Juan Pablo Amorocho, Diego Fernando Trujillo, Jose Alejandro López, Nicolás Pérez Fonseca

Materia: Análisis Numérico

30/08/2020

Profesor: Eddy Herrera Daza

Método del punto fijo

Introducción al método

Es un algoritmo iterativo que dada una función g su punto fijo es tal que *g(p)=p*, y es equivalente a resolver ecuaciones porque puedo expresar una función de la forma *f(x)=0* como *g(p)=p-f(p)*.

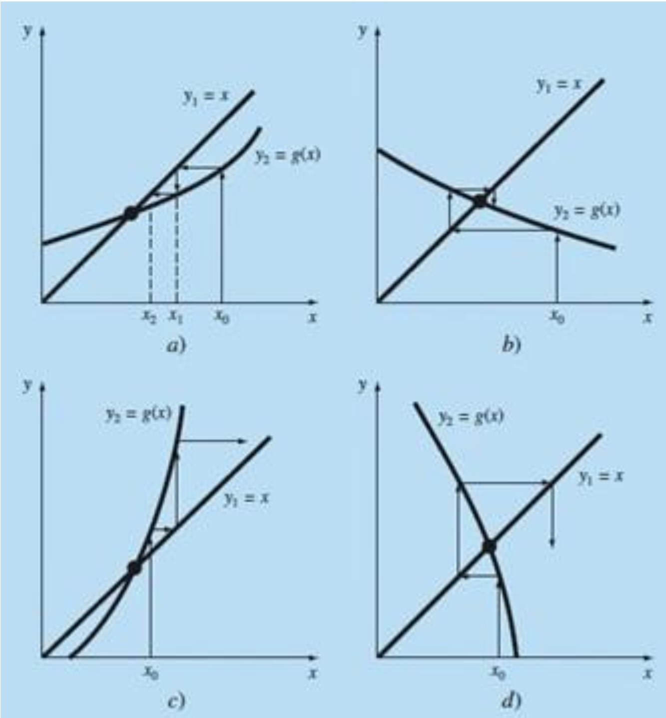
Se considera el Método del Punto Fijo una opción eficaz de resolución de ecuaciones debido a que es efectivo. Su aplicación consiste en reemplazos sucesivos de la imagen de la función partiendo de un valor p_0 que es deseable sea lo más cercano a la raíz. De tal forma que la sucesión $p_n = g(p_{n-1})$ sea convergente.

Preguntas:

¿Cuáles son las condiciones para aplicar el método?

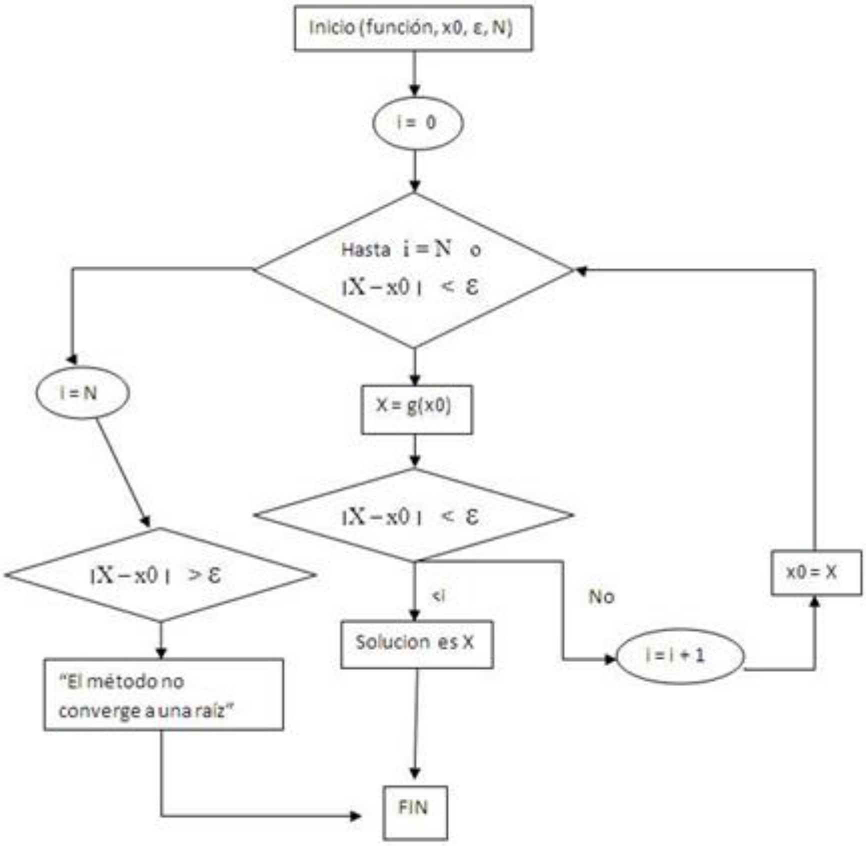
R/Si cumple con el teorema de punto fijo: si g:[a,b]->[a,b] es continua y derivable en [a,b], con |g´(x)|<=k<=1 que para todo x perteneciente en [a,b], y dado un x inicial perteneciente a [a,b] y la sucesión X sub n es igual a g en la posición X sub n menos 1 sea convergente, talque el límite de X sub n sea igual a el punto fijo entonces ahí seria la raíz.

Explicación geométrica del algoritmo:



La resolución de la ecuación g(c)=x consiste en encontrar la intersección de la gráfica de y=g(x) con la recta pendiente unidad y=x. A esta intersección se le conoce como punto fijo de g(x). Se puede comprobar que cuando la derivada de g(x) es mayor que la unidad en un intervalo que contiene al punto fijo, la sucesión de valores calculados diverge, alejándose de la solución.

Diagrama de Flujo



Raíces y su validación

Las raíces que salen de la ecuación acos(x)/2 con con 1e-12 daría como raíz

```
x_ 30 =  0.5149333
x_ 31 =  0.5149333
x_ 32 =  0.5149333
x_ 33 =  0.5149333
x_ 34 =  0.5149333
x_ 35 =  0.5149333
x_ 36 =  0.5149333
x_ 37 =  0.5149333
x_ 38 =  0.5149333
x_ 39 =  0.5149333
x_ 40 =  0.5149333
x_ 41 =  0.5149333
x_ 42 =  0.5149333
x_ 43 =  0.5149333
x_ 44 =  0.5149333
x_ 45 =  0.5149333
x_ 46 =  0.5149333
x_ 47 =  0.5149333
x_ 48 =  0.5149333
x_ 49 =  0.5149333
x_ 50 =  0.5149333
x_ 51 =  0.5149333
x* es aproximadamente  0.5149333  con error menor que  1e-12
```

Pérdida de Significancia

Como se sabe se genera pérdida de significancia cuando algún calculo numérico envuelve la resta de dos cantidades o en el cálculo de un número por medio de una sumatoria donde el resultado es mucho menor que los términos de la sumatoria los cuales alternan en signo. Para este método primero dependerá de la función a la que se le ha de encontrar las raíces, donde se pueden encontrar los dos casos de perdida de significancia, lo cual, el algoritmo no tiene total control, por otro lado, la única operación donde puede llegar a sufrir una pérdida de significancia es en el cálculo del error, el cual es una resta.

La parte del algoritmo es el siguiente:

```
dx = abs(x1 - x0)
```

Siendo:
dx el error calculado
x1 el valor de la función g(x0)
x0 el valor inicial o anterior de la iteración

Puede llegar a haber perdida de significancia cuando x1 y x0 sean muy parecidos siendo la resta un número muy cercano a cero. La forma en que solventamos el problema fue por medio de la misma herramienta de RStudio, simplemente teniendo en cuenta las primeras 16 cifras significativas donde es sometida al error su última cifra, ya que esta aplicación asegura controlar estas 16 primeras cifras, después de esta no serán tomadas en cuenta.

En cuanto al número de iteraciones del método depende de la función g(x), del valor inicial y de la tolerancia.
De la función g(x) depende de lo siguiente:

$|g'(x)| \leq k < 1$ para todo $x \in (a, b)$.

Entre más cercano sea el valor de k a 1, mayor iteración requerirá el algoritmo.

la dependencia de la tolerancia se puede demostrar con un ejemplo:
Teniendo la función g(x) = arccos(x)/2 con valor inicial x0 = 0.75
Se da una tolerancia inicial de 1e-12 como resultado tenemos

```
x_ 51 =  0.514933264660834
x* es aproximadamente  0.514933264660834  con error menor que  1e-12
```

Termina en la iteración número 51, ahora se muestra con una tolerancia de 1e-16 manteniendo el resto de las variables iguales, tenemos

```
x_ 67 =  0.5149332646611294
x* es aproximadamente  0.5149332646611294  con error menor que  1e-16
```

Terminando en la iteración 67, aumentando 16 iteraciones.

En caso de más de dos raíces

El método de punto fijo no admite hallar más de una raíz, esto según el teorema que dice lo siguiente:

Teorema 1.
{1} Si $g \in C[a,b]$ y $g(x) \in [a, b]$ para todo $x \in [a, b]$, entonces g tiene un punto fijo en [a,b]. Este punto fijo no tiene por qué ser único.

{2} Pero si además, $g'(x)$ existe en (a, b) y $|g'(x)| \leq k < 1$ para todo $x \in (a, b)$, Entonces g tiene un único punto fijo en [a, b].

Según el punto [2] solo permite que exista un único punto fijo en los intervalos, por tanto, el método solo detectará una única raíz. Como ejemplo tenemos a la siguiente función f(x) = x*sin(x)-1, el cual tiene múltiples raíces; siendo g(x) = 1/sin(x), el algoritmo da la siguiente solución:

```
x_ 42 =  1.114157140872201
x* es aproximadamente  1.114157140872201  con error menor que  1e-12
>
```

Dando la solución de la raíz más cercana a cero, pero según wolfram tiene las siguientes soluciones:

$x \approx \pm 9.31724294141481\ldots$

$x \approx \pm 6.43911723841725\ldots$

$x \approx \pm 2.77260470826599\ldots$

$x \approx \pm 1.11415714087193\ldots$

Como es de notar solo detectó la primera raíz.
Toca explicar que pasa cuando hay más de dos raíces, y cómo se solucionó el problema

Comportamiento del método frente a funciones par y periódicas

Como se dijo en el punto anterior el método de punto fijo solo detecta una raíz, pero si es una función par o impar con dos únicas raíces, al hallar solo una raíz ya tenemos la segunda raíz, que se ha de encontrar en el mismo punto de x, pero la parte opuesta del eje.

Un ejemplo ocurre con la función f(x) = cos^2(2x)-x^2, donde es una función par, teniendo dos raíces, según el método de punto fijo arroja lo siguiente con la función g(x) = arccos(x)/2:

```
x_ 51 =  0.514933264660834
x* es aproximadamente  0.514933264660834  con error menor que  1e-12
```

Y según wolfram es:

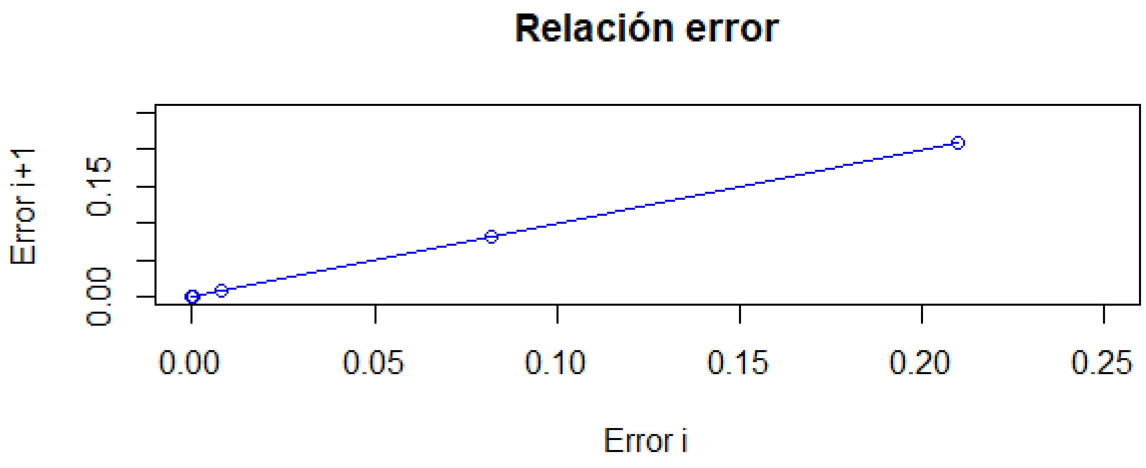
$x \approx 0.514933$

$x \approx -0.514933$

Lo que se ha de hacer para encontrar la otra raíz es cambiar el signo de x, obteniendo como resultado las dos raíces de la función.

Gráfica de relación ε_{i+1} y ε_i

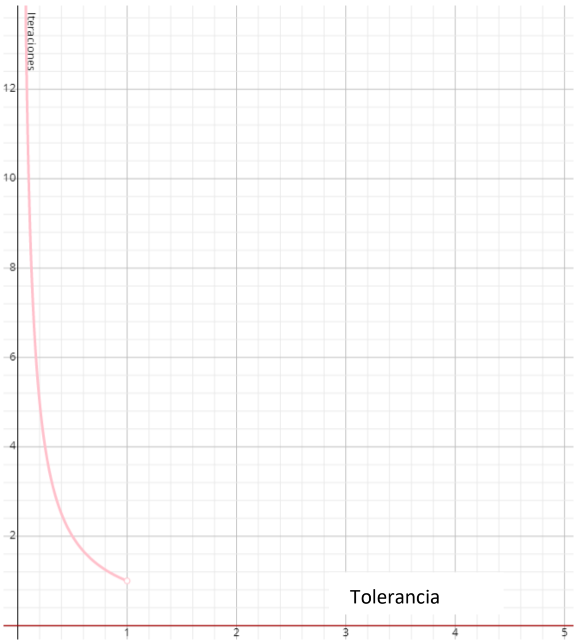
La relación del error presenta una tendencia lineal, con tendencia a cero.



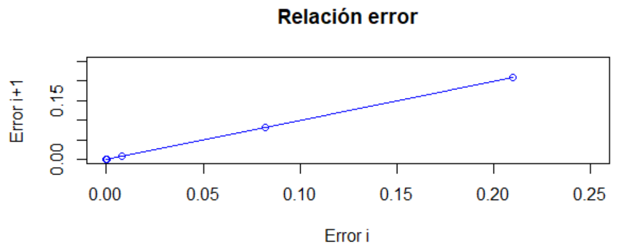
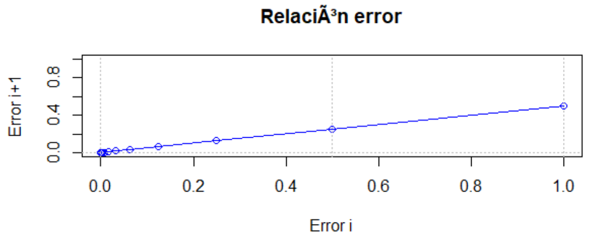
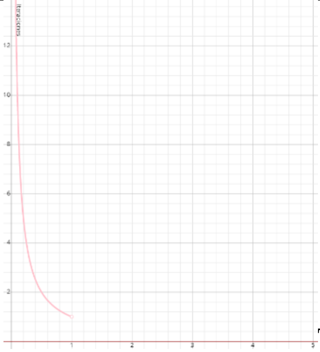
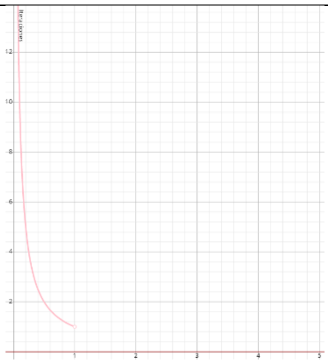
Gráfica de Tolerancia y Número de Iteraciones

Tolerancia es un umbral que al ser superado el detiene las iteraciones del algoritmo que, al ser contrastado con las iteraciones, muestra el comportamiento de las iteraciones cuando la diferencia del error varía.

En este caso entre más pequeño es el valor de la tolerancia, más iteraciones se requieren, Trazando así una curva asíntota al eje de las ordenadas.



Punto Fijo vs Bisección

Punto Fijo	Bisección
	
 Tolerancia vs Iteraciones	 Tolerancia vs Iteraciones
x_ 2 = 0.0081 x_ 3 = 6.561e-05 x_ 4 = 4.3046721e-09 x_ 5 = 1.85302018885e-17 x* es aproximadamente 1.85302018885e-17 con error menor que 1e-08	Cero de f en [0 , 2] las raÃces son: 0.785398162901 y 1.63848876953 con error <= 7.45058059692e-09 El numero de iteraciones es: 28

Resultados

Primero, se debe de tener en cuenta que las restricciones planteadas por el método de punto fijo son más sugerencias que restricciones en sí. Habrá algunas funciones que no cumplen con las restricciones estrictamente, pero si se aproximan. Esto puede ser suficiente para que se puedan realizar por el método.

Resultados -> g(x) = cos(2x) con intervalo (0, 3/2)

```
> f <- expression (cos(2*x))
> f.1<- expression (cos(2*x)^2 - x^2 + x)
> f2 <- expression (1 / (sin(x)))
> f2.1 <- expression (asin(1/x))
> f3 <- expression(sqrt(2*x^2 - (4/3)*x + (8/27)))
> f3.1 <- expression ((-3/7)* x^3 + (6/7)*x^2 + (8/63))
> puntofijo(IntervaloI, IntervaloR, f,0, 1e-8, 100)
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -1.82589050146
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  1
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.408082061813
Comprobación Intervalo Medio (Tolerancia de 0,5):  -0.839071529076
La función ingresada con los parametros no es valida
```

Tomando en cuenta las restricciones de punto fijo, los resultados se aproximan, pero no cumplen con ellas.

Calculando la raíz sin tener en cuenta el método de Stefenssen, nos un valor indeterminado:

```
x_ 66 =  0.841139916656
x_ 67 =  -0.111252719846
x_ 68 =  0.975347625779
x_ 69 =  -0.370826602521
x_ 70 =  0.737352813913
x_ 71 =  0.0959428928871
x_ 72 =  0.981646341816
x_ 73 =  -0.382496131152
x_ 74 =  0.721387716056
x_ 75 =  0.127671484605
x_ 76 =  0.96757672637
x_ 77 =  -0.356348696827
x_ 78 =  0.756600866953
x_ 79 =  0.0575627566432
x_ 80 =  0.993380374256
x_ 81 =  -0.404072300796
x_ 82 =  0.690841086651
x_ 83 =  0.187988915967
x_ 84 =  0.930149024139
x_ 85 =  -0.285474717289
x_ 86 =  0.841388250457
x_ 87 =  -0.111746290466
x_ 88 =  0.975129314208
x_ 89 =  -0.370421074348
x_ 90 =  0.737900446245
x_ 91 =  0.0948526235258
x_ 92 =  0.982059859175
x_ 93 =  -0.383260144847
x_ 94 =  0.720328667322
x_ 95 =  0.129771960688
x_ 96 =  0.966507126566
x_ 97 =  -0.354349115335
x_ 98 =  0.759209770401
x_ 99 =  0.0523528414964
x_ 100 =  0.994523366208
No hubo convergencia
>
```

Pero, al usar cierto método, podemos encontrar la raíz positiva:

```
x_ 1 =  0.673181412986
x_ 2 =  0.859081391038
x_ 3 =  0.993718951669
x_ 4 =  0.893991127465
x_ 5 =  0.611897435867
x_ 6 =  0.518850402929
x_ 7 =  0.514939105863
x_ 8 =  0.514933264674
x* es aproximadamente  0.514933264674  con error menor que  1e-08
```

(Tomando X0 = 0) con Error: 1e-8

```
x_ 1 =  0.673181412986
x_ 2 =  0.859081391038
x_ 3 =  0.993718951669
x_ 4 =  0.893991127465
x_ 5 =  0.611897435867
x_ 6 =  0.518850402929
x_ 7 =  0.514939105863
x_ 8 =  0.514933264674
x_ 9 =  0.514933264661
```

(Tomando X0 = 0) con Error: 1e-12

```
x_ 1 =  0.699763307706
x_ 2 =  0.739070465668
x_ 3 =  0.760834434476
x_ 4 =  0.777880928468
x_ 5 =  0.79453854181
x_ 6 =  0.814227976496
x_ 7 =  0.842639959645
x_ 8 =  0.894730098906
x_ 9 =  0.997448245591
x_ 10 =  0.748032242656
x_ 11 =  0.924255067944
x_ 12 =  0.994742674639
x_ 13 =  0.778445490774
x_ 14 =  0.551249279433
x_ 15 =  0.515450190471
x_ 16 =  0.514933366087
x_ 17 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-08
```

(Tomando X0 = 3/2) con Error: 1e-8

```
x_ 1 =  0.699763307706
x_ 2 =  0.739070465668
x_ 3 =  0.760834434476
x_ 4 =  0.777880928468
x_ 5 =  0.79453854181
x_ 6 =  0.814227976496
x_ 7 =  0.842639959645
x_ 8 =  0.894730098906
x_ 9 =  0.997448245591
x_ 10 =  0.748032242656
x_ 11 =  0.924255067944
x_ 12 =  0.994742674639
x_ 13 =  0.778445490774
x_ 14 =  0.551249279433
x_ 15 =  0.515450190471
x_ 16 =  0.514933366087
x_ 17 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-12
```

(Tomando X0 = 3/2) con Error: 1e-12

Resultados -> g(x) = cos(2x) con intervalo (0, 10)

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -1.82589050146
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  1
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.408082061813
Comprobación Intervalo Medio (Tolerancia de 0,5):  -0.839071529076
La función ingresada con los parametros no es valida
```

)

Resultados de restricciones -> Igual al intervalo (0, 3/2), ya que no hay ninguna desviación grande pero no cumple con restricciones.

(Tomando X0 = 10):

```
x_ 1 =  0.199409022606
x_ 2 =  0.546272790001
x_ 3 =  0.515316448628
x_ 4 =  0.514933320387
x_ 5 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-08
```

```
x_ 1 =  0.199409022606
x_ 2 =  0.546272790001
x_ 3 =  0.515316448628
x_ 4 =  0.514933320387
x_ 5 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-12
```

Resultados -> g(x) = cos(2*x) ^2 - x^2 + x con intervalo (0, 3/2)

```
Derivada con Cota Izquierda:  1
Derivada con Cota Derecha:  -1.4411690036
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  1
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.230085143325
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.1925037517
La función ingresada con los parametros no es valida
```

Igual que los anteriores resultados en restricciones, no los cumple, pero se acercan al resultado. Esto se debe a que las restricciones son más sugerencias.

```
x_ 1 =  0.230085143325
x_ 2 =  0.979920550568
x_ 3 =  0.163548354676
x_ 4 =  1.03356979363
x_ 5 =  0.192082334353
x_ 6 =  1.01472304832
x_ 7 =  0.181076965699
x_ 8 =  1.02276720987
x_ 9 =  0.185659900204
x_ 10 =  1.01953337028
x_ 11 =  0.183796090392
x_ 12 =  1.0208686427
x_ 13 =  0.18456224503
x_ 14 =  1.02032308152
x_ 15 =  0.184248625668
x_ 16 =  1.02054696506
x_ 17 =  0.184377228848
x_ 18 =  1.0204552533
x_ 19 =  0.184324531314
x_ 20 =  1.02049284974
x_ 21 =  0.184346131446
x_ 22 =  1.02047744204
x_ 23 =  0.184337278857
x_ 24 =  1.02048375717
x_ 25 =  0.184340907177
x_ 26 =  1.02048116893
x_ 27 =  0.184339420105
x_ 28 =  1.02048222974
x_ 29 =  0.184340029589
x_ 30 =  1.02048179496
```

Igual, como se puede ver, sin usar el método de Stefenssen, no encuentra la raíz dentro de 100+ iteraciones. Con Stefenssen, si se cumplen:

(Tomando X0 = 0):

```
x_ 1 =  1.027946
x_ 2 =  0.1862181
x_ 3 =  0.4239127
x_ 4 =  0.5118215
x_ 5 =  0.5149302
x_ 6 =  0.5149333
x* es aproximadamente  0.5149333  con error menor que  1e-08>
```

```
x_ 1 =  1.02794570812
x_ 2 =  0.186218050982
x_ 3 =  0.423912662088
x_ 4 =  0.511821477438
x_ 5 =  0.514930203753
x_ 6 =  0.514933264658
x* es aproximadamente  0.514933264658  con error menor que  1e-12
```

(Tomando X0 = 3/2):

```
x_ 1 =  0.699763307706
x_ 2 =  0.739070465668
x_ 3 =  0.760834434476
x_ 4 =  0.777880928468
x_ 5 =  0.79453854181
x_ 6 =  0.814227976496
x_ 7 =  0.842639959645
x_ 8 =  0.894730098906
x_ 9 =  0.997448245591
x_ 10 =  0.748032242656
x_ 11 =  0.924255067944
x_ 12 =  0.994742674639
x_ 13 =  0.778445490774
x_ 14 =  0.551249279433
x_ 15 =  0.515450190471
x_ 16 =  0.514933366087
x_ 17 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-08
```

```
x_ 1 =  0.699763307706
x_ 2 =  0.739070465668
x_ 3 =  0.760834434476
x_ 4 =  0.777880928468
x_ 5 =  0.79453854181
x_ 6 =  0.814227976496
x_ 7 =  0.842639959645
x_ 8 =  0.894730098906
x_ 9 =  0.997448245591
x_ 10 =  0.748032242656
x_ 11 =  0.924255067944
x_ 12 =  0.994742674639
x_ 13 =  0.778445490774
x_ 14 =  0.551249279433
x_ 15 =  0.515450190471
x_ 16 =  0.514933366087
x_ 17 =  0.514933264661
x* es aproximadamente  0.514933264661  con error menor que  1e-12
```

Resultados -> g(x) = cos(2*x) ^2 - x^2 + x con intervalo (0, 10)

```
Derivada con Cota Izquierda:  1
Derivada con Cota Derecha:  -20.490226321
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  1
Comprobación Intervalo Derecho (Tolerancia de 0,5):  -89.8334690308
Comprobación Intervalo Medio (Tolerancia de 0,5):  -19.2959589691
La función ingresada con los parametros no es valida
```

Se puede ver que estas al calcular las restricciones, están varían en una gran medida, llevando que al tomar X0=10, diverge.

