

# RODRIGO PLAUCHÚ RODRÍGUEZ

## Primer Examen Parcial

Primavera 2022

Sube tu archivo resultado (con las respuestas visibles) a Canvas → Examen Parcial 1 antes de las 9:55 horas.

Cada pregunta vale 5 puntos.

Los exámenes son trabajos individuales. Está estrictamente prohibido dar o recibir ayuda de cualquier persona.

El artículo 29 del Reglamento de Alumnos establece que "se calificará como no acreditada (N.A.) cualquier materia cuando el alumno incurra en alguna práctica fraudulenta".

Recuerda salvar frecuentemente tu archivo.

Puede ser conveniente limpiar periódicamente el espacio de trabajo (comando clear).

En el caso extremo de gráficas que aparecen mal, puedes salir y volver a entrar a la sesión.

### Pregunta 1.

¿Qué número real  $x$  está representado en punto flotante de doble precisión por la siguiente cadena hexadecimal?

```
hex = '3fe921fb54442d18';    % estándar IEEE 754
bin = '00111111110100100100001111101101010100010001000010110100011000';
```

Extrae los diferentes campos del número binario *bin*

```
format long
s = bin(1)
```

```
s =
'0'
```

```
exp = bin(2:12)
```

```
exp =
'01111111110'
```

```
m = bin(13:64)
```

```
m =
'1001001000011111101101010100010001000010110100011000'
```

Calcula el signo  $s$  (-1 o 1)

```
sig = ((-1)^s)
```

```
sig =  
1
```

Calcula el exponente con bias  $be$

```
be = bin2dec(exp)
```

```
be =  
1022
```

Calcula el exponente  $e$

```
e = be - 1023
```

```
e =  
-1
```

Calcula la mantisa  $f < 1$

```
man = (bin2dec(m)/2^52) + 1
```

```
man =  
1.570796326794897
```

Calcula el número real  $x$

```
x1 = (sig)*(2^e)*(man) %pi/4
```

```
x1 =  
0.785398163397448
```

## Pregunta 2.

Observa el resultado de la siguiente expresión.

```
res = (7/100)*100 - 7
```

```
res =  
8.881784197001252e-16
```

¿Por qué  $res$  no es igual a 0?

```
% Por errores de redondeo de MatLab
```

Encuentra  $n$  tal que  $res = \text{eps}(n)$  y verifica tu respuesta usando la función de MATLAB *isequal*

```
e = eps(7)
```

```
e =  
8.881784197001252e-16
```

```
isequal(res, e)
```

```
ans = logical  
1
```

### Pregunta 3.

Calcula el error de la aproximación de la primera derivada centrada de cinco puntos

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$$

de la función  $f(x) = \sin(x^2)$  en el punto  $x = 2$  como función del espaciado  $h$  entre los valores de  $x$ .

Calcula el valor exacto  $dftrue$  de la derivada de la función  $f(x) = \sin(x^2)$  en el punto  $x = 2$ .

```
f = @(x) sin(x.^2);  
fd = matlabFunction(diff(sym(f)));           % = @(x) 2*x.*cos(x.^2)  
x = 2;  
dftrue = fd(x)
```

```
dftrue =  
-2.614574483454448
```

Calcula el valor absoluto  $te$  del error para  $h = [1, 10^{-1}, 10^{-2}, 10^{-3}, \dots, 10^{-9}]$ .

```
h = 10
```

```
h =  
10
```

```
i = 0:1:h-1
```

```
i = 1x10  
0 1 2 3 4 5 6 7 8 9
```

```
h = 1./10.^i
```

```
h = 1x10  
1.0000000000000000 0.1000000000000000 0.0100000000000000 0.0010000000000000 ...
```

```
apr = (f(x-2*h)-8*f(x-h)+8*(x+h)-f(x+2*h))./(12*h)
```

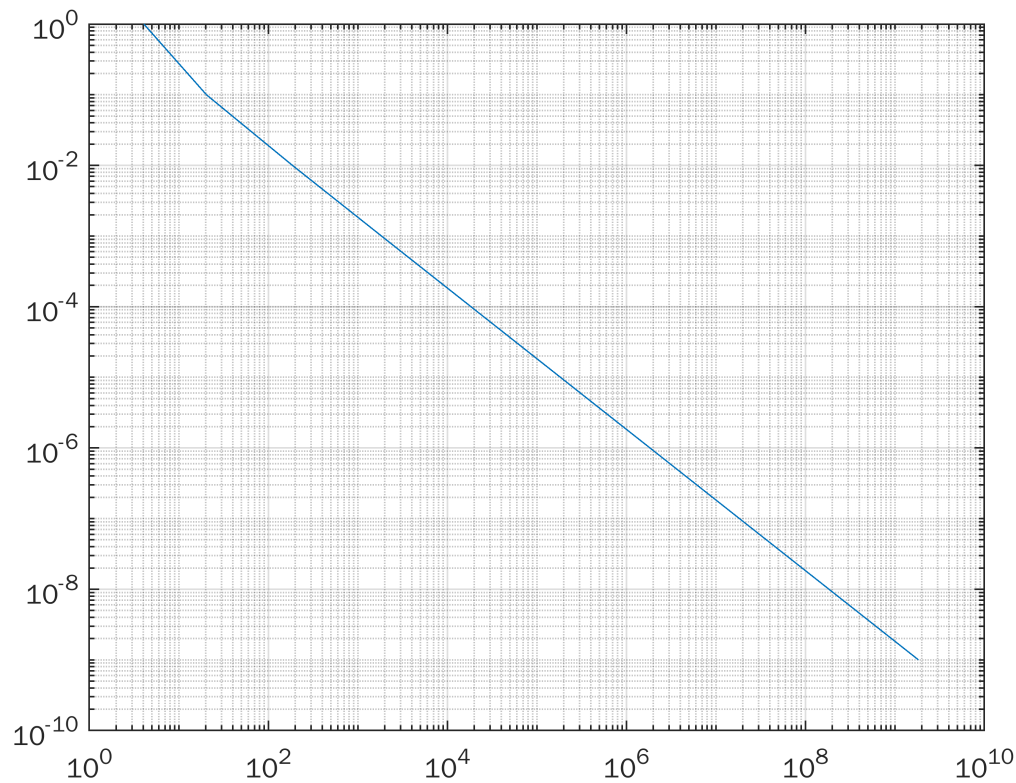
```
apr = 1x10  
109 ×  
0.000000001463011 0.000000017754455 0.000000183545305 0.000001837659865 ...
```

```
te = abs(dftrue - apr)
```

```
te = 1x10  
109 ×  
0.000000004077586 0.000000020369030 0.000000186159880 0.000001840274439 ...
```

Gráfica el valor absoluto del error  $te$  vs.  $h$  en una escala logarítmica ( $\log\log$ )

```
loglog(te, h)  
grid on
```



Calcula el valor de  $h$  que debe usarse para tener la mejor exactitud.

```
usar = min(te)
```

```
usar =  
4.077585769971272
```

Dado que el error de truncamiento de la aproximación es  $\frac{1}{30} f^{(5)}(\xi) h^4$ , donde  $\xi$  es un punto en el dominio, el error debería disminuir siempre que se reduzca  $h$ . Explica entonces ¿por qué aumenta el error al reducir  $h$  a partir del mínimo encontrado?

```
% Por errores de redondeo de Matlab  
%
```

## Pregunta 4.

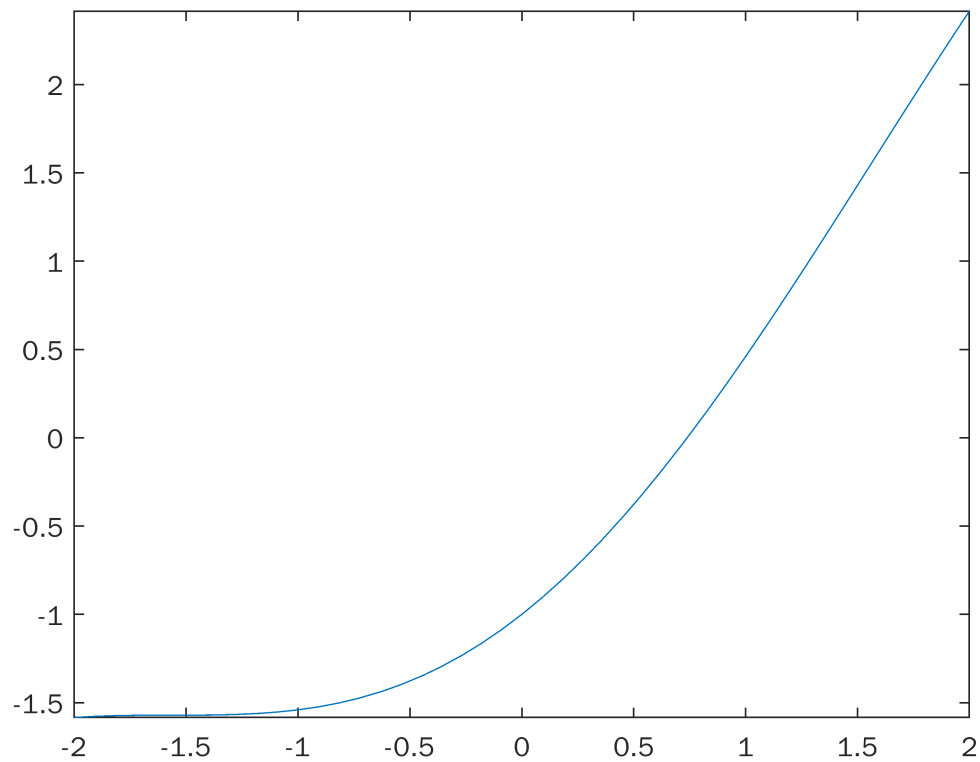
Resuelve la ecuación  $x = \cos(x)$ .

La ecuación puede reformularse para encontrar las raíces de una función (anónima)  $f1$ .

```
f1 = @(x) x - cos(x);
```

Grafica la función  $f1$  en el intervalo  $[-2, 2]$

```
fplot(f1, [-2, 2])
```



¿Cuántas raíces (ceros) tiene  $f1$ ?

Encuentra una raíz de  $f1$  usando la función de MATLAB  $fzero$  y el valor inicial  $x=0.5$ .

```
fzero(f1, 0.5)
```

```
ans =  
0.739085133215161
```

Encuentra una raíz de  $f1$  usando la función de MATLAB  $fzero$  y el valor inicial  $x=1$ .

```
fzero(f1, 1)
```

```
ans =  
0.739085133215161
```

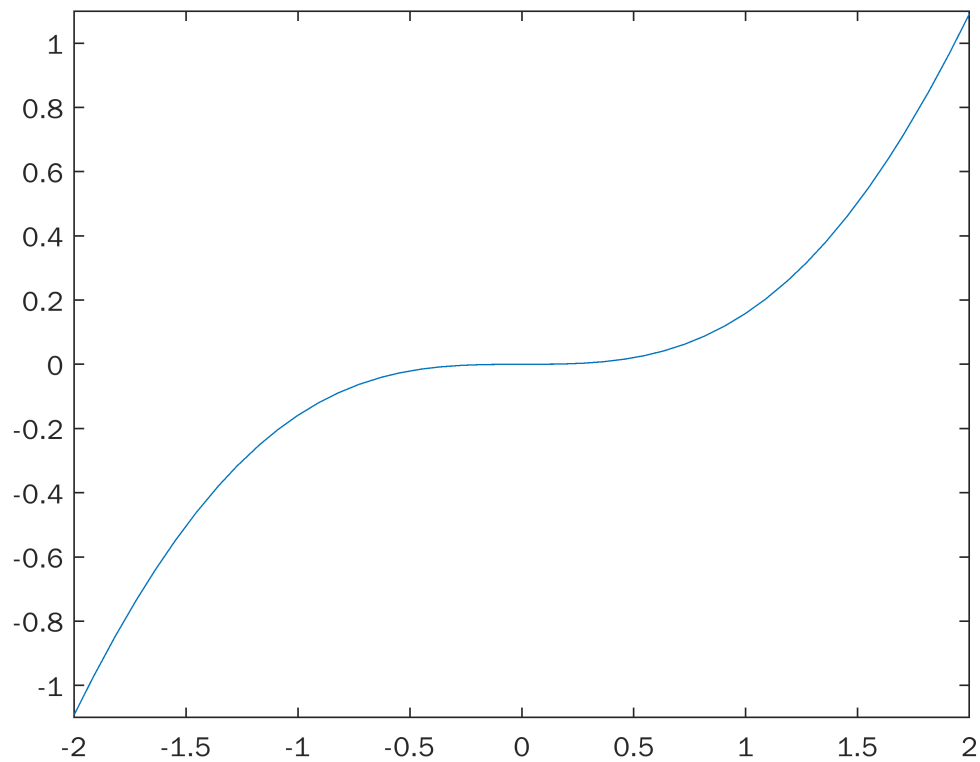
Resuelve la ecuación  $x=\sin(x)$ .

