x + 2 - x^2 + 4x^2 = 0 \rightarrow 3x^2 - 16x + 2 = 0 \rightarrow$$

$$\rightarrow x = \frac{16 \pm \sqrt{16^2 - 4 \cdot 3 \cdot 2}}{6} = \frac{16 \pm \sqrt{232}}{6}$$

$$\rightarrow x = \frac{16 \pm 2\sqrt{58}}{6} = \frac{8 \pm \sqrt{58}}{3} \rightarrow x_1 = 5.205257703 \quad \wedge \quad x_2 = 0.1280756314$$

Vemos que nuestra segunda solución utilizando McLaurin sí es una buena aproximación a una de las raíces de la ecuación de partida del problema. Desgraciadamente la primera solución se desvía bastante de la otra raíz, una de las posibles causas de que ocurra esto es que $f(x)$ no es una función simétrica. Una forma de intentar solucionarlo es desarrollar McLaurin hasta grado 4, veamos qué pasa:

$$P(x) = h(0) + \frac{h'(0)}{1!}x + \frac{h''(0)}{2!}x^2 + \frac{h'''(0)}{3!}x^3 + \frac{h^{iv}(0)}{4!}x^4 = 1 + 0 - \frac{1}{2}x^2 + 0 + \frac{1}{24}x^4 \rightarrow$$

$$\rightarrow P(x) = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4$$

Sustituimos nuestro nuevo $P(x)$ en la ecuación:

$$8x - 1 + \frac{1}{2}x^2 - \frac{1}{24}x^4 - 2x^2 = 0 \rightarrow -192x + 24 - 12x^2 + x^4 + 48x^2 = 0 \rightarrow x^4 + 36x^2 - 192x + 24 = 0$$

Desgraciadamente esta nueva ecuación polinómica no es fácil de resolver, así que para estimar la otra raíz deberíamos aplicar alguno de los métodos iterativos o dibujar ambas gráficas y compararlas:

```
1 x=-0.5:0.1:4.5;
2 clf(), plot(x,8*x-cos(x)-2*x.^2,x,x.^4+36*x.^2-192*x+24), grid on
```

Y vemos que la nueva aproximación que hemos intentado es peor que la primera. Bueno, pues en éste caso nuestro experimento ha resultado sólo parcialmente bueno en cuanto a resultados, pero enriquecedor en ampliación de conocimientos. Te animo a que experimentes, a que te preguntes

¹³Recuerda que el desarrollo de McLaurin no es mas que el polinomio de Taylor centrado alrededor del punto cero.

por ti mismo cuáles pueden ser las causas de que no haya funcionado del todo bien y si hay otros caminos alternativos. Por mi parte, para no alargar mucho el ejercicio lo voy a dejar aquí y pasar al siguiente apartado.

c) Vamos a averiguar la primera ecuación/función de punto fijo en nuestro problema:

$$8x - \cos x - 2x^2 = 0 \rightarrow x = \frac{1}{8} \cos x - \frac{1}{4}x^2 \rightarrow g_1(x) = \frac{1}{8} \cos x - \frac{1}{4}x^2$$

$$\text{Estudiamos su criterio de convergencia: } g_1'(x) = \frac{-1}{8} \sin x - \frac{1}{2}x$$

$$|g_1'(x)| = \left| \frac{-1}{8} \sin x - \frac{1}{2}x \right| = \left| \frac{-\sin x - 4x}{8} \right| = \frac{1}{8} |\sin x + 4x| \leq \frac{1}{8} [|\sin x| + 4|x|] \leq \frac{1}{8} [1 + 4|x|]$$

$$|g_1'(x)| < 1 \Leftrightarrow \frac{1}{8} [1 + 4|x|] < 1 \Leftrightarrow [1 + 4|x|] < 8 \Leftrightarrow 4|x| < 7 \Leftrightarrow |x| < \frac{7}{4} = 1.75 \Leftrightarrow -1.75 < x < 1.75$$

Luego el método de punto fijo para $g_1(x)$ converge en el intervalo $] -1.75, 1.75[$. Por tanto, al aplicar el método sólo conseguiremos aproximaciones a la raíz 1.028..., pero para la raíz 4.07... el método diverge porque ni la raíz ni valores iniciales para aplicar el método caen dentro del intervalo $] -1.75, 1.75[$. Aplicamos el método para el cálculo de ambas raíces y obtenemos los resultados esperados.

```
pfijs('1/8*cos(x)-(1/4)*x.^2','8*x-cos(x)-2*x.^2',3.5,4.5,4.25,1e-5,100)
```

La solución es NaN

```
pfijs('1/8*cos(x)-(1/4)*x.^2','8*x-cos(x)-2*x.^2',-0.5,0.5,0.25,1e-5,100)
```

La solución es 0.120465620630844, y se ha alcanzado en 6 iteraciones.

Pero además vemos que la aproximación obtenida de la raíz es bastante pobre.

Veamos una segunda función de punto fijo:

$$8x - \cos x - 2x^2 = 0 \rightarrow 2x^2 = 8x - \cos x \rightarrow x = \sqrt{4x - \frac{\cos x}{2}} \rightarrow g_2(x) = \sqrt{4x - \frac{\cos x}{2}} \rightarrow g_2(x) = \frac{\sqrt{2}}{2} \sqrt{8x - \cos x}$$

Tenemos que: $g_2'(x) = \frac{\sqrt{2}}{4} \frac{8 + \sin x}{\sqrt{8x - \cos x}}$ Esta derivada es algo complicada de acotar, su gráfica siempre es positiva. Para ver dónde $|g_2'(x)| < 1$ lo que haremos es dibujar directamente su gráfica en Octave mediante el código:

```
1 function[] = ecuacioncoseno()
2   x = -0.5:0.1:60;
3   c = sqrt(2)/4;
```

```

4   num = 8 + sin(x);
5   den = sqrt(8*x-cos(x));
6   fun = (c*num)./den;
7   clf();
8   plot(x,fun)
9   grid;
10  end

```

Al ejecutarlo podemos apreciar en su grafica que hay 2 zonas en las que su gráfica es menor que 1 y que, supuestamente nuestro método de punto fijo converge a la raíz.

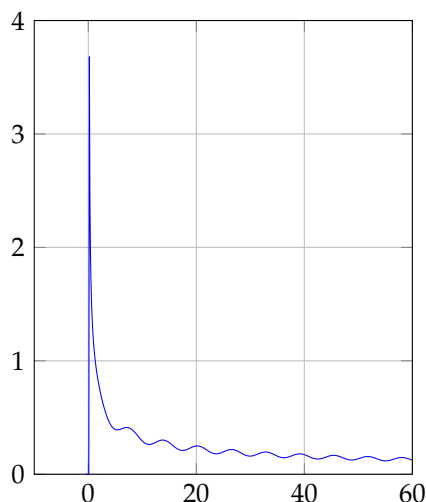


Figura 25: Derivada de $g_1(x)$

La zona, aproximadamente es: $]0,0'11[\cup]1'3,+\infty[$. Vamos a aplicar el método de punto fijo para los valores iniciales 0 y 4 a ver qué ocurre:

```

pfijo('sqrt(2)/2.*(sqrt(8*x-cos(x)))','8*x-cos(x)-2*x.^2',3.5,4,5,4,1e-5,100)
La solucion es 4.073220999225586, y se ha alcanzado en 13 iteraciones.
pfijo('sqrt(2)/2.*(sqrt(8*x-cos(x)))','8*x-cos(x)-2*x.^2',-0.5,0.5,0,1e-5,100)
La solucion es 4.073221928782237, y se ha alcanzado en 20 iteraciones.

```

Finalmente hay una tercera función candidata si despejamos el coseno: $g_3(x) = \arccos(8x - 2x^2)$, si aplicamos el método de punto fijo con ésta función vemos que el método diverge:

```

pfijo('acos(8*x-2*x.^2)','8*x-cos(x)-2*x.^2',-0.5,0.5,0.5,1e-5,500)
Insuficientes iteraciones

```

`pfijo('acos(8*x-2*x.^2)','8*x-cos(x)-2*x.^2',3,4.5,4.5,1e-5,500)`

Insuficientes iteraciones

Calculamos su derivada: $g'_3(x) = \frac{4x - 8}{\sqrt{1 - (8x - 2x^2)^2}}$ Si dibujamos su gráfica vemos claramente que ésta función alrededor del 0 y del 4 sus valores en módulo son mayores que 1, además también podemos apreciar que no es una función suave¹⁴. Por estas razones el método del punto fijo diverge para ésta tercera función de punto fijo.

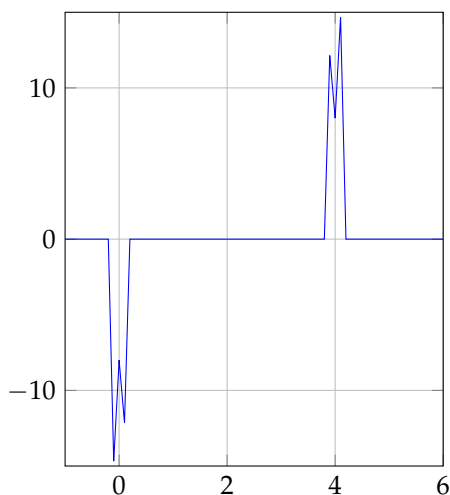


Figura 26: Derivada de $g_3(x)$

La podemos implementar en Octave con las siguientes líneas de código:

```
1  clf();
2  x=-1:0.1:6;
3  grid on;
4  plot(x, (4*x-8) ./ sqrt(1 - (8*x-2*x.^2).^2))
```

¹⁴Presenta picos, no es derivable, o lo que es lo mismo, hay puntos donde no hay una única recta tangente.

Ejercicio 3.0.12 ► Identidad De Raabe

Vamos a buscar los valores de a que verifican la Identidad de Raabe:

$$a \ln a - a + \ln(\sqrt{2\pi}) = a$$

- Obtén gráficamente una aproximación inicial de las raíces reales de esta ecuación no lineal.
- Utiliza el método de Newton para obtener la menor de sus raíces positivas con tolerancia una diezmilésima. ¿Cuántas iteraciones has necesitado?
- Utiliza ahora el método de punto fijo para obtener las soluciones.

Resolución

a) Consideramos la función $f(a) = a \ln(a) - 2a + \ln(\sqrt{2\pi})$ y la dibujamos con Octave:

Debes tener en cuenta que el dominio de la función es $D(f) =]0, +\infty[$ debido al logaritmo neperiano, así que al definir nuestra variable independiente en Octave lo haremos a partir de un valor cercano a 0, sin ser 0, para evitar problemas.

Aunque en el problema no lo piden vamos a calcular los posibles extremos de la función para servirnos de guía.

$$f'(a) = \ln(a) - 1 = 0 \rightarrow a = e$$

$f''(a) = \frac{1}{a} \rightarrow f''(e) = \frac{1}{e} > 0 \rightarrow$ Para $a = e$ tenemos un Mínimo Absoluto, por lo que la función Decrece en $]0, e[$ y Crece en $]e, +\infty[$. Además, como $f''(a) = \frac{1}{a} > 0; \forall a \in D(f)$, la función siempre es Convexa. Los comandos en Octave para dibujar la función pueden ser los siguientes:

```
1 a=0:0.1:10;
2 plot(a,a.*log(a)-2*a+log(sqrt(2*pi)))
3 grid;
```

Obteniendo la figura adjunta.

Con todo esto que hemos hecho y con la ayuda de la gráfica podemos aseverar que nuestra función tiene 2 raíces, una alrededor del 0, y otra alrededor del 6.

b) La menor de las raíces es la que está alrededor del 0, y esa es la que vamos a calcular por Newton. Tomamos como punto de partida 0.25:

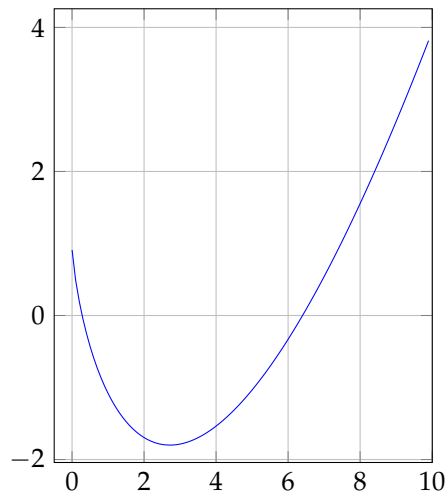


Figura 27: Función De Raabe $f(a) = a \ln a - 2a + \ln(\sqrt{2\pi})$

```
newton('a.*log(a)-2*a+log(sqrt(2*pi))','log(a)-1',0.25,0.00001,1,1e-4,20)
```

El algoritmo tiene un coste en tiempo de:

Elapsed time is 0.0525141 seconds.

La solución es 0.281104424618856, y se ha alcanzado en 4 iteraciones.

Vemos que obtenemos como aproximación de la raíz el valor esperado y el método converge muy rápido.

c) Para el método del punto fijo vamos a tomar la función $g(a) = \frac{2a - \ln(\sqrt{2\pi})}{\ln(a)}$

```
pfijo('(2*a-log(sqrt(2*pi)))/log(a)','a.*log(a)-2*a+log(sqrt(2*pi))',0.00001,1,0.25,1e-4,300)
```

La solución es 0.281069334437776, y se ha alcanzado en 29 iteraciones.

```
pfijo('(2*a-log(sqrt(2*pi)))/log(a)','a.*log(a)-2*a+log(sqrt(2*pi))',6,8,0.7,1e-4,300)
```

La solución es 6.400796607782891, y se ha alcanzado en 22 iteraciones.

Con éste método obtenemos las raíces buscadas pero la convergencia es muy lenta, en vez de ser cuadrática es lineal.

Ejercicio 3.0.13 ► Problema De La Caja

Se construye una caja sin tapadera a partir de una hoja metálica rectangular, que mide 10 por 16 centímetros. Queremos averiguar cuál debe ser el lado de los cuadrados que hay que recortar en cada esquina para que el volumen de la caja sea 100 centímetros cúbicos.

- Plantea la ecuación que nos proporciona el volumen de la caja en función del lado x de los cuadrados a recortar.
- Representa gráficamente la ecuación a resolver y obtén un intervalo inicial en el que esté el valor de x que estamos buscando.
- Resuelve dicha ecuación por el método de bisección, empleando una tolerancia de 10^{-9} y tomando como intervalo inicial el obtenido en el apartado b).
- ¿Mejora el método de regula falsi el resultado obtenido por bisección?
- Obtén el lado de dichos cuadrados empleando el método de punto fijo, con estimación inicial el punto medio del intervalo utilizado en apartados anteriores.
- Compara los resultados obtenidos por ambos métodos (solución, número de iteraciones, velocidad de convergencia).

Resolución

a) Lo primero que debes saber es que x es la altura de la caja, así como que el volumen es el mismo con o sin tapa. Aunque el área total de la caja no es la misma con o sin tapa.

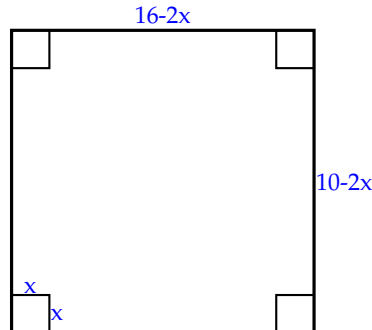


Figura 28: Problema de la caja

$$100 = V(x) = x(16 - 2x)(10 - 2x) = 4x(8 - x)(5 - x) = 4x(40 - 8x - 5x + x^2) = 160x - 32x^2 - 20x^2 + 4x^3$$

$$\rightarrow 4x^3 - 52x^2 + 160x - 100 = 0 \rightarrow x^3 - 13x^2 + 40x - 25 = 0$$

Por tanto, nuestra función va a ser:¹⁵

$$f(x) = x^3 - 13x^2 + 40x - 25$$

b) Nuestra incógnita cumple que $x > 0$ por ser la medida de un lado, además, $10 - 2x > 0 \Leftrightarrow 5 - x > 0 \Leftrightarrow x < 5$. Luego sin dibujar nada ya sabemos que $x \in]0,5[$ debido a la naturaleza del problema. La dibujamos en Octave con el siguiente código:

```
1 x=0:0.1:10;  
2 plot(x,x.^3-13*x.^2+40*x-25)  
3 grid;  
4 legend('f(x)');
```

En la gráfica de la función vemos que de las 3 raíces reales que tiene la ecuación por ser de grado 3 sólo 2 se ajustan al dominio de nuestra función para nuestro problema. Así que tomaremos los intervalos $[0,1]$ y $[3,4]$

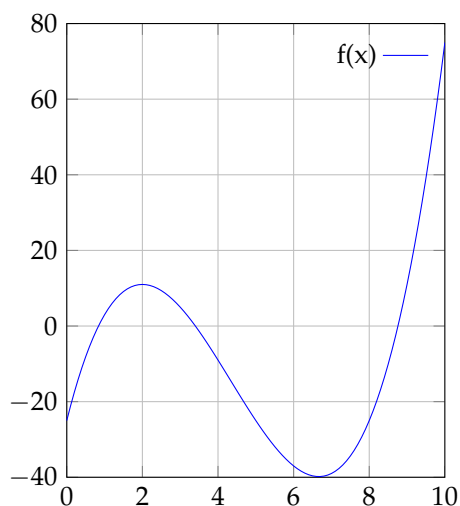


Figura 29: Función $f(x) = x^3 - 13x^2 + 40x - 25$

c) Apliquemos bisección a ambos intervalos:

¹⁵Date cuenta que hemos simplificado dividiendo por 4 la ecuación porque para obtener x no influye en el resultado por ser ecuaciones equivalentes, de esa forma simplificamos cálculos, sobretodo determinando el intervalo inicial para aplicar por ejemplo la bisección.

```
biseccion('x.^3-13*x.^2+40*x-25',0,1,1e-9,100)
La solucion es 0.839018882252276.
Se ha alcanzado en 30 iteraciones.
Con un error de 1.862645e-09
```

Para el otro intervalo obtenemos:

```
biseccion('x.^3-13*x.^2+40*x-25',3,4,1e-9,100)
La solucion es 3.401748646982014.
Se ha alcanzado en 30 iteraciones.
Con un error de 1.862645e-09
```

d) Y ahora para regla falsi:

```
regulafalsi('x.^3-13*x.^2+40*x-25',0,1,1e-9,100)
La solucion es 0.839018883302484, y se ha alcanzado en 18 iteraciones.
Con un error de 4.274464e-10
```

Y en el segundo intervalo:

```
regulafalsi('x.^3-13*x.^2+40*x-25',3,4,1e-9,100)
La solucion es 3.401748647478851, y se ha alcanzado en 10 iteraciones.
Con un error de 1.420744e-10
```

Las soluciones obtenidas son prácticamente las mismas a las obtenidas en la bisección, pero el número de iteraciones es mucho menor, especialmente para la segunda solución.

e) Calculemos una función de punto fijo:

$$x^3 - 13x^2 + 40x - 25 = 0 \rightarrow 40x = -x^3 + 13x^2 + 25 \rightarrow x = \frac{-1}{40}(x^3 - 13x^2 - 25)$$

Luego tenemos como función de punto fijo:

$$g(x) = \frac{-1}{40}(x^3 - 13x^2 - 25)$$

Los puntos iniciales van a ser 0.5 y 3.5 respectivamente:


```
pfijs('(-1/40)*(x.^3-13*x.^2-25)', 'x.^3-13*x.^2+40*x-25', 0, 1, 0.5, 1e-9, 100)
```

La solución es 0.839018882515451, y se ha alcanzado en 29 iteraciones.

```
pfijs('(-1/40)*(x.^3-13*x.^2-25)', 'x.^3-13*x.^2+40*x-25', 3, 4, 3.5, 1e-9, 100)
```

La solución es 8.759232469400999, y se ha alcanzado en 23 iteraciones.

Como vemos aquí con esta función de punto fijo sólo obtenemos la primera solución y no con muy buenos resultados, de hecho perdemos la convergencia cuadrática del método, sólo llegamos a convergencia lineal. Para obtener la otra solución del problema debemos buscar otra función de punto fijo, eso lo dejo como tarea.

Claramente el método de la Regula Falsi ha sido el vencedor.

Ejercicio 3.0.14 ► Método De Halley

El método de Halley, para encontrar una raíz de la ecuación $f(x) = 0$, fue diseñado para acelerar la convergencia del método de Newton. Puede interpretarse como un método de punto fijo, cuya función $g(x)$ es:

$$g(x) = x - \frac{f(x)}{f'(x)} \left(1 - \frac{f(x)f''(x)}{2(f'(x))^2} \right)^{-1}$$

A partir de $f(x) = x^2 - A$, con $A = 2, 3, 5, 7$, determina la función de iteración de Halley para hallar \sqrt{A} . Empieza con $x_0 = 1.4$. ¿Cuál es la velocidad de convergencia del método en este caso? Compara estos resultados con los obtenidos aplicando otros métodos: bisección, regla falsi, punto fijo y Newton.

Resolución

Vamos a averiguar la función g :

$$f(x) = x^2 - A$$

$$f'(x) = 2x \quad f''(x) = 2 \quad (f'(x))^2 = 4x^2 \quad f(x)f''(x) = 2x^2 - 2A$$

$$\left(1 - \frac{f(x)f''(x)}{2(f'(x))^2} \right)^{-1} = \left(1 - \frac{2x^2 - 2A}{8x^2} \right)^{-1} = \left(\frac{6x^2 + 2A}{8x^2} \right)^{-1} = \frac{8x^2}{6x^2 + 2A} = \frac{4x^2}{3x^2 + A}$$

$$\frac{f(x)}{f'(x)} = \frac{x^2 - A}{2x}$$

$$\frac{f(x)}{f'(x)} \left(1 - \frac{f(x)f''(x)}{2(f'(x))^2} \right)^{-1} = \frac{x^2 - A}{2x} \cdot \frac{4x^2}{3x^2 + A} = \frac{2x(x^2 - A)}{3x^2 + A}$$

$$g(x) = x - \frac{2x(x^2 - A)}{3x^2 + A} = \frac{3x^3 + Ax - 2x^3 + 2Ax}{3x^2 + A} = \frac{x^3 + 3Ax}{3x^2 + A} = \frac{x(x^2 + 3A)}{3x^2 + A}$$

Luego hemos obtenido la función de Halley:

$g(x) = \frac{x(x^2 + 3A)}{3x^2 + A}$, dicha función de Halley le vamos a aplicar nuestro método de punto fijo: ¹⁶

I) Para A=2:

```
pfijo('(x.^3+6*x)./(3*x.^2+2)', 'x.^2-2', 1, 2, 1.4, 1e-6, 50)
```

La solución es 1.414213562373095, y se ha alcanzado en 3 iteraciones.

II) Para A=3:

```
pfijo('(x.^3+9*x)./(3*x.^2+3)', 'x.^2-3', 1, 2, 1.7, 1e-6, 50)
```

La solución es 1.732050807568877, y se ha alcanzado en 4 iteraciones.

III) Para A=5:

```
pfijo('(x.^3+15*x)./(3*x.^2+5)', 'x.^2-5', 2, 3, 2.2, 1e-6, 50)
```

La solución es 2.236067977499790, y se ha alcanzado en 4 iteraciones.

IV) Para A=7:

```
pfijo('(x.^3+21*x)./(3*x.^2+7)', 'x.^2-7', 2, 3, 2.6, 1e-6, 50)
```

La solución es 2.645751311064591, y se ha alcanzado en 4 iteraciones.

Como vemos aquí el método de Halley es muy rápido. Pero tiene varios inconvenientes, nuestra función de partida debe ser de clase C^2 en un entorno de x_0 . Además no debe tener una expresión muy complicada, en el caso de nuestra función de partida es un polinomio de grado muy bajo, por lo que el coste computacional es óptimo.

¹⁶ Esto lo hacemos porque el Método de Halley no es más que el Método de Punto Fijo pero en donde nuestra función de punto fijo es la función de Halley. De esta forma nos evitamos tener que programar un código específico para el Método de Halley.

Ejercicio 3.0.15 ► Polinomio Grado Dos

Dada la función $f(x) = x^2 - x - 2$

- a) ¿Converge la fórmula $x_{n+1} = (x_n)^2 - 2$ a una raíz de $f(x) = 0$?
- b) Escribir una fórmula de Newton-Raphson que resuelva el problema del cálculo de los ceros de f .

Resolución

a) Observa la fórmula

$$x_{n+1} = (x_n)^2 - 2$$

, nos esta diciendo que se esta aplicando la iteración de punto fijo sobre la función de punto fijo $x = g(x) = x^2 - 2$.

Tomemos un intervalo cualquiera $I =]a, b[$ y veamos si $g(I) \subset I$

$$\begin{aligned} a &< g(a) = a^2 - 2 \Leftrightarrow 0 < a^2 - a - 2 \Leftrightarrow 0 < (a - 2) \cdot (a + 1) \Leftrightarrow a > 2 \quad \wedge \quad a < -1 \\ g(b) &= b^2 - 2 < b \Leftrightarrow b^2 - b - 2 < 0 \Leftrightarrow (b - 2) \cdot (b + 1) < 0 \Leftrightarrow -1 < b < 2 \end{aligned}$$

Tenemos:

$$g(a) = a^2 - 2 < b_{\max} = 2 \Rightarrow |a| < \sqrt{2} \quad g(b) = b^2 - 2 > a_{\min} = -\sqrt{2} \Rightarrow |b| > \sqrt{2 - \sqrt{2}}$$

Luego no existe ningún intervalo $]a, b[/ g(]a, b[) \subset]a, b[$. Luego la sucesión dada es divergente.

b) $f'(x) = 2x - 1$, el esquema es:

$$x_{n+1} = x_n - \frac{x_n^2 - x_n - 2}{2x_n - 1} = \frac{2x_n^2 - x_n - x_n^2 + x_n + 2}{2x_n - 1} = \frac{x_n^2 + 2}{2x_n - 1}; \forall n \in \mathbb{N}$$

Ejercicio 3.0.16 ► Sistema No Lineal

Escribir las ecuaciones del esquema de Newton-Raphson para resolver el sistema:

$$\begin{cases} x - 3x^2y = 0 \\ y - x^3 = 0 \end{cases}$$

Resolución

Tenemos el sistema de ecuaciones $\begin{cases} x - 3x^2y = 0 \\ y - x^3 = 0 \end{cases}$, que en general es de la forma:

$$\begin{pmatrix} g_1(x, y) \\ g_2(x, y) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Tiene como solución en el método Newton-Raphson:

$$\begin{pmatrix} x \\ y \end{pmatrix}_{n+1} = \begin{pmatrix} x \\ y \end{pmatrix}_n - \begin{pmatrix} \frac{\partial g_1}{\partial x} & \frac{\partial g_1}{\partial y} \\ \frac{\partial g_2}{\partial x} & \frac{\partial g_2}{\partial y} \end{pmatrix}_n^{-1} \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}_n$$

Por tanto, hemos de calcular la matriz inversa de la matriz jacobiana:

$$g_1(x, y) = x - 3x^2y \quad g_2(x, y) = y - x^3$$

La matriz jacobiana es:

$$A = \begin{pmatrix} 1 - 6xy & -3x^2 \\ -3x^2 & 1 \end{pmatrix}$$

Tenemos que: $\det(A) = 1 - 6xy - 9x^4$. Por tanto:

$$A^{-1} = \frac{1}{1 - 6xy - 9x^4} \cdot \begin{pmatrix} 1 & 3x^2 \\ 3x^2 & 1 - 6xy \end{pmatrix}$$

Entonces:

$$\begin{aligned}
 A^{-1} \cdot \begin{pmatrix} x - 3x^2y \\ y - x^3 \end{pmatrix} &= \frac{1}{1 - 6xy - 9x^4} \cdot \begin{pmatrix} 1 & 3x^2 \\ 3x^2 & 1 - 6xy \end{pmatrix} \cdot \begin{pmatrix} x - 3x^2y \\ y - x^3 \end{pmatrix} = \\
 &= \frac{1}{1 - 6xy - 9x^4} \cdot \begin{pmatrix} x - 3x^5 \\ 2x^3 - 3x^4y - 6xy^2 + y \end{pmatrix}
 \end{aligned}$$

Por tanto, en forma de sucesiones es:

$$\left\{ \begin{array}{l} x_{n+1} = x_n - \frac{x_n - 3x_n^5}{1 - 6x_n y_n - 9x_n^4} \\ y_{n+1} = y_n - \frac{2x_n^3 - 3x_n^4 y_n - 6x_n y_n^2 + y_n}{1 - 6x_n y_n - 9x_n^4} \end{array} \right\}; \forall n \in \mathbb{N}$$

Ejercicio 3.0.17 ► Esfera Sumergida

Se tienen una esfera de radio unidad cuya densidad es la cuarta parte de la del agua dulce y una pileta muy grande, de fondo plano, con un agujero de radio 0.5 unidades en el cual se ha quedado la esfera después de haber rodado por el fondo. Se empieza a llenar de agua la pileta.

- Obtener la ecuación que proporciona la altura de agua a la que hay que llenar la pileta para que la bola deje su agujero y empiece a flotar.
- Utilizar el método de Newton con estimador inicial 0.1660 para encontrar dicha altura. Dar 3 pasos en el esquema iterativo detallando los valores intermedios obtenidos.
- Convertir el problema en uno de punto fijo que se resolverá por el método de aproximaciones sucesivas.
- Aplicar la iteración de punto fijo con el mismo estimador inicial y dando el mismo número de pasos que en b). Valorar y justificar los resultados.

El volumen de un casquete esférico es:

$$V(z) = \pi R z^2 - \frac{\pi}{3} z^3$$

Resolución

a) Se tiene $Peso - E + \rho g h \pi r^2 = 0$. Luego:

$$\frac{4}{3}\pi R^3 \frac{\rho}{4} g + \pi \left(\frac{h^3}{3} + (c - R)h^2 + (c^2 - 2cR)h \right) \rho g + \rho g h \pi \left(\frac{R}{2} \right)^2 = 0$$

Operamos:

$$h^3 + 3(c - R)h^2 + 3\left(c^2 - 2cR + \frac{R^2}{12}\right)h + R^3 = 0$$

Sustituimos $c = 1 - \sin(\pi/3) = 0.1340$ $R = 1$:

$$h^3 - 2.5980h^2 - 0.5001h + 1 = 0$$

b) Definimos la función: $f(h) := h^3 - 2.5980h^2 - 0.5001h + 1$, tenemos que:

$$f'(h) = 3h^2 - 5.196h - 0.5001$$

Aplicamos Newton con $h_0 = 0.1660$

```
newton('x.^3-2.5980*x.^2-0.5001*x+1', '3*x.^2-5.196*x-0.5001', 0.1660, 0.1, 0.000001, 20)
```

La solución es 0.592360397221254, y se ha alcanzado en 6 iteraciones.

c) Para aplicar punto fijo despejamos h:

$$g(h) = 1.9996h^3 - 5.1950h^2 + 1 - 9996$$

d) Aplicamos el algoritmo de punto fijo:

```
pfijo('1.9996*x.^3-5.1950*x.^2+1.9996', 'x.^3-2.5980*x.^2-0.5001*x+1', 0.1, 0.1660, 0.000001, 20)
```

La solución es -Inf, y se ha alcanzado en 10 iteraciones.

En éste caso el método de punto fijo diverge, ya que:

$$g'(h) = 3 \cdot 1.9996h^2 - 2 \cdot 5.1950h \Rightarrow |g'(h)| > 1 \forall h \in]0, 1[$$

Ejercicio 3.0.18 ► Problema Partícula

Una partícula parte del reposo descendiendo sobre un plano inclinado empujada por su propio peso. El ángulo que forma el plano inclinado con la horizontal θ cambia con el tiempo con una velocidad constante ω , siendo $\theta = 0$ en el instante inicial. La partícula se encuentra en dicho instante sobre el eje de giro del plano. Planteando este problema en un sistema de coordenadas polares con origen en dicho eje se llega a la ecuación diferencial:

$$\frac{d^2 r}{dt^2} - r\omega^2 + g \sin(\omega t) = 0$$

que describe la variación de la posición de la partícula sobre la rampa. Esta ecuación admite una integral analítica cuyas constantes se fijan utilizando las condiciones iniciales de posición y velocidad nulas

$$r(t) = -\frac{g}{2\omega^2}(\sinh(\omega t) - \sin(\omega t))$$

Se tomará la gravedad $g = 10 \text{ m/s}^2$. Se supone que en un segundo la partícula ha recorrido 0.5m , lo que significa, con el sentido dado a los ejes, que $\omega = \frac{d\theta}{dt}$ es negativa. Nuestro objetivo es calcular el valor de ω correspondiente a ese desplazamiento.

1. Formular el problema como la búsqueda de la raíz de una ecuación $f(\omega) = 0$.
2. Variar ω empezando por -1 , evaluando la función f en los distintos valores hasta detectar un cambio de signo que permita definir un intervalo que contenga a la raíz buscada.
3. Tomando como estimador inicial -0.375 , iterar con el método de Newton hasta que la diferencia entre dos iteraciones consecutivas sea en módulo menor que 10^{-2} .
4. Reformular el problema $f(\omega) = 0$ en una ecuación de punto fijo $\omega = T(\omega)$.
Asumiendo que el valor de ω obtenido en el apartado anterior es la raíz buscada, estudiar la convergencia local del esquema $\omega = T(\omega)$.
5. Tomando como estimador inicial $\omega_0 = -0.375$, dar tres pasos en éste esquema.
6. Caso de que no se observen indicios claros de convergencia, relajar el esquema 4 tomando como factor de relajación el óptimo correspondiente a la raíz obtenida en el apartado 3.
Escribir el esquema resultante $\omega_{i+1} = T(\omega_i)$.
7. Tomando el mismo estimador inicial $\omega_0 = -0.375$, iterar con este esquema relajado hasta que la diferencia entre dos iteraciones consecutivas sea en módulo menor que 0.05 .

Resolución

1.) Sustituimos $r = 0.5$ para $t = 1$ y obtenemos la ecuación:

$$10 (\sinh \omega - \sin \omega) + \omega^2 = 0$$

2.) La tabla de valores con $f(\omega) = 10 (\sinh \omega - \sin \omega) + \omega^2$ es la siguiente:

ω	-1	-2	-0.5	-0.25
$f(\omega)$	-2.33	-23.1756	-0.1666	0.0104

Luego hay cambio de signo en el intervalo $[-0.5, -0.25]$, es decir; en dicho intervalo hay una raíz.

3.) Calculemos la sucesión de Newton en nuestro caso:

$$f'(\omega) = 10 (\cosh \omega - \cos \omega) + 2\omega$$

Por lo que la sucesión es:

$$\omega_{n+1} = \omega_n - \frac{10 (\sinh \omega_n - \sin \omega_n) + \omega_n^2}{10 (\cosh \omega_n - \cos \omega_n) + 2\omega_n}; \forall n \in \mathbb{N}$$

Debemos iterar dicha sucesión hasta que se cumpla:

$$|\omega_n - \omega_{n-1}| < 10^{-2}$$

Lo ponemos en una tabla:

n	0	1	2	3
ω_n	-0.375	-0.3214	-0.3025	-0.3000
$ \omega_n - \omega_{n-1} $		0.0536	0.0189	0.0025

Luego obtenemos con el método de Newton que la velocidad angular es $\omega = -0.3 \text{ rad/seg}$, con 3 iteraciones.

4.) Tomamos $T(\omega) = \omega + f(\omega)$

$$T(\omega) = \omega + 10 (\sinh \omega - \sin \omega) + \omega^2$$

Veamos si converge o diverge:

$$T'(\omega) = 1 + 10 (\cosh \omega - \cos \omega) + 2\omega$$

Entonces, $|T'(-0.3000)| = 1.3 > 1$. Luego el método diverge, por lo que nuestra raíz es un punto de repulsión.

5.) Planteamos la iteración:

$$\omega_{n+1} = T(\omega_n) = \omega_n + 10 (\sinh \omega_n - \sin \omega_n) + (\omega_n)^2 \omega_n + f(\omega_n) ; \forall n \in \mathbb{N}$$

Introducimos en Octave el siguiente código:

```

1 w = -0.375
2 for n=1:6
3     w=w+10*(sinh(w)-sin(w))+w^2
4 endfor

```

Y obtenemos los siguientes resultados:

```

w = -0.37500
w = -0.41016
w = -0.47194
w = -0.59962
w = -0.95882
w = -2.9807
w = -90.748

```

Claramente, viendo el resultado de la última iteración podemos concluir, como en el apartado anterior, que el método diverge.

6.) Nombramos α el factor de relajación a buscar para que el método converja. El esquema es:

$$\omega = (1 - \alpha) \omega + \alpha T(\omega) = T(\omega)$$

Para $\omega = -0.3000$

$$T'(-0.3000) = 10 (\cosh(-0.3000) - \cos(-0.3000)) + 0.4000 = 1.3$$

Luego:

$$\alpha = \frac{1}{1 - T'(-0.3)} = \frac{1}{1 - 1.3} = -3.333 \Rightarrow \alpha = -3.333$$

El esquema relajado es:

$$\begin{aligned}\omega_{n+1} &= 4.3333\omega_n - 3.3333 \cdot T(\omega_n) = T(\omega_n) \\ \omega_{n+1} &= 4.3333\omega_n - 3.333 \cdot (\omega_n + f(\omega_n)) \\ \omega_{n+1} &= \omega_n - 3.3333 \cdot f(\omega_n)\end{aligned}$$

Iteramos dicho proceso hasta que $|\omega_n - \omega_{n-1}| < 10^{-2}$

n	0	1	2	3	4
ω_n	-0.375	-0.2578	-0.2890	-0.2992	-0.3000
$ \omega_n - \omega_{n-1} $		0.1172	0.0312	0.0102	0.0008

Y ahora al relajar el método obtenemos un comportamiento razonable.

Ejercicio 3.0.19 ► Problema Reynolds-Colebrook

Se considera un flujo turbulento en una tubería, sea $\mathcal{R} \geq 3500$ el número de Reynolds asociado a su diámetro D . Se desea calcular el coeficiente λ de pérdida de carga lineal utilizando la relación de Colebrook:

$$\lambda^{-\frac{1}{2}} = -2 \log_{10} \left(\frac{\epsilon}{3.71 \cdot D} + \frac{2.51}{\mathcal{R} \lambda^{\frac{1}{2}}} \right)$$

donde:

- λ es el coeficiente de pérdida de carga lineal o factor de fricción de Moody. Es adimensional.
- ϵ es la rugosidad de la tubería.

Para obtener una estimación inicial de λ se usará el valor suministrado por la fórmula empírica de Hermann

$$\lambda_0 = 0.0054 + 0.395 \mathcal{R}^{-0.3}$$

El objetivo del ejercicio es calcular aproximadamente λ para los siguientes valores de \mathcal{R} y de ϵ/D

\mathcal{R}	ϵ/D
10^4	0.05
10^5	0.003
10^6	0.003

Se utilizarán para ello los siguientes métodos:

1. Método de las aproximaciones sucesivas.
 - 1.1 Efectuar un análisis previo de la existencia y unicidad de la solución y de la convergencia de la sucesión aproximadamente.
 - 1.2 Hacer una estimación a priori del número de iteraciones necesarias para obtener la solución con un error $\epsilon < 10^{-6}$ para cada pareja de valores $(\mathcal{R}, \epsilon/D)$.
 - 1.3 Resolver la ecuación.
 - 1.4 Efectuar estimaciones **a priori** y **a posteriori** del error en cada caso comparando los resultados.
 - 1.5 Efectuar una aceleración de la convergencia por el método Δ^2 de Aitkens en el caso $(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$.
 - 1.6 Aplicar el método de Steffensen al caso $(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$.
2. Método de Wegstein con estimadores iniciales sugeridos en cada caso, por el apartado anterior y test de parada $\epsilon < 10^{-6}$ comparando el número de iteraciones.
3. Método de Newton-Raphson.
 - 3.1 Hallar un intervalo que contenga a la raíz λ^* con el estimador inicial λ_0 y en el que el método de Newton-Raphson converja independientemente del estimador inicial.
 - 3.2 Determinar la raíz λ^* con el estimador inicial λ_0 obtenido mediante la fórmula de Hermann, y test de parada $\epsilon < 10^{-6}$.
4. Con el mismo estimador inicial que en el apartado 3.2, determinar la raíz λ^* en el caso $(\mathcal{R}, \epsilon/D) = (10^5, 0.003)$ utilizando:
 - 4.1 El método Illinois.
 - 4.2 El método Pegasus.

Resolución

1. Para simplificar cálculos tomamos $x = \frac{1}{\sqrt{\lambda}} > 0$ con lo que reescribimos la ecuación del enunciado como:

$$x = -2 \log_{10} \left(\frac{\epsilon}{3.71 \cdot D} + \frac{2.51}{\mathcal{R}} \cdot x \right)$$

Veamos el caso $(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$. Tenemos:

$$x = -2 \log_{10} (0.013477 + 0.000251x) = F_1(x)$$

El dominio de la función es $x > -53.69322709$, que es cuando se cumple que lo que hay dentro del logaritmo en base 10 es estrictamente positivo. Como $\lim_{x \rightarrow -53.69322709} F_1(x) = +\infty$ tenemos que la función tiene una asíntota vertical $x = -53.69322709$

$$F'_1(x) = -\frac{0.000218015}{0.013477 + 0.000251x} < 0 \quad \forall x > -53.69322709$$

Luego la función es Decreciente, además, $F'_1(x) = -1$, por tanto:

$$|F'_1(x)| < 1 \quad \forall x > -53.69322709$$

Con todo esto concluimos que F_1 satisface el teorema del punto fijo, por lo que el Método de Punto Fijo siempre converge sea cual sea el punto inicial que se tome. Aplicamos la fórmula de Hermann para determinar el mejor punto inicial de partida en el método de Punto Fijo, que viene en el enunciado del problema:

$$\lambda_0 = 0.0054 + 0.395\mathcal{R}^{-0.3} \rightarrow \lambda_0 = 0.03032281$$

Luego como hemos de determinar x_0 :

$$x_0 = \frac{1}{\sqrt{\lambda_0}} = 5.742688256 \in [-52.82464143, +\infty[$$

Por tanto, es un valor inicial válido. Ahora aplicamos Bolzano:

$$F_1(3) \cdot F_1(\lambda_0) < 0 \rightarrow \exists x^* \in [3, \lambda_0] / F_1(x^*) = 0$$

Luego aplicaremos el método del punto fijo en el intervalo $I_1 = [3, \lambda_0]$ para encontrar nuestra raíz aproximada.

1.2 Para calcular el número k de iteraciones necesarias para que $\|x^* - x_k\| < \epsilon = 10^{-6}$ a priori es necesario la constante de Lipschitz de F_1 en I_1 . La constante L viene dada por:

$$L = \max_{x \in I_1} |F'_1(x)| = |F'_1(3)| = 0.015320801$$

Ahora consideramos:

$$x_0 = 5.742688256 \quad x_1 = F(x_0) = 3.652554646$$

Utilizamos el siguiente resultado teórico:

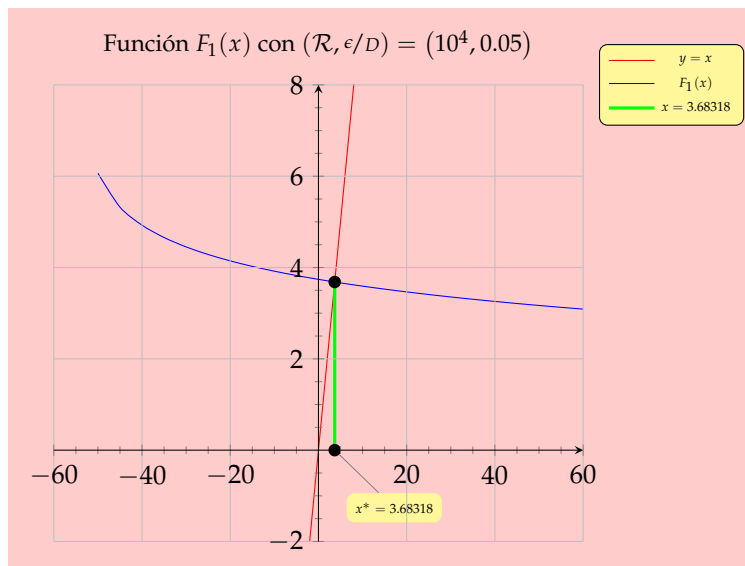


Figura 30: Función Reynolds Punto Fijo

Corolario 3.0.1 ► Número Iteraciones

El número k de iteraciones necesarias para que $\|x_k - x^*\| < \epsilon$ satisface la siguiente desigualdad:

$$k \geq \frac{\ln \left(\frac{\epsilon(1-L)}{\|x_1 - x_0\|} \right)}{\ln L}$$

Vamos sustituyendo nuestros datos:

$$\begin{aligned} \|x_1 - x_0\| &= \|3.652554646 - 5.742688256\| = 2.09013361 \\ \epsilon(1-L) &= 10^{-6}(1 - 0.015320801) = 0.00000984679 \\ \ln \left(\frac{\epsilon(1-L)}{\|x_1 - x_0\|} \right) &= \ln \frac{0.00000984679}{2.09013361} = \ln(0.00000471108) \\ k &\geq \frac{\ln(0.00000471108)}{\ln(0.015320801)} \approx 3.48643 \rightarrow k = 4 \end{aligned}$$

Por tanto, necesitamos **al menos** 4 iteraciones para que el método de punto fijo tenga un error menor que $\epsilon = 10^{-6}$.

1.3 Vamos a resolverlo aplicando nuestro algoritmo de punto fijo. Con 4 iteraciones vemos que no son suficientes:

```
pfijo('2*log10(0.013477+0.000251*x)', 'x+2*log10(0.013477+0.000251*x)', 3, 5.742688256, 10^(-6), 4)
```

Insuficientes iteraciones

Vemos que son suficientes 5 iteraciones:

```
pfijs('2*log10(0.013477+0.000251*x)', 'x+2*log10(0.013477+0.000251*x)', 3, 5,
742688256, 5.742688256, 10^(-6), 5)
La solución es 3.683185769546551, y se ha alcanzado en 6 iteraciones.
```

Pero esta no es la solución del problema, hemos obtenido x^* y nosotros buscamos λ^* , sabemos que:

$$x^* = \frac{1}{\sqrt{\lambda^*}} \rightarrow \lambda^* = \frac{1}{(x^*)^2} = \frac{1}{(3.683185769546551)^2} \approx 0.07371447078918$$

Por tanto, nuestra solución aproximada al problema utilizando el método del punto fijo es $\lambda^* \approx 0.07371447078918$

1.4

Estimación a priori: Tomando x_5 como solución es:

$$|x^* - x_5| \leq \frac{L^5}{1 - L} \cdot |x_1 - x_0| \approx 1.79179 \cdot 10^{-9}$$

Estimación a posteriori: Tomando x_5 como solución es:

$$|x^* - x_5| \leq \frac{L}{1 - L} \cdot |x_5 - x_4| < 1.67893 \cdot 10^{-9}$$

Para obtener los valores de x_4 y x_5 debes correr el algoritmo de punto fijo, por ello son a posteriori.

1.5

Para solucionarlo te adjunto el código realizado con Octave del método de Aitken:

```
1 function [] =aitken(f,x0,epsilon,max1)
2   X=zeros(1,max1+1);
3   Z=zeros(1,max1+1);
4   X(1)=x0;
5   X(2)=feval('colebrook1',X(1));
6   X(3)=feval('colebrook1',X(2));
7   Z(3)=X(1)-(X(2)-X(1))^2/(X(3)-2*X(2)+X(1));
8   for k=4:max1+1
9     X(k)=feval('colebrook1',X(k-1));
10    denom=X(k)-2*X(k-1)+X(k-2);
11    if denom==0
12      'Lo siento, division por cero'
13      break;
14    else
```

```

15         Z(k)=X(k-2)-(X(k-1)-X(k-2))^2/denom;
16     endif
17     err=abs(Z(k-1)-Z(k));
18     relerr=err/(abs(Z(k-1))+epsilon);
19     if (err<epsilon) or (relerr<epsilon)
20         break;
21     endif
22 endfor

23
24 printf ('La solucion es %.15f, y se ha alcanzado en %d iteraciones.\n',
25         X(k), k);
26 printf('Con error absoluto ');
27 err
28 printf('y error relativo ');
29 relerr
30 clear;
31 end

```

aitken.m

Se ha aplicado en nuestro caso particular con la función:

```

1 function f = colebrook1(x)
2     f=x+2*log10(0.013477+(0.000251)*x);
3 end

```

colebrook1.m

Consideramos el caso $(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$

Con los datos dados en el problema lo aplicamos a nuestro código con los siguientes resultados:

```

aitken('colebrook1',5.742688256,1e-7,6)
La solucion es 3.683185877452450, y se ha alcanzado en 5 iteraciones.
Con error absoluto err = 1.8466e-009
y error relativo relerr = 5.0135e-010

```

Como vemos, obtenemos una solución muy parecida a la que obtuvimos con el método de punto fijo. Rehaciendo el cambio de variable obtenemos que nuestra aproximación a la solución es $\lambda^* \approx 0.07371446651$

Los otros 2 casos te los dejo como ejercicio para que practiques y entiendas mejor el problema.

1.6

Tan sólo hemos de aplicar el método de Steffensen con los datos del problema, dándonos el siguiente resultado:

```
steffensen('x+2*log10(0.013477+(0.000251)*x)',5.742688256,3,5.742688256,1e-7,5)
```

La solución es 3.683185770874608, y se ha alcanzado en 2 iteraciones.

Apreciamos que los resultados mediante éste método son muy buenos, tan sólo hacen falta 2 iteraciones para obtener lo deseado.

Veamos el caso $(\mathcal{R}, \epsilon/D) = (10^5, 0.003)$.

1.1

Tenemos:

$$x = -2 \log_{10} (0.00080862 + 0.0000251x) = F_2(x)$$

Veamos el dominio:

$$0.00080862 + 0.0000251x > 0 \Leftrightarrow x > -\frac{0.00080862}{0.0000251} \Leftrightarrow x > -32.21513944$$

Además, $\lim_{x \rightarrow -32.21513944} F_2(x) = +\infty$, luego $F_2(x)$ tiene una asíntota vertical de ecuación $x = -32.21513944$. Y su dominio es el intervalo $[-32.21513944, +\infty[$.

$$F_2'(x) = -\frac{0.00002180158}{0.0008062 + 0.0000251x}$$

Veamos el dominio de la derivada:

$$0.0008062 + 0.0000251x = 0 \rightarrow x = \frac{-0.0008062}{0.0000251} \rightarrow x = -32.34262948$$

Luego el dominio de la derivada es todos los reales menos -32.34262948 . Además, la derivada siempre es negativa en el dominio de $F_2(x)$, por lo que $F_2(x)$ es estrictamente decreciente. Ahora:

$$\begin{aligned} F_2'(x) = -1 \rightarrow 0.00002180158 = 0.0008062 + 0.0000251x \rightarrow x &= \frac{0.00002180158 - 0.0008062}{0.0000251} \\ \rightarrow x &= -31.25093307 \end{aligned}$$

Concluimos pues:

$$|F_2'(x)| < 1 \quad \forall x \in [-31.25093307, +\infty]$$

Por tanto, $F_2(x)$ cumple el Teorema del punto fijo en cualquier subintervalo cerrado de $[-31.25093307, +\infty]$.

La estimación inicial por la fórmula de Hermann es:

$$\lambda_0 = 0.0054 + 0.395 \cdot \frac{1}{10^{5 \cdot 0.3}} = 0.017891 \rightarrow x_0 = \frac{1}{\sqrt{\lambda_0}} \rightarrow x_0 = 7.476231341$$

Como $x_0 \in [-31.25093307, +\infty]$, podemos asegurar que este estimador inicial es válido. Aplicamos Bolzano:

$$F_2(5) \cdot F_2(x_0) < 0 \rightarrow \exists x^* \in [5, x_0] / F_2(x^*) = 0$$

Por tanto, en el intervalo $I_2 = [5, x_0]$ la función F_2 tiene una raíz.

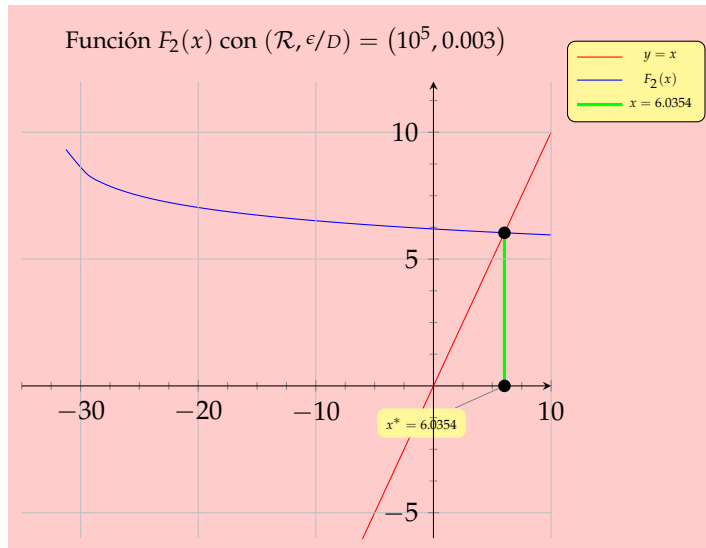


Figura 31: Función Reynolds2 Punto Fijo

1.2

Calculemos la constante de Lipschitz de F_2 en I_2

$$L = \max_{x \in I_2} |F_2'(x)| = |F_2'(5)| = 0.02334$$

Ahora, $x_1 = F_2(x_0) = 6.003242924$, por lo que para que se cumpla $|x^* - x_k| < \epsilon = 10^{-6}$ tenemos:

$$k \geq \frac{\ln \left(\frac{\epsilon(1-L)}{\|x_1 - x_0\|} \right)}{\ln L} = \frac{\ln \left(\frac{10^{-6}(1-0.02334)}{\|6.003242924 - 7.476231341\|} \right)}{\ln 0.02334}$$

$$k \geq \frac{-14.22642052}{-3.757586652} \approx 3.786048524 \rightarrow k = 4$$

Luego necesitamos, al menos, 4 iteraciones para que el método del punto fijo converja.

1.3

Aplicamos el método de punto fijo para $x_0 = 7.476231341$ y $\epsilon = 10^{-6}$

```
pfxj('x+2*log10(0.00080862+0.0000251*x)', 'x+2*log10(0.00080862+0.0000251*x)',
5, 7.476231341, 7.476231341, 10^(-6), 4)
Insuficientes iteraciones
```

Como era de esperar, el método diverge para 4 iteraciones, y converge a partir de 5 iteraciones:

```
pfijo('-2*log10(0.00080862+0.0000251*x)', 'x+2*log10(0.00080862+0.0000251*x)',
5,7.476231341,7.476231341,10^(-6),5)
La solucion es 6.035360228993012, y se ha alcanzado en 6 iteraciones.
```

Si deshacemos el cambio de variable, obtenemos que nuestra solución es $\lambda^* \approx 0.02745323999234$.

1.4

Estimación a priori: Tomando x_5 como solución es:

$$|x^* - x_5| \leq \frac{L^5}{1-L} \cdot |x_1 - x_0| \approx 1.044297 \cdot 10^{-8}$$

Estimación a posteriori: Tomando x_5 como solución es:

$$|x^* - x_5| \leq \frac{L}{1-L} \cdot |x_5 - x_4| < 0.919397 \cdot 10^{-10}$$

Para obtener los valores de x_4 y x_5 debes correr el algoritmo de punto fijo, por ello son a posteriori.

2

$$(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$$

Tenemos $x_0 = 5.742688256$, $\epsilon = 10^{-7}$, aplicamos el algoritmo de Wegstein:

```
wegstein('-2*log10(0.013477+(0.000251)*x)',5.742688256,1e-7,2)
La solucion es 3.683185771155717, y se ha alcanzado en 2 iteraciones.
Con un error de 6.06449e-08
```

Por tanto, $\lambda^* \approx 0.07371447557970$

$$(\mathcal{R}, \epsilon/D) = (10^5, 0.003)$$

Tenemos $x_0 = 7.476231341$, $\epsilon = 10^{-7}$, aplicamos el algoritmo de Wegstein:

```
wegstein('-2*log10(0.00080862+0.0000251*x)',7.476231341,1e-7,3)
La solucion es 6.035360237535613, y se ha alcanzado en 3 iteraciones.
Con un error de 0.000000000000000e+00
```

Por tanto, $\lambda^* \approx 0.02745323991462$

$$(\mathcal{R}, \epsilon/D) = (10^6, 0.003)$$

Tenemos $x_0 = 9.260717036$, $\epsilon = 10^{-7}$, aplicamos el algoritmo de Wegstein:

```
wegstein('-2*log10(0.00080862+0.00000251*x)',9.260717036,1e-7,3)
La solucion es 6.168038299704970, y se ha alcanzado en 2 iteraciones.
Con un error de 1.330255905429567e-10
```

Por tanto, $\lambda^* \approx 0.02628487260934$

3

$$(\mathcal{R}, \epsilon/D) = (10^4, 0.05)$$

Tenemos la función $f_1(x) = x + 2 * \log_{10}(0.013477 + 0.000251 * x)$, que tiene una raíz en el intervalo $I_1 = [3, 5.742688256]$ Para tomar un buen estimador lo que haremos es saber en cuál de los 2 extremos de I_1 la función f_1 y su f'' tienen el mismo signo.

$$f_1'(x) = 1 + \frac{0.000218015}{0.013477 + 0.000251 * x}$$

$$f_1''(x) = -\frac{5.4721765 \cdot 10^{-8}}{(0.013477 + 0.000251 * x)^2} < 0; \forall x \in I_1$$

En éste caso da igual qué extremo de los dos tomemos, porque la segunda derivada siempre tiene signo negativo, así que tomaremos el más cercano a la raíz; en éste caso nuestro estimador (valor) inicial va a ser $x_0 = 3$, para aplicar el método de Newton.

3.2

```
newton('x+2*log10(0.013477+0.000251*x)',
'1+(0.000218015/(0.013477+0.000251*
x))',3,3,5,1e-7,10)
```

El algoritmo tiene un coste en tiempo de:

Elapsed time is 0.022152 seconds.

La solucion es 3.683185771155717, y se ha alcanzado en 4 iteraciones.

Apreciamos que no hay mucha diferencia con los anteriores métodos, obtenemos muy buena aproximación con a penas iteraciones.

Por tanto, $\lambda^* \approx 0.07371447072477$.

Te dejo como ejercicios los restantes casos.

4

$$(\mathcal{R}, \epsilon/D) = (10^5, 0.003)$$

Nuestra función es:

$$f_2(x) = x + 2 * \log_{10}(0.00080862533 + 0.0000251 * x)$$

Su única raíz esta en el intervalo $I_2 = [5, 7.476231431]$.

4.1

Aplicamos el método Illinois:

```
illinois('x+2*log10(0.00080862533+0.0000251*x)',
5,7.476231431,1e-9,1e-8,30)
```

La solución es 6.035355517748460, y se ha alcanzado en 18 iteraciones.
 Con un error de 7.386344e-09
 Con $f(\text{raiz}) = -5.036084e-09$

4.2

Hacemos lo mismo pero con el método Pegasus:

alpha = 0.553348381381023
 alpha = 0.500032212843228
 alpha = 0.500012332434644
 alpha = 0.500008620338480
 alpha = 0.500004041974341
 alpha = 0.500002215625791
 alpha = 0.500001090307565
 alpha = 0.500000558175124
 alpha = 0.500000277980678
 alpha = 0.500000139818658
 alpha = 0.500000069839944
 alpha = 0.500000034971973
 alpha = 0.500000017481646
 alpha = 0.500000008744077
 alpha = 0.500000004371767
 alpha = 0.500000002186087
 alpha = 0.500000001093026
 pegasus('x+2*log10(0.00080862533+0.0000251*x)',
 5,7.4776231431,1e-9,1e-8,30)
 La solución es 6.035355531468648, y se ha alcanzado en 17 iteraciones.
 Con un error de 1.319402e-08
 Con $f(\text{raiz}) = 8.995652e-09$

Con el método Pegasus obtenemos resultados parecidos al de Illinois, tan sólo hemos ganado una iteración. Esto se puede apreciar viendo los valores que hemos obtenido con el método Pegasus, que son casi todos 0.5 como el que se toma en el método Illinois.

Ejercicio 3.0.20 ► Método Savitsky

Método de Savitsky En el cálculo de la resistencia hidrodinámica de lanchas planeadoras por el método de Savitsky, es crucial la estimación del ángulo de inclinación del barco en la dirección de la marcha τ . Esto es así porque existen fórmulas empíricas que relacionan τ con todas las variables que intervienen en el diseño.

La ecuación base del método y cuya resolución es el objetivo del correspondiente algoritmo es:

$$\Delta \left(\frac{1 - \sin \tau \sin(\tau + \epsilon)c}{\cos \tau} - f \sin \tau \right) + D_f(a - f) = 0$$

Las variables c, D_f dependen de τ , dependencia que no es expresable mediante funciones elementales pero que se puede describir a través de las siguientes relaciones empíricas.

$$c = LCG - C_p \lambda b \quad (3.1)$$

$$D_f = \frac{\rho V M^2 \lambda b^2 (C_F + \Delta C_F)}{\cos \beta} \quad (3.2)$$

$$C_p = 0.75 - \frac{1}{5.21 \frac{C_V^2}{\lambda^2} + 2.39} \quad (3.3)$$

$$C_V = \frac{V}{\sqrt{gb}} \quad (3.4)$$

donde λ y C_F se obtienen de las siguientes ecuaciones:

- Coeficiente de empuje para ángulo de astilla muerta cero.

$$C_{L0} - 0.0065\beta(C_{L0})^{0.6} - C_{L\beta} = 0 \quad (3.5)$$

donde $C_{L\beta}$ y C_{L0} son coeficientes de empuje adimensionalizados relativos a un ángulo de astilla muerta igual a β y 0 respectivamente, y donde:

$$C_{L\beta} = \frac{\Delta}{\frac{1}{2}\rho V^2 b^2}$$

- Línea de fricción de Schoenherr para flujo turbulento.

$$F(\mathcal{R}, C_F) = \frac{0.242}{\sqrt{C_F}} - \log_{10}(\mathcal{R} \cdot C_F) \quad (3.6)$$

donde \mathcal{R} es el número de Reynolds.

- Relación entre la eslora mojada y la manga.

$$\frac{\tau^{1.1}}{C_{L0}} \left(0.012\sqrt{\lambda} + 0.0055 \frac{\lambda^{0.25}}{C_V^2} \right) - 1 = 0 \quad (3.7)$$

Existen además una ecuación que relaciona el número de Reynolds \mathcal{R} con las otras variables:

$$\mathcal{R} = \frac{V_M \lambda b}{\nu}$$

y una última ecuación que relaciona V_M con V

$$V_M = kV \quad (3.8)$$

donde $k \in [0.8, 1.0]$ se obtiene mediante unas curvas en función de τ , β , y λ .

El resto de las variables se incluye a continuación con una pequeña definición que permita situarse correctamente en el problema global y una figura donde se aclara su significado físico.

- Δ Peso del buque en libras.
- D_f Componente viscosa de la resistencia (libras). Se supone que actúa paralelamente a la línea de la quilla.
- τ Ángulo de trimado, en grados.
- ϵ Inclinação de la línea de empuje relativa a la quilla (grados).
- CG Centro de gravedad.
- LCG Distancia longitudinal del centro de gravedad desde la popa medida a lo largo de la quilla (pies).
- a Distancia entre D_f y CG, medida normalmente a D_f pies.
- T Empuje del propulsor (libras).
- N Resultante de las fuerzas de presión actuando normalmente a la base (libras).
- f Distancia entre T y CG medida normalmente a la línea de ejes en pies.
- c Distancia entre N y CG medida normalmente a N pies.
- β Ángulo transversal a que da lugar la astilla muerta (grados).
- b Manga (pies)
- V Velocidad (pies/segundo).
- d Calado de la quilla en la popa (pies).

Enunciado

En un momento del proceso numérico asociado al método de Savisky se debe resolver la ecuación:

$$C_{L0} - 0.0065\beta(C_{L0})^{0.6} - C_{L\beta} = 0$$

donde $C_{L0} > 0$ es la incógnita y $C_{L\beta} > 0$, con $0^\circ < \beta < 20^\circ$.

El objetivo es analizar la existencia y unicidad de la solución de la ecuación, determinándola después por varios métodos cuyos comportamientos se compararán.

1. Buscar una escritura conveniente de la ecuación del tipo:

$$C_{L0} = T(C_{L0})$$

que permita demostrar la existencia y unicidad de la solución para valores estrictamente positivos de $C_{L\beta}$ y β .

2. ¿Para qué valores de C_{L0} será necesariamente convergente el método de punto fijo asociado a 1.?
3. Se supone que $C_{L\beta} = 0.07$ y $\beta = 20^\circ$
 - a) Resolver la ecuación por el método de las aproximaciones sucesivas seleccionando un estimador inicial que cumpla con los valores obtenidos en 2. con test de parada $\epsilon < 10^{-7}$.
 - b) Usar el método de Wegstein con el mismo estimador inicial y test de parada y comparar el número de iteraciones.
 - c) Determinar un intervalo que contenga a la raíz de la ecuación y usar para determinarla:
 - El método Illinois.
 - El método Pegasus.
 - d) Comprobar el mejor comportamiento del segundo método respecto al primero y compararlo con el de Wegstein.

Resolución

1. Vamos a convertir la ecuación en una de punto fijo, tenemos:

$$C_{L0} - 0.0065\beta(C_{L0})^{0.6} - C_{L\beta} = 0$$

Se consigue fácilmente despejando el primer C_{L0}

$$C_{L0} = 0.0065\beta(C_{L0})^{0.6} + C_{L\beta} \quad (3.9)$$

$$T(C_{L0}) = 0.0065\beta(C_{L0})^{0.6} + C_{L\beta} \quad (3.10)$$

Para no liarnos vamos a considerar $x = C_{L0}, \alpha = C_{L\beta}, k_1 = 0.0065, \beta > 0$, de forma que:

$$T_{\alpha, \beta}(x) = \alpha + k_1 \beta x^{\frac{3}{5}}$$

Veamos cuándo $T_{\alpha, \beta}(x) = 0$, es decir; cuándo corta al eje OX.

$$x = \left(-\frac{\alpha}{k_1 \beta} \right)^{\frac{5}{3}}$$

Observa que $x < 0$ para $\alpha, \beta, k_1 \in \mathbb{R}^+$ y $T_{\alpha, \beta}(0) = \alpha$. Además,

$$\lim_{x \rightarrow +\infty} T_{\alpha, \beta}(x) = +\infty$$

Derivamos para estudiar su crecimiento:

$$T'_{\alpha, \beta}(x) = \frac{3}{5} k_1 \beta x^{-\frac{2}{5}} = \frac{3}{5} k_1 \beta \frac{1}{x^{\frac{2}{5}}} > 0$$

$$\lim_{x \rightarrow 0^+} T'_{\alpha, \beta}(x) = +\infty$$

Luego $T_{\alpha, \beta}(x)$ son todas Estrictamente Crecientes. Volvamos a derivar para estudiar su concavidad.

$$T''_{\alpha, \beta}(x) = -\frac{6}{25} k_1 \beta x^{-\frac{7}{5}} < 0$$

Por tanto, $T_{\alpha, \beta}(x)$ son cóncavas hacia abajo. Todas parten de $(0, \alpha)$ con tangente vertical. Dibujamos la gráfica de una de las funciones para parámetros particulares.

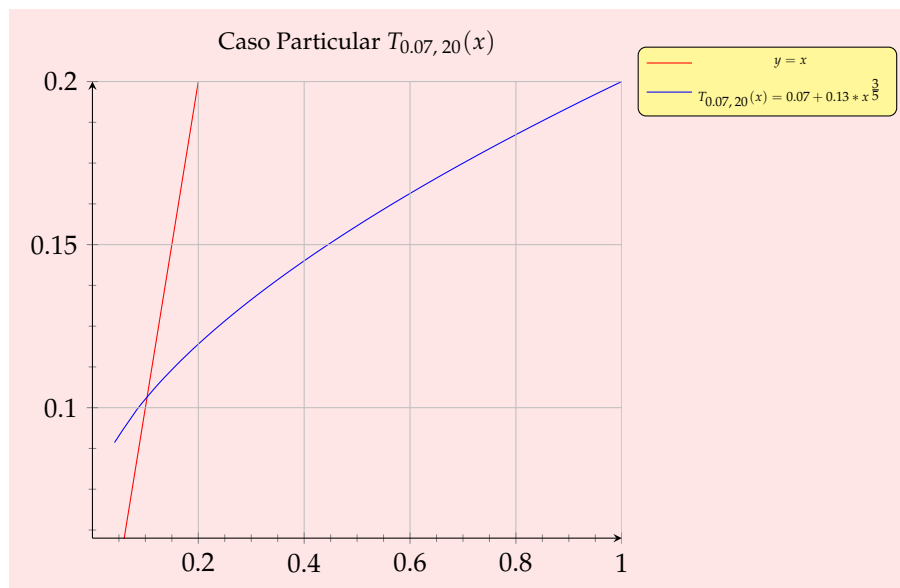


Figura 32: Caso particular para $(\alpha, \beta) = (0.07, 20)$

2. Demostremos la existencia y unicidad de la raíz x^*

$$|T'_{\alpha,\beta}(x)| = T'_{\alpha,\beta}(x) < 1 \Leftrightarrow \frac{3}{5}k_1\beta x^{-\frac{2}{5}} < 1 \Leftrightarrow \frac{3}{5}k_1\beta x^{-\frac{2}{5}} < 1$$

$$\Leftrightarrow x^{-\frac{2}{5}} < \frac{5}{3 \cdot k_1 \cdot \beta} \Leftrightarrow x^{\frac{2}{5}} > \frac{3 \cdot k_1 \cdot \beta}{5} \Leftrightarrow x > \left(\frac{3 \cdot k_1 \cdot \beta}{5} \right)^{\frac{5}{2}}$$

Sustituimos $k_1 = 0.0065 \rightarrow \frac{3 \cdot k_1}{5} = \frac{3 \cdot 0.0065}{5} = 0.0039$ y tenemos:

$$x > (0.0039 \cdot \beta)^{\frac{5}{2}}$$

Por tanto, la familia de funciones $T_{\alpha,\beta}$ son contractivas en el intervalo

$$I_1 = [(0.0039 \cdot \beta)^{\frac{5}{2}}, +\infty[$$

Por tanto, cumplen el Teorema del Punto Fijo, aseguramos pues, que el método del punto fijo converge y que las soluciones $x_{\alpha,\beta}^*$ de cada $T_{\alpha,\beta}$ están contenidas en cualquier intervalo cerrado contenido en su respectivo I_1 .

3. Vamos a tomar, por ejemplo, $\beta = 20, \alpha = 0.07$, puedes probar con cualesquiera otros valores. Por tanto; $I_1 = [1.69916872 \cdot 10^{-3}, +\infty[$. Tomamos, por ejemplo, el estimador inicial $x_0 = 0.02$ y aplicamos el método:

```
pfijo('0.07+0.13*x.^(3/5)', 'x-0.07+0.13*x.^(3/5)', 0, 1, 0.02, 1e-7, 15)
```

La solución es 0.103296123978309, y se ha alcanzado en 11 iteraciones.

Fíjate en las gráficas de convergencia que aparecen al ejecutar el método, que la velocidad de convergencia es lineal, no cuadrática. Por lo que podemos mejorar los resultados, en éste caso los intentaremos mejorar con el método de Wegstein.

4. Aplicamos el método de Wegstein a la misma función:

```
wegstein('0.07+0.13*x.^(3/5)', 0.02, 1e-7, 15)
```

La solución es 0.103296132315191, y se ha alcanzado en 3 iteraciones.

Con un error de 2.890849435527310e-08

Claramente hemos mejorado, hemos necesitado tan sólo 3 iteraciones.

5. Vamos a aplicar los métodos Illinois y Pegasus para

$$f_{\alpha,\beta}(x) = x - 0.07 + 0.13x^{\frac{3}{5}}$$

con $\delta = 10^{-7}$ y $\epsilon = 10^{-6}$, con un número máximo de iteraciones de 30, en el intervalo $[0.02, 1]$

Método Illinois

```
illinois('x-0.07+0.13*x.^(3/5)',0.02,1,1e-7,1e-6,30)
```

La solución es 0.048773734273443, y se ha alcanzado en 14 iteraciones.

Con un error de 1.083560e-06

Con $f(\text{raiz}) = -9.110032e-07$

Método Pegasus

```
pegasus('x-0.07+0.13*x.^(3/5)',0.02,1,1e-7,1e-6,30)
```

$\alpha = 0.949177854265324$

$\alpha = 0.523590708370158$

$\alpha = 0.523047478743633$

$\alpha = 0.500343770075646$

$\alpha = 0.500205654511487$

$\alpha = 0.500075176684662$

$\alpha = 0.500039349670936$

$\alpha = 0.500018320371238$

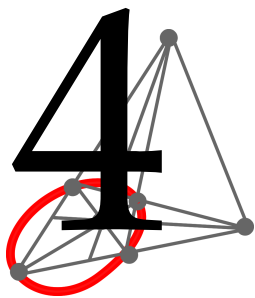
$\alpha = 0.500009266335892$

La solución es 0.048775049973069, y se ha alcanzado en 9 iteraciones.

Con un error de 8.832061e-07

Con $f(\text{raiz}) = 7.482342e-07$

El método Pegasus es más rápido que el método de Illinois, éste último es el más lento de los 4 que hemos utilizado. El mejor en cuanto a rapidez ha sido el de Wegstein.



MÉTODOS PARA POLINOMIOS

4.1 Propiedades De Los Polinomios

Definición 4.1.1 ► Ecuación Algebraica

Sea el polinomio de grado n $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \in \mathbb{R}[x]$, se define Ecuación Algebraica a la ecuación $P(x) = 0$.

Teorema 4.1.1 ► Teorema Fundamental Del Álgebra

Todo polinomio $P(x) \in \mathbb{C}[x]$ de grado $n \geq 1$ tiene al menos una raíz real o dos complejas (la compleja y su conjugada).

Nota 4.1.1 ► Galois

Las ecuaciones algebraicas se pueden resolver mediante cálculos algebraicos siempre que $\partial P \leq 4$, para grados a partir de 5 no se pueden resolver algebraicamente, es necesario utilizar la *Teoría de Resolubilidad de Galois*. De todas formas, también podemos resolver aproximadamente las raíces de los polinomios de cualquier grado con los métodos que puedes ver aquí, pero lo normal es que no podamos resolver las ecuaciones algebraicas a partir de grado 5 de forma exacta con los métodos vistos aquí ^a.

^aPuedes consultar sobre la teoría de Galois en [Galois Theory](#)

Teorema 4.1.2 ► División De Polinomios

Sean $P(x), Q(x) \in \mathbb{R}[x]$ con $\partial P \geq \partial Q$. Entonces: $\exists! C(x), R(x) \in \mathbb{R}[x]$, $\partial C = \partial P - \partial Q$, $\partial R < \partial Q$ tales que:

$$P(x) = C(x) \cdot Q(x) + R(x)$$

$C(x)$ es el Cociente y $R(x)$ es el Resto.

Definicion 4.1.2 ► Polinomio Divisor

Si en el teorema anterior se cumple que $R(x) = 0$, se dice que $Q(x)$ es un Divisor de (o Divide a) $P(x)$, y se denota por $Q(x)|P(x)$

Corolario 4.1.1 ► El Resto

Si $P(x) \in \mathbb{R}[x]$ y $r \in \mathbb{R} \implies \exists! C(x) \in \mathbb{R}[x]$, $\partial C = \partial P - 1$, $P(x) = (x - r)C(x) + P(r)$

Luego tenemos que:

$$P(r) = 0 \Leftrightarrow (x - r)|P(x)$$

Se tiene como consecuencia la multiplicidad de raíces:

Definicion 4.1.3 ► Multiplicidad De Una Raíz

Sea $P(x) \in \mathbb{R}[x]$, $r \in \mathbb{R}$. Se dice que r es de Multiplicidad m de P si:

$$P(r) = P'(r) = P''(r) = \dots = P^{(m-1)}(r) = 0, P^{(m)}(r) \neq 0$$

Definicion 4.1.4 ► MCD Polinomios

Sean $P(x), Q(x) \in \mathbb{R}[x]$ se define el Máximo Común Divisor de ambos polinomios al polinomio $S(x) \in \mathbb{R}[x]$ que cumple: $S(x)|P(x), S(x)|Q(x)$ Si $S_1(x)|P(x) \wedge S_1(x)|Q(x) \Rightarrow S_1(x)|S(x) \forall S_1(x) \in \mathbb{R}[x]$ Lo denotamos por $S(x) = \text{mcd}(P(x), Q(x))$

4.2 Método McLaurin De Acotación De Raíces

Con el método de McLaurin se pretende determinar un conjunto fuera del cual se puede asegurar que la ecuación polinómica $P(x) = 0$ no posee raíces. Consideraremos:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 + a_0 \in \mathbb{R}[x], a_n \neq 0$$

Teorema 4.2.1 ► Teorema Acotación McLaurin

Si $\{r_1, r_2, \dots, r_n\} \subset \mathbb{C}$ son las n raíces de P , entonces:

$$|r_k| < 1 + \lambda \quad 1 \leq k \leq n$$

donde: $\lambda = \max_{0 \leq k \leq n-1} \left| \frac{a_k}{a_n} \right|$.

Es decir, las raíces se encuentran en el interior de la bola abierta:

$$B_{1+\lambda}(0) = \{z \in \mathbb{C} : |z| < 1 + \lambda\}$$

Corolario 4.2.1 ► Corolario Acotación McLaurin

Si $a_0 \neq 0$ y $\{r_1, r_2, \dots, r_n\} \subset \mathbb{C}$ son las n raíces de P , entonces:

$$|r_k| > \frac{1}{1 + \mu} \quad 1 \leq k \leq n$$

donde: $\mu = \max_{1 \leq k \leq n} \left| \frac{a_k}{a_0} \right|$.

Luego las raíces se encuentran en el exterior de la bola cerrada:

$$\overline{B}_{\frac{1}{1+\mu}}(0) = \{z \in \mathbb{C} : |z| \leq \frac{1}{1+\mu}\}$$

Nota 4.2.1 ► Corona Circular

Si unimos todo ello, y siempre que $a_0 \neq 0$, tenemos que las raíces se encuentran en el interior de la corona circular:

$$C = \left\{ z \in \mathbb{C} : \frac{1}{1+\mu} < |z| < 1+\lambda \right\}$$

Ejemplo 4.2.1 ► Acotación De Raíces De Un Polinomio

Consideremos la ecuación polinómica $2x^4 + 4x^3 - 59x^2 - 61x + 30 = 0$, determinar el conjunto de acotación de las raíces.

Resolución

Tenemos que los coeficientes del polinomio son:

$$a_4 = 2, a_3 = 4, a_2 = -59, a_1 = -61, a_0 = 30 \neq 0$$

Al tener que el término independiente es no nulo podemos aseverar que las raíces se encuentran en el interior de una corona circular.

$$\lambda = \max \left\{ \frac{4}{2}, \frac{59}{2}, \frac{61}{2}, \frac{30}{2} \right\} = \frac{61}{2} = 30'5 \quad \mu = \max \left\{ \frac{2}{30}, \frac{4}{30}, \frac{59}{30}, \frac{61}{30} \right\} = \frac{61}{30}$$

Luego: $1 + \lambda = 1 + \frac{61}{2} = \frac{63}{2} = 31'5$, $\frac{1}{1+\mu} = \frac{1}{1+\frac{61}{30}} = \frac{1}{\frac{91}{30}} = \frac{30}{91} \approx 0'329$. Luego la corona circular es:

$$C = \{ z \in \mathbb{C} : 0'329 < |z| < 31'5 \}$$

Lo puedes apreciar en la figura 1:

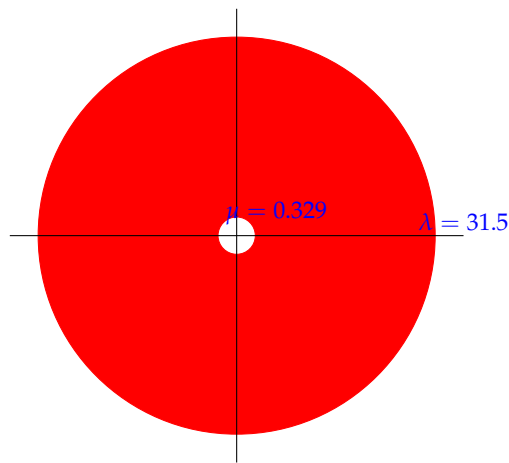


Figura 1: Corona Circular Acotación McLaurin

4.3 Separación De Raíces Reales

Se trata de saber cuántas raíces reales tiene un polinomio en el interior de cada intervalo, y que estas sean únicas en sus respectivos intervalos.

Proposición 4.3.1 ► Separación Raíces En Intervalos

- a) Si $P(a)P(b) < 0 \Rightarrow P$ tiene un número impar de raíces en $]a, b[$, contadas con su multiplicidad. (Se debe al Teorema de Bolzano).
- b) Si $P(a)P(b) > 0 \Rightarrow P$ no tiene raíces reales en $]a, b[$, o bien tiene un número par de raíces en $]a, b[$, contadas con su multiplicidad.

4.3.1. Regla De Los Signos De Descartes

Definición 4.3.1 ► Cambios De Signo

Dada la secuencia $\{c_1, c_2, \dots, c_k\} \subset \mathbb{R}$, se define el Número De Cambios De Signo al número total de cambios de signo en cada par consecutivo de elementos eliminando previamente los elementos nulos.

Ejemplo 4.3.1 ► Determinamos Cambios De Signo

En la secuencia $\{1, 0, -2, -3, 4, 0\}$, eliminando ceros tenemos $\{1, -2, -3, 4\}$, su número de cambios de signo es 2 porque hay un cambio de signo entre el primero y el segundo; y otro entre el tercero y el cuarto.

Teorema 4.3.1 ► Regla de los signos de Descartes

Dado $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$; $a_n \neq 0$, se cumple:

El número de raíces positivas (contando cada raíz con su multiplicidad) en $P(x) = 0$ es igual al Número De Cambios De Signo en $\{a_n, a_{n-1}, \dots, a_2, a_1, a_0\}$ o menor que dicho número en un natural par.

Para contar las raíces negativas basta considerar $P(-x) = 0$.

Ejemplo 4.3.2 ► Hallamos Raíces

Determinar el número de raíces reales del polinomio $P(x) = x^5 + x^3 - x^2 - 10x + 1$

Resolución

La secuencia de coeficientes del polinomio, eliminando los nulos, es: $\{1, 1, -1, -10, 1\}$, que tiene 2 cambios de signo, luego el polinomio tiene 2 raíces positivas o ninguna. Pero como $P(0) \cdot P(1) = 1 \times (-8) < 0$, el Teorema de Bolzano nos dice que hay al menos una raíz en $]0, 1[$. Luego por la Regla de Descartes el polinomio tiene 2 raíces reales positivas.

Para las negativas consideramos $P(-x) = -x^5 - x^3 - x^2 + 10x + 1$, con secuencia:

$\{-1, -1, -1, 10, 1\}$, y que tiene un único cambio de signo (impar), luego por la Regla de Descartes el polinomio tiene una única raíz real negativa.

En resumen, el polinomio tiene 2 raíces reales positivas y una negativa, al ser de grado 5 y las raíces reales simples, se tiene que también posee una raíz compleja y su conjugada.

4.3.2. Método De Sturm

Con este método queremos determinar el número de raíces reales distintas y separarlas de la ecuación polinómica $P(x) = 0$. Consiste en construir una sucesión de polinomios que surge de calcular el $\text{mcd}(P(x), P'(x))$ al aplicar el *Algoritmo de División de Euclides*. Primero tomamos como primer polinomio de la sucesión a $P_1(x) = P'(x)$ y seguidamente aplicamos el algoritmo de Euclides hasta que lleguemos a una división exacta:

$$\begin{cases} P(x) = P_1(x)Q_1(x) - P_2(x) & \partial P_2 < \partial P_1 \\ P_1(x) = P_2(x)Q_2(x) - P_3(x) & \partial P_3 < \partial P_2 \\ \dots\dots\dots & \dots\dots\dots \\ P_{m-2}(x) = P_{m-1}(x)Q_{m-1}(x) - P_m(x) & \partial P_m < \partial P_{m-1} \\ P_{m-1}(x) = P_m(x)Q_m(x) & \end{cases}$$

Notar que $\text{mcd}(P(x), P'(x)) = P_m(x)$

Definición 4.3.2 ► Secuencia De Sturm

La sucesión $\{P(x), P_1(x), \dots, P_m(x)\}$ se denomina Secuencia De Sturm para el polinomio P .