```
> f2 <- expression (1 / (sin(x)))
> f2.1 <- expression (asin(1/x))
> f3 <- expression(sqrt(2*x^2 - (4/3)*x + (8/27)))
> f3.1 <- expression ((-3/7)* x^3 + (6/7)*x^2 + (8/63))
> puntofijo(IntervaloI, IntervaloR, f.1,10, 1e-8, 100)
x_ 1 =  -66580717.7447
x_ 2 =  -4064484357603657
x_ 3 =  -1.65034925044e+31
x_ 4 =  -2.72365230657e+62
x_ 5 =  -7.41828188708e+124
x_ 6 =  -5.50309061562e+249
x_ 7 =  -Inf
Error in if (dx < tol || k > maxIter) break :
  valor ausente donde TRUE/FALSE es necesario
```

Resultados -> g(x) = 1 / (sin(x) con intervalo (-1, 2)

Cumple con las restricciones.

Tomando X0 = -1

```
Derivada con Cota Izquierda:  -0.763059722233
Derivada con Cota Derecha:  0.503308973344
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  -1.18839510578
Comprobación Intervalo Derecho (Tolerancia de 0,5):  1.09975017029
Comprobación Intervalo Medio (Tolerancia de 0,5):  2.08582964293
x_ 1 =  -1.13514685679
x_ 2 =  -1.11392287918
x_ 3 =  -1.11415711157
x_ 4 =  -1.11415714087
x* es aproximadamente  -1.11415714087  con error menor que  1e-08
```

```
Derivada con Cota Izquierda:  -0.763059722233
Derivada con Cota Derecha:  0.503308973344
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  -1.18839510578
Comprobación Intervalo Derecho (Tolerancia de 0,5):  1.09975017029
Comprobación Intervalo Medio (Tolerancia de 0,5):  2.08582964293
x_ 1 =  -1.13514685679
x_ 2 =  -1.11392287918
x_ 3 =  -1.11415711157
x_ 4 =  -1.11415714087
x* es aproximadamente  -1.11415714087  con error menor que  1e-12
```

Se observa que tanto optimiza la respuesta comparado sin usar Stefenssen:

```
x_ 1 =  1.09975017029
x_ 2 =  1.12221604932
x_ 3 =  1.10979942063
x_ 4 =  1.11655808124
x_ 5 =  1.11284773486
x_ 6 =  1.11487526422
x_ 7 =  1.11376450044
x_ 8 =  1.11437218073
x_ 9 =  1.11403947659
x_ 10 =  1.11422155608
x_ 11 =  1.11412188652
x_ 12 =  1.11417643844
x_ 13 =  1.11414657862
x_ 14 =  1.11416292223
x_ 15 =  1.11415397646
x_ 16 =  1.11415887293
x_ 17 =  1.11415619283
x_ 18 =  1.11415765978
x_ 19 =  1.11415685684
x_ 20 =  1.11415729634
x_ 21 =  1.11415705578
x_ 22 =  1.11415718745
x_ 23 =  1.11415711538
x_ 24 =  1.11415715483
x_ 25 =  1.11415713323
x_ 26 =  1.11415714505
x_ 27 =  1.11415713858
x* es aproximadamente  1.11415713858  con error menor que  1e-08
```

(sin uso de Stefenssen)

Tomando X0 = 2

```
Derivada con Cota Izquierda:  -0.763059722233
Derivada con Cota Derecha:  0.503308973344
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  -1.18839510578
Comprobación Intervalo Derecho (Tolerancia de 0,5):  1.09975017029
Comprobación Intervalo Medio (Tolerancia de 0,5):  2.08582964293
x_ 1 =  1.10979942063
x_ 2 =  1.11414696643
x_ 3 =  1.11415714082
x* es aproximadamente  1.11415714082  con error menor que  1e-08
```

```
Derivada con Cota Izquierda:  -0.763059722233
Derivada con Cota Derecha:  0.503308973344
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  -1.18839510
Comprobación Intervalo Derecho (Tolerancia de 0,5):  1.0997501702
Comprobación Intervalo Medio (Tolerancia de 0,5):  2.08582964293
x_ 1 =  1.10979942063
x_ 2 =  1.11414696643
x_ 3 =  1.11415714082
x_ 4 =  1.11415714087
x* es aproximadamente  1.11415714087  con error menor que  1e-12
```

Resultados -> g(x) = arcsin(1/x) con intervalo (-1, 2)

```
Derivada con Cota Izquierda:  -Inf
Derivada con Cota Derecha:  -0.288675134595
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  -1.57079632679
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.523598775598
Comprobación Intervalo Medio (Tolerancia de 0,5):  NaN
```

En este caso, las restricciones varían de gran manera. Por esta razón, no se puede usar el método del punto fijo para este g(x) en este intervalo:

```
x_ 1 =  NaN
```

Resultados -> g(x) = raíz(2*x^2 - (4/3)*x + (8/27)) con intervalo (0, 3/2)

