

# Informe Tarea 5

Ronald Cardona      Anderson Grajales  
Sebastian Valencia      Julian Sanchez

10 de septiembre de 2018

## 1. Solución numérica de ecuaciones de una variable

En el presente documento se describen algunos de los algoritmos numéricos principales para resolver ecuaciones de una variable. Estos algoritmos nos permiten ya sea encontrar intervalos en los que existe una raíz de la ecuación  $f(x) = 0$  ó encontrar propiamente la raíz.

En algunos casos es necesario optimizar algunos métodos numéricos con la finalidad de alcanzar una tolerancia  $\xi$  en el menor numero de iteraciones posible. Es por esto, que acá se presentan algunos algoritmos para obtener mejores resultados en el menor tiempo posible.

## 1.1. Método de Müller

```

/* Tomado de:
/* http://kilyos.ee.bilkent.edu.tr/ microwa-
ve/programs/utilities/numeric1/infoMuller.htm
*/
Leer  $x_0, x_1, x_2, niter, tol$ 
 $h_1 = x_1 - x_0$ 
 $h_2 = x_2 - x_1$ 
 $\psi_1 = \frac{f(x_1) - f(x_0)}{h_1}$ 
 $\psi_2 = \frac{f(x_2) - f(x_1)}{h_2}$ 
 $d = \frac{\psi_2 - \psi_1}{h_2 + h_1}$ 
 $contador = 1$ 
 $error = tol + 1$ 
 $p = -\infty$ 
mientras  $contador < niter$  y  $error > tol$  hacer
     $b = \psi_2 + h_2 * d$ 
     $D = \sqrt{b^2 - 4 * f(x_2) * d}$ 
     $E = -\infty$ 
    si  $|b - D| < |b + d|$  entonces
         $E = b + D$ 
    en otro caso
         $E = b - D$ 
    fin
     $h = -2 * \frac{f(x_2)}{E}$ 
     $p = x_2 + h$ 
     $x_0 = x_1$ 
     $x_1 = x_2$ 
     $x_2 = p$ 
     $h_1 = x_1 - x_0$ 
     $h_2 = x_2 - x_1$ 
     $\psi_1 = \frac{f(x_1) - f(x_0)}{h_1}$ 
     $\psi_2 = \frac{f(x_2) - f(x_1)}{h_2}$ 
     $d = \frac{\psi_2 - \psi_1}{h_2 + h_1}$ 
     $error = |h|$ 
     $contador = contador + 1$ 
fin
si  $error \leq tol$  entonces
    | Hay una raíz en  $p$ 
en otro caso
    | El método fracasó después de  $niter$  iteraciones
fin

```

**Algoritmo 1:** Método de Müller

## 1.2. Método de Steffensen

```
// Tomado de:
    https://en.wikipedia.org/wiki/Steffensen%27s_method
Leer  $x_0$ ,  $niter$ ,  $tol$ 
 $error = 1 + tol$ 
 $contador = 0$ 
mientras  $contador < niter$  y  $error > tol$  hacer
    |  $x_1 = f(x_0)$ 
    |  $x_2 = f(x_1)$ 
    |  $f(x) = f(x_0)$ 
    |  $p = x_0 - \frac{(x_1 - x_2)^2}{x_2 - 2 * x_1 + x_0}$ 
    |  $error = |x - x_0|$ 
    |  $x_0 = p$ 
    |  $contador = contador + 1$ 
fin
si  $error \leq tol$  entonces
    | Hay una raíz en  $p$ 
en otro caso
    | El método fracasó después de  $niter$  iteraciones
fin
```

**Algoritmo 2:** Método de Steffensen

## 2. Ejemplos

### 2.1. Método de Müller como extensión del Método de la Secante

Este método fue propuesto por D.E Müller en 1956. Esta técnica puede ser usada por cualquier programa de búsqueda de raíces, pero es particularmente útil para la aproximación de raíces de polinomios. El método de Müller es una extensión del método de la secante, que inicia con dos aproximaciones iniciales que determinan la siguiente aproximación como la intersección del eje  $x$  con la línea a través de  $(x_0, f(x_0))$  y  $(x_1, f(x_1))$ . Pero el método de Müller hace uso de tres aproximaciones iniciales y determina la siguiente aproximación considerando la intersección del eje  $x$  con la parábola  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$  and  $(x_2, f(x_2))$ . El siguiente ejemplo muestra que la convergencia del método de Müller es mucho mas rápida con respecto al

método de la secante, debido a la optimización ya mencionada haciendo uso de  $f(x) = e^x - 5x + 2$  en el intervalo  $[0, 2]$  y con una tolerancia al error de  $0,5 * 10^{-7}$

Tabla resultante de la ejecución del método de la secante con los argumentos mencionados

Iteracion	$x_n$	$f(x_n)$	Error
0	0	3.00000000000000	
1	2	-0.610943901069350	
2	1.66161540150850	-1.04026339858352	0.819923424147482
3	2.48153882565598	1.55195995923909	0.490886836614247
4	1.99065198904173	-0.632954979834611	0.142206573909908
5	2.13285856295164	-0.225337163478609	0.0786138993612666
6	2.21147246231290	0.0717865172704462	0.0189934980273785
7	2.19247896428553	-0.00500415521214353	0.00123773381685233
8	2.19371669810238	-9.90948323020544e-5	2.50054057477911e-5
9	2.19374170350813	1.4153htop4102600865e-7	3.56635152520823e-8

Tabla resultante de la ejecución del método de Müller con los argumentos mencionados

Iteracion	$x_n$	$f(x_n)$	Error
0	-9223372036854775807	4.61168601842739e+19	
1	2.46578862771510	1.44381968047063	0.465788627715100
2	2.15026705630200	-0.164183937636073	0.315521571413104
3	2.19023185069732	-0.0138742632229052	0.0399647943953204
4	2.19338492210219	-0.00141524916212999	0.00315307140487347
5	2.19372270388581	-7.52608127419234e-5	0.000337781783616939
6	2.19374052323720	-4.54261109412252e-6	1.78193513943864e-5
7	2.19374159809482	-2.76820541156754e-7	1.07485762153721e-6
8	2.19374166359098	-1.68854239354194e-8	6.54961557673982e-8
9	2.19374166758608	-1.03002576084185e-9	3.99510298946957e-9

Se puede concluir que para esta función los métodos convergen en igual numero de iteraciones, mas sin embargo el método de Müller obtiene un resultado certero, mientras que el método de la secante solo puede entregar una aproximación

## 2.2. Método de Steffensen vs Método de Newton vs Método de Punto Fijo

El método de Steffensen es una técnica para la búsqueda de raíces similar al método de Newton, este método alcanza convergencia cuadrática, pero con una optimización respecto al método de Newton, ya que no usa la derivada para calcular la siguiente aproximación, así:

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)}$$

$$g(x_n) = \frac{f(x_n + f(x_n))}{f(x_n)} - 1$$

El siguiente ejemplo muestra que la convergencia del método de Steffensen es mucho mas rápida con respecto al metodo de Newton, haciendo uso de  $f(x) = x^3 + 4 * x^2 - 10$ , dada una aproximacion inicial de 1,5 y con una tolerancia al error de  $0,5 * 10^{-8}$

Tabla resultante de la ejecución del método de Newton con los argumentos mencionados

Iteracion	$x_n$	$f(x_n)$	Error
0	1.5	2.375000000000000	
1	1.373333333333333	0.134345481481481	0.126666666666667
2	1.36526201487463	0.000528461179515671	0.00807131845870668
3	1.36523001391615	8.29054833008246e-9	3.20009584799941e-5
4	1.36523001341410	6.96088026123474e-16	5.02049735118248e-10

Tabla resultante de la ejecución del método de Steffensen con los argumentos mencionados

$i$	$x_n$	$x_{n+1}$	$f(x_n)$	Error
0	1,5	1,36526522395726	0,000581455787873405	
1	1,36526522395726	1,36523001341659	$4,10971949237575e - 11$	$3,52105406746794e - 5$
2	1,36523001341659	1,36523001341410	$6,96088026123474e - 16$	$2,48867593199975e - 12$

Se evidencia una diferencia notable en la convergencia de ambos métodos, mientras el método de Newton converge en 5 iteraciones, el método de Steffensen lo hace solo en 3, además, este ultimo se acerca de manera muy rápida a la raíz, e inclusive en dos iteraciones menos, entrega un error mucho menor