Taller Análisis Numérico

Jenifer Medina Yepez
Oscar Andrés Pacheco Turizo
Pontificia Universidad Javeriana
Bogotá D.C

Febrero 2021

Resumen

El siguiente trabajo contiene la investigación e implementación de dos métodos: Newton Raphson y $\triangle^2$ de Aitken, estos métodos nos permiten hallar las raíces de una función en especial, se hará un análisis teniendo en cuenta parámetros como la convergencia, la pérdida de significancia, el número de errores y el comportamiento propio del método bajo ciertas cirscuntancias.

1. Newton Raphson

1.1. Explicación Geométrica

El método consiste en acercarse a la raíz p de una función continua, mediante la siguiente fórmula recursiva

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \tag{1}$$

Este método tiene la ventaja de ser muy rápido, es decir, converge a la solución en menos iteraciones en comparación a otros métodos como el de bisección. Una desventaja importante es que este método cerca de los máximos y mínimos locales tiene una convergencia lenta, y presenta conflictos a la hora de trabajar con raíces múltiples.

Se toma un X_1 cercano a la raíz y se traza la recta tangente a la gráfica con el fin de ver el punto de corte en el eje X y así se obtiene un punto X_2. El método se repite varias veces hasta obtener un X con una gran aproximación a la raíz.

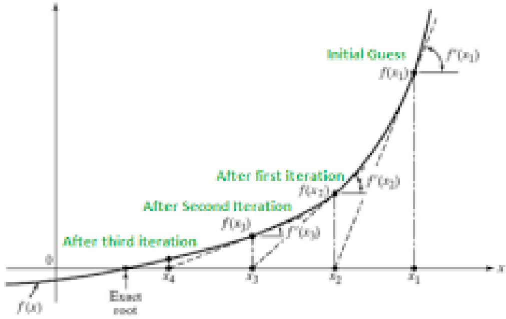


Figura 1: Interpretación Geométrica.

1.2. Condiciones del método

Este método tiene tres condiciones, la primera de ellas es que as funciones deben ser diferenciables dos veces en el intervalo $[a, b]$, la segunda corresponde a que la expresión $f'(s) = 0$ no debe cumplirse y por último, la única forma de alcanzar la convergencia es seleccionar un valor lo suficientemente cercano a la raíz buscada.

1.3. Diagrama de flujo

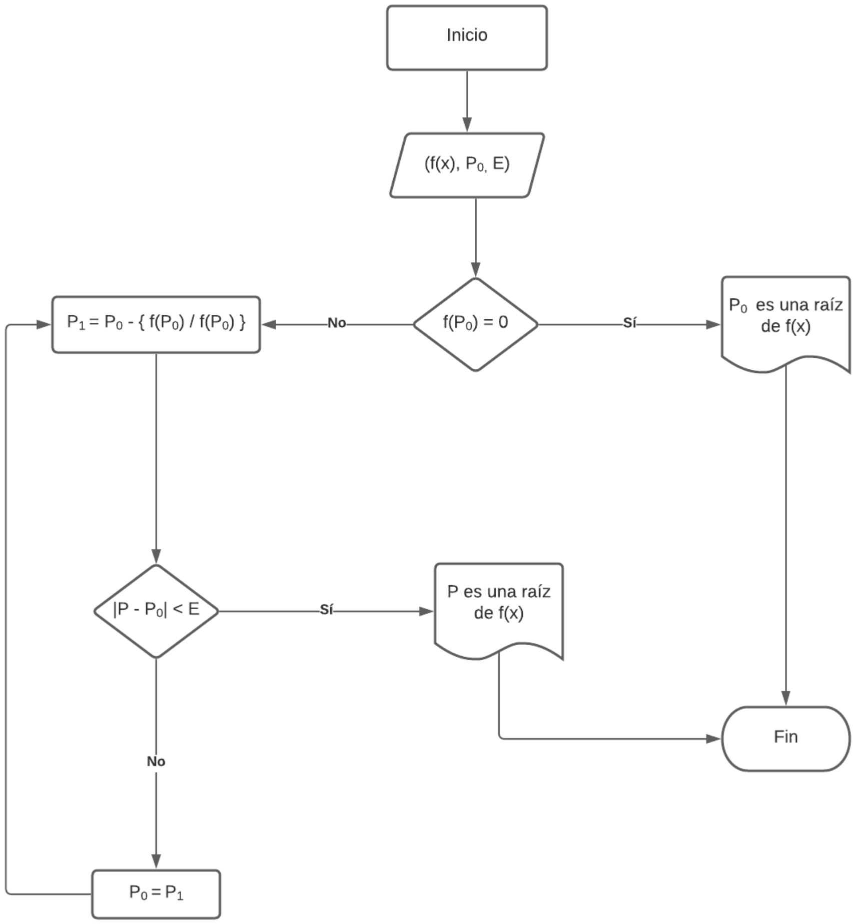


Figura 2: Diagrama de flujo.

1.4. Ejercicios

1.4.1. Ejercicio 1

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

Figura 3: Función 1

Roots:

$x \approx 0.739085$

$x \approx -0.739085$

Figura 4: Raíces Función 1 por Wolfram

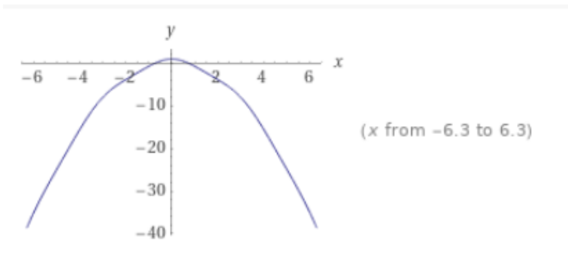


Figura 5: Gráfica Función 1

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.2433829596605205736485549607288268157047653932602921936

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.017401058983126756969742368746374326365820128635820184388

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.0001308405823451406787057718362290943885183788183221101353

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 7.5588468618157375933441066983217146864837665028416895673e-9

Solucion converge en  4  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.73908513321516066688725930534446306521918141280179900903
```

Figura 6: Resultado Tolerancia 10^{-8}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.2433829596605205736485549607288268157047653932602921936

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.017401058983126756969742368746374326365820128635820184388

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.0001308405823451406787057718362290943885183788183221101353

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 7.5588468618157375933441066983217146864837665028416895673e-9

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 2.5231947217670589380053746671610166104508830547921237545e-17

Solucion converge en  5  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.73908513321516064165531208767387368516543474119163290452
```

Figura 7: Resultado Tolerancia 10^{-16}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.2433829596605205736485549607288268157047653932602921936

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.017401058983126756969742368746374326365820128635820184388

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.0001308405823451406787057718362290943885183788183221101353

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 7.5588468618157375933441066983217146864837665028416895673e-9

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 2.5231947217670589380053746671610166104508830547921237545e-17

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 2.8115202298229087543949681447612859957062649327990054452e-34

Solucion converge en  6  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.73908513321516064165531208767387340401341175890075746503
```

Figura 8: Resultado Tolerancia 10^{-32}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.24338295966052057364855496072882681570476539326029219936

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.017401058983126756969742368746374326365820128635820184388

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.00013084058234514067870577183622909438851837881832211011353

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 7.5588468618157375933441066983217146864837665028416895673e-9

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 2.5231947217670589380053746671610166104508830547921237545e-17

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 2.8115202298229087543949681447612859957062649327990054452e-34

Iteracion 7
1 'mpfr' number of precision  180   bits
[1] 0

Solucion converge en  7  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.73908513321516064165531208767387340401341175890075746503
```

Figura 9: Resultado Tolerancia 10^{-56}

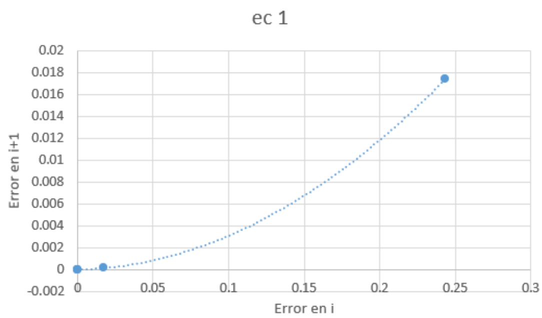


Figura 10: Convergencia en Ejercicio 1

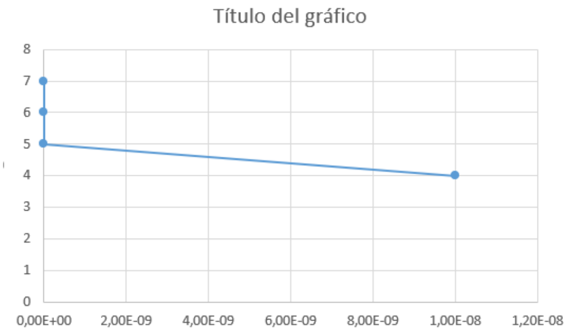


Figura 11: Gráfica Tolerancia Vs Iteraciones

1.4.2. Ejercicio 2

$$\boldsymbol{f(x) = xsin(x) - 1 \;\; [-1, 2]}$$

Figura 12: Función 2

Roots:

$x \approx 0.739085$

$x \approx -0.739085$

Figura 13: Raíces Función 2 por Wolfram

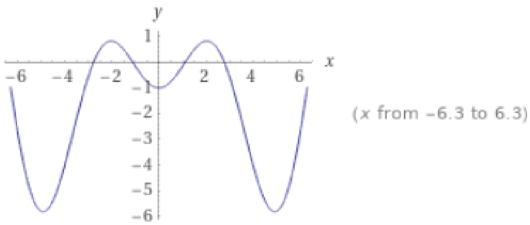


Figura 14: Gráfica Función 2

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.44965759793290057973070279447791013671576739957779202347

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.063838848560016041484437716031573934464327073191866470081

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 2.4109779917900954989847393355686638180986834169459730038e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 2.4732526501754088587840233202846104966851410896538350256e-11

Solucion converge en  4  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 1.1141571408719300873004991627481483443132247917467563978
```

Figura 15: Resultado Tolerancia 10^{-8}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.44965759793290057973070279447791013671576739957779202347

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.063838848560016041484437716031573934464327073191866470081

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 2.4109779917900954989847393355686638180986834169459730038e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 2.4732526501754088587840233202846104966851410896538350256e-11

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 2.601542105555964087658427383490131942483006607755826893e-23

Solucion converge en  5  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 1.1141571408719300873005251781692039039541013760205912991
```

Figura 16: Resultado Tolerancia 10^{-16}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.44965759793290057973070279447791013671576739957779202347

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.063838848560016041484437716031573934464327073191866470081

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 2.4109779917900954989847393355686638180986834169459730038e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 2.4732526501754088587840233202846104966851410896538350256e-11

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 2.601542105555964087658427383490131942483006607755826893e-23

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 2.8784296478328412220848631195046763000686395172607218698e-47

Solucion converge en  6  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 1.1141571408719300873005251781692039039541013760493755956
```

Figura 17: Resultado Tolerancia 10^{-32}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.24338295966052057364855496072882681570476539326029219 36
```

Figura 18: Resultado Tolerancia 10^{-56}

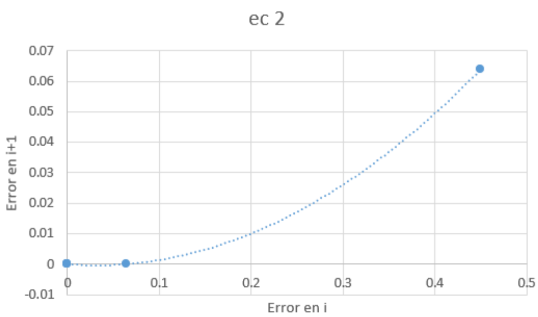


Figura 19: Convergencia en Ejercicio 2

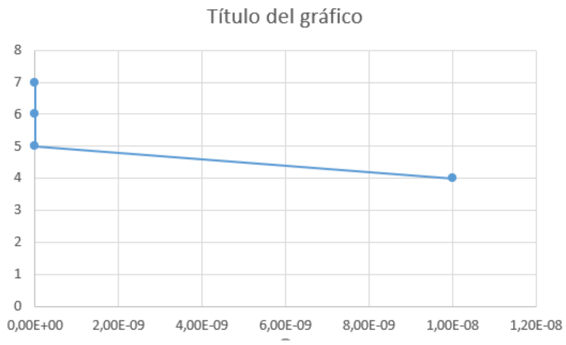


Figura 20: Gráfica Tolerancia Vs Iteraciones

1.4.3. Ejercicio 3

$$f(x) = x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$$

Figura 21: Función 3

Root:

$$x = \frac{2}{3}$$

Figura 22: Raíces Función 3 por Wolfram

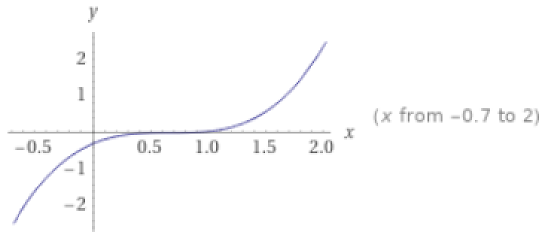


Figura 23: Gráfica Función 3

```
Iteracion 21
1 'mpfr' number of precision  180   bits
[1] 3.341343397053579039478870600052206769469990992158806092 1e-5

Iteracion 22
1 'mpfr' number of precision  180   bits
[1] 2.2274258652266850468381231093944866187382699418603614982e-5

Iteracion 23
1 'mpfr' number of precision  180   bits
[1] 1.4846437370106487706186973198886549985281546056096807261e-5

Iteracion 24
1 'mpfr' number of precision  180   bits
[1] 9.8907251219370550120679085826283506673252510874080126995e-6

Iteracion 25
1 'mpfr' number of precision  180   bits
[1] 6.5783227669468312024969883031507071664474835225538240225e-6

Iteracion 26
1 'mpfr' number of precision  180   bits
[1] 4.3509168295340962991720658739780495155908550870280024291e-6

Iteracion 27
1 'mpfr' number of precision  180   bits
[1] 2.8243991489461893362023422440729438909422008676990790473e-6

Iteracion 28
1 'mpfr' number of precision  180   bits
[1] 1.7236049610622755729262875084455777222058855025266912046e-6

Iteracion 29
1 'mpfr' number of precision  180   bits
[1] 8.6539946356357073542673168575510261732286054631145799131e-7

Iteracion 30
1 'mpfr' number of precision  180   bits
[1] 2.5120768790971342470790035803657158803419331301687181884e-7

Iteracion 31
1 'mpfr' number of precision  180   bits
[1] 2.0584582325511351030762427539204960861968538129680461728e-8

Iteracion 32
1 'mpfr' number of precision  180   bits
[1] 1.3280363327781248040646724850024084820285506065325361414e-10

Solucion converge en  32  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.66666987081929470460568798140634690816869431650638054886
```

Figura 24: Resultado Tolerancia 10^{-8}

```
Iteracion 23
1 'mpfr' number of precision  180   bits
[1] 1.4846437370106487706186973198886549985281546056096807261e-5

Iteracion 24
1 'mpfr' number of precision  180   bits
[1] 9.8907251219370550120679085826283506673252510874080126995e-6

Iteracion 25
1 'mpfr' number of precision  180   bits
[1] 6.5783227669468312024969883031507071664474835225538240225e-6

Iteracion 26
1 'mpfr' number of precision  180   bits
[1] 4.3509168295340962991720658739780495155908550870280024291e-6

Iteracion 27
1 'mpfr' number of precision  180   bits
[1] 2.8243991489461893362023422440729438909422008676990790473e-6

Iteracion 28
1 'mpfr' number of precision  180   bits
[1] 1.7236049610622755729262875084455777222058855025266912046e-6

Iteracion 29
1 'mpfr' number of precision  180   bits
[1] 8.6539946356357073542673168575510261732286054631145799131e-7

Iteracion 30
1 'mpfr' number of precision  180   bits
[1] 2.5120768790971342470790035803657158803419331301687181884e-7

Iteracion 31
1 'mpfr' number of precision  180   bits
[1] 2.0584582325511351030762427539204960861968538129680461728e-8

Iteracion 32
1 'mpfr' number of precision  180   bits
[1] 1.3280363327781248040646724850024084820285506065325361414e-10

Iteracion 33
1 'mpfr' number of precision  180   bits
[1] 5.5045239025838886795994391788531038562590488110798503118e-15

Iteracion 34
1 'mpfr' number of precision  180   bits
[1] 9.4564335135697318495983610682963429760068891052150100062e-24

Solucion converge en  34  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.66666987081928920008177594108415373899766586504145599347
```

Figura 25: Resultado Tolerancia 10^{-16}

```
1 'mpfr' number of precision  180   bits
[1] 1.4846437370106487706186973198886549985281546056096807261e-5

Iteracion 24
1 'mpfr' number of precision  180   bits
[1] 9.8907251219370550120679085826283506673252510874080126995e-6

Iteracion 25
1 'mpfr' number of precision  180   bits
[1] 6.5783227669468312024969883031507071664474835225538240225e-6

Iteracion 26
1 'mpfr' number of precision  180   bits
[1] 4.3509168295340962991720658739780495155908550870280024291e-6

Iteracion 27
1 'mpfr' number of precision  180   bits
[1] 2.8243991489461893362023422440729438909422008676990790473e-6

Iteracion 28
1 'mpfr' number of precision  180   bits
[1] 1.7236049610622755729262875084455777222058855025266912046e-6

Iteracion 29
1 'mpfr' number of precision  180   bits
[1] 8.6539946356357073542673168575510261732286054631145799131e-7

Iteracion 30
1 'mpfr' number of precision  180   bits
[1] 2.5120768790971342470790035803657158803419331301687181884e-7

Iteracion 31
1 'mpfr' number of precision  180   bits
[1] 2.0584582325511351030762427539204960861968538129680461728e-8

Iteracion 32
1 'mpfr' number of precision  180   bits
[1] 1.3280363327781248040646724850024084820285506065325361414e-10

Iteracion 33
1 'mpfr' number of precision  180   bits
[1] 5.5045239025838886795994391788531038562590488110798503118e-15

Iteracion 34
1 'mpfr' number of precision  180   bits
[1] 9.4564335135697318495983610682963429760068891052150100062e-24

Iteracion 35
1 'mpfr' number of precision  180   bits
[1] 2.7923483135325383763638585744442425693332407316181285764e-41

Solucion converge en  35  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.66666987081928920008177594108415373899763794155832066809
```

Figura 26: Resultado Tolerancia 10^{-32}

```
Iteracion 32
1 'mpfr' number of precision  180   bits
[1] 1.3280363327781248040646724850024084820285506065325361414e-10

Iteracion 33
1 'mpfr' number of precision  180   bits
[1] 5.5045239025838886795994391788531038562590488110798503118e-15

Iteracion 34
1 'mpfr' number of precision  180   bits
[1] 9.456433513569731849598361068296342976006889105215010062e-24

Iteracion 35
1 'mpfr' number of precision  180   bits
[1] 2.7923483135325383763638585744442425693332407316181285764e-41

Iteracion 36
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 37
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 38
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 39
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 40
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 41
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 42
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 43
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 44
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44

Iteracion 45
1 'mpfr' number of precision  180   bits
[1] 2.1186254275594863050900308346789439738848873786344058774e-44
```

Figura 27: Resultado Tolerancia 10^{-56}

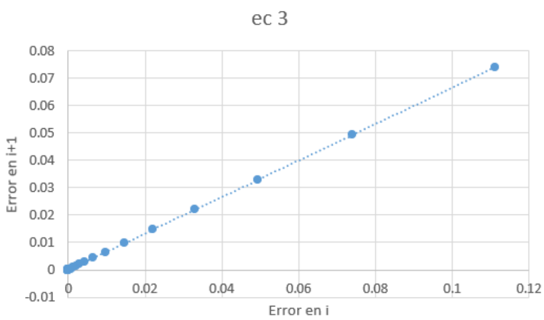


Figura 28: Convergencia en Ejercicio 3

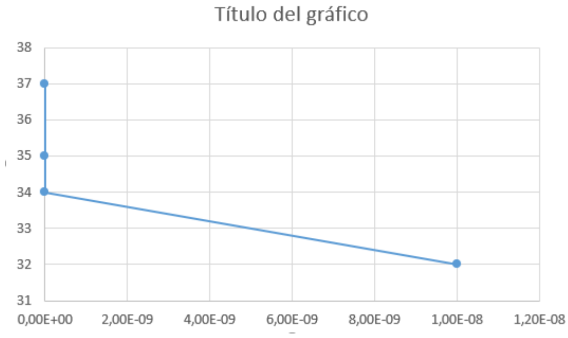


Figura 29: Gráfica Tolerancia Vs Iteraciones

1.4.4. Ejercicio 4

$$\frac{9.80665(68.1)}{w}\left(1 - e^{-\left(\frac{w}{68.1}\right)10}\right) - 40$$

Figura 30: Función 4

Numerical root:

$x \approx 14.7802038316611\ldots$

Figura 31: Raíces Función 4 por Wolfram

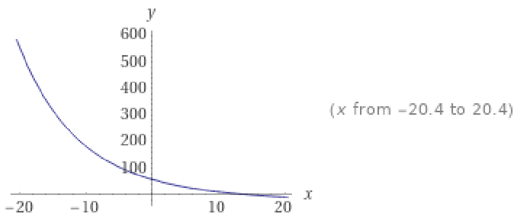


Figura 32: Gráfica Función 4

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 7.6366417148580276177624986594114110643768910744536523066

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 4.3083791984794243650243027741367869423254010186747849588

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.98428550980226801781714461939521142197894980674607615576

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 0.040314855144972385770168659424527963909693337590236717591

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 6.3596345772582428208905252260734870131255335362756387171e-5

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.6564549653096512268591702676716450825997817117581423968e-10

Solucion converge en  6  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 14.969684874796110465333288740306115154228231001060091066
```

Figura 33: Resultado Tolerancia 10^{-8}

```
Iteracion 23
1 'mpfr' number of precision  180   bits
[1] 1.4846437370106487706186973198886549985281546056096807261e-5

Iteracion 24
1 'mpfr' number of precision  180   bits
[1] 9.8907251219370550120679085826283506673252510874080126995e-6

Iteracion 25
1 'mpfr' number of precision  180   bits
[1] 6.5783227669468312024969883031507071664474835225538240225e-6

Iteracion 26
1 'mpfr' number of precision  180   bits
[1] 4.3509168295340962991720658739780495155908550870280024291e-6

Iteracion 27
1 'mpfr' number of precision  180   bits
[1] 2.8243991489461893362023422440729438909422008676990790473e-6

Iteracion 28
1 'mpfr' number of precision  180   bits
[1] 1.7236049610622755729262875084455777222058855025266912046e-6

Iteracion 29
1 'mpfr' number of precision  180   bits
[1] 8.6539946356357073542673168575510261732286054631145799131e-7

Iteracion 30
1 'mpfr' number of precision  180   bits
[1] 2.5120768790971342470790035803657158803419331301687181884e-7

Iteracion 31
1 'mpfr' number of precision  180   bits
[1] 2.0584582325511351030762427539204960861968538129680461728e-8

Iteracion 32
1 'mpfr' number of precision  180   bits
[1] 1.3280363327781248040646724850024084820285506065325361414e-10

Iteracion 33
1 'mpfr' number of precision  180   bits
[1] 5.5045239025838886795994391788531038562590488110798503118e-15

Iteracion 34
1 'mpfr' number of precision  180   bits
[1] 9.4564335135697318495983610682963429760068891052150100062e-24

Solucion converge en  34  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 0.66666987081928920008177594108415373899766586504145599347
```

Figura 34: Resultado Tolerancia 10^{-16}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 7.6366417148580276177624986594114110643768910744536523066

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 4.3083791984794243650243027741367869423254010186747849588

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.98428550980226801781714461939521142197894980674607615576

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 0.040314855144972385770168659424527963909693337590236717591

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 6.3596345772582428208905252260734870131255335362756387171e-5

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.6564549653096512268591702676716450825997817117581423968e-10

Iteracion 7
1 'mpfr' number of precision  180   bits
[1] 2.0334361079660852032808240184988088378837781653787791129e-17

Iteracion 8
1 'mpfr' number of precision  180   bits
[1] 2.4960803354101729855984875904898835929638787492991371287e-24

Iteracion 9
1 'mpfr' number of precision  180   bits
[1] 3.0639846594506317855846573895107804126183106975872992239e-31

Iteracion 10
1 'mpfr' number of precision  180   bits
[1] 3.7610976939194163081465835074955404924767729529975734923e-38

Solucion converge en  10  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 14.969684874796110485667652316047608995713012824653522564
```

Figura 35: Resultado Tolerancia 10^{-32}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 7.636641714858027617762498659411411064376891074453652 3066

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 4.3083791984794243650243027741367869423254010186747849588

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 0.98428550980226801781714461939521142197894980674607615576

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 0.040314855144972385770168659424527963909693337590236717591

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 6.3596345772582428208905252260734870131255335362756387171e-5

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.6564549653096512268591702676716450825997817117581423968e-10

Iteracion 7
1 'mpfr' number of precision  180   bits
[1] 2.0334361079660852032808240184988088378837781653787791129e-17

Iteracion 8
1 'mpfr' number of precision  180   bits
[1] 2.4960803354101729855984875904898835929638787492991371287e-24

Iteracion 9
1 'mpfr' number of precision  180   bits
[1] 3.0639846594506317855846573895107804126183106975872992239e-31

Iteracion 10
1 'mpfr' number of precision  180   bits
[1] 3.7610976939194163081465835074955404924767729529975734923e-38

Iteracion 11
1 'mpfr' number of precision  180   bits
[1] 4.6168167849952436303930870739411496208912685693425052733e-45

Iteracion 12
1 'mpfr' number of precision  180   bits
[1] 5.6378630603507251910776941103976570881258059189715870973e-52

Iteracion 13
1 'mpfr' number of precision  180   bits
[1] 2.0880974297595278485472941149620952178243725625820692953e-53
```

Figura 36: Resultado Tolerancia 10^{-56}

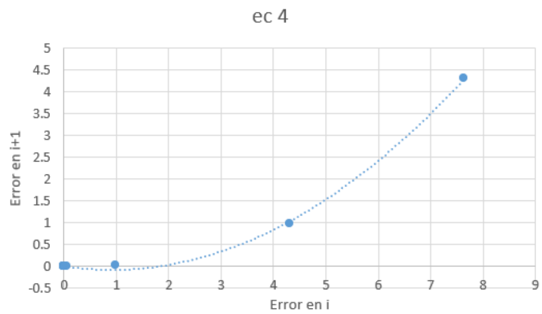


Figura 37: Convergencia en Ejercicio 4

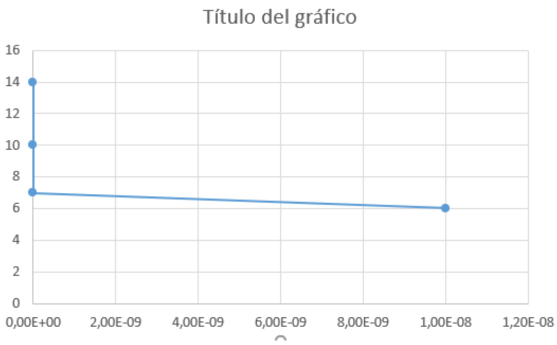


Figura 38: Gráfica Tolerancia Vs Iteraciones

1.4.5. Ejercicio 5

$$\boldsymbol{x^3 - 2x - 5 = 0}$$

Figura 39: Función 5

Real root:

$$x \approx 2.0946$$

Figura 40: Raíces Función 5 por Wolfram

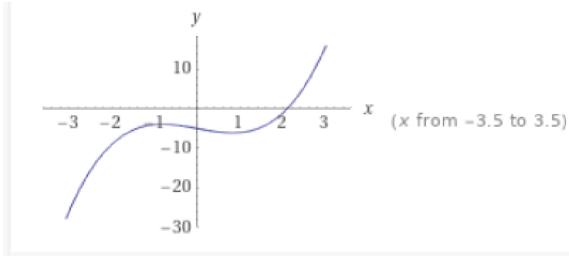


Figura 41: Gráfica Función 5

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.09999999999999999999999999999999999999999999999999999998956

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.0054318788958147818343722172751558325912733748886910048773

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 1.6639405985915281804079180578301928063379556124291091393e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 1.5587271138775885639055187001120200549664129412336220293e-10

Solucion converge en  4  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 2.094551481542326591496064847153713995469461240058543408 9
```

Figura 42: Resultado Tolerancia 10^{-8}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.09999999999999999999999999999999999999999999999999999998956

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.0054318788958147818343722172751558325912733748886910048773

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 1.6639405985915281804079180578301928063379556124291091393e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 1.5587271138775885639055187001120200549664129412336220293e-10

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 1.3678306574411031612049803281637822090281812636221633648e-20

Solucion converge en  5  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 2.0945514815423265914823865405793029638574114367769055868
```

Figura 43: Resultado Tolerancia 10^{-16}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.09999999999999999999999999999999999999999999999999999998956

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.0054318788958147818343722172751558325912733748886910048773

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 1.6639405985915281804079180578301928063379556124291091393e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 1.5587271138775885639055187001120200549664129412336220293e-10

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 1.3678306574411031612049803281637822090281812636221633648e-20

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.0533114866640520985246917317923258586580631809606026012e-40

Solucion converge en  6  iteraciones.
Raiz 1 'mpfr' number of precision  180   bits
[1] 2.0945514815423265914823865405793029638573061056282391816
```

Figura 44: Resultado Tolerancia 10^{-32}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.09999999999999999999999999999999999999999999999999999998956

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.0054318788958147818343722172751558325912733748886910048773

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 1.6639405985915281804079180578301928063379556124291091393e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 1.5587271138775885639055187001120200549664129412336220293e-10

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 1.3678306574411031612049803281637822090281812636221633648e-20

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.0533114866640520985246917317923258586580631809606026012e-40

Iteracion 7
1 'mpfr' number of precision  180   bits
[1] 0
```

Figura 45: Resultado Tolerancia 10^{-56}

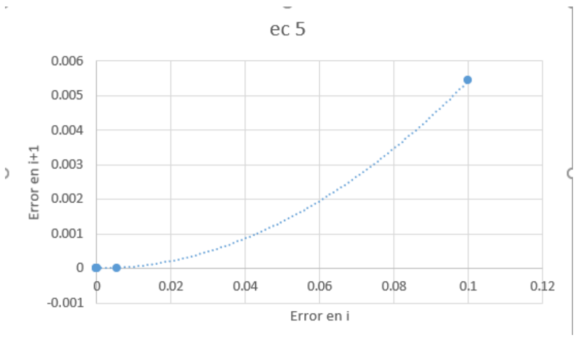


Figura 46: Convergencia en Ejercicio 4

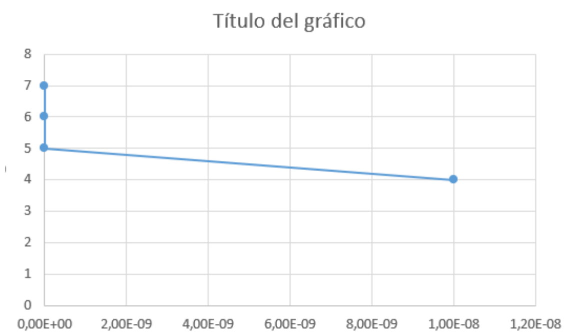


Figura 47: Gráfica Tolerancia Vs Iteraciones

1.5. Análisis de resultados

Para la implementación de este método decidimos usar la libreria Rmpfr de R, que nos permite alcanzar niveles de presión más alto en los datos, por lo que el problema de pérdida de significancia se soluciona, sin embargo, debemos aclarar que este método es bastante preciso. En todos los ejercicios, se pudo observar que el método efectivamente converge cuadráticamente. Además, el número de iteraciones realizadas para encontrar la raíces son pocas, por lo que es un método rápido.

1.6. Newton Raphson y Raíces Múltiples

El método tiene un orden de convergencia es cuadrática, pero pierde su convergencia cuadrática cuando la raíz buscada es de multiplicidad algebraica mayor a uno y pasa a ser lineal de constante asintótica de convergencia $(1-1)/m$, con m=multiplicidad de la raíz. El problema radica en que tanto $f(x)$ como $f'(x)$ se aproximen a 0, debido a que tendríamos en el denominador una división entre cero cuando la solución converge cerca de la raíz.

1.7. Newton Raphson y Bisección

El método de Newton Raphson puede llegar a ser más eficiente que el método de Bisección, esto sólo si el método converge, debido a que la convergencia de Bisección es muy lenta en comparación a la de Newton Raphson. También, presenta mejores niveles de presión, pero, con el riesgo de que bajo ciertas circunstancias el método no converja, en cambio, bisección siempre converge, que además es una ventaja que tiene en comparación a Polinomios de Taylor, ya que no todas las funciones pueden ser escritas como una serie de Taylor.

2. $\triangle^2$ de Aitken

2.1. Explicación

Es un método de aceleración de la convergencia, por lo que es útil para acelerar la convergencia de una sucesión que converge lineal e independientemente de su origen.

$$x' = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

Figura 48: Formula de Aitken

Como ya se mencionó anteriormente, el método de $\triangle^2$ de Aitken converge linealmente, sin embargo, para un método de punto fijo, o sea para una sucesión $x_(n+1) = f(x_n)$ la convergencia es cuadrática y pasa a ser llamado Método de Steffense. Este método no se ve afectado cuando hay más de dos raíces, es decir, cuando hay raíces múltiples, de hecho, una forma de solucionar el problema que se genera en el método de Newton Raphson se soluciona aplicando este método. Tampoco cuando la función es par, impar o periódica.

2.2. Condiciones del método

2.2.1. Condición 1

El método acelera la sucesión x_n sólo si

$$\lim_{n\to\infty} \frac{\widehat{x_n} - l}{x_n - l} = 0$$

Figura 49: Formula de limite

Debe mantenerse un caso de convergencia lineal para poder realizar la aceleración de la misma.

2.3. Diagrama de flujo

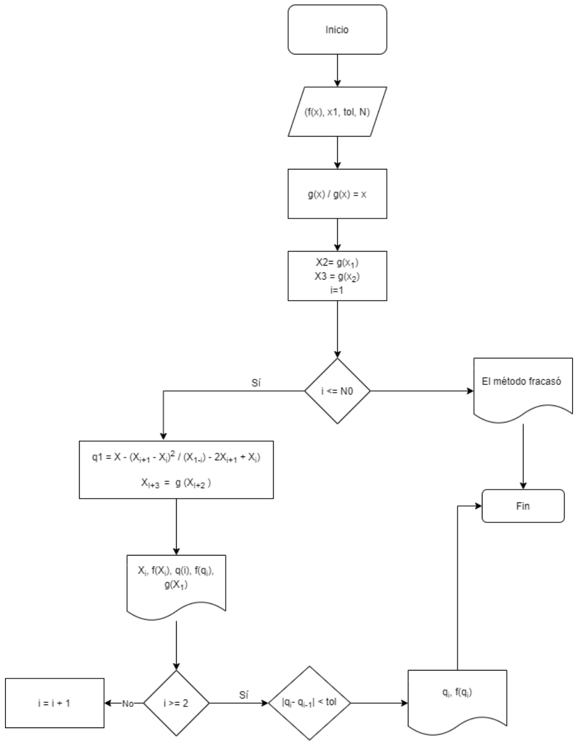
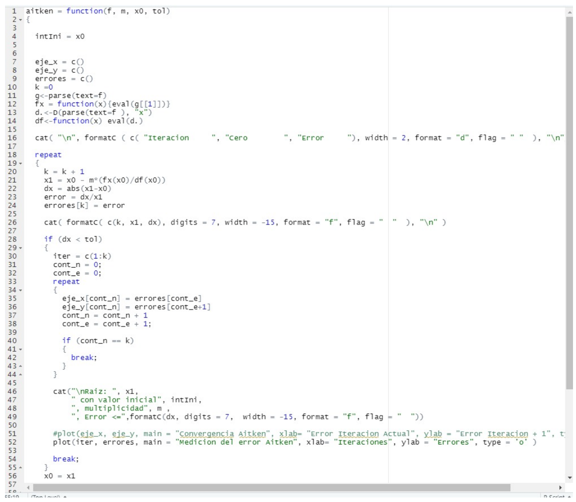


Figura 50: Diagrama de Flujo Aitken



```r
aitken = function(f, m, x0, tol)
{

  intIni = x0


  eje_x = c()
  eje_y = c()
  errores = c()
  k =0
  g<-parse(text=f)
  fx = function(x){eval(g[[1]])}
  d.<-D(parse(text=f ), "x")
  df<-function(x) eval(d.)

  cat( "\n", formatC ( c( "Iteracion     ", "Cero         ", "Error      "), width = 2, format = "d", flag = " "  ), "\n"

  repeat
  {
    k = k + 1
    x1 = x0 - m*(fx(x0)/df(x0))
    dx = abs(x1-x0)
    error = dx/x1
    errores[k] = error

    cat( formatC( c(k, x1, dx), digits = 7, width = -15, format = "f", flag = "   "  ), "\n" )

    if (dx < tol)
    {
      iter = c(1:k)
      cont_n = 0;
      cont_e = 0;
      repeat
      {
        eje_x[cont_n] = errores[cont_e]
        eje_y[cont_n] = errores[cont_e+1]
        cont_n = cont_n + 1
        cont_e = cont_e + 1;

        if (cont_n == k)
        {
          break;
        }
      }

      cat("\nRaiz: ", x1,
          " con valor inicial", intIni,
          ", multiplicidad", m ,
          ", Error <=",formatC(dx, digits = 7,  width = -15, format = "f", flag = "  "))

      #plot(eje_x, eje_y, main = "Convergencia Aitken", xlab= "Error Iteracion Actual", ylab = "Error Iteracion + 1", t
      plot(iter, errores, main = "Medicion del error Aitken", xlab= "Iteraciones", ylab = "Errores", type = 'o' )

      break;
    }
    x0 = x1
```

Figura 51: Implementación Método Aitken

```
  }
}



limInf = 0
limSup = 2
limMitad = 1
tolerancia = 1e-32
#f = function(x) ((exp(1)^x) -(pi*x))

aitken("x*x*x-2*x-5", 1, 2, tolerancia)

```

Figura 52: Implementación Método Aitken Parte 2

2.4. Ejercicios

2.4.1. Ejercicio 1

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

Figura 53: Función 1

```
 Iteracion        Cero          Error
 1.0000000       0.7566170      0.2433830
 2.0000000       0.7392160      0.0174011
 3.0000000       0.7390851      0.0001308
 4.0000000       0.7390851      0.0000000

Raiz:  0.7390851  con valor inicial 1 , multiplicidad 1 , Error <=  0.0000000
```

Figura 54: Resultado Tolerancia 10^{-8}

Iteracion	Cero	Error
1.0000000	0.7566170	0.2433830
2.0000000	0.7392160	0.0174011
3.0000000	0.7390851	0.0001308
4.0000000	0.7390851	0.0000000
5.0000000	0.7390851	0.0000000

Figura 55: Resultado Tolerancia 10^{-16}

Iteracion	Cero	Error
1.0000000	0.7566170	0.2433830
2.0000000	0.7392160	0.0174011
3.0000000	0.7390851	0.0001308
4.0000000	0.7390851	0.0000000
5.0000000	0.7390851	0.0000000

Figura 56: Resultado Tolerancia 10^{-32}

Iteracion	Cero	Error
1.0000000	0.7566170	0.2433830
2.0000000	0.7392160	0.0174011
3.0000000	0.7390851	0.0001308
4.0000000	0.7390851	0.0000000
5.0000000	0.7390851	0.0000000

Figura 57: Resultado Tolerancia 10^{-56}

2.4.2. Ejercicio 2

$$f(x) = xsin(x) - 1 \ \ [-1, 2]$$

Figura 58: Función 2

Iteracion	Cero	Error
1.0000000	1.0503424	0.4496576
2.0000000	1.1141813	0.0638388
3.0000000	1.1141571	0.0000241
4.0000000	1.1141571	0.0000000

Figura 59: Resultado Tolerancia 10^{-8}

2.4.3. Ejercicio 3

$$f(x) = x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$$

Figura 60: Función 3

Iteracion	Cero	Error
1.0000000	0.8888889	0.1111111
2.0000000	0.8148148	0.0740741
3.0000000	0.7654321	0.0493827
4.0000000	0.7325103	0.0329218
5.0000000	0.7105624	0.0219479
6.0000000	0.6959305	0.0146319
7.0000000	0.6861759	0.0097546
8.0000000	0.6796728	0.0065031
9.0000000	0.6753374	0.0043354
10.0000000	0.6724472	0.0028903
11.0000000	0.6705203	0.0019268
12.0000000	0.6692358	0.0012846
13.0000000	0.6683794	0.0008564
14.0000000	0.6678085	0.0005709
15.0000000	0.6674279	0.0003806
16.0000000	0.6671741	0.0002537
17.0000000	0.6670050	0.0001692
18.0000000	0.6668922	0.0001128
19.0000000	0.6668170	0.0000752
20.0000000	0.6667669	0.0000501
21.0000000	0.6667335	0.0000334
22.0000000	0.6667112	0.0000223
23.0000000	0.6666964	0.0000149
24.0000000	0.6666865	0.0000099
25.0000000	0.6666799	0.0000066
26.0000000	0.6666756	0.0000042
27.0000000	0.6666729	0.0000028
28.0000000	0.6666709	0.0000019
29.0000000	0.6666709	0.0000000

Figura 61: Resultado Tolerancia 10^{-8}

Iteracion	Cero	Error
1.0000000	0.8888889	0.1111111
2.0000000	0.8148148	0.0740741
3.0000000	0.7654321	0.0493827
4.0000000	0.7325103	0.0329218
5.0000000	0.7105624	0.0219479
6.0000000	0.6959305	0.0146319
7.0000000	0.6861759	0.0097546
8.0000000	0.6796728	0.0065031
9.0000000	0.6753374	0.0043354
10.0000000	0.6724472	0.0028903
11.0000000	0.6705203	0.0019268
12.0000000	0.6692358	0.0012846
13.0000000	0.6683794	0.0008564
14.0000000	0.6678085	0.0005709
15.0000000	0.6674279	0.0003806
16.0000000	0.6671741	0.0002537
17.0000000	0.6670050	0.0001692
18.0000000	0.6668922	0.0001128
19.0000000	0.6668170	0.0000752
20.0000000	0.6667669	0.0000501
21.0000000	0.6667335	0.0000334
22.0000000	0.6667112	0.0000223
23.0000000	0.6666964	0.0000149
24.0000000	0.6666865	0.0000099
25.0000000	0.6666799	0.0000066
26.0000000	0.6666756	0.0000042
27.0000000	0.6666729	0.0000028
28.0000000	0.6666709	0.0000019
29.0000000	0.6666709	0.0000000

Figura 62: Resultado Tolerancia 10^{-16}

Iteracion	Cero	Error
1.0000000	0.8888889	0.1111111
2.0000000	0.8148148	0.0740741
3.0000000	0.7654321	0.0493827
4.0000000	0.7325103	0.0329218
5.0000000	0.7105624	0.0219479
6.0000000	0.6959305	0.0146319
7.0000000	0.6861759	0.0097546
8.0000000	0.6796728	0.0065031
9.0000000	0.6753374	0.0043354
10.0000000	0.6724472	0.0028903
11.0000000	0.6705203	0.0019268
12.0000000	0.6692358	0.0012846
13.0000000	0.6683794	0.0008564
14.0000000	0.6678085	0.0005709
15.0000000	0.6674279	0.0003806
16.0000000	0.6671741	0.0002537
17.0000000	0.6670050	0.0001692
18.0000000	0.6668922	0.0001128
19.0000000	0.6668170	0.0000752
20.0000000	0.6667669	0.0000501
21.0000000	0.6667335	0.0000334
22.0000000	0.6667112	0.0000223
23.0000000	0.6666964	0.0000149
24.0000000	0.6666865	0.0000099
25.0000000	0.6666799	0.0000066
26.0000000	0.6666756	0.0000042
27.0000000	0.6666729	0.0000028
28.0000000	0.6666709	0.0000019
29.0000000	0.6666709	0.0000000

Figura 63: Resultado Tolerancia 10^{-32}

Iteracion	Cero	Error
1.0000000	0.7566170	0.2433830
2.0000000	0.7392160	0.0174011
3.0000000	0.7390851	0.0001308
4.0000000	0.7390851	0.0000000
5.0000000	0.7390851	0.0000000

Figura 64: Resultado Tolerancia 10^{-56}

2.4.4. Ejercicio 4

$$\frac{9.80665(68.1)}{w}\left(1 - e^{-\left(\frac{w}{68.1}\right)10}\right) - 40$$

Figura 65: Función 4

Iteracion	Cero	Error
1.0000000	14.9334946	0.9334946
2.0000000	14.9696338	0.0361392
3.0000000	14.9696849	0.0000511
4.0000000	14.9696849	0.0000000

Figura 66: Resultado Tolerancia 10^{-8}

Iteracion	Cero	Error
1.0000000	14.9334946	0.9334946
2.0000000	14.9696338	0.0361392
3.0000000	14.9696849	0.0000511
4.0000000	14.9696849	0.0000000
5.0000000	14.9696849	0.0000000

Figura 67: Resultado Tolerancia 10^{-16}

Iteracion	Cero	Error
1.0000000	14.9334946	0.9334946
2.0000000	14.9696338	0.0361392
3.0000000	14.9696849	0.0000511
4.0000000	14.9696849	0.0000000
5.0000000	14.9696849	0.0000000

Figura 68: Resultado Tolerancia 10^{-32}

```
Iteracion 1
1 'mpfr' number of precision  180   bits
[1] 0.0999999999999999999999999999999999999999999999999999998956

Iteracion 2
1 'mpfr' number of precision  180   bits
[1] 0.0054318788958147818343722172751558325912733748886910048773

Iteracion 3
1 'mpfr' number of precision  180   bits
[1] 1.663940598591528180407918057830192806337955612429109 1393e-5

Iteracion 4
1 'mpfr' number of precision  180   bits
[1] 1.5587271138775885639055187001120200549664129412336220293e-10

Iteracion 5
1 'mpfr' number of precision  180   bits
[1] 1.3678306574411031612049803281637822090281812636221633648e-20

Iteracion 6
1 'mpfr' number of precision  180   bits
[1] 1.0533114866640520985246917317923258586580631809606026012e-40

Iteracion 7
1 'mpfr' number of precision  180   bits
[1] 0
```

Figura 69: Resultado Tolerancia 10^{-56}

2.4.5. Ejercicio 5

$$x^3 - 2x - 5 = 0$$

Figura 70: Función 5

Iteracion	Cero	Error
1.0000000	2.1000000	0.1000000
2.0000000	2.0945681	0.0054319
3.0000000	2.0945515	0.0000166
4.0000000	2.0945515	0.0000000

Figura 71: Resultado Tolerancia 10^{-8}

Iteracion	Cero	Error
1.0000000	2.1000000	0.1000000
2.0000000	2.0945681	0.0054319
3.0000000	2.0945515	0.0000166
4.0000000	2.0945515	0.0000000
5.0000000	2.0945515	0.0000000

Figura 72: Resultado Tolerancia 10^{-16}

Iteracion	Cero	Error
1.0000000	2.1000000	0.1000000
2.0000000	2.0945681	0.0054319
3.0000000	2.0945515	0.0000166
4.0000000	2.0945515	0.0000000
5.0000000	2.0945515	0.0000000

Figura 73: Resultado Tolerancia 10^{-32}

Iteracion	Cero	Error
1.0000000	2.1000000	0.1000000
2.0000000	2.0945681	0.0054319
3.0000000	2.0945515	0.0000166
4.0000000	2.0945515	0.0000000
5.0000000	2.0945515	0.0000000

Figura 74: Resultado Tolerancia 10^{-56}

2.5. Análisis de resultado

Los valores suelen converger luego de pocas iteraciones, pero pasado un punto de iteraciones empieza a diverger con un empeoramiento de la estimación. Esto nos indica que pasado cierto punto no se gana precisión al utilizar más nodos de interpelación

Luego de que empieza a bajar la precisión del método, seguirá perdiendo significancia sin solución alguna.

2.6. $\triangle^2$ de Aitken y Raíces Múltiples

Por definición y condiciones de uso del método, agregar una tercera raíz va a hacer que el método se vuelva invalido.

2.7. $\triangle^2$ de Aitken y funciones especiales

En cuanto a la pariedad de la función es irrelevante para encontrar la raíz. En cambio, si es periodica, hace quen aumente el número de iteraciones, lo cual hace que pierda precisión y perdida de significancia.

2.8. $\triangle^2$ de Aitken y Bisección

Este método suele ser mejor que bisección en la mayoria de casos ya que necesita muy pocas iteraciones para llegar a un resultado final. Pero cuando la función se va a muchas más iteraciones (Como la periodica) este método se queda por detras del método de bisección.