Nota 4.3.1 ► Multiplicidad Polinomios

Se tiene lo siguiente:

- Si $P_m(x) = c$ (cte.) \Rightarrow Las raíces de P son Simples.
- Si $P_m(x) = c(x - r_1)^{m_1}(x - r_2)^{m_2} \dots (x - r_s)^{m_s} \Rightarrow$ Cada r_k es raíz de P de multiplicidad $m_k + 1$, y además:

$$P(x) = (x - r_1)^{m_1+1}(x - r_2)^{m_2+1} \dots (x - r_s)^{m_s+1}Q(x); Q(r_k) \neq 0; 1 \leq k \leq s$$

Y las raíces de $Q(x) = 0$ son las **Raíces Simples** de $P(x) = 0$

Notación

Denotemos por:

- $N(\alpha)$ el número de cambios de signo en la sucesión de Sturm $\{P(\alpha), P_1(\alpha), \dots, P_m(\alpha)\}$
- $N(a, b)$ al número de raíces reales distintas de $P(x) = 0$, en el intervalo $]a, b[$, SIN CONTAR LA MULTIPLICIDAD.
- $N(\pm\infty) = \lim_{x \rightarrow \pm\infty} P(x)$

Teorema 4.3.2 ► Teorema De Sturm

Sea $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 \in \mathbb{R}[x]$, $a_n \neq 0$. Si $P(a)P(b) \neq 0$ en el intervalo $]a, b[$, entonces:

$$N(a, b) = N(a) - N(b)$$

Corolario 4.3.1 ► Número Raíces

Sea $P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 \in \mathbb{R}[x]$, $a_n \neq 0$.

a) Si $P(0) \neq 0$ entonces:

$$\begin{cases} N_+ = N(0) - N(+\infty) & \text{es el número de raíces positivas de } P \\ N_- = N(-\infty) - N(0) & \text{es el número de raíces negativas de } P \end{cases}$$

b) Las n raíces de P son reales y simples $\Leftrightarrow N(-\infty) - N(+\infty) = n$

Ejemplo 4.3.3 ► Determinamos Raíces

Determinar el número de raíces reales positivas y negativas de:

$$P(x) = x^4 - 4x + 1$$

Resolución

$P'(x) = 4x^3 - 4$, para simplificar cálculos dividimos por 4, $P_1(x) = x^3 - 1$. Dividimos $P(x)$ entre $P_1(x)$:

$$x^4 - 4x + 1 = (x^3 - 1)x + (-3x + 1)$$

Tomamos $P_2(x) = 3x - 1$ y dividimos $P_1(x)$ entre $P_2(x)$:

$$x^3 - 1 = (3x - 1) \left(\frac{1}{3}x^2 + \frac{1}{9}x + \frac{1}{27} \right) - \frac{26}{27}$$

Tomamos $P_3(x) = 1$ en vez de $\frac{-26}{27}$ para simplificar cálculos. La secuencia de Sturm es:

$$\{x^4 - 4x + 1, x^3 - 1, 3x - 1, 1\}$$

Realizamos la tabla de signos:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$N(x)$
$-\infty$	+	-	-	+	2
0	+	-	-	+	2
$+\infty$	+	+	+	+	0

Luego tenemos que:

$$\begin{cases} N_- = N(-\infty) - N(0) = 0 \\ N_+ = N(0) - N(+\infty) = 2 \end{cases}$$

Como $N(-\infty) - N(+\infty) = 2 - 0 = 2$, todas sus raíces son simples.

Luego P tiene 2 raíces positivas, ninguna negativa y 2 complejas conjugadas por ser de grado 4 todas simples.

4.3.3. Método de Sturm

Combinando el teorema de Sturm y la bisección obtenemos el método de Sturm para poder separar en intervalos las raíces de un polinomio. Supongamos que todas las raíces están en $]a, b[$, dicho intervalo lo podemos tomar como $]a, b[=]-1 - \lambda, 1 + \lambda[$, siendo $1 + \lambda$ la cota de McLaurin.

a) Si $N(a) - N(b) = 1$, proceso terminado.

b) Si $N(a) - N(b) > 1$, se toma $c = \frac{a+b}{2}$ y los intervalos $]a, c[$ y $]c, b[$. Calcular $N(a) - N(c)$ y $N(c) - N(b)$. Si alguno es mayor que 1 se le repite el proceso.

Se llega así a tantos intervalos como raíces reales y distintas tenga el polinomio, intervalos con sólo una raíz en cada uno de ellos.

Ejemplo 4.3.4 ► Separamos En Intervalos

Separar en intervalos las raíces de:

$$P(x) = x^4 - 4x + 1$$

Resolución

$\lambda = \max \left\{ \frac{1}{1}, \frac{4}{1}, \frac{1}{1} \right\} = 4 \Rightarrow 1 + \lambda = 5$ es la cota de McLaurin. Para las raíces positivas consideramos $]0, 5[$, por el Teorema de Sturm sabemos que éste intervalo contiene las 2 raíces positivas.

$$N(0) - N(5) = N(0) - N(+\infty) = 2 - 0 = 2 > 1 \Rightarrow c = \frac{a+b}{2} = 2.5$$

Luego tenemos $]0, 2.5[,]2.5, 5[$

$N(0) - N(2.5) = 2 - 0 = 2 > 1$ y $N(2.5) - N(5) = 0 - 0 = 1 \Rightarrow$ Las 2 raíces positivas están en $]0, 2.5[$

Calculamos $c_2 = \frac{0+2.5}{2} = 1.25$ Obtenemos $]0, 1.25[,]1.25, 2.5[$

$$N(0) - N(1.25) = 2 - 1 = 1 \text{ y } N(1.25) - N(2.5) = 1 - 0 = 1$$

Luego una raíz positiva está en $]0, 1.25[$ y la otra en $]1.25, 2.5[$

Al no haber raíz negativa no consideramos el intervalo abierto $] -1 - \lambda, 0[=] -6, 0[$

4.4 Polinomios Con Coeficientes Racionales

Consideremos el polinomio con coeficientes racionales:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 + a_0 \in \mathbb{Q}[x], a_n \neq 0$$

Antes de pasar adelante con la sección hay que decir que dicho polinomio lo podemos simplificar a coeficientes enteros. Tenemos

$$a_i = \frac{p_i}{q_i}; 0 \leq i \leq n; p_i \in \mathbb{Z}, q_i \in \mathbb{N}$$

Luego podemos considerar el mínimo común múltiplo de todos los denominadores:

$$m = mcm(q_n, q_{n-1}, \dots, q_1, q_0)$$

Obtenemos el nuevo polinomio en los enteros:

$$Q(x) = b_n x^n + b_{n-1} x^{n-1} + \dots + b_2 x^2 + b_1 + b_0 \in \mathbb{Z}[x], b_n \neq 0; b_i = m \frac{p_i}{q_i} \in \mathbb{Z}; 0 \leq i \leq n$$

Se verifica que $Q(x) = mP(x)$, luego $P(x) = 0$ y $Q(x) = 0$ son ecuaciones equivalentes. Luego en toda la sección podemos considerar, sin pérdida de generalidad:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 + a_0 \in \mathbb{Z}[x], a_n \neq 0$$

Proposición 4.4.1 ► Divisores Término Independiente

Si $r \in \mathbb{Z} / P(r) = 0 \Rightarrow r | a_0$

Proposición 4.4.2 ► Estudio Raíces Racionales En Polinomios

Si $r = \frac{p}{q} \in \mathbb{Q} \text{ mcd}(p, q) = 1 / P(r) = 0 \Rightarrow p | a_0 \text{ y } q | a_n$

Si $a_n = \pm 1 \Rightarrow$ sus posibles raíces racionales se reducen a las enteras.

Ejemplo 4.4.1 ► Resolvemos Polinomio De Grado Cuatro

Resolver $10x^4 - 11x^3 - 41x^2 + x + 6 = 0$

Resolución

Tenemos que $P(-x) = 10x^4 + 11x^3 - 41x^2 - x + 6$, las secuencias de coeficientes son:

$$\{10, -11, -41, 1, 6\} \Rightarrow 2 \text{ cambios de signo}$$

$$\{10, 11, -41, -1, 6\} \Rightarrow 2 \text{ cambios de signo}$$

Aplicando la Regla de Descartes nuestro polinomio P puede tener:

$$\begin{cases} 2 \text{ raíces positivas o ninguna} \\ 2 \text{ raíces negativas o ninguna} \end{cases}$$

Vamos a acotar las raíces con el método de McLaurin buscando las cotas de la corona circular:

$$\lambda = \max \left\{ \frac{11}{10}, \frac{41}{10}, \frac{1}{10}, \frac{6}{10} \right\} = \frac{41}{10} \Rightarrow 1 + \lambda = 1 + \frac{41}{10} \Rightarrow 1 + \lambda = \frac{51}{10} = 5.1$$

$$\mu = \max \left\{ \frac{10}{6}, \frac{11}{6}, \frac{41}{6}, \frac{1}{6} \right\} = \frac{41}{6} \Rightarrow \frac{1}{1+\mu} = \frac{1}{1+\frac{41}{6}} \Rightarrow \frac{1}{1+\mu} = \frac{6}{47} \approx 0.127$$

Por tanto, las raíces se encuentran en la corona circular:

$$C = \{z \in \mathbb{C} : 0.127 < |z| < 5.1\}$$

Veamos si el polinomio tiene raíces enteras, de $a_0 = 6$ sus divisores y candidatos a ser raíces enteras son:

$$\{\pm 1, \pm 2, \pm 3, \pm 6\}$$

Como $\pm 6 \notin C$, ya sabemos que no son raíces enteras del polinomio, además; de los que pertenecen a C , como $P(\pm 1) \neq 0$, $P(\pm 2) \neq 0$, $P(\pm 3) \neq 0$, concluimos que nuestro polinomio no tiene raíces enteras.

Ahora hemos de ver si el polinomio posee raíces racionales, los divisores de $a_4 = 10$ son:

$$\{\pm 1, \pm 2, \pm 5, \pm 10\}$$

Para saber cuáles son nuestros candidatos a raíces racionales lo que hemos de hacer es dividir cada uno de nuestros candidatos de las raíces enteras por cada uno de los divisores de $a_4 = 10$. Simplificando fracciones, quitando las repetidas y las fracciones que nos dan enteros tenemos que nuestro conjunto de candidatos a ser raíces racionales del polinomio son:

$$\left\{ \pm \frac{1}{2}, \pm \frac{1}{5}, \pm \frac{1}{10}, \pm \frac{2}{5}, \pm \frac{3}{2}, \pm \frac{3}{5}, \pm \frac{3}{10}, \pm \frac{6}{5} \right\}$$

Como $\pm \frac{1}{10} \notin C$, podemos decir que no son raíces racionales del polinomio. De los restantes que sí pertenecen a C tenemos:

$$\begin{aligned} P\left(\pm \frac{1}{2}\right) &\neq 0, P\left(\pm \frac{1}{5}\right) \neq 0, P\left(\pm \frac{2}{5}\right) \neq 0, P\left(\frac{3}{2}\right) \neq 0, P\left(-\frac{3}{2}\right) = 0, P\left(\pm \frac{3}{5}\right) \neq 0, \\ P\left(\pm \frac{3}{10}\right) &\neq 0, P\left(\pm \frac{6}{5}\right) \neq 0 \end{aligned}$$

Luego la única raíz racional de P es $-\frac{3}{2}$. Si dividimos $P(x) \mid \left(x + \frac{3}{2}\right)$, por la Regla de Ruffini:

$$\begin{array}{r|rrrrr} -\frac{3}{2} & 10 & -11 & -41 & 1 & 6 \\ & \downarrow & -15 & 39 & 3 & -6 \\ \hline & 10 & -26 & -2 & 4 & 0 \end{array}$$

Con lo cual tenemos que podemos factorizar el polinomio como:

$$P(x) = 20 \left(x + \frac{3}{2}\right) (5x^3 - 13x^2 - x + 2)$$

Ahora trabajamos con $Q(x) = 5x^3 - 13x^2 - x + 2$. Los divisores de $a_0 = 2$ son $\{\pm 1, \pm 2\}$ y los de $a_3 = 5$ son $\{\pm 1, \pm 5\}$.

Luego los candidatos a raíces racionales de Q son $\left\{\pm \frac{1}{5}, \pm \frac{2}{5}\right\}$. De ellos sólo $-\frac{2}{5}$ es raíz racional de Q .

Notar que Q NO tiene raíces enteras porque P no las tiene, ya que si Q tuviese raíces enteras éstas serían a su vez raíces enteras de P , pero eso ya hemos comprobado que no es cierto.

Si hacemos $Q(x) \mid \left(x + \frac{2}{5}\right)$ por Ruffini obtenemos:

$$\begin{array}{r|rrrr} -\frac{2}{5} & 5 & -13 & -1 & 2 \\ & \downarrow & -2 & 6 & -2 \\ \hline & 5 & -15 & 5 & 0 \end{array}$$

Ahora factorizamos Q como:

$$Q(x) = 25 \left(x + \frac{2}{5}\right) (x^2 - 3x + 1)$$

Finalmente, podemos hallar las raíces de $x^2 - 3x + 1$ por el método tradicional de la fórmula de segundo grado $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ y obtenemos:

$$Q(x) = 25 \left(x + \frac{2}{5}\right) \left(x - \frac{3 + \sqrt{5}}{2}\right) \left(x - \frac{3 - \sqrt{5}}{2}\right)$$

Como teníamos que $P(x) = 20 \left(x + \frac{3}{2}\right) Q(x)$, si sustituimos $Q(x)$ por lo que acabamos de obtener:

$$P(x) = 500 \left(x + \frac{3}{2}\right) \left(x + \frac{2}{5}\right) \left(x - \frac{3 + \sqrt{5}}{2}\right) \left(x - \frac{3 - \sqrt{5}}{2}\right)$$

Luego las raíces son:

$$r_1 = \frac{-3}{2}, r_2 = \frac{-2}{5}, r_3 = \frac{3 + \sqrt{5}}{2}, r_4 = \frac{3 - \sqrt{5}}{2}$$

Como r_3 y r_4 son irracionales viene ahora muy bien aproximarlas mediante alguno de los métodos iterativos del capítulo anterior: bisección, regula-falsi, punto fijo, etc. Además, para aplicar estos métodos no los aplicaremos directamente al polinomio P , para simplificar cálculos los aplicaremos directamente al polinomio $x^2 - 3x + 1$.

Ejemplo 4.4.2 ► Otro Más De Hallar Raíces

Resolver la ecuación $P(x) = 0$ siendo

$$P(x) = x^4 - 4x^3 - x^2 + 12x - 6$$

Resolución

Primero aplicamos la regla de los signos de Descartes, tenemos $P(-x) = x^4 + 4x^3 - x^2 - 12x - 6$, obtenemos las secuencias de coeficientes:

$$\begin{cases} \{1, -4, -1, 12, -6\} & \Rightarrow 3 \text{ cambios de signo} \\ \{1, 4, -1, -12, -6\} & \Rightarrow 1 \text{ cambio de signo} \end{cases}$$

Luego nuestro polinomio tiene 3 ó 1 raíces reales positivas y 1 raíz real negativa. Seguidamente acotamos las raíces por el método de McLaurin:

$$\begin{aligned} \lambda &= \max \{4, 1, 12, 6\} = 12 \Rightarrow 1 + \lambda = 13 \\ \mu &= \max \left\{ \frac{1}{6}, \frac{2}{3}, \frac{1}{6}, 2 \right\} = 2 \Rightarrow \frac{1}{1 + \mu} = \frac{1}{3} \end{aligned}$$

Por tanto, las raíces se encuentran en el interior de la corona circular:

$$C = \left\{ z \in \mathbb{C} : \frac{1}{3} < |z| < 13 \right\}$$

Veamos si posee raíces enteras, los divisores de $a_0 = -6$ son: $\{\pm 1, \pm 2, \pm 3, \pm 6\}$. Sustituyendo en el polinomio tenemos que ninguna de ellos es raíz. Luego, el polinomio no tiene raíces enteras.

Pasemos a ver si tiene raíces racionales, los divisores de $a_4 = 1$ son ± 1 , luego sus posibles raíces racionales se reducen a raíces enteras, pero ya sabemos que no tiene; luego el polinomio tampoco tiene raíces racionales. Por tanto, sus raíces sólo pueden ser irracionales o complejas.

El siguiente paso es hacer la separación de raíces, vamos a aplicar el algoritmo de Euclides hasta obtener el máximo común divisor de P y P'

$$\begin{aligned}P'(x) &= 4x^3 - 12x^2 - 2x + 12 = 2(2x^3 - 6x^2 - x + 6) \\P_1(x) &= 2x^3 - 6x^2 - x + 6\end{aligned}$$

Si dividimos $P(x)|P_1(x)$ obtenemos:

$$\begin{aligned}P(x) &= P_1(x) \left(\frac{x}{2} - \frac{1}{2} \right) + \left(-\frac{7}{2}x^2 + \frac{17}{2}x - 3 \right) \\P_2(x) &= 7x^2 - 17x + 6\end{aligned}$$

Pasamos a hacer la división $P_1(x)|P_2(x)$

$$\begin{aligned}P_1(x) &= P_2(x) \left(\frac{2}{7}x - \frac{8}{49} \right) + \left(-\frac{269}{49}x + \frac{342}{49} \right) \\P_3(x) &= 269x - 342\end{aligned}$$

Y finalmente $P_2(x)|P_3(x)$

$$\begin{aligned}P_2(x) &= P_3(x) \left(\frac{7}{269}x - \frac{2179}{72361} \right) - \frac{311052}{72361} \\P_4(x) &= 1\end{aligned}$$

Así pues, nuestra secuencia de Sturm es:

$$\left\{ x^4 - 4x^3 - x^2 + 12x - 6, 2x^3 - 6x^2 - x + 6, 7x^2 - 17x + 6, 269x - 342, 1 \right\}$$

Realizamos la tabla de signos de Sturm:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$N(x)$
$-\infty$	+	-	+	-	+	4
0	-	+	+	-	+	3
$+\infty$	+	+	+	+	+	0

Aplicando el Teorema de Sturm podemos concluir:

$$\begin{cases} N_+ = N(0) - N(+\infty) = 3 \\ N_- = N(-\infty) - N(0) = 1 \end{cases}$$

Luego nuestro polinomio tiene 3 raíces positivas y una raíz negativa. Seguidamente aplicamos el método de Sturm (NO el Teorema), donde los extremos van a ser todos los divisores de $a_0 = -6$ junto con el 0 del Teorema de Sturm:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$N(x)$
-6	+	-	+	-	+	4
-3	+	-	+	-	+	4
-2	+	-	+	-	+	4
-1	-	-	+	-	+	3
0	-	+	+	-	+	3
1	+	+	-	-	+	2
2	-	-	0	+	+	1
3	-	+	+	+	+	1
6	+	+	+	+	+	0

Concluimos que el polinomio tiene exactamente 1 raíz en cada intervalo:

$$]-2, -1[,]0, 1[,]1, 2[,]3, 6[$$

Con todo esto no hemos podido saber cuáles son las raíces, sólo hemos sabido por dónde caen, al ser irracionales deberemos aplicar algún método iterativo, por ejemplo el de Newton. Dibujemos la función con Octave primero para determinar los puntos iniciales del método de Newton, en la figura 2

Luego viendo la figura podemos tomar como valor inicial del método de Newton los siguientes valores:

$$x_0 \in \{-2, 0, 2, 6\}$$

Y si aplicamos en cada uno de ellos el método de Newton obtenemos las siguientes raíces aproximadas:

$$\{-1'732050807\dots, 0'585786437\dots, 1'732050807\dots, 3'414213562\dots\}$$

Si quieres las raíces exactas para comparar lo puedes hacer con el programa **wxMaxima** con el siguiente comando:

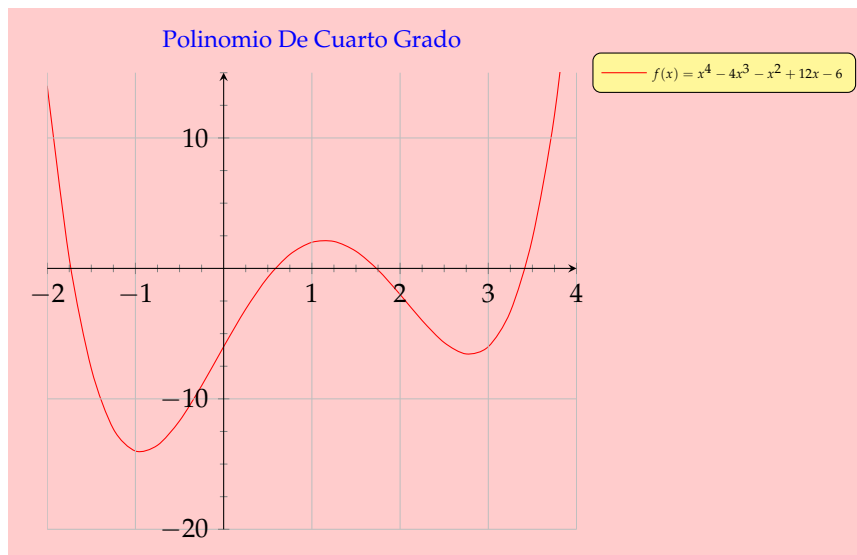


Figura 2: Polinomio $f(x) = x^4 - 4x^3 - x^2 + 12x - 6$

```
(%i1) solve([x^4-4*x^3-x^2+12*x-6], [x]);
```

Dando como soluciones:

$$\left[x = 2 - \sqrt{2}, x = \sqrt{2} + 2, x = -\sqrt{3}, x = \sqrt{3} \right]$$

Ejemplo 4.4.3 ► Ejemplo Deflación

Encontrar una aproximación de las raíces de la ecuación algebraica:

$$2x^4 - x^3 + 2x^2 - 7x + 3 = 0$$

Resolución

$$P(x) = 2x^4 - x^3 + 2x^2 - 7x + 3$$

$$P(-x) = 2x^4 + x^3 + 2x^2 + 7x + 3$$

Tenemos las secuencias de coeficientes:

$$\begin{cases} P(x) \longrightarrow \{2, -1, 2, -7, 3\} \Rightarrow 4 \text{ cambios de signo} \\ P(-x) \longrightarrow \{2, 1, 2, 7, 3\} \Rightarrow 0 \text{ cambios de signo} \end{cases}$$

Luego por la regla de signos de Descartes el polinomio no tiene raíces negativas porque $P(-x)$ no tiene cambios de signo, y al ser de grado 4 puede tener 4, 2 ó 0 raíces positivas.

Apliquemos el método de McLaurin para acotar las raíces:

$$\begin{aligned} \lambda &= \max \left\{ \frac{3}{2}, \frac{7}{2}, 1, \frac{1}{2} \right\} = \frac{7}{2} \Rightarrow 1 + \lambda = 1 + \frac{7}{2} = \frac{9}{2} = 4'5 \\ \mu &= \max \left\{ \frac{2}{3}, \frac{1}{3}, \frac{2}{3}, \frac{7}{3} \right\} = \frac{7}{3} \Rightarrow \frac{1}{1 + \mu} = \frac{1}{1 + \frac{7}{3}} = \frac{3}{10} = 0'3 \end{aligned}$$

Luego las raíces se encuentran en el interior de la corona circular:

$$C = \left\{ z \in \mathbb{C} : \frac{3}{10} < |z| < \frac{9}{2} \right\}$$

Los divisores positivos de $a_0 = 3$ son $\{1, 3\}$. Como $P(1) \neq 0$, $P(3) \neq 0$, se tiene que el polinomio no tiene raíces enteras.