Reformula la ecuación para encontrar las raíces de una función  $f2$

```
f2 = @(x) x - sin(x);
```

Grafica la función  $f2$  en el intervalo  $[-2,2]$

```
fplot(f2, [-2, 2]);
```



¿Cuántas raíces (ceros) tiene  $f_2$ ?

Encuentra una raíz de  $f_2$  usando la función de MATLAB  $fzero$  y el valor inicial  $x=-0.5$ .

```
fzero(f2, -0.5)
```

```
ans =  
4.646889931104623e-09
```

Encuentra una raíz de  $f_2$  usando la función de MATLAB  $fzero$  y el valor inicial  $x=0$ .

```
fzero(f2, 0)
```

```
ans =  
0
```

Encuentra una raíz de  $f_2$  usando la función de MATLAB  $fzero$  y el valor inicial  $x=0.5$ .

```
fzero(f2, 0.5)
```

```
ans =  
-4.646889931104623e-09
```

¿Cuántas raíces (ceros) tiene  $f_2$ ? ¿En qué puntos  $\frac{\sin(x)}{x} = 1$ ? Explica qué observas

```
% Tiene 1 raíz  
% Esto fue demostrado en una tarea anterior, solo se  
% cumple cuando el lim x -> 0
```

## Pregunta 5.

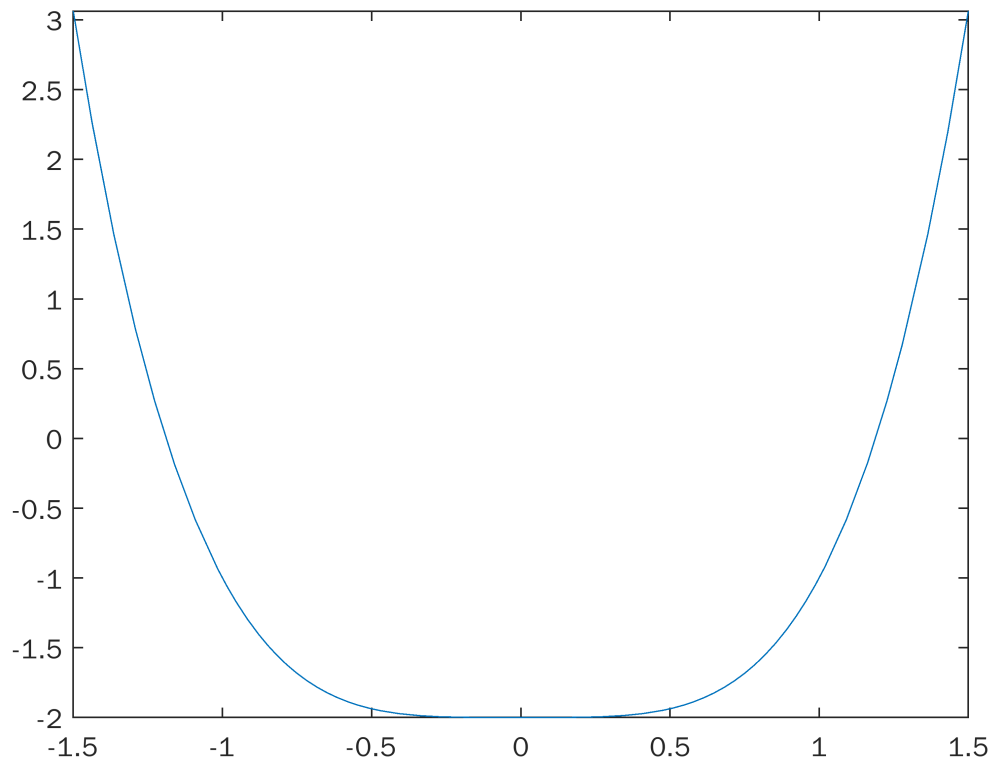
Resuelve la ecuación  $x^4 = 2$ .

Reformula la ecuación para encontrar las raíces de una función  $f$ .

```
f4 = @(x) x.^4 - 2;
```

Grafica la función entre -1.5 y 1.5.

```
fplot(f4, [-1.5, 1.5])
```



Encuentra la raíz real positiva usando el método de Newton-Raphson y la aproximación inicial  $x_0 = 1$ .

```
raiz = newtonRaphson(f4, 1, 3)
```

```
raiz =  
1.189207115002721
```

¿Para qué valor real  $x_0$  el método no converge?

```
noraiz = newtonRaphson(f4, 0, 0)
```

```
noraiz =  
Inf
```

```
% Al parecer se acerca al cero pero nunca lo toca
```

## Pregunta 6.

Encuentra en qué valor de  $x$  ocurre el máximo de la función  $f(x) = 2\sin(x) - \frac{x^2}{10}$  y cuál es el valor máximo, usando alguno de los métodos vistos en clase.

```
f = @(x) 2*sin(x) - x.^2/10
```

```
f = function_handle with value:  
@(x)2*sin(x)-x.^2/10
```

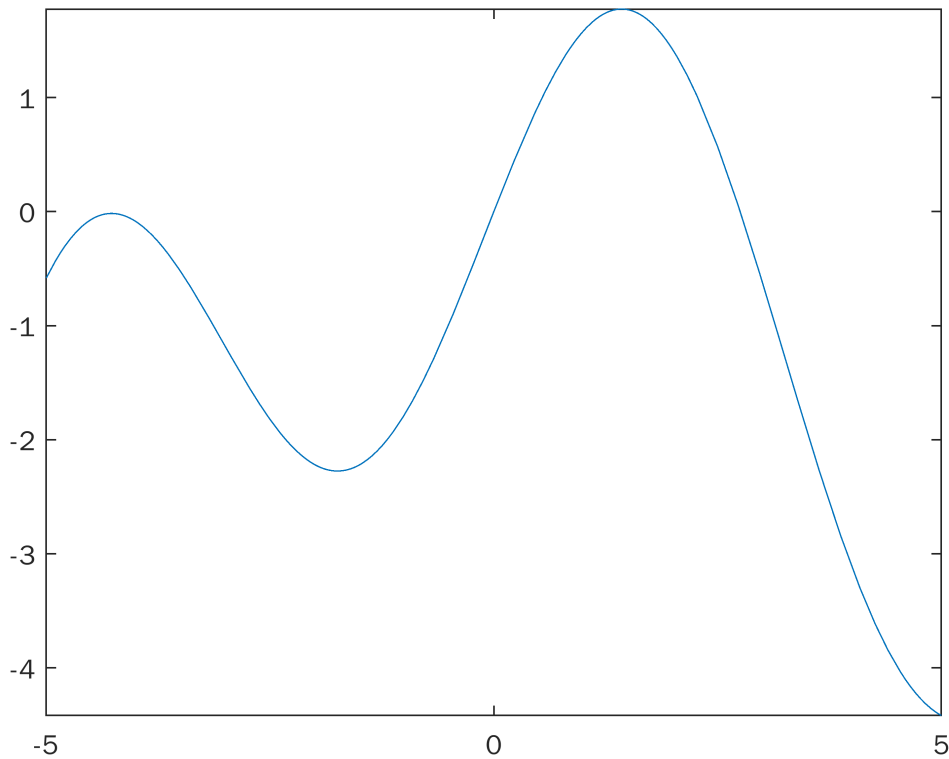
```
g = @(x) -2*sin(x) + x.^2/10
```

```
g = function_handle with value:  
@(x)-2*sin(x)+x.^2/10
```

```
max = newtonOpt(@(x) -f(x), 0, 1)
```

```
max =  
1.427551778764594
```

```
fplot(f)
```



Verifica tu solución usando la función de MATLAB *fminbnd*.

```
fmin = fminbnd(g, -5 , 5)
```

```
fmin =
```



1.427552078010970

Encuentra la derivada  $df$  de la función  $f(x) = 2\sin(x) - \frac{x^2}{10}$

```
df = matlabFunction(diff(sym(f)));
```

Encuentra en dónde vale cero  $df$  usando el algoritmo de Muller

```
cero = Muller(df, 0, 1, 2)
```

```
cero =  
1.427551778764594
```

## Pregunta 7.

Encuentra una fórmula para encontrar la siguiente aproximación en el método de interpolación parabólica sucesiva que calcula un punto crítico de una función  $f$ . Puedes usar el polinomio de interpolación de Lagrange (para obtener la fórmula del libro de texto) o el de Newton (para obtener la fórmula de la presentación del curso).

