Practica nº3: Raices de Ecuaciones no lineales

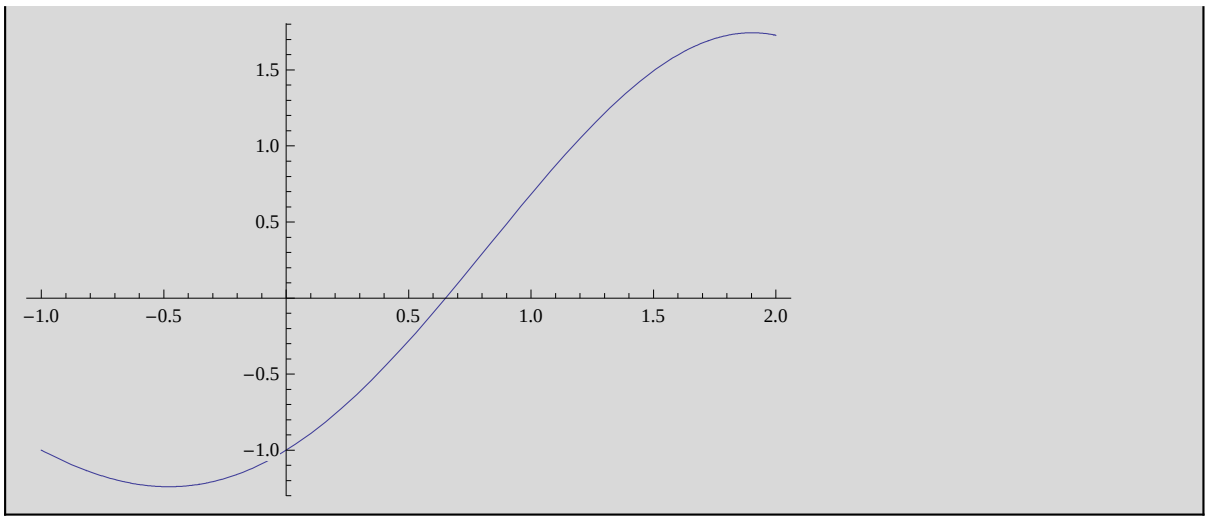
Resolución de una ecuación no lineal

Calcular la raiz de la ecuacion f(x)=(1+x)Sin[x]-1 en el intervalo (0. , 1.)
Utilizando el método de Bisección, Punto fijo, Newton-Raphson y Secante
Con un error menor que 10^{-6}

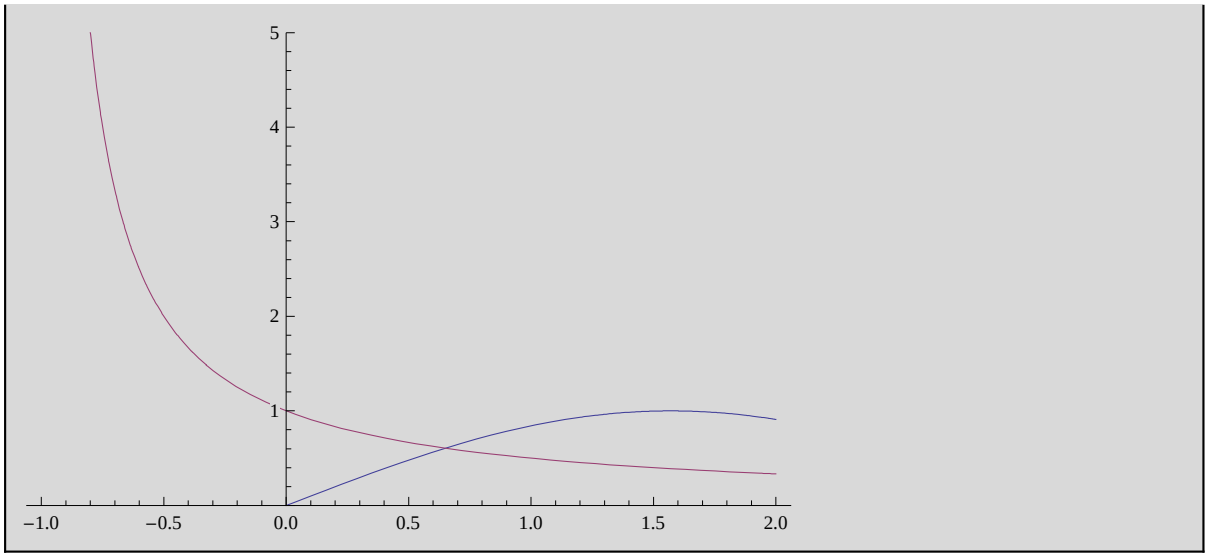
```
Clear[f]
f[x_] = (1 + x) * Sin[x] - 1
```

```
-1 + (1 + x) Sin[x]
```

```
Plot[f[x], {x, -1, 2}]
```



```
Plot[{Sin[x], 1/(1 + x)}, {x, -1, 2}, PlotRange → {0, 5}]
```