Los divisores positivos de $a_4 = 2$ son $\{1, 2\}$, sus candidatas a raíces racionales positivas son $\left\{ \frac{1}{2}, \frac{3}{2} \right\}$. De aquí tenemos:

$$P\left(\frac{1}{2}\right) = 0, P\left(\frac{3}{2}\right) \neq 0$$

Luego $\frac{1}{2}$ es una raíz racional positiva del polinomio. Por lo que por la Regla de Ruffini vamos a factorizar el polinomio:

$$\begin{array}{r|rrrrr} & 2 & -1 & 2 & -7 & 3 \\ \frac{1}{2} & \downarrow & 1 & 0 & 1 & -3 \\ \hline & 2 & 0 & 2 & -6 & 0 \end{array}$$

$$P(x) = 4 \left(x - \frac{1}{2} \right) (x^3 + x - 3)$$

Seguimos trabajando con $Q(x) = x^3 + x - 3$, que no tiene raíces negativas por no tenerlas P . La secuencia de signos es $\{+, +, -\}$, luego sólo hay un cambio de signo, por lo que sólo posee una raíz positiva por la regla de signos de Descartes. Así pues, P tiene 2 raíces positivas, ninguna

negativa y 2 raíces complejas. Q no posee raíces enteras porque P no tampoco las tiene. Como en Q tenemos que $a_3 = 1$ también se tiene que Q no tiene raíces racionales, en consecuencia, P sólo tiene 1 raíz racional. Por tanto, la raíz real que nos falta averiguar de P sabemos que es una raíz irracional positiva; y la vamos a averiguar de Q , por ser más sencillo. Aplicando el Teorema de Bolzano:

$$Q(1) \cdot Q(2) = -1 \cdot 7 < 0 \longrightarrow \exists r \in]1, 2[/ Q(r) = 0$$

Luego la única raíz positiva de Q se encuentra en el intervalo $]1, 2[$. Antes de pasar a calcular las otras 2 raíces complejas conjugadas hemos de calcular por un método iterativo ésta raíz irracional de Q , si aplicamos el Método de Newton con $x_0 = 2$ obtenemos que la raíz aproximada es 1'21341166276223. Esto que acabamos de hacer es la **DEFLACIÓN**, es decir; en cada paso vamos reduciendo el polinomio de partida de grado, y si vemos que la raíz o raíces del nuevo polinomio reducido de grado son irracionales las calculamos por medio de un método iterativo de forma aproximada.

Por tanto, Q queda factorizado de forma aproximada como:

$$Q(x) = (x - 1'21341166276223) \cdot \tilde{Q}(x), \partial \tilde{Q} = 2$$

Con un poco de paciencia, por la Regla de Ruffini obtenemos de forma aproximada:

$$\tilde{Q}(x) = x^2 + 1'21341166278x + 2'47236765561$$

Como es de grado 2 basta aplicar la fórmula de todos conocida para ecuaciones de segundo grado y obtener que las aproximaciones a las 2 raíces complejas conjugadas son:

$$\alpha \pm \beta i \simeq -0'606705831781 \pm 1'45061217736i$$



EJERCICIOS RESUELTOS

POLINOMIOS

Ejercicio 5.0.1 ► Polinomio Grado Tres

Calcular las raíces de la ecuación $x^3 - x^2 + 3x = 3$

Resolución

Tenemos el polinomio $P(x) = x^3 - x^2 + 3x - 3$, con $P(-x) = -x^3 - x^2 - 3x - 3$. Las secuencias de signos son:

$$\begin{cases} P(x) \rightarrow \{1, -1, 3, -3\} \Rightarrow 3 \text{ cambios de signo} \\ P(-x) \rightarrow \{-1, -1, -3, -3\} \Rightarrow 0 \text{ cambios de signo} \end{cases}$$

Luego por la regla de signos de Descartes el polinomio no tiene raíces negativas, y puede tener 3 ó 1 raíz positivas.

Los divisores positivos de $a_0 = -3$ son $\{1, 3\}$, como $P(1) = 0$, $P(3) = 24 \neq 0$, concluimos que 1 es una raíz entera del polinomio. Como $a_1 = 1$, sabemos que no tiene raíces racionales. Aplicamos Ruffini para factorizar el polinomio:

$$\begin{array}{r|rrrr} & 1 & -1 & 3 & -3 \\ 1 & \downarrow & 1 & 0 & 3 \\ \hline & 1 & 0 & 3 & 0 \end{array}$$

Luego factorizamos el polinomio como:

$$P(x) = (x - 1) \cdot (x^2 + 3) = (x - 1) \cdot (x - \sqrt{3}i) \cdot (x + \sqrt{3}i)$$

Así pues todas las raíces son: $1, \pm \sqrt{3}i$, todas ellas simples.

Una cosa importante que debes saber es que no tenemos porqué aplicar todos los métodos si hay un camino más corto, en este ejemplo omitimos McLaurin y Sturm. Incluso podríamos haber omitido los pasos anteriores y haberlo solucionado directamente por la Regla de Ruffini, ya que 1 es uno de los primeros candidatos a ser raíz cuando hacemos Ruffini por tanteo.

Otra cosa, las dos raíces conjugadas que hemos obtenido su parte imaginaria es un número irracional, con lo que podríamos haber obtenido sus aproximaciones por un método iterativo como el método de Müller, pero en éste caso es innecesario porque obtenemos las raíces exactas.

Ejercicio 5.0.2 ► Polinomio Grado Trece

Hallar las raíces, determinando su multiplicidad, de $x^{13} - x^{11} + x^2 - 1 = 0$

Resolución

Consideramos $P(x) = x^{13} - x^{11} + x^2 - 1$, los candidatos a raíces enteras son 1 y -1. No tiene raíces racionales por ser polinomio mónico¹. Como $P(1) = P(-1) = 0$, tenemos que 1 y -1 son raíces del polinomio. Factorizemos el polinomio aplicando la Regla de Ruffini:

	1	0	-1	0	0	0	0	0	0	0	0	0	1	0	-1
1	↓	1	1	0	0	0	0	0	0	0	0	0	0	1	1
	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0
-1	↓	-1	0	0	0	0	0	0	0	0	0	0	0	-1	
	1	0	0	0	0	0	0	0	0	0	0	0	1	0	

Por tanto, el polinomio queda factorizado como:

$$P(x) = (x - 1) \cdot (x + 1) \cdot (x^{11} + 1)$$

Vamos a calcular las raíces de $x^{11} + 1 = 0 \Rightarrow x^{11} = -1 = 1_{\pi} = |-1|_{\theta}$. Pero esto es calcular las raíces onceavas (undécimas) de -1, para ello debemos recordar que todas ellas están dentro de una circunferencia formando un polígono de 11 lados y radio el módulo de -1, que es 1. Es decir:

$$z = re^{i\theta} = r \cdot (\cos \theta + i \sin \theta), r > 0, \theta \in [0, 2\pi[\Rightarrow z_k = \sqrt[n]{r} e^{i \frac{\theta + 2k\pi}{n}}; 0 \leq k \leq n - 1$$

En nuestro caso particular, $n = 11$ y las raíces son:

¹Un polinomio mónico es aquel polinomio en el que su coeficiente director es 1. El coeficiente director es el coeficiente que acompaña al monomio de mayor grado.

$$\zeta_k = \sqrt[11]{e^{i \frac{\pi + 2k\pi}{11}}} = e^{i \frac{\pi(2k+1)}{11}} = \cos\left(\frac{\pi(2k+1)}{11}\right) + i \sin\left(\frac{\pi(2k+1)}{11}\right); 0 \leq k \leq 10$$

Notar que para $k = 5$ obtenemos $\zeta_5 = -1$. Por tanto, -1 es una raíz de multiplicidad doble del polinomio P .

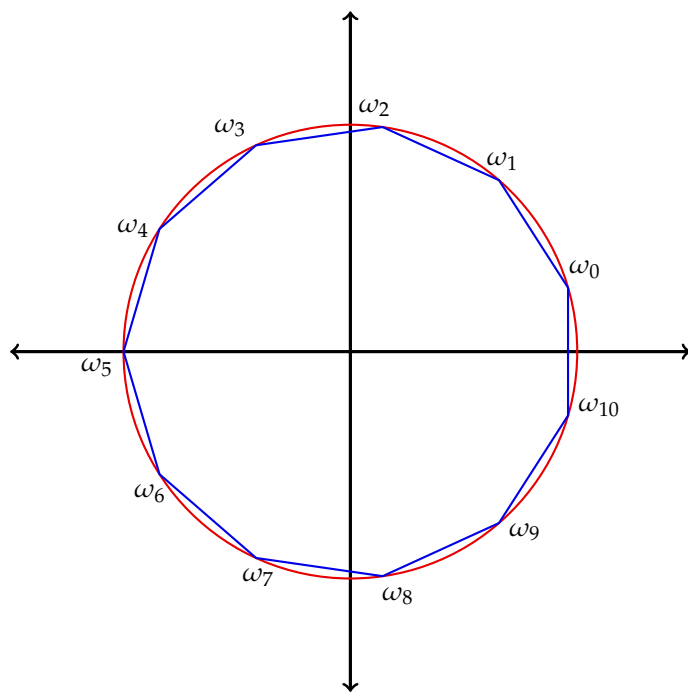


Figura 1: Raíces Unidad De $x^{11} + 1 = 0$

Te pongo como extra el código Octave para dibujar el polígono regular de 11 lados que se forma con dichas raíces.

```

1  % PARA QUE FUNCIONE HAY QUE TENER INSTALADO EL PAQUETE GEOMETRY DE OCTAVE-SOURCEFORGE
2  % ESTA EN LA WEB http://octave.sourceforge.net/geometry/index.html
3  % ANTES DE EJECUTAR EL CODIGO DESDE OCTAVE EJECUTA LA ORDEN
4  % pkg load all
5
6  clear;
7  clf();
8  hold on; axis equal;
9  axis ([-1.5, 1.5,-1.5,1.5], 'square');
10 grid;
11 n=11;
12

```

```

13 for k =1:11
14     h(k)=0;
15 end
16
17 for k=0:10
18     x(k+1)=cos(((2*k+1)*pi)/n);
19     y(k+1)=sin(((2*k+1)*pi)/n);
20     if (x(k+1)>0 && y(k+1)>0 || x(k+1)>0 && y(k+1)<0)
21         h(k+1) = drawLabels(x(k+1)+0.1,y(k+1)+0.1,['\omega_' num2str(k)]);
22     else
23         h(k+1) = drawLabels(x(k+1)-0.25,y(k+1)+0.1,['\omega_' num2str(k)]);
24     end
25 end
26
27 for k = 8:11
28     delete(h(k));
29 end
30
31 drawLabels(x(8)-0.25,y(8),'\omega_{7}');
32 drawLabels(x(9),y(9)-0.1,'\omega_{8}');
33 drawLabels(x(10),y(10)-0.1,'\omega_{9}');
34 drawLabels(x(11)+0.1,y(11),'\omega_{10}');
35
36 p1 = [1 0];
37 p2 = [0 1];
38 p3 = [-1 0];
39 origin = [0 0];
40
41 circle = createCircle(p1, p2,p3);
42
43 drawCircle (circle,'r','LineWidth',2);
44 patch(x,y,'b','EdgeColor','b','Marker','o','MarkerFaceColor','flat','MarkerSize',8,'
    LineWidth',2,'FaceAlpha',0.15);
45
46 for k=0:10
47     edge = [0 0 x(k+1) y(k+1)];
48     drawEdge(edge, 'linewidth', 1.15,'color','b');
49 end
50
51 title('Raices onceavas de la unidad');

```

raicesunidad.m

Ejercicio 5.0.3 ► Secuencia Sturm

Dados $a > 0, b \in \mathbb{R}$, se considera el polinomio

$$P(x) = x^3 - bx^2 + ax - ab$$

- a) Encontrar una relación entre a y b que garantice que la secuencia de Sturm de P tenga sólo tres términos $\{P_0(x), P_1(x), P_2(x)\}$.
- b) Decidir, en el caso en que a y b verifiquen la relación anterior, el número de raíces reales y distintas de P . ¿Son simples?

Resolución

Tomamos $P_1(x) = P'(x)$ y dividimos $P(x)$ entre $P_1(x)$, obteniendo:

$$P(x) = \left(\frac{x}{3} - \frac{b}{9}\right) \cdot P_1(x) + \frac{2}{9} \cdot [(3a - b^2)x - 4ab]$$

- a) Para que la secuencia de Sturm tenga sólo tres términos se ha de cumplir que $(3a - b^2)x - 4ab$ sea un polinomio constante y ello se cumple si $3a - b^2 = 0$. Es decir:

$$3a - b^2 = 0 \Leftrightarrow b^2 = 3a \Leftrightarrow b = \pm\sqrt{3a}$$

De esta forma la secuencia de Sturm es:

$$\left\{ x^3 - bx^2 + ax - ab, 3x^2 - 2bx + a, \frac{8ab}{9} \right\}$$

- b) Distinguimos dos casos:

CASO I: $b = \sqrt{3a}$

x	$P(x)$	$P_1(x)$	$P_2(x)$	$N(x)$
$-\infty$	-	+	+	1
0	-	+	+	1
$+\infty$	+	+	+	0

$$\begin{cases} N_- = N(-\infty) - N(0) = 0 \\ N_+ = N(0) - N(+\infty) = 1 \end{cases}$$

Por tanto, P tiene una raíz positiva y dos complejas conjugadas, porque $\partial P = 3$ y las raíces de P son simples.

CASO II: $b = -\sqrt{3a}$

x	$P(x)$	$P_1(x)$	$P_2(x)$	$N(x)$
$-\infty$	-	+	-	2
0	+	+	-	1
$+\infty$	+	+	-	1

$$\begin{cases} N_- = N(-\infty) - N(0) = 1 \\ N_+ = N(0) - N(+\infty) = 0 \end{cases}$$

Ahora P tiene una raíz negativa y dos complejas conjugadas por el mismo razonamiento de antes.

En ambos casos tenemos:

$$P(b) = b^3 - b^3 + ab - ab = 0$$

Luego b es raíz del polinomio, aplicando Ruffini:

	1	$-b$	a	$-ab$
b	↓	b	0	ab
	1	0	a	0
\sqrt{ai}	↓	\sqrt{ai}	$-a$	
	1	\sqrt{ai}	0	
$-\sqrt{ai}$	↓	$-\sqrt{ai}$		
	1	0		

Por lo que: $P(x) = (x - b) \cdot (x - \sqrt{ai}) \cdot (x + \sqrt{ai})$, luego las raíces del polinomio son $\{b, \pm \sqrt{ai}\}$

Ejercicio 5.0.4 ► Polinomio Grado Nueve

Consideremos el polinomio

$$P(x) = 9x^3 + 9x^2 + 9\lambda x + \lambda$$

a) Estudiar, en función del parámetro λ , el número de raíces (reales y complejas) de la ecuación $P(x) = 0$. ¿Para qué valores de λ las raíces de P son múltiples? Hallar todas las raíces de P para esos valores de λ .

b) Fijado $\lambda = \sqrt{3}$ encontrar un intervalo donde pueda aplicarse el método de Newton para calcular una raíz negativa de P . Determinar los primeros términos de la sucesión definida por dicho método.

Resolución

a) $P'(x) = 27x^2 + 18x + 9\lambda = 9(3x^2 + 2x + \lambda) \Rightarrow P_1(x) = 3x^2 + 2x + \lambda$

Dividiendo $P(x)$ entre $P_1(x)$ obtenemos:

$$P(x) = (3x + 1)P_1(x) + 2(3\lambda - 1)x$$

Luego tomamos $P_2(x) = (1 - 3\lambda)x$ si $\lambda \neq \frac{1}{3}$. Dividimos $P_1(x)$ entre $P_2(x)$:

$$P_1(x) = \frac{3x+2}{1-3\lambda}P_2(x) + \lambda$$

Distinguiremos casos según el valor de λ

i) Si $\lambda = \frac{1}{3} \Rightarrow \{P(x), P_1(x)\}$ es la secuencia de Sturm, y tenemos:

$$P_1(x) = 3x^2 + 2x + \frac{1}{3} = 0 \Rightarrow P_1(x) = 9\left(x + \frac{1}{3}\right)^2$$

Pero $P\left(\frac{1}{3}\right) = 0 \Rightarrow P(x) = 9\left(x + \frac{1}{3}\right)^3$. Luego para $\lambda = \frac{1}{3}$ tenemos que $-\frac{1}{3}$ es una raíz triple del polinomio P

ii) Si $\lambda = 0 \Rightarrow \{P(x), P_1(x), P_2(x)\}$ es la secuencia de Sturm. Tenemos $P_2(x) = (1 - 3\lambda)x = (1 - 3 \cdot 0)x = x$. Tenemos $r_1 = 0$ es raíz de $P_2(x)$, además:

$P_1(x) = 3x^2 + 2x = x(3x + 2)$, luego $r_1 = 0$ es raíz de $P_1(x)$. Pero además:

$P(x) = 9x^3 + 9x^2 = 9x^2(x + 1)$, luego $r_1 = 0$ es raíz doble de $P(x)$ y $r_2 = -1$ es raíz simple de $P(x)$.

iii) $\lambda \notin \left\{0, \frac{1}{3}\right\} \Rightarrow \{P(x), P_1(x), P_2(x), P_3(x)\}$ es la secuencia de Sturm, con $P_3(x) = -\lambda$ constante, luego en éste caso todas las raíces de P son simples. Distinguimos 3 casos:

j) Si $\lambda < 0$, obtenemos la siguiente tabla:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$N(x)$
$-\infty$	-	+	-	+	3
0	-	-	0	+	1
$+\infty$	+	+	+	+	0

$$\begin{cases} N_- = N(-\infty) - N(0) = 2 \\ N_+ = N(0) - N(+\infty) = 1 \end{cases}$$

Luego por la regla de signos P tiene dos raíces positivas y una negativa (recuerda que todas son simples).

jj) Si $0 < \lambda < \frac{1}{3}$, tenemos:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$N(x)$
$-\infty$	-	+	-	-	2
0	+	+	0	-	1
$+\infty$	+	+	+	-	1

$$\begin{cases} N_- = N(-\infty) - N(0) = 1 \\ N_+ = N(0) - N(+\infty) = 0 \end{cases}$$

Luego por la regla de signos P tiene una raíz negativa y las otras son complejas conjugadas.

jjj) Si $\lambda > \frac{1}{3}$, tenemos:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$N(x)$
$-\infty$	-	+	+	-	2
0	+	+	0	-	1
$+\infty$	+	+	+	-	1

$$\begin{cases} N_- = N(-\infty) - N(0) = 1 \\ N_+ = N(0) - N(+\infty) = 0 \end{cases}$$

Luego por la regla de signos P tiene una raíz negativa y las otras son complejas conjugadas.

b) Si $\lambda = \sqrt{3} > \frac{1}{3}$ Luego por jjj) del apartado a) tenemos que P tiene una raíz negativa y 2 complejas conjugadas.

$$P(x) = 9x^3 + 9x^2 + 9\sqrt{3}x + \sqrt{3}$$

Notar que 2 de los coeficientes de P son irracionales, luego no podemos aplicar los anteriores métodos de separación de raíces.

$$\begin{cases} P'(x) = 9(3x^2 + 2x + \sqrt{3}) > 0, & x \in \mathbb{R} \\ P''(x) = 18(3x + 1) \implies & \begin{cases} < 0, & x < -\frac{1}{3} \\ = 0, & x = -\frac{1}{3} \\ > 0, & x > -\frac{1}{3} \end{cases} \end{cases}$$

Como $P\left(-\frac{1}{6}\right) < 0 < P(0)$ y $P'(x) > 0$ y $P''(x) > 0$, $x \in \left[-\frac{1}{6}, 0\right]$ la sucesión del método de Newton converge de forma cuadrática a la raíz negativa de P con valor inicial $x_0 = 0$.

$$x_n = x_{n-1} - \frac{9x_{n-1}^3 + 9x_{n-1}^2 + 9\sqrt{3}x_{n-1} + \sqrt{3}}{9(3x_{n-1}^2 + 2x_{n-1} + \sqrt{3})}; \forall n \in \mathbb{N}$$

Si aplicas el método de Newton, con 5 iteraciones obtienes la raíz negativa aproximada $r_1 \approx -0.11822700646197$

Ejercicio 5.0.5 ► Polinomio Coeficientes PI

Calcular las raíces del polinomio:

$$P(x) = 2x^5 - \pi x^4 - 8x^3 + 4\pi x^2 + 8x - 4\pi$$

Resolución

Calculemos la secuencia de Sturm:

$$P_1(x) = \frac{P'(x)}{2} = 5x^4 - 2\pi x^3 - 12x^2 + 4\pi x + 4$$

Dividimos $P(x)$ entre $P_1(x)$:

$$P(x) = \left(\frac{2x}{5} - \frac{\pi}{25}\right) P_1(x) + \frac{2}{25} \left(-(\pi^2 + 40)x^3 + 24\pi x^2 + 2(\pi^2 + 40)x - 48\pi\right)$$

Por tanto, tomamos:

$$P_2(x) = (\pi^2 + 40)x^3 - 24\pi x^2 - 2(\pi^2 + 40)x + 48\pi$$

Dividimos $P_1(x)$ entre $P_2(x)$:

$$P_1(x) = \left(\frac{5}{\pi^2 + 40}x + \frac{2\pi(20 - \pi^2)}{(\pi^2 + 40)^2} \right) P_2(x) + 50 \frac{\pi^4 - 16\pi^2 + 64}{(\pi^2 + 40)^2} (2 - x^2)$$

Así pues, tenemos:

$$P_3(x) = x^2 - 2$$

Dividimos $P_2(x)$ entre $P_3(x)$:

$$P_2(x) = \left((\pi^2 + 40)x - 24\pi \right) P_3(x)$$

Luego $P_4(x) = 0$. Luego la secuencia de Sturm es: $\{P(x), P_1(x), P_2(x), P_3(x)\}$

Sabemos que $\text{mcd}(P(x), P'(x)) = P_3(x) = x^2 - 2 = (x - \sqrt{2})(x + \sqrt{2})$. Luego $\pm\sqrt{2}$ son raíces dobles de $P(x)$ porque $P(\pm\sqrt{2}) = 0$. Aplicamos Ruffini:

	2	$-\pi$	-8	4π	8	-4π
$\sqrt{2}$	↓	$2\sqrt{2}$	$4 - \sqrt{2}\pi$	$-4\sqrt{2} - 2\pi$	$-8 + 2\sqrt{2}\pi$	4π
$\sqrt{2}$	2	$2\sqrt{2} - \pi$	$-4 - \sqrt{2}\pi$	$-4\sqrt{2} + 2\pi$	$2\sqrt{2}\pi$	0
	↓	$2\sqrt{2}$	$8 - \sqrt{2}\pi$	$4\sqrt{2} - 4\pi$	$-2\sqrt{2}\pi$	
$-\sqrt{2}$	2	$4\sqrt{2} - \pi$	$4 - 2\sqrt{2}\pi$	-2π	0	
	↓	$-2\sqrt{2}$	$-4 + \sqrt{2}\pi$	2π		
$-\sqrt{2}$	2	$2\sqrt{2} - \pi$	$-\sqrt{2}\pi$	0		
	↓	$-2\sqrt{2}$	$\sqrt{2}\pi$			
$\pi/2$	2	$-\pi$	0			
	↓	π				
	2	0				

Factorizamos el polinomio $P(x)$:

$$P(x) = 2 \cdot (x - \sqrt{2}) \cdot^2 (x + \sqrt{2}) \cdot^2 \left(x - \frac{\pi}{2}\right)$$

Luego tenemos como raíces de $P(x)$: $\pm\sqrt{2}$, ambas son raíces dobles y $\frac{\pi}{2}$, que es raíz simple.