```
Derivada con Cota Izquierda:  -1.22474487139
Derivada con Cota Derecha:  1.39535653779
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.544331053952
Comprobación Intervalo Derecho (Tolerancia de 0,5):  1.6722129937
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.649073413642
La función ingresada con los parametros no es valida
```

No cumple con las restricciones dentro del intervalo, pero se aproxima a ellas suficientemente para poder ser realizado en el método de Stefenssen.

Tomando X0 = 0

```
x_ 1 =  0.289872429293
x_ 2 =  0.609173010986
x_ 3 =  1.17063031092
x_ 4 =  1.05354921574
x_ 5 =  1.05156752741
x_ 6 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-08
```

```
x_ 1 =  0.289872429293
x_ 2 =  0.609173010986
x_ 3 =  1.17063031092
x_ 4 =  1.05354921574
x_ 5 =  1.05156752741
x_ 6 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-12
```

Tomando X0 = 3/2

```
x_ 1 =  2.25039472403
x_ 2 =  1.10750714033
x_ 3 =  1.05205654796
x_ 4 =  1.05156688782
x_ 5 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-08
```

```
x_ 1 =  2.25039472403
x_ 2 =  1.10750714033
x_ 3 =  1.05205654796
x_ 4 =  1.05156688782
x_ 5 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-12
```

Resultados -> g(x) = raíz(2*x^2 - (4/3)*x + (8/27)) con intervalo (0, 10)

```
Derivada con Cota Izquierda:  -1.22474487139
Derivada con Cota Derecha:  1.41393338138
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.5443310539
Comprobación Intervalo Derecho (Tolerancia de 0,5):  13.6734400559
Comprobación Intervalo Medio (Tolerancia de 0,5):  6.60527286565
La función ingresada con los parametros no es valida>
>
```

No cumple con las restricciones dentro del intervalo, pero se aproxima a ellas suficientemente para poder ser realizado en el método de Stefenssen.

Tomando X0 = 10

```
x_ 1 =  26.2129981118
x_ 2 =  1.25452996314
x_ 3 =  1.05657970355
x_ 4 =  1.05157117683
x_ 5 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-08
>
```

```
x_ 1 =  26.2129981118
x_ 2 =  1.25452996314
x_ 3 =  1.05657970355
x_ 4 =  1.05157117683
x_ 5 =  1.05156684613
x_ 6 =  1.05156684613
x* es aproximadamente  1.05156684613  con error menor que  1e-12
```

Resultados -> g(x) = (-3/7) * x^3 + (6/7)*x^2 + (8/63) con intervalo (0, 3/2)

La ecuación cumple con las restricciones, y se pudieron encontrar las raíces:

Tomando X0 = 0

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -0.321428571429
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.126984126984
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.609126984127
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.428323412698
x_ 1 =  0.14259266044
x_ 2 =  0.143330654568
x_ 3 =  0.143331157837
x* es aproximadamente  0.143331157837  con error menor que  1e-08
```

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -0.321428571429
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.126984126984
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.609126984127
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.428323412698
x_ 1 =  0.14259266044
x_ 2 =  0.143330654568
x_ 3 =  0.143331157837
x_ 4 =  0.143331157838
```

Tomando X0 = 3/2

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -0.321428571429
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.126984126984
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.609126984127
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.428323412698
x_ 1 =  0.21279375322
x_ 2 =  0.142752006179
x_ 3 =  0.143330853117
x_ 4 =  0.143331157838
x* es aproximadamente  0.143331157838  con error menor que  1e-08
```

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -0.321428571429
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.126984126984
Comprobación Intervalo Derecho (Tolerancia de 0,5):  0.609126984127
Comprobación Intervalo Medio (Tolerancia de 0,5):  0.428323412698
x_ 1 =  0.21279375322
x_ 2 =  0.142752006179
x_ 3 =  0.143330853117
x_ 4 =  0.143331157838
x_ 5 =  0.143331157838
```

Resultados -> g(x) = (-3/7) * x^3 + (6/7)*x^2 + (8/63) con intervalo (0, 10)

```
Derivada con Cota Izquierda:  0
Derivada con Cota Derecha:  -111.428571429
Comprobación Intervalo Izquierdo (Tolerancia de 0,5):  0.126984126984
Comprobación Intervalo Derecho (Tolerancia de 0,5):  -342.73015873
Comprobación Intervalo Medio (Tolerancia de 0,5):  -32.0158730159
La función ingresada con los parametros no es valida
> 
```

Los resultados para las restricciones varían demasiado con este intervalo. Esto significa que no es posible aplicar el método dentro de este.

```
x_ 1 =  -2.23998231171e+21
x_ 2 =  4.81678183926e+63
x_ 3 =  -4.78954375413e+190
x_ 4 =  Inf
```