Biseccion

```
a = 0; b = 1; i = 0; error = 1;
While[error > 10^-6, x = (b + a) / 2;
 If[f[x] * f[a] < 0, b = x; error = Abs[b - a], a = x; error = Abs[b - a]];
 i++; Print["iteración nº ", i, "  a=", a // N, "  b=", b // N]]
```

```
iteración nº 1  a=0.5  b=1.
iteración nº 2  a=0.5  b=0.75
iteración nº 3  a=0.625  b=0.75
iteración nº 4  a=0.625  b=0.6875
iteración nº 5  a=0.625  b=0.65625
iteración nº 6  a=0.640625  b=0.65625
iteración nº 7  a=0.648438  b=0.65625
iteración nº 8  a=0.648438  b=0.652344
iteración nº 9  a=0.650391  b=0.652344
iteración nº 10  a=0.650391  b=0.651367
iteración nº 11  a=0.650391  b=0.650879
iteración nº 12  a=0.650635  b=0.650879
iteración nº 13  a=0.650635  b=0.650757
iteración nº 14  a=0.650696  b=0.650757
iteración nº 15  a=0.650726  b=0.650757
iteración nº 16  a=0.650742  b=0.650757
iteración nº 17  a=0.650749  b=0.650757
iteración nº 18  a=0.650749  b=0.650753
iteración nº 19  a=0.650751  b=0.650753
iteración nº 20  a=0.650751  b=0.650752
```

```
a = 0; b = 1;
Do[x = (b + a) / 2;
 If[f[x] * f[a] < 0, b = x; error = Abs[b - a], a = x; error = Abs[b - a]];
 Print["iteración nº ", i, "  a=", a // N, "  b=", b // N];
 If[error ≤ 10^-6, Break[ ]], {i, 1, 50}]
```

```
iteración nº 1  a=0.5  b=1.
iteración nº 2  a=0.5  b=0.75
iteración nº 3  a=0.625  b=0.75
iteración nº 4  a=0.625  b=0.6875
iteración nº 5  a=0.625  b=0.65625
iteración nº 6  a=0.640625  b=0.65625
iteración nº 7  a=0.648438  b=0.65625
iteración nº 8  a=0.648438  b=0.652344
iteración nº 9  a=0.650391  b=0.652344
iteración nº 10  a=0.650391  b=0.651367
iteración nº 11  a=0.650391  b=0.650879
iteración nº 12  a=0.650635  b=0.650879
iteración nº 13  a=0.650635  b=0.650757
iteración nº 14  a=0.650696  b=0.650757
iteración nº 15  a=0.650726  b=0.650757
iteración nº 16  a=0.650742  b=0.650757
iteración nº 17  a=0.650749  b=0.650757
iteración nº 18  a=0.650749  b=0.650753
iteración nº 19  a=0.650751  b=0.650753
iteración nº 20  a=0.650751  b=0.650752
```

Newton-Raphson

```
Clear[x]
```

```
f1[x_] = D[f[x], x]
```

```
(1 + x) Cos[x] + Sin[x]
```

```
x0 = 0.5; error = 1; Itermax = 100; i = 0;
```

```
While[error > 10^-6 && i < Itermax, x1 = x0 - f[x0]/f1[x0]; error = Abs[x1 - x0];
 x0 = x1; i++; Print["iteración ", i, "    x̃=", x0];]
```

```
iteración 1    x̃=0.656399
iteración 2    x̃=0.650756
iteración 3    x̃=0.650752
iteración 4    x̃=0.650752
```

```
x0 = 0.5; Itermax = 100;
For[i = 1, i ≤ Itermax, x1 = x0 - f[x0]/D[f[x], x] /. x → x0;
 error = Abs[x1 - x0]; x0 = x1; Print["iteración ", i, "    x̃=", x0];
 If[error <= 10^-6, Break[ ]]; i++;]
```

```
iteración 1    x̃=0.656399
iteración 2    x̃=0.650756
iteración 3    x̃=0.650752
iteración 4    x̃=0.650752
```

Secante

```
x0 = 0.; x1 = 1.; error = 1; Itermax = 100; i = 0;
```

```
While[error > 10^-6 && i < Itermax,
 x2 = x1 - (x1 - x0)/(f[x1] - f[x0]) * f[x1]; error = Abs[x2 - x1]; x0 = x1;
 x1 = x2; i++; Print["iteración ", i, "    x̃=", x1];]
```

```
iteración 1    x̃=0.594198
iteración 2    x̃=0.649386
iteración 3    x̃=0.650765
iteración 4    x̃=0.650752
iteración 5    x̃=0.650752
```

```
x0 = 0.; x1 = 1.; Itermax = 100;
For[i = 1, i ≤ Itermax, x2 = x1 - (x1 - x0)/(f[x1] - f[x0]) * f[x1];
 error = Abs[x2 - x1]; x0 = x1; x1 = x2;
 Print["iteración ", i, "    x̃=", x0]; If[error <= 10^-6, Break[ ]]; i++;]
```

```
iteración 1    x̃=1.
iteración 2    x̃=0.594198
iteración 3    x̃=0.649386
iteración 4    x̃=0.650765
iteración 5    x̃=0.650752
```