Inserta aquí una foto del procedimiento.

### Derivada

Utilizando diferencias divididas de Newton, entonces:

Sean:

$$f = ax^2 + bx + c$$

$$f' = 2ax + b$$

$$a = 2ax + b \Rightarrow x = -b/2a$$

Entonces,

$$P_2(x) = f[x_1] + f[x_1, x_2](x - x_1) + f[x_1, x_2, x_3](x - x_1)(x - x_2)$$

$$f[x_1] = f(x_1)$$

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$$

$$= \frac{f(x_3) - f(x_2)}{x_3 - x_2} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

10

$$= \frac{(f(x_3) - f(x_2))(x_2 - x_1) - (f(x_2) - f(x_1))(x_3 - x_2)}{(x_3 - x_2)(x_2 - x_1)}$$

$$= \frac{x_3 - x_1}{(f(x_3) - f(x_2))(x_2 - x_1) - (f(x_2) - f(x_1))(x_3 - x_2)}$$

$$P_2(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1} \cdot x - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \cdot x_1 + \frac{(f(x_3) - f(x_2))(x_2 - x_1)}{(x_3 - x_2)(x_2 - x_1)(x_3 - x_1)}$$

$$- \frac{(f(x_2) - f(x_1))(x_3 - x_2)}{(x_3 - x_2)(x_2 - x_1)(x_3 - x_1)} \cdot (x^2 - x_2 \cdot x - x_1 \cdot x + x_2 \cdot x_1)$$

$$\therefore a = f[x_1, x_2, x_3]$$

$$b = f[x_1, x_2] - x_2 f[x_1, x_2, x_3] - x_1 f[x_1, x_2, x_3]$$

$$= f[x_1, x_2] - f[x_1, x_2, x_3] \cdot (x_2 + x_1)$$

Es

Escribe aquí las funciones que necesitas para correr el script.

## Pregunta 8.

Newton-Raphson

```
function [x, i] = newtonRaphson(f, a, b)
    REL_TOL = sqrt(eps);
    MAX_ITER = 53;
    x = (a + b) / 2;
    df = matlabFunction(diff(sym(f)));

    if nargin(df) == 0
        df = @(x) df();
    end

    i = 0;
    flag = true;
    while flag
        xp = x;
        dx = f(xp) / df(xp);
        x = xp - dx;
        i = i + 1;
        flag = abs(dx / x) > REL_TOL && i < MAX_ITER;
    end
end
```

## Pregunta 9.

Método de optimización

```
function [x, i, m] = newtonOpt(f, a, b)
    TOL = eps;
    MAX_ITER = 53;
    x = (a + b) / 2;
    df = diff(sym(f));
    ddf = matlabFunction(diff(df));
    df = matlabFunction(df);

    if nargin(ddf) == 0
        ddf = @(x) ddf();
    end

    i = 0;
    flag = true;
    while flag
        xp = x;
        dx = df(xp) / ddf(xp);
        x = xp - dx;
```

```

        i = i + 1;
        flag = abs((dx / x)) > TOL && i < MAX_ITER;
    end
    m = ddf(x) > 0;
end

```

## Pregunta 10.

Muller

```

function [x,i]= Muller(f,p1,p2,p3)
    iMAX=52;
    TolMax=sqrt(eps);
    i=0;
    f1=f(p1);
    f2=f(p2);
    condition=true;
    while(condition)
        f3=f(p3);
        Pol2=polyfit([p1,p2,p3],[f1,f2,f3],2);
        r=roots(Pol2);
        if(abs(p3-r(1))<abs(p3-r(2)))
            x=r(1);
        else
            x=r(2);
        end
        f1=f2; f2=f3;
        p1=p2; p2=p3; p3=x;
        i=i+1;
        condition=abs((x-p2)/x)>TolMax && i<iMAX;
    end
end

```