Ejercicio 5.0.6 ► Polinomio Grado Cinco

Aproximar las raíces reales de la ecuación algebraica:

$$2x^5 - 100x^2 + 2x - 1 = 0$$

Resolución

Definimos $P(x) = 2x^5 - 100x^2 + 2x - 1$, luego $P(-x) = -2x^5 - 100x^2 - 2x - 1$. Tenemos las secuencias de signos:

$$\begin{cases} P(x) \longrightarrow \{2, -100, 2, -1\} \Rightarrow 3 \text{ cambios de signo} \\ P(-x) \longrightarrow \{-2, -100, -2, -1\} \Rightarrow 0 \text{ cambios de signo} \end{cases}$$

Por el método de Descartes sabemos que no hay raíces negativas y puede tener 1 ó 3 raíces positivas. Acotemos las raíces:

$$\begin{aligned} \lambda &= \max \left\{ \frac{1}{2}, 1, 50 \right\} = 50 \Rightarrow 1 + \lambda = 51 \\ \mu &= \max \{2, 100, 2\} = 100 \Rightarrow \frac{1}{1 + \mu} = \frac{1}{101} \end{aligned}$$

Luego las raíces están en la corona circular:

$$C = \left\{ z \in \mathbb{C} : \frac{1}{101} < |z| < 51 \right\}$$

Esta acotación nos da muy poca información porque el tamaño de la corona es demasiado grande.

No tiene raíces enteras porque $P(1) = -97 \neq 0$. Como $a_5 = 2$, el único candidato a raíz racional es $\frac{1}{2}$, que no lo es porque $P\left(\frac{1}{2}\right) \neq 0$. Luego las raíces son o irracionales o complejas, pero al ser de grado 5 (impar) podemos asegurar que existe al menos una raíz irracional positiva. Realicemos la separación de raíces.

$$P'(x) = 10x^4 - 200x + 2 = P_1(x)$$

Antes de empezar a hacer las divisiones polinomiales, como van a ser muy engorrosas hacerlas a mano las vamos a hacer con el comando *deconv* de Octave. Lo que debes saber es que todo polinomio lo podemos expresar como el producto matricial de una matriz fila formada por los

coeficientes del polinomio y una matriz columna con sus correspondientes monomios de la incógnita. Por ejemplo, para $P(x)$ lo podemos expresar como:

$$P(x) = 2x^5 - 100x^2 + 2x - 1 = \begin{pmatrix} 2 & 0 & 0 & -100 & 2 & -1 \end{pmatrix} \cdot \begin{pmatrix} x^5 \\ x^4 \\ x^3 \\ x^2 \\ x \\ 1 \end{pmatrix}$$

Para el comando *deconv* sólo necesitamos el de los coeficientes. Luego en nuestro caso, para obtener la secuencia de Sturm ejecutamos los siguientes comandos en Octave:

```
1 [c1 r1]=deconv([2,0,0,-100,2,-1], [10,0,0,-200,2])
2 [c2 r2]=deconv([10,0,0,-200,2], [60,-1.6,1])
3 [c3 r3]=deconv([60,-1.6,1], [200.0087,-2.0027])
```

Luego nuestra secuencia de Sturm es $\{P(x), P_1(x), P_2(x), P_3(x), P_4(x)\}$:

$$\begin{aligned} P(x) &= 2x^5 - 100x^2 + 2x - 1 \\ P_1(x) &= 10x^4 - 200x + 2 \\ P_2(x) &= 60x^2 - 1.6x + 1 \\ P_3(x) &= 200.0087x - 2.0027 \\ P_4(x) &= -0.98999 \end{aligned}$$

Ahora montamos la tabla de signos sin considerar $-\infty$ porque ya sabemos que no hay raíces negativas:

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$N(x)$
0	-	+	+	-	-	2
$+\infty$	+	+	+	+	-	1

Luego $N_+ = N(0) - N(+\infty) = 2 - 1 = 1$. Por tanto, sólo hay una raíz racional positiva simple, las otras 4 son complejas conjugadas simples o dobles. Debemos encontrar un intervalo lo más pequeño posible en el que se encuentre la raíz real que buscamos. Lo más sencillo es dibujar su gráfica y ver que podemos establecer como intervalo el $]3, 4[$. Aplicando el método de Newton con $x_0 = 4$ obtenemos como raíz aproximada $\omega_1 = 3.67825295048808$. Con esto terminamos el problema.

Ejercicio 5.0.7 ► Raíces Complejas

Hallar las raíces reales y complejas, determinando su multiplicidad, de la ecuación:

$$x^{13} - x^{12} - x^{11} + x^{10} - x^3 + x^2 + x - 1 = 0$$

Resolución

Tomamos:

$$P(x) = x^{13} - x^{12} - x^{11} + x^{10} - x^3 + x^2 + x - 1 = 0$$

Vamos a empezar de forma distinta, nos fijamos que $a_0 = -1$, luego ± 1 son los candidatos a ser raíces enteras, de hecho lo son, ya que $P(\pm 1) = 0$. Como $a_{13} = 1$, no tenemos raíces racionales. Ahora para simplificar cálculos vamos a factorizar nuestro polinomio y determinar las multiplicidades de ± 1 . Para ello vamos a aplicar Ruffini:

	1	-1	-1	1	0	0	0	0	0	0	-1	1	1	-1
1	↓	1	0	-1	0	0	0	0	0	0	0	-1	0	1
	1	0	-1	0	0	0	0	0	0	0	-1	0	1	0
1	↓	1	1	0	0	0	0	0	0	0	0	-1	-1	
	1	1	0	0	0	0	0	0	0	0	-1	-1	0	
1	↓	1	2	2	2	2	2	2	2	2	2	1		
	1	2	2	2	2	2	2	2	2	2	1	0		
-1	↓	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1			
	1	1	1	1	1	1	1	1	1	1	0			
-1	↓	-1	0	-1	0	-1	0	-1	0	-1				
	1	0	1	0	1	0	1	0	1	0				

Por tanto, tenemos la siguiente factorización del polinomio $P(x)$:

$$P(x) = (x-1)^3 \cdot (x+1)^2 \cdot (x^8 + x^6 + x^4 + x^2 + 1)$$

Luego 1 es raíz triple y -1 es raíz doble. Aplicando la deflación vamos a pasar a calcular las raíces de $Q(x) := x^8 + x^6 + x^4 + x^2 + 1$, las cuales es obvio que serán también las raíces de $P(x)$. De esta forma hemos pasado de trabajar con un polinomio de grado 13 a trabajar con un polinomio de grado 8. Es obvio que $Q(\pm 1) \neq 0$, y al ser su término independiente 1 aseguramos que $Q(x)$ no tiene raíces enteras. Pero $a_8 = 1$, luego tampoco tiene raíces racionales, lo cual ya sabíamos porque $P(x)$ no las tiene. Luego sus raíces son irracionales o complejas.

Como $Q(x) = Q(-x)$ no poseen cambios de signo, por la Regla de Signos de Descartes aseguramos que no tiene raíces reales positivas o negativas. Luego todas las raíces que quedan por

calcular son complejas. La cosa se nos complica bastante. Fíjate que todos los grados de $Q(x)$ son pares y van de 2 en 2, por tanto; vamos a realizar el siguiente cambio de variable:

$$t = x^2$$

De esa forma obtenemos un nuevo polinomio de grado 4 que es con el que vamos a trabajar. Posteriormente, para saber cuáles son las raíces de $Q(x)$ tendremos que deshacer el cambio de variable.

$$R(t) := t^4 + t^3 + t^2 + t + 1$$

Este polinomio no es bicuadrado, luego no podemos aplicarle el típico cambio de variable para resolver su ecuación algebraica. Una forma de hacerlo es introduciendo una raíz de forma que podamos convertir el polinomio a la forma $t^5 + a = 0$, el cual podemos resolver. Fíjate:

$$\begin{aligned} (t+a)(t^4+t^3+t^2+t+1) &= t^5+a \\ t^5+t^4+t^3+t^2+t+at^4+at^3+at^2+at+a &= t^5+a \\ t^5+(1+a)t^4+(1+a)t^3+(1+a)t^2+(1+a)t+a &= t^5+a \end{aligned}$$

Identificando (igualando) coeficientes tenemos:

$$1+a=0 \implies a = -1$$

Por tanto, hemos de resolver la ecuación $t^5 - 1 = 0 \implies t^5 = 1 \implies t = \sqrt[5]{1}$. Es decir, hemos de calcular las raíces quintas del complejo (real en nuestro caso) $z = 1$.

Tenemos: $|z| = 1$, $\arg(z) = 0$. Así pues:

$$z_k = r \frac{\arg(z) + 2k\pi}{5} = 1 \frac{2k\pi}{5}; 0 \leq k \leq 4$$

Y obtenemos las siguientes raíces:

$$\begin{aligned}
 z_0 &= 1_0 = 1 \\
 z_1 &= 1_{\frac{2\pi}{5}} \\
 z_2 &= 1_{\frac{4\pi}{5}} \\
 z_3 &= 1_{\frac{6\pi}{5}} \\
 z_4 &= 1_{\frac{8\pi}{5}}
 \end{aligned}$$

Luego las raíces de $R(t)$ son:

$$\begin{aligned}
 t_1 &= 1_{\frac{2\pi}{5}} \\
 t_2 &= 1_{\frac{4\pi}{5}} \\
 t_3 &= 1_{\frac{6\pi}{5}} \\
 t_4 &= 1_{\frac{8\pi}{5}}
 \end{aligned}$$

Fíjate que $z_0 = 1$ no es raíz de $R(t)$, ya que es la que hemos introducido de forma ficticia para poder hallar de forma algebraica las raíces de $R(t)$.

Nos queda deshacer el cambio de variable para poder saber cuáles son las 8 raíces complejas simples de $Q(x) = x^8 + x^6 + x^4 + x^2 + 1$. Tenemos:

$$t = x^2 \implies x = \pm\sqrt{t}$$

Sustituyendo de 1 en 1 tenemos, $k = 0, 1$:

Notar que $-1 = 1_\pi$, y que $(r_\alpha) \cdot (r'_\beta) = (r \cdot r')_{\alpha+\beta}$

$$\begin{aligned}
x_{1,2} &= \pm \left[1 \frac{2\pi}{5} \right]^{1/2} = \pm 1 \frac{\pi}{5+k\pi} = \begin{cases} 1 \frac{\pi}{5+k\pi} = \left\{ 1 \frac{\pi}{5}, 1 \frac{6\pi}{5} \right\} \\ 1_{\pi} \cdot 1 \frac{\pi}{5+k\pi} = 1 \frac{6\pi}{5+k\pi} = \left\{ 1 \frac{6\pi}{5}, 1 \frac{11\pi}{5} \right\} = \left\{ 1 \frac{6\pi}{5}, 1 \frac{\pi}{5} \right\} \end{cases} \\
x_{3,4} &= \pm \left[1 \frac{4\pi}{5} \right]^{1/2} = \pm 1 \frac{2\pi}{5+k\pi} = \begin{cases} 1 \frac{2\pi}{5+k\pi} = \left\{ 1 \frac{2\pi}{5}, 1 \frac{7\pi}{5} \right\} \\ 1_{\pi} \cdot 1 \frac{2\pi}{5+k\pi} = 1 \frac{7\pi}{5+k\pi} = \left\{ 1 \frac{7\pi}{5}, 1 \frac{12\pi}{5} \right\} = \left\{ 1 \frac{7\pi}{5}, 1 \frac{2\pi}{5} \right\} \end{cases} \\
x_{5,6} &= \pm \left[1 \frac{6\pi}{5} \right]^{1/2} = \pm 1 \frac{3\pi}{5+k\pi} = \begin{cases} 1 \frac{3\pi}{5+k\pi} = \left\{ 1 \frac{3\pi}{5}, 1 \frac{8\pi}{5} \right\} \\ 1_{\pi} \cdot 1 \frac{3\pi}{5+k\pi} = 1 \frac{8\pi}{5+k\pi} = \left\{ 1 \frac{8\pi}{5}, 1 \frac{13\pi}{5} \right\} = \left\{ 1 \frac{8\pi}{5}, 1 \frac{3\pi}{5} \right\} \end{cases} \\
x_{7,8} &= \pm \left[1 \frac{8\pi}{5} \right]^{1/2} = \pm 1 \frac{4\pi}{5+k\pi} = \begin{cases} 1 \frac{4\pi}{5+k\pi} = \left\{ 1 \frac{4\pi}{5}, 1 \frac{9\pi}{5} \right\} \\ 1_{\pi} \cdot 1 \frac{4\pi}{5+k\pi} = 1 \frac{9\pi}{5+k\pi} = \left\{ 1 \frac{9\pi}{5}, 1 \frac{14\pi}{5} \right\} = \left\{ 1 \frac{9\pi}{5}, 1 \frac{4\pi}{5} \right\} \end{cases}
\end{aligned}$$

Quitando las repetidas tenemos que las raíces de $Q(x) = x^8 + x^6 + x^4 + x^2 + 1$ son:

$$\left\{ 1 \frac{\pi}{5}, 1 \frac{6\pi}{5}, 1 \frac{2\pi}{5}, 1 \frac{7\pi}{5}, 1 \frac{3\pi}{5}, 1 \frac{8\pi}{5}, 1 \frac{4\pi}{5}, 1 \frac{9\pi}{5} \right\}$$

Como las raíces de $Q(x)$ lo son también de $P(x)$, tenemos que las raíces de $P(x)$ son:

$$\left\{ \pm 1, 1 \frac{\pi}{5}, 1 \frac{6\pi}{5}, 1 \frac{2\pi}{5}, 1 \frac{7\pi}{5}, 1 \frac{3\pi}{5}, 1 \frac{8\pi}{5}, 1 \frac{4\pi}{5}, 1 \frac{9\pi}{5} \right\}$$

Todas las raíces complejas conjugadas son simples, 1 es triple y -1 es doble. Con esto ya hemos calculado todas las raíces de $P(x)$ porque la suma de todas las multiplicidades nos da 13, que es el grado de nuestro polinomio $P(x)$.

Ejercicio 5.0.8 ► Acotar Raíces

Acotar las raíces de las siguientes ecuaciones:

a) $x^5 - x^4 + x^3 - x^2 + 1 = 0$

b) $x^7 - 5x^6 - 27x^5 + 3x^3 + 4x^2 + 7x - 2 = 0$

Resolución

a) $P(x) := x^5 - x^4 + x^3 - x^2 + 1$

$$\lambda = 1 \Rightarrow 1 + \lambda = 2$$

$$\mu = 1 \Rightarrow \frac{1}{1 + \mu} = \frac{1}{2}$$

Por tanto, tenemos: $C = \left\{ z \in \mathbb{C} : \frac{1}{2} < |z| < 2 \right\}$

b) $P(x) := x^7 - 5x^6 - 27x^5 + 3x^3 + 4x^2 + 7x - 2$

$$\lambda = \max \left\{ \left| \frac{-5}{1} \right|, \left| \frac{-27}{1} \right|, \left| \frac{3}{1} \right|, \left| \frac{4}{1} \right|, \left| \frac{7}{1} \right|, \left| \frac{-2}{1} \right| \right\} = 27 \Rightarrow 1 + \lambda = 28$$

$$\begin{aligned} \mu &= \max \left\{ \left| \frac{1}{-2} \right|, \left| \frac{-5}{-2} \right|, \left| \frac{-27}{-2} \right|, \left| \frac{3}{-2} \right|, \left| \frac{4}{-2} \right|, \left| \frac{7}{-2} \right| \right\} = \\ &= \max \left\{ \frac{1}{2}, \frac{5}{2}, \frac{27}{2}, \frac{3}{2}, 2, \frac{7}{2} \right\} = \frac{27}{2} = 13.5 \end{aligned}$$

$$\Rightarrow \frac{1}{1 + \mu} = \frac{1}{1 + \frac{27}{2}} = \frac{2}{29}$$

Por tanto, tenemos: $C = \left\{ z \in \mathbb{C} : \frac{2}{29} < |z| < 28 \right\}$

Ejercicio 5.0.9 ► Separación De Raíces

Separar en intervalos de longitud uno, mediante el método de Sturm, las raíces reales de las ecuaciones:

a) $x^5 - 3x^4 + 2x^3 - 3x^2 + 4x + 1 = 0$

b) $x^5 - x^4 + x^3 - x^2 + 1 = 0$

Resolución

a) $P(x) := x^5 - 3x^4 + 2x^3 - 3x^2 + 4x + 1$

$P(-x) = -x^5 - 3x^4 - 2x^3 - 3x^2 - 4x + 1$

Tenemos que en $P(x)$ hay 4 cambios de signo y en $P(-x)$ hay sólo un cambio de signo. Por tanto, hay una raíz negativa y pueden haber 4 ó 2 ó ninguna raíz positiva.

Tenemos que $a_0 = a_5 = 1$, con $P(\pm 1) \neq 0$. Luego todas las raíces reales que tiene son irracionales.

$$P'(x) = 5x^4 - 12x^3 + 6x^2 - 6x + 4 = P_1(x)$$

Vamos a aplicar el algoritmo de Euclides para calcular la secuencia de Sturm. En esta ocasión te voy a enseñar a hacerlo con wxMaxima ya que así vas a obtener los polinomios en forma simbólica (algebraica) que son exactos. Con Octave no se puede realizar el cálculo simbólico si no es instalando un paquete extra. En wxMaxima vamos a utilizar el comando **divide** para dividir polinomios, el resultado nos da el cociente y el resto de la división polinómica. Aquí tienes lo que he utilizado:

```
(%i1) divide(x^5-3*x^4+2*x^3-3*x^2+4*x+1,5*x^4-12*x^3+6*x^2-6*x+4,x);
```

```
(%o1) [5x-3, -16x^3+27x^2-62x-37]
      25      25
```

```
(%i2) divide(5*x^4-12*x^3+6*x^2-6*x+4,(16*x^3+27*x^2-62*x-37)/25,x);
```

```
(%o2) [2000x-8175, 15325x^2-18850x-11075]
      256      256
```

```
(%i3) divide((16*x^3+27*x^2-62*x-37)/25,(-15325*x^2+18850*x+11075)/(256),x);
```

```
(%o3) [-2510848x+7325440, 2622976x-1227008]
      234855625      9394225
```

```
(%i4) divide((-15325*x^2+18850*x+11075)/(256),(-2622976*x+1227008)/(9394225),x);
```

```
(%o4) [1475080739788750x-1124341947734375, 1736306424075]
      6880003096576      26875012096
```


De esta forma, obtenemos la secuencia de Sturm:

$$\left\{ P(x), P_1(x), \frac{16}{25}x^3 + \frac{27}{25}x^2 - \frac{62}{25}x - \frac{37}{25}, -\frac{15325}{256}x^2 + \frac{18850}{256}x + \frac{11075}{256}, -\frac{2622976}{9394225}x + \frac{1227008}{9394225}, -1 \right\}$$

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$P_5(x)$	$N(x)$
$-\infty$	-	+	-	-	+	-	4
0	+	+	-	+	+	-	3
$+\infty$	+	+	+	-	-	-	1

Tenemos pues:

$$\begin{cases} N_- = N(-\infty) - N(0) = 4 - 3 = 1 \\ N_+ = N(0) - N(+\infty) = 3 - 1 = 2 \end{cases}$$

Luego obtenemos que nuestro polinomio tiene 1 raíz negativa y 2 raíces positivas, las 3 son simples porque $P_5(x) = -1$ es constante. Además, tiene 2 raíces complejas conjugadas simples. Nos falta determinar los intervalos de longitud 1 para cada una de las 3 raíces reales (irracionales).

$$\lambda = \max\{3, 2, 3, 4, 1\} = 4 \Rightarrow 1 + \lambda = 5$$

Por tanto, tomamos como intervalo de partida:

$$[-1 - \lambda, 1 + \lambda] = [-5, 5]$$

Como sólo hay 1 raíz negativa, sabemos que estará en $[-5, 0]$, y las 2 positivas en $[0, 5]$. Y ahora vamos a hacerlo fácil, te puedes ayudar dibujando la gráfica del polinomio:

$P(-1) \cdot P(0) < 0 \Rightarrow [-1, 0]$ es nuestro intervalo de longitud 1 en el que se encuentra la raíz negativa (Teorema Bolzano).

$P(1) \cdot P(2) < 0$ $P(2) \cdot P(3) < 0 \Rightarrow [1, 2]$ y $[2, 3]$ son los 2 intervalos de longitud 1 en los que se encuentran las positivas (Teorema Bolzano).

$$\text{b) } P(x) := x^5 - x^4 + x^3 - x^2 + 1$$

$$P(-x) = -x^5 - x^4 - x^3 - x^2 + 1$$

Tenemos que $P(x)$ tiene 4 cambios de signo y que $P(-x)$ tiene un cambio de signo, luego nuestro polinomio tiene una raíz negativa y puede tener 0, 2 ó 4 raíces positivas.

Como $a_0 = a_5 = 1$ y $P(1) \neq 0$, $P(-1) \neq 0$, podemos asegurar que sus raíces son irracionales o complejas.

$$P'(x) = 5x^4 - 4x^3 + 3x^2 - 2x = P_1(x)$$

Halleemos como en el apartado anterior la secuencia de Sturm:

```
(%i6) divide(x^5-x^4+x^3-x^2+1,5*x^4-4*x^3-2*x,x);
```

```
(%o6) [5x-1, 21x^3-15x^2-2x+25]
```

```
(%i7) divide(5*x^4-4*x^3-2*x,-21*x^3+15*x^2+2*x-25,x);
```

```
(%o7) [-35x-3, 25x^2-1175x+75]
```

```
(%i8) divide(-21*x^3+15*x^2+2*x-25,-x^2+47*x-3,x);
```

```
(%o8) [21x+972,2891-45619x]
```

```
(%i9) divide(-x^2+47*x-3,931*x-59,x);
```

```
(%o9) [-931x-43698, -22101]
```

Obtenemos la siguiente secuencia de Sturm:

$$\{P(x), P_1(x), -21x^3 + 15x^2 + 2x - 25, -x^2 + 47x - 3, 931x - 59, 1\}$$

x	$P(x)$	$P_1(x)$	$P_2(x)$	$P_3(x)$	$P_4(x)$	$P_5(x)$	$N(x)$
$-\infty$	-	+	+	-	-	+	3
0	+	0	-	-	-	+	2
$+\infty$	+	+	-	-	+	+	2

$$\begin{cases} N_- = N(-\infty) - N(0) = 3 - 2 = 1 \\ N_+ = N(0) - N(+\infty) = 2 - 2 = 0 \end{cases}$$

Luego tenemos que nuestro polinomio sólo tiene una raíz negativa y 4 raíces complejas conjugadas, todas ellas simples. Aplicando el Teorema de Bolzano:

$P(-1) \cdot P(0) = -3 < 0 \Rightarrow [-1, 0]$ es un intervalo de longitud 1 en el que se encuentra la raíz negativa.

Ejercicio 5.0.10 ► Aproximamos Raíces

Aproximar las raíces del polinomio:

$$P(x) = 5x^5 - 17x^4 - 79x^3 + 269x^2 - 34x - 24$$

Resolución

$$P(-x) = -5x^5 - 17x^4 + 79x^3 + 269x^2 + 34x - 24$$

En $P(x)$ hay 3 cambios de signo y en $P(-x)$ hay 2 cambios de signo, luego por la regla de signos de Descartes pueden haber 2 raíces negativas y 3 positivas, o pueden haber 2 negativas, 1 positiva y 2 complejas conjugadas.

Aplicamos la acotación de McLaurin:

$$\begin{aligned}\lambda &= \max \left\{ \frac{17}{5}, \frac{79}{5}, \frac{269}{5}, \frac{34}{5}, \frac{24}{5} \right\} = \frac{269}{5} \Rightarrow 1 + \lambda = 1 + \frac{269}{5} = 54.8 \\ \mu &= \max \left\{ \frac{5}{24}, \frac{17}{24}, \frac{79}{24}, \frac{269}{24}, \frac{34}{24} \right\} = \frac{269}{24} \Rightarrow \frac{1}{1 + \mu} = \frac{1}{1 + \frac{269}{24}} \approx 0.082\end{aligned}$$

Luego las raíces están dentro de la siguiente corona circular:

$$C = \{z \in \mathbb{C} : 0.082 < |z| < 54.8\}$$

Tenemos $a_0 = -24$, luego todos sus divisores son:

$$\{\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 8, \pm 12, \pm 24\}$$

Se tiene que $P(-4) = 0$ y que $P(3) = 0$, todos los demás son no cero, luego $\omega_0 = -4, \omega_1 = 3$ son raíces enteras del polinomio.

Tenemos que $a_5 = 5$, luego sus divisores son $\{\pm 1, \pm 5\}$, por los que los candidatos a raíces racionales son:

$$\left\{ \pm \frac{1}{5}, \pm \frac{2}{5}, \pm \frac{3}{5}, \pm \frac{4}{5}, \pm \frac{6}{5}, \pm \frac{8}{5}, \pm \frac{12}{5}, \pm \frac{24}{5} \right\}$$

De todas ellas sólo $P\left(\frac{2}{5}\right) = 0$, las demás no se anulan, luego $\omega_2 = \frac{2}{5}$ es la única raíz racional del polinomio. Vamos a aplicar Ruffini con las 3 raíces para obtener un polinomio de grado 2 cuyas raíces van a ser también de $P(x)$, es decir; vamos a aplicar la deflación.

	5	-17	-79	269	-34	-24
-4	↓	-20	148	-276	28	24
	5	-37	69	-7	-6	0
2/5	↓	2	-14	22	6	
	5	-35	55	15	0	
3	↓	15	-60	-15		
	5	-20	-5	0		

Factorizamos $P(x)$:

$$P(x) = 25 \cdot (x+4) \cdot \left(x - \frac{2}{5}\right) \cdot (x-3) \cdot (x^2 - 4x - 1)$$

Por tanto, vamos a calcular las raíces del polinomio:

$$Q(x) := x^2 - 4x - 1$$

Ahora

$$Q(x) = x^2 - 4x - 1 = 0 \Rightarrow x = \frac{4 \pm \sqrt{20}}{2} = \frac{4 \pm 2\sqrt{5}}{2} = 2 \pm \sqrt{5}$$

Luego hemos obtenido todas las raíces de $P(x)$ de forma exacta. Si queremos obtener valores aproximados de las irracionales, tan sólo hemos de aplicarles el método de Newton.

Ejercicio 5.0.11 ► Estudiamos Raíces

Se considera la ecuación algebraica:

$$x^5 + x^4 + 5x^3 + 2x^2 - 13x - 10 = 0$$

- Determinar el número de raíces positivas.
- Encontrar una raíz racional negativa.
- Hallar el número de raíces reales y complejas de la ecuación anterior.

Resolución

$$\text{Definimos } P(x) := x^5 + x^4 + 5x^3 + 2x^2 - 13x - 10$$

$$P(-x) = -x^5 + x^4 - 5x^3 + 2x^2 + 13x - 10$$

Luego $P(x)$ tiene 1 cambio de signo y $P(-x)$ tiene 4 cambios de signo. Por la regla de signos de Descartes tenemos que $P(x)$ tiene 1 raíz positiva y puede tener 4 ó 2 raíces negativas.

Tenemos que $a_0 = -10$, luego sus divisores son $\{\pm 1, \pm 2, \pm 5, \pm 10\}$

$P(-1) = 0$ y todos los demás son no nulos, por tanto; tenemos que -1 es la única raíz entera del polinomio. Vamos a aplicar la deflación para simplificar cálculos:

$$\begin{array}{r|rrrrrr} & 1 & 1 & 5 & 2 & -13 & -10 \\ -1 & \downarrow & -1 & 0 & -5 & 3 & 10 \\ \hline & 1 & 0 & 5 & -3 & -10 & 0 \end{array}$$

Factorizamos: $P(x) = (x + 1) \cdot (x^4 + 5x^2 - 3x - 10)$

Trabajamos pues con $Q(x) = x^4 + 5x^2 - 3x - 10$. Se tiene que $a_4 = 1$, luego no tiene raíces racionales, en consecuencia, tampoco $P(x)$ tiene raíces racionales. Pero cuidado, porque podemos considerar que $-1 = \frac{-1}{1}$, de esa forma vemos que -1 es una raíz racional negativa.

Averigüemos la secuencia de Sturm de $Q(x)$

$$Q'(x) = 4x^3 + 10x - 3 = Q_1(x)$$

```
(%i33) divide(x^4+5*x^2-3*x-10,4*x^3+10*x-3,x);
```

```
(%o33) [ $\frac{x}{4}, \frac{10x^2 - 9x - 40}{4}$ ]
```

```
(%i34) divide(4*x^3+10*x-3,-10*x^2+9*x+40,x);
```

```
(%o34) [ $-\frac{10x + 9}{25}, \frac{731x + 285}{25}$ ]
```

```
(%i35) divide(-10*x^2+9*x+40,-731*x-285,x);
```

```
(%o35) [ $\frac{7310x - 9429}{534361}, \frac{18687175}{534361}$ ]
```

La secuencia de Sturm es la siguiente:

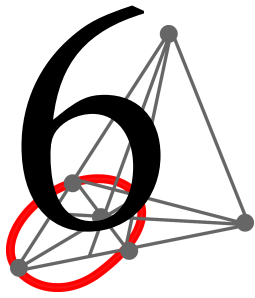
$$\{x^4 + 5x^2 - 3x - 10, 4x^3 + 10x - 3, -10x^2 + 9x + 40, -731x - 285, -1\}$$

x	$Q(x)$	$Q_1(x)$	$Q_2(x)$	$Q_3(x)$	$Q_4(x)$	$N(x)$
$-\infty$	+	-	-	+	-	3
0	-	-	+	-	-	2
$+\infty$	+	+	-	-	-	1

Por tanto, de la tabla se extrae la siguiente información:

$$\begin{cases} N_- = N(-\infty) - N(0) = 3 - 2 = 1 \\ N_+ = N(0) - N(+\infty) = 2 - 1 = 1 \end{cases}$$

Luego nuestro polinomio $Q(x)$ tiene 1 raíz negativa, 1 raíz positiva y 2 complejas conjugadas, porque al ser $Q_4(x)$ constante sabemos que todas las raíces de $Q(x)$ son simples. Como sabemos que $P(x)$ tiene una raíz negativa, y uniendo la información que hemos obtenido con $Q(x)$ concluimos: $P(x)$ tiene 1 raíz real positiva, 2 raíces reales negativas y 2 raíces complejas conjugadas, todas ellas simples.



CÁLCULO DE RAÍCES COMPLEJAS

6.1 Método De Müller

6.1.1. Introducción

El método de Müller es el primer método que se estudia para aproximar raíces complejas, se usa para funciones polinómicas, y aproxima tanto raíces complejas como reales. El método se inicia con 3 aproximaciones iniciales: x_0, x_1, x_2 , lo suficientemente cercanas a la raíz que se desea aproximar. Al ser 3 valores iniciales lo que hacemos es calcular la parábola que pasa por las 3 como segundo paso. Consideremos que queremos calcular las raíces de

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

La parábola a calcular es de la siguiente forma general:

$$P(x) = a(x - x_2) + b(x - x_2) + c$$

Dicha parábola es un polinomio de grado 2 centrado en x_2 , es decir; se considera como base $B = \{1, x - x_2, (x - x_2)^2\}$, en detrimento de la base canónica $B' = \{1, x, x^2\}$.

Para saber cuál es debemos calcular a, b, c , que los podemos determinar gracias a que la parábola debe pasar por los puntos: $(x_0, f(x_0))$, $(x_1, f(x_1))$, y $(x_2, f(x_2))$. Si sustituimos los 3 puntos en $P(x)$ obtenemos el siguiente sistema no lineal a resolver:

$$\begin{cases} f(x_0) = a(x_0 - x_2) + b(x_0 - x_2) + c \\ f(x_1) = a(x_1 - x_2) + b(x_1 - x_2) + c \\ f(x_2) = c \end{cases}$$

De la tercera ecuación del sistema ya tenemos determinada c , por sustitución y con un poco de paciencia tenemos:

$$a = \frac{[f(x_0) - f(x_2)](x_1 - x_2) - [f(x_1) - f(x_2)](x_0 - x_2)}{(x_0 - x_1)(x_0 - x_2)(x_1 - x_2)}$$

$$b = \frac{(x_0 - x_2)^2 [f(x_1) - f(x_2)] - (x_1 - x_2)^2 [f(x_0) - f(x_2)]}{(x_0 - x_1)(x_0 - x_2)(x_1 - x_2)}$$

$$c = f(x_2)$$

Una vez calculado el polinomio $P(x)$, debemos calcular la primera aproximación a la raíz a buscar, eso lo hacemos calculando el punto $(x_3, 0)$, lo que es equivalente a calcular las raíces del polinomio, es decir;

$$P(x_3) = a(x_3 - x_2)^2 + b(x_3 - x_2) + c = 0$$

Las soluciones son:

$$x_3 - x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para evitar la cancelación catastrófica racionalizamos y simplificamos:

$$x_3 - x_2 = \frac{(-b + \sqrt{b^2 - 4ac})(-b - \sqrt{b^2 - 4ac})}{2a(-b - \sqrt{b^2 - 4ac})} = \frac{b^2 - b^2 + 4ac}{2a(-b - \sqrt{b^2 - 4ac})} = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

Considerando las 2 opciones de más y menos, tenemos:

$$x_3 - x_2 = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

Para que la raíz aproximada x_3 esté lo más cercana a x_2 en el método de Müller se considera el signo de b , despejando x_3 en la anterior expresión y teniendo en cuenta el signo obtenemos finalmente:

$$x_3 = x_2 - \frac{2c}{b + \text{signo}(b) \sqrt{b^2 - 4ac}}$$

Eso sería la obtención de nuestra primera aproximación de la raíz, para obtener las siguientes iteraciones es suficiente hacer $x_0 = x_1, x_1 = x_2, x_2 = x_3$. Añadiendo las pertinentes condiciones de parada tenemos ya el método totalmente desglosado.

6.1.2. Ejemplo Gráfico Del Método De Müller

Para entender mejor el método vamos a ver un ejemplo gráfico del método. Consideremos la función polinómica $f(x) = 16x^4 - 40x^3 + 5x^2 + 20x + 6$. Tomemos como aproximaciones iniciales $x_0 = 0.5, x_1 = -0.5, x_2 = 0$. Como los cálculos a mano son muy laboriosos podemos crear con Octave el siguiente código:

```

1  function [] = mullergrafico(f)
2  x0 = 0.5;
3  x1 = -0.5;
4  x2 = 0;
5  x = -3:0.1:3;
6
7  fx = inline(f);
8  color = ['r','c','g','m'];
9  clf();
10 hold on;
11 for i=1:4
12     fx0 = feval(fx,x0);
13     fx1 = feval(fx,x1);
14     fx2 = feval(fx,x2);
15
16     delta1 = fx0-fx2;
17     delta2 = fx1-fx2;
18     delta3 = (x0-x1)*(x0-x2)*(x1-x2);
19
20     a = (delta1*(x1-x2)-delta2*(x0-x2))/delta3;
21     b = (delta2*(x0-x2)^2-delta1*(x1-x2)^2)/delta3;
22     c = fx2;
23
24     P = a*(x-x2).^2+b*(x-x2).^2+c;
25
26     plot(x,P,color(i))
27
28     x3 = x2 - (2*c)/(b+sign(b)*sqrt(b^2-4*a*c))
29
30     x0 = x1; x1 = x2; x2 = x3;
31 end
32
33
34 fplot(fx,[-1.5,1.5],'b','linewidth=3')
35 legend({'P1(x)', 'P2(x)', 'P3(x)', 'P4(x)', 'f(x)'});
36
37 legend boxon;
38 set(gca, 'axislocation', 'zero');

```

```

39 set(gca, 'yaxislocation', 'zero');
40 set(gca, 'box', 'off');
41 xlabel('y');
42 ylabel('x');
43 set(get(gca, 'ylabel'), 'rotation', 360);
44 grid;
45 hold off;
46
47 end

```

mullergrafico.m

Con dicho código dibujamos la función polinómica y las 4 parábolas que nos salen al iterar el método 4 veces, se puede ver en la siguiente figura:

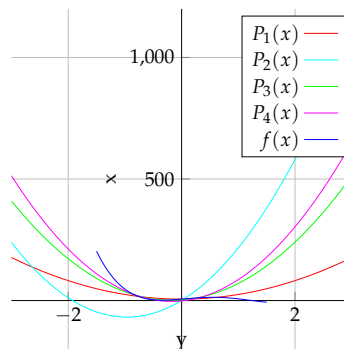


Figura 1: Müller para $f(x) = 16x^4 - 40x^3 + 5x^2 + 20x + 6$

El algoritmo también nos da las 4 aproximaciones a la raíz:

```

mullergrafico('16*x.^4-40*x.^3+5*x.^2+20*x+6')
x3 = -0.55556 + 0.59835i
x3 = -0.22806 + 0.11688i
x3 = -0.29146 + 0.19761i
x3 = -0.36873 + 0.19696i

```

Una cosa importante es que cuando la raíz aproximada es una raíz compleja sabemos que sin aplicar el método conocemos que otra de las raíces aproximadas es su raíz conjugada, esto se debe gracias al Teorema Fundamental del Algebra.

Como vemos en la figura el punto donde se cortan aproximadamente las 4 parábolas es la aproximación de la raíz. Aunque estamos representando en el plano real sabemos que todo número complejo lo “podemos” representar en dicho plano real identificando su parte real como la abscisa y su parte imaginaria como la ordenada.

El principal problema para buscar aproximaciones a raíces complejas en la gráfica de la función es que estas NO cortan al eje de abscisas, así que gráficamente sólo podemos apreciar las raíces reales y no las complejas. Tampoco nos sirve aplicar el Teorema De Bolzano. Afortunadamente hay técnicas para solventar estos inconvenientes y que se vió en los 2 capítulos anteriores.

6.1.3. Algoritmo De Müller

El algoritmo que se ha realizado bajo el programa Octave para éste documento es el siguiente:

```
1  % Probar con muller('fun_muller',-1,-0.5,0,0.00001,30)
2  function []=muller(f,p0,p1,p2,tol,maxiter)
3      a = p0;
4      b = p2;
5      fx = inline(f);
6      fplot(fx,[a-3,b+3])
7      set(gca,'axislocation','zero');
8      set(gca,'yaxislocation','zero');
9      set(gca,'box','off');
10     grid;
11     title('La Funcion');
12     legend({f},'location','southoutside');
13     legend boxon;
14     xlabel('y');
15     ylabel('x');
16     set(get(gca,'ylabel'),'rotation',360);
17
18     fx = inline(f);
19
20     h1=p1-p0;
21     h2=p2-p1;
22
23     d1=(feval(fx,p1)-feval(fx,p0))/h1;
24     d2=(feval(fx,p2)-feval(fx,p1))/h2;
25     d=(d2-d1)/(h2+h1);
26     count=3;
27     h=1;
28
29     while count < maxiter && abs(h)>tol
30         b=d2+h2*d;
31         D=sqrt(b^2-4*feval(fx,p2)*d);
32         if abs(b-D)<abs(b+D)
33             E=b+D;
34         else
35             E=b-D;
```

```
36     end
37     h=-2*feval(fx,p2)/E;
38     sol=p2+h;
39
40     if abs(h)>tol
41         p0=p1;
42         p1=p2;
43         p2=sol;
44         h1=p1-p0;
45         h2=p2-p1;
46         d1=(feval(fx,p1)-feval(fx,p0))/h1;
47         d2=(feval(fx,p2)-feval(fx,p1))/h2;
48         d=(d2-d1)/(h2+h1);
49         count++;
50     end
51 end
52 printf ('La solucion es %s.\nY su solucion conjugada es %s\nSe ha alcanzado en %d
53 iteraciones.\n',num2str(sol),num2str(conj(sol)),count);
end
```

muller.m

Bibliografía

- [1] Richard L Burden and J Douglas Faires. *Análisis numérico*, chapter 2, pages 74–88. Cengage Learning Brooks Cole, 3 edition, 1996. ISBN 9789706861344.
- [2] Alicia Cordero Barbero, José Luis Hueso Pagoaga, Eulalia Martínez Molada, and Juan Ramón Torregrosa Sánchez. *Problemas Resueltos De Métodos Numéricos*, chapter 2, pages 37–69. Editorial Paraninfo, 1 edition, 2006. ISBN 84-9732-409-9. URL www.paraninfo.es.
- [3] Juan Antonio Infante Del Río and Jose María Rey Cabezas. *Métodos Numéricos Teoría, problemas y prácticas con MATLAB*, chapter 10, pages 435–483. Ediciones Pirámide, 3 edition, 2007. ISBN 978-84-368-2090-4.
- [4] Juan Miguel Sánchez and Antonio Souto. *Problemas de cálculo numérico para ingenieros con aplicaciones Matlab*, chapter 1, pages 2–60. Schaum, 1 edition, 2005. ISBN 84-481-2951-2.
- [5] Wikipedia. Root finding algorithms, 2013. URL https://en.wikipedia.org/wiki/Category:Root-finding_algorithms.