Calcular la raiz de la ecuacion f(x)=(1+x)Sin[x]-1 en el intervalo (0. , 1.) Utilizando el método de Punto fijo, estudiando las condiciones de convergencia para la función g(x) utilizada. Resolver el problema con un error menor que 10^{-6}

Punto fijo

Función utilizada

```
g[x_] := ArcSin[1/(1 + x)]
```

Condiciones de convergencia

```
0 ≤ g[0] ≤ 1
```

```
False
```

```
N[g[0]]
```

```
1.5708
```

```
0 <= g[1] ≤ 1
```

```
True
```

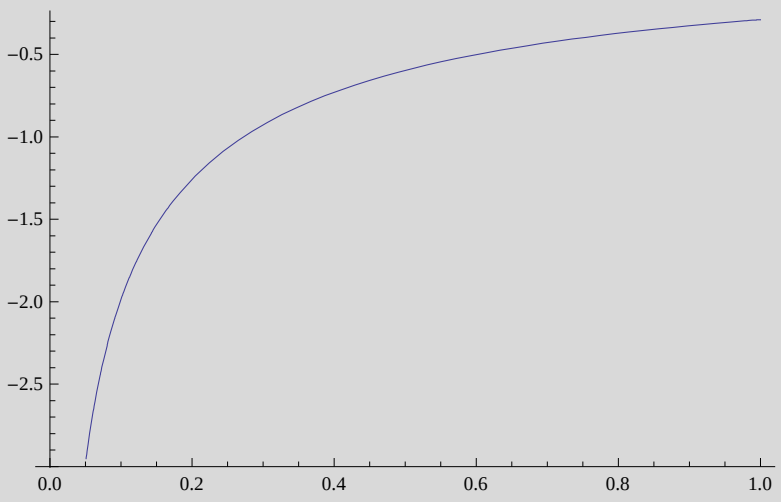
```
0.2 ≤ g[0.2] ≤ 1
```

```
True
```

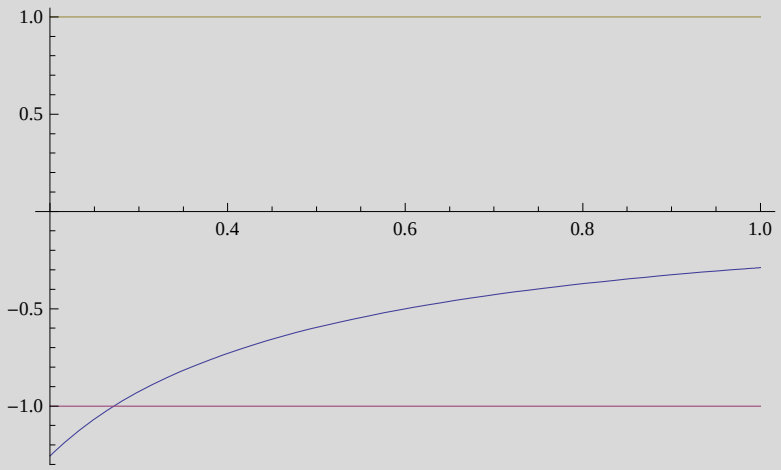
```
Clear[x]
gprima[x_] = D[g[x], x]
```

$$-\frac{1}{(1+x)^2\sqrt{1-\frac{1}{(1+x)^2}}}$$

```
Plot[gprima[x], {x, 0, 1}]
```




```
Plot[{gprima[x], -1, 1}, {x, 0.2, 1}]
```



```
gprima[0.02] // N
```

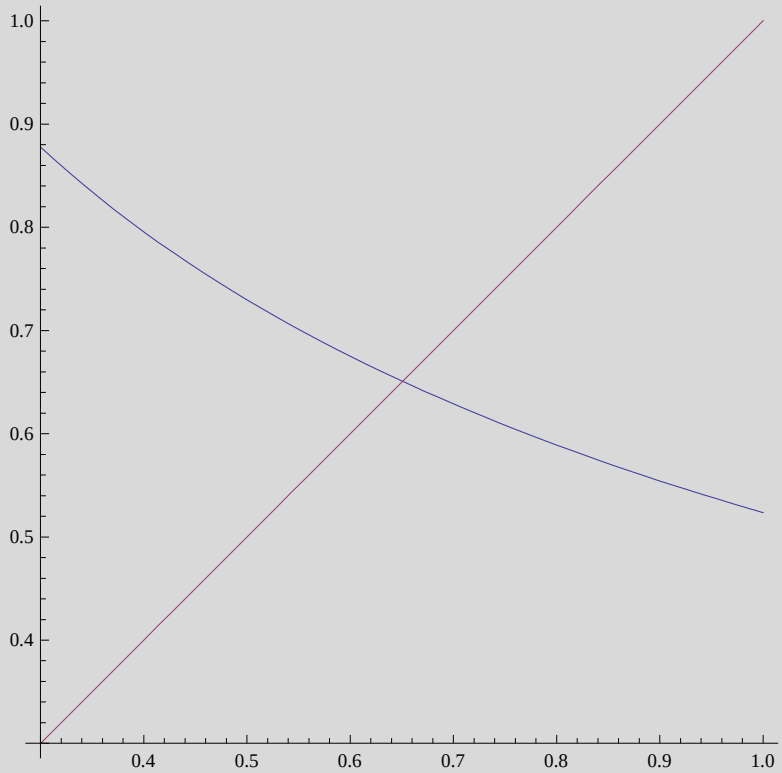
```
-4.87763
```

```
Abs[gprima[0.4]] < 1
```

```
True
```

Cumple las condiciones en el intervalo[0.4,1]

```
grafico = Plot[{g[x], x}, {x, 0.3, 1}, AspectRatio → Automatic]
```



```
x0 = 0.5; Itermax = 100;
For[i = 1, i ≤ Itermax, x1 = g[x0]; error = Abs[x1 - x0]; x0 = x1;
 Print["iteración ", i, "    x̃=", x0]; If[error <= 10^-6, Break[ ]]; i++;]
```

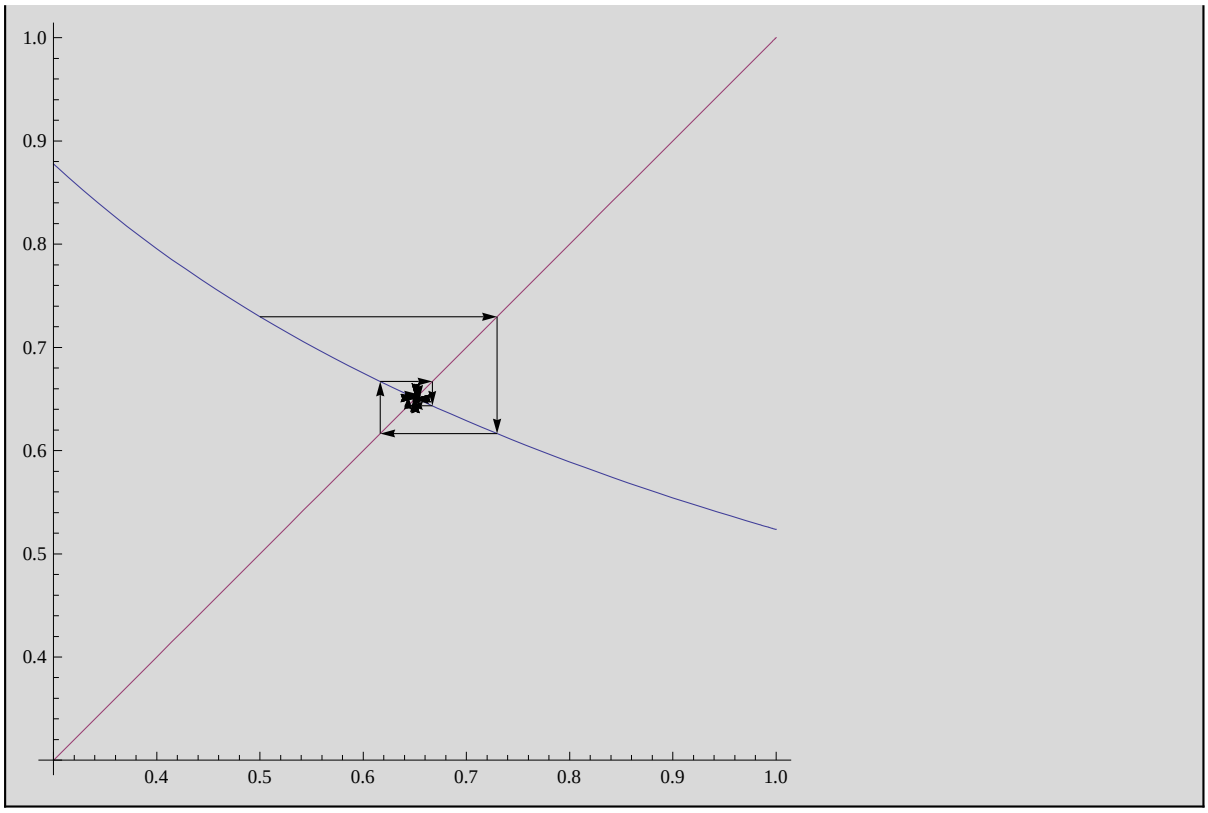
```
iteración 1    x̃=0.729728
iteración 2    x̃=0.61643
iteración 3    x̃=0.66702
iteración 4    x̃=0.643342
iteración 5    x̃=0.654189
iteración 6    x̃=0.64917
iteración 7    x̃=0.651482
iteración 8    x̃=0.650415
iteración 9    x̃=0.650907
iteración 10    x̃=0.65068
iteración 11    x̃=0.650785
iteración 12    x̃=0.650736
iteración 13    x̃=0.650759
iteración 14    x̃=0.650748
iteración 15    x̃=0.650753
iteración 16    x̃=0.650751
iteración 17    x̃=0.650752
iteración 18    x̃=0.650752
```

```
valor = {};
x0 = 0.5; Itermax = 100; error = 1; i = 0;
While[error > 10^-6 && i < Itermax, x1 = g[x0]; valor = Append[valor, {x0, x1}];
 valor = Append[valor, {x1, x1}]; error = Abs[x1 - x0]; x0 = x1; i++;
 Print["iteración ", i, "    x̃=", x0, "  error=", error // N];]
```

```
iteración 1    x̃=0.729728  error=0.229728
iteración 2    x̃=0.61643  error=0.113298
iteración 3    x̃=0.66702  error=0.0505902
iteración 4    x̃=0.643342  error=0.0236777
iteración 5    x̃=0.654189  error=0.0108469
iteración 6    x̃=0.64917  error=0.0050187
iteración 7    x̃=0.651482  error=0.00231148
iteración 8    x̃=0.650415  error=0.00106685
iteración 9    x̃=0.650907  error=0.000491924
iteración 10    x̃=0.65068  error=0.000226927
iteración 11    x̃=0.650785  error=0.000104661
iteración 12    x̃=0.650736  error=0.000048275
iteración 13    x̃=0.650759  error=0.000022266
iteración 14    x̃=0.650748  error=0.00001027
iteración 15    x̃=0.650753  error=4.73692×10^-6
iteración 16    x̃=0.650751  error=2.18485×10^-6
iteración 17    x̃=0.650752  error=1.00774×10^-6
iteración 18    x̃=0.650752  error=4.64809×10^-7
```

```
lis = Table[{Arrowheads[.02], Arrow[{valor[[i]], valor[[i + 1]]}]},
   {i, 1, Length[valor] - 1}];
```

```
Show[{grafico, Graphics[lis]}]
```



Raices Múltiples

Sea p(x) el polinomio de grado 4,

$p(x)=7.14292992-20.0471\,x+19.0956\,x^2-7.45\,x^3+x^4$,

Se pide:

1) Representar gráficamente p(x).

2) Utilizar el comando Factor para obtener sus raices.

3) Utilizar el comando FindRoot para obtener sus raices dando distintas aproximaciones iniciales. ¿Qué ocurre cuando éstas están cercanas a la raiz múltiple?, ¿Y si están próximas a las otras raices?

4) Utilizar el mdo. de Newton-Raphson con una aproximación inicial x0=1.4, una precisión de 10^{-6} y un nº máximo de iteraciones igual a 25. Hacer los cálculos con una precisión de 12 dígitos (N[expr,12]).

La fórmula de recursión es: $x_{i+1} = x_i - p(x_i) / p'(x_i)$

5) Repetir el apdo. anterior con el mdo. de NewtonRaphson acelerado (valor de m=2).

La fórmula de recursión es: $x_{i+1} = x_i - m \,.\, p(x_i) / p'(x_i)$

6) Idem. con el mdo. de Halley.

La fórmula de recursión es:

$x_{i+1} = x_i - 2.\, p(x_i).\, p'(x_i) /(2 \,.\, p'(x_i).\, p'(x_i) - p(x_i).\, p''(x_i))$

7) Idem. con el mdo. de Newton-Raphson modificado para raices múltiples.

La fórmula de recursión es:

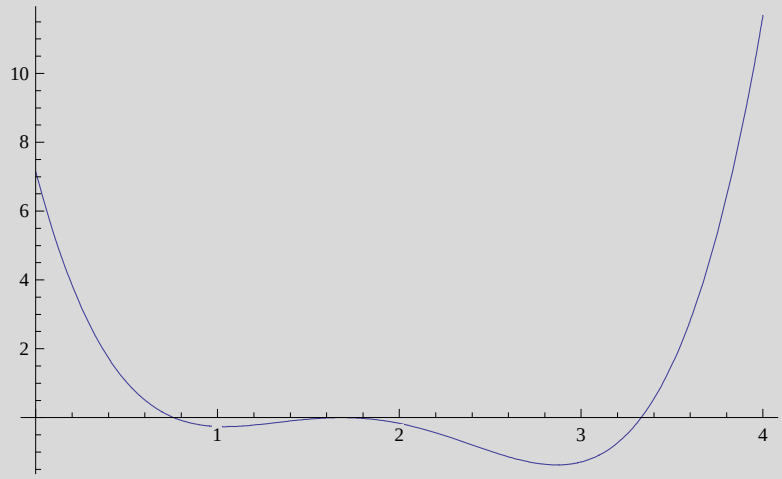
$x_{i+1} = x_i - p(x_i).\, p'(x_i) /(p'(x_i).\, p'(x_i) - p(x_i).\, p''(x_i))$

1) Representar gráficamente p(x).

```
p[x_] = 7.14292992 - 20.047104 x + 19.0956 x^2 - 7.45 x^3 + x^4
```

$7.14293 - 20.0471\, x + 19.0956\, x^2 - 7.45\, x^3 + x^4$

```
Plot[p[x], {x, 0, 4}]
```



2) Utilizar el comando Factor para obtener sus raices.

```
Factor[p[x]]
```

$(-3.33 + 1.\, x)\, (-1.68 + 1.\, x)\, (-1.68 + 1.\, x)\, (-0.76 + 1.\, x)$

3) Utilizar el comando FindRoot para obtener sus raices dando distintas aproximaciones iniciales. ¿Qué ocurre cuando éstas están cercanas a la raiz múltiple?, ¿Y si están próximas a las otras raices?

```
FindRoot[p[x], {x, 2}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 1.9}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 1.8}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 1.7}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 1.6}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 1.5}]
```

```
{x → 1.68}
```

```
FindRoot[p[x], {x, 0.8}]
```

```
{x → 0.76}
```

```
FindRoot[p[x], {x, 0.9}]
```

```
{x → 0.76}
```

```
FindRoot[p[x], {x, 0.6}]
```

```
{x → 0.76}
```

```
FindRoot[p[x], {x, 3.1}]
```

```
{x → 3.33}
```

```
FindRoot[p[x], {x, 3.2}]
```

```
{x → 3.33}
```

```
FindRoot[p[x], {x, 3.4}]
```

```
{x → 3.33}
```

4) Utilizar el mdo. de Newton-Raphson con una aproximación inicial x0=1.4, una precisión de 10^{-6} y un nº máximo de iteraciones igual a 25. Hacer los cálculos con una precisión de 12 dígitos (N[expr,12]).

La fórmula de recursión es:$x_{i+1} = x_i - p(x_i) / p'(x_i)$

```
p1[x_] = D[p[x], {x, 1}];
```

```
Print["Métodos de Newton-Raphson"];
Print["Iteración     Aproximación       Error"];
i = 0; x_i = 1.4; Print["   ", i, "          ", x_i];
For[error = 1, error > 10^-6 && i <= 25, i++, {a = N[p[x] /. x -> x_i, 12];
  b = N[p1[x] /. x -> x_i, 12]; x_i+1 = N[x_i - a / b, 12]; error = N[Abs[x_i+1 - x_i], 12];
  Print["    ", i + 1, "        ", x_i+1, "     ", error]}]
```

Métodos de Newton-Raphson

Iteración	Aproximación	Error
0	1.4	
1	1.56397	0.163975
2	1.62436	0.0603862
3	1.65263	0.0282682
4	1.66641	0.0137842
5	1.67323	0.00681649
6	1.67662	0.00339063
7	1.67831	0.00169106
8	1.67916	0.000844488
9	1.67958	0.000421985
10	1.67979	0.000210928
11	1.67989	0.000105448
12	1.67995	0.00005272
13	1.67997	0.000026359
14	1.67999	0.0000131792
15	1.67999	6.58973×10^{-6}
16	1.68	3.2948×10^{-6}
17	1.68	1.64737×10^{-6}
18	1.68	8.24074×10^{-7}

5) Repetir el apdo. anterior con el mdo. de NewtonRaphson acelerado (valor de m=2).
La fórmula de recursión es:$x_{i+1} = x_i - m \,.\, p(x_i) / p'(x_i)$

```
Print["Método de Newton-Raphson acelerado"];
Print["Iteración       Aproximación       Error"];
i = 0; x_i = 1.4; Print["   ", i, "          ", x_i];
For[error = 1, error > 10^-6 && i <= 25, i++,
 {a = N[p[x] /. x -> x_i, 12]; b = N[p1[x] /. x -> x_i, 12];
  x_{i+1} = N[x_i - 2 * a / b, 12]; error = N[Abs[x_{i+1} - x_i], 12];
  Print["    ", i + 1, "        ", x_{i+1}, "    ", error]}]
```

```
Método de Newton-Raphson acelerado
Iteración       Aproximación       Error
   0         1.4
    1       1.72795    0.32795
    2       1.68047    0.0474844
    3       1.68    0.000465468
    4       1.68    0.
```

6) Idem. con el mdo. de Halley.
La fórmula de recursión es:
$x_{i+1} = x_i - 2.\, p(x_i).\, p'(x_i) /(2 \,.\, p'(x_i).\, p'(x_i) - p(x_i).\, p''(x_i))$

```
p2[x_] = D[p[x], {x, 2}];
```

```
Print["Método de Halley"];
Print["Iteración       Aproximación       Error"];
i = 0; x_i = 1.4; Print["   ", i, "          ", x_i];
For[error = 1, error > 10^-6 && i <= 25, i++, {a = N[p[x] /. x -> x_i, 12];
  b = N[p1[x] /. x -> x_i, 12]; c = N[p2[x] /. x -> x_i, 12];
  x_(i+1) = N[x_i - 2 * a * b / (2 * b * b - a * c), 12]; error = N[Abs[x_(i+1) - x_i], 12];
  Print["    ", i + 1, "        ", x_(i+1), "     ", error]}]
```

Método de Halley

Iteración	Aproximación	Error
0	1.4	
1	1.58646	0.186465
2	1.64926	0.0627914
3	1.6698	0.0205458
4	1.67661	0.00680414
5	1.67887	0.00226312
6	1.67962	0.000753825
7	1.67987	0.000251214
8	1.67996	0.0000837313
9	1.67999	0.0000279097
10	1.68	9.30343×10^{-6}
11	1.68	3.1009×10^{-6}
12	1.68	1.03353×10^{-6}
13	1.68	3.4804×10^{-7}

7) Idem. con el mdo. de Newton-Raphson modificado para raices múltiples.
La fórmula de recursión es:

$$x_{i+1} = x_i - p(x_i).\, p'(x_i) / (\, p'(x_i).\, p'(x_i) - p(x_i).\, p''(x_i))$$

```
Print["Método de Newton Raphson modificado para raices múltiples"];
Print["Iteración       Aproximación       Error"];
i = 0; x_i = 1.4; Print["   ", i, "          ", x_i];
For[error = 1, error > 10^-6 && i <= 25, i++, {a = N[p[x] /. x -> x_i, 12];
  b = N[p1[x] /. x -> x_i, 12]; c = N[p2[x] /. x -> x_i, 12];
  x_{i+1} = N[x_i - a * b / (b * b - a * c), 12]; error = N[Abs[x_{i+1} - x_i], 12];
  Print["    ", i + 1, "        ", x_{i+1}, "    ", error]}]
```

Método de Newton Raphson modificado para raices múltiples

Iteración	Aproximación	Error
0	1.4	
1	1.6161	0.216105
2	1.67859	0.0624844
3	1.68	0.00141035
4	1.68	4.8625×10^{-7}

```
Roots[p[x] == 0, x]
```

```
x == 0.76 || x == 1.68 || x == 1.68 || x == 3.33
```

```
FindRoot[p[x], {x, 1.6}, AccuracyGoal → 50, WorkingPrecision → 3000,
 MaxIterations → 250]
```

FindRoot::precw: The precision of the argument function $(7.14293 - 20.0471\,x + 19.0956\,x^2 - 7.45\,x^3 + x^4)$ is less than WorkingPrecision (3000.`). ≫

```
{x →
  1.6799999778515924907365126244262592831251286792640407071092824143793 52
   6759993630865509594185928022304880978304710507094352894745619439036184
   0581883497505078367605315402788741357441124732172333800611257609875967
   7166867268541521902378874523909588212297913083370868314076711782562964
   4875891680286704007234525941412093144020473303796663805049912348655963
   2180620571862161328042587547443256911570897894042676144621292655751473
   6629218940782636524567109614663952751725163299306420373547266717661731
   3172145731818718092114613131744774052601659276022735395221223680231604
   3943342851973825784378333964600619289181158084018460645494360532138416
   0247495398934428988345426277593088247260094605681020969803192804385666
   8903725669558141404789918332671697925002864270044771740187524694072256
   1882644917631916641932121062697112793040329729816689869589293684883116
   2322674292211157404727641253214593198407560220996791137222811311266283
   2774256061557908471375296902129679637521668920765294723708958670992951
   9147157628291626275021830132971446383909251307973152260290632350179235
   8745747545281158851084417776072686215971386183179166853187577691244823
   8101255200861025540839882459839109630065409673731171298379322533107725
   8138481660391049058966563024165495382739898315798954220204053452327827
   3139162644267752785473480045001424923982732963662121933698181950019487
   5696312262843203325128403858600218046725737530083929764887930844979216
   3753091338086836236839304714829989244819060419470489589235471642324067
   1803901253510096965152341175904174542274525296162397765263397704268437
   7578426859973976948806057707809724887701765119103740712643033130618703
   4320404864537229510673395278014806680946627160206687867470253199630008
   7404300035923543450475485124484907705003685117571103053427041140284241
   1771530538763615166606161621129657531281581020763995381882793597187224
   0013655490631803829490803891640117394941406985059206308893666091459868
   8236975558179256690501021368389773625762114475572789880986000852369115
   7069470540205184170727174253726385240379429254136543165078774070210413
   4379783372750830601915231620349399572190104583408733730258263388762852
   0985968091385403862778753507456112230403676575308826199341726129452104
   7881322139580699650986397901585436738033288474621511053240993529107320
   8705445423705135017149757622431358820273849252801375452120634471726332
   8654100482170481776074371966342360274681335114611090728627689567386976
   2048193020559263841077348360584075513409837075867393803518345658799443
   7252095372031139412019113456127038937544807105310291153188541052590575
   7816438636281102955235962968059326641255463992766571569319117001876231
   3751383661893130313358957167315861342009135434668393212738643101454488
   2118136742906606237479080189166297216113925315453626411403120253415738
   5364555351367295234762941706151391820476806126458917788647098546905083
   5307314855735125458894529291571320991415945392394992289415912763589826
   9698286538992366686818075819460557243076124791291811566392089908144220
   7158303817729614985411342242769802261539381448255463181037 43}
```

Resolución de un sistema de ecuaciones no lineales

Resolver el sistema de tres ecuaciones con tres incognitas:
$3\,x_1$-cos[x_1.x_2]-1/2 = 0
x_1^2-81.(x_2 + 0.1)^2+sen(x_3)+1.06 = 0
Exp(-x_1. x_2) + 20.x_3 + (10π - 3)/6 = 0
mediante el mdo. de Newton, siguiendo los siguientes pasos:
1º.- Definición del sistema
2º.- Definir la matriz Jacobiana
3º.- Realizando las iteraciones a mano.(ejecutando progresivamente los comandos, sin ciclo)
4º.- Realizando las iteraciones automáticamente con criterio de parada que el max. valor absoluto de la diferencia de las aproximaciones para cada una de las incognitas sea menor que 10^{-5}
5º.-Resolver con el comando FindRoot.

1º.- Definición del sistema

```
Clear[x1, x2, x3, f1, f2, f3]
```

```
f1[x1_, x2_, x3_] = 3 x1 - Cos[x1 * x2] - 1 / 2;
```

```
f2[x1_, x2_, x3_] = x1^2 - 81 * (x2 + .1) ^2 + Sin[x3] + 1.06;
```

```
f3[x1_, x2_, x3_] = Exp[-x1 * x2] + 20 x3 + (10 Pi - 3)/6;
```

2º.- Definir la matriz Jacobiana

```
matjacob[x1_, x2_, x3_] =
 {{∂x1 f1[x1, x2, x3], ∂x2 f1[x1, x2, x3], ∂x3 f1[x1, x2, x3]},
  {∂x1 f2[x1, x2, x3], ∂x2 f2[x1, x2, x3], ∂x3 f2[x1, x2, x3]},
  {∂x1 f3[x1, x2, x3], ∂x2 f3[x1, x2, x3], ∂x3 f3[x1, x2, x3]}}
```

$$\{\{3 + x2\,\mathrm{Sin}[x1\,x2],\ x1\,\mathrm{Sin}[x1\,x2],\ 0\},\ \{2\,x1,\ -162\,(0.1 + x2),\ \mathrm{Cos}[x3]\},\ \{-e^{-x1\,x2}\,x2,\ -e^{-x1\,x2}\,x1,\ 20\}\}$$

```
sisecu[x1_, x2_, x3_] = {f1[x1, x2, x3], f2[x1, x2, x3], f3[x1, x2, x3]}
```

$$\left\{-\frac{1}{2} + 3\,x1 - \mathrm{Cos}[x1\,x2],\ 1.06 + x1^2 - 81\,(0.1 + x2)^2 + \mathrm{Sin}[x3],\ e^{-x1\,x2} + \frac{1}{6}\,(-3 + 10\,\pi) + 20\,x3\right\}$$

3º.- Realizando las iteraciones a mano.(ejecutando progresivamente los comandos, sin ciclo)

```
a0 = .1; b0 = .1; c0 = -.1;
```

```
mat = matjacob[a0, b0, c0]
```

```
{{3.001, 0.000999983, 0}, {0.2, -32.4, 0.995004}, {-0.099005, -0.099005, 20}}
```

```
b = -sisecu[a0, b0, c0]
```

```
{1.19995, 2.26983, -3.72604}
```

```
solu = LinearSolve[mat, b]
```

```
{0.399874, -0.0732599, -0.184685}
```

```
Max[Abs[solu]]
```

```
0.399874
```

```
a0 = a0 + solu[[1]]
b0 = b0 + solu[[2]]
c0 = c0 + solu[[3]]
```

```
0.499874
```

```
0.0267401
```

```
-0.284685
```

4º.- Realizando las iteraciones automáticamente con criterio de parada que el max. valor absoluto de la diferencia de las aproximaciones para cada una de las incognitas sea menor que 10^{-5}

```
a0 = .1; b0 = .1; c0 = -.1; it = 1.;
```

```
While[it < 10,
    mat = matjacob[a0, b0, c0]; b = -sisecu[a0, b0, c0];
    sol = LinearSolve[mat, b] // N;
    m = Max[Abs[sol]];
    a0 = a0 + sol[[1]]; b0 = b0 + sol[[2]]; c0 = c0 + sol[[3]];
    Print["Itera.", it, "  error  ", m];
    Print["x1=", a0, "   x2=", b0, "   x3=", c0];
    If[m < .00001, Break[], it = it + 1]]
```

Itera.1. error 0.399874

x1=0.499874 x2=0.0267401 x3=-0.284685

Itera.2. error 0.013329

x1=0.5 x2=0.0134111 x3=-0.286464

Itera.3. error 0.000784677

x1=0.499993 x2=0.0126264 x3=-0.286485

Itera.4. error 2.73846×10^{-6}

x1=0.499993 x2=0.0126237 x3=-0.286485

5º.-Con el comando FindRoot.

```
Clear[x1, x2, x3]
```

```
FindRoot[{f1[x1, x2, x3] == 0, f2[x1, x2, x3] == 0, f3[x1, x2, x3] == 0},
 {x1, .1}, {x2, .1}, {x3, -.1}]
```

```
{x1 → 0.499993, x2 → 0.0126237, x3 → -0.286485}
```