UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
Departamento de Ciências de Computação

SME0602 - Cálculo Numérico – 1º semestre 2020
Prof. Elias Salomão Helou Neto

Projeto Prático 1
Zeros de Funções de Uma Variável

Alunos:

Paulo Katsuyuki Muraishi Kamimura 10277040
Guilherme Eiji Ichibara 10310700

Data: 30/05/2020

1. Introdução

Neste trabalho será implementado quatro métodos de aproximação de raízes, cada funcionamento será observado e os resultados serão comparados com três diferentes equações. Cada equação envolve um caso especial que será discutido ao decorrer do relatório. Será também analisado a ordem de convergência de cada método aplicado nas diferentes equações

O desenvolvimento do programa foi feito em linguagem C e além de ser responsável pelo cálculo das raízes também possui a função de converter os dados para o formato CSV para a facilitação da análise e interpretação dos dados.

2. Questões

2.1. Método de Newton

O d-ésimo método de Householder é dado por:

$$x_{k+1} = x_k + d\frac{\left(\frac{1}{f}\right)^{(d-1)}(x_k)}{\left(\frac{1}{f}\right)^{(d)}(x_k)} \qquad (1)$$

Substituindo $d = 1$ em (1) temos:

$$x_{k+1} = x_k + \frac{\left(\frac{1}{f}\right)(x_k)}{\left(\frac{1}{f}\right)^{(1)}(x_k)} \qquad (2)$$

Usando a regra do quociente em $\left(\frac{1}{f}\right)^{(1)}$:

$$\left(\frac{1}{f}\right)^{(1)} = \frac{f1' - 1f'}{f^2}$$

$$= \frac{0-f'}{f^2} \qquad (3)$$

Substituindo (3) em (2):

$$x_{k+1} = x_k + \frac{\left(\frac{1}{f}\right)(x_k)}{1}\frac{1}{\left(\frac{-f'}{f^2}\right)(x_k)}$$

$$= x_k + \frac{\left(\frac{1}{f}\right)(x_k)}{1} \frac{1}{\left(\frac{-f'}{f}\right)(x_k)} \frac{1}{\left(\frac{1}{f}\right)(x_k)}$$

Cortando o elemento em comum no numerador e denumerador obtemos:

$$= x_k + \frac{1}{\left(\frac{-f'}{f}\right)(x_k)}$$

$$= x_k + \frac{f(x_k)}{-f'(x_k)}$$

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

2.2. Método de Halley

Substituindo $d = 2$ em (1) temos:

$$x_{k+1} = x_k + 2\frac{\left(\frac{1}{f}\right)^{(1)}(x_k)}{\left(\frac{1}{f}\right)^{(2)}(x_k)} \qquad (4)$$

Sabemos $\left(\frac{1}{f}\right)^{(1)}$ dado pela equação (3), para calcular $\left(\frac{1}{f}\right)^{(2)}$ derivamos novamente (3) também utilizando regra do quociente:

$$\left(\frac{1}{f}\right)^{(2)} = \left(\frac{-f'}{f^2}\right)^{(1)} = \frac{-f^2 f'' + 2f(f')^2}{f^4} \qquad (5)$$

Obs: No passo acima foi necessário realizar a regra da cadeia na hora de realizar $(f^2)'$, resultando em $2ff'$.

Substituindo (3) e (5) em (4):

$$x_{k+1} = x_k + 2\frac{\left(\frac{-f'}{f^2}\right)(x_k)}{\left(\frac{2f(f')^2 - f^2 f''}{f^4}\right)(x_k)}$$

$$= x_k - 2\frac{f'(x_k)}{f^2(x_k)} \frac{f^4(x_k)}{(2f(f')^2 - f^2 f'')(x_k)}$$

Cortando os termos $f(x_k)$ e 2 do numerador e enumerador:

$$= x_k - \frac{f'(x_k)}{1} \frac{f(x_k)}{((f')^2 - \frac{1}{2} f f'')(x_k)}$$

Distribuindo os termos (x_k) para deixar com uma aparência da equação do enunciado do trabalho:

$$= x_k - \frac{f'(x_k)}{1} \frac{f(x_k)}{(f')(x_k)^2 - \frac{1}{2} f(x_k) f''(x_k)}$$

$$x_{k+1} = x_k - \frac{f(x_k) f'(x_k)}{f'(x_k)^2 - \frac{1}{2} f(x_k) f''(x_k)}$$

2.3. Implementações

A implementação se baseia nos arquivos *main.c, equations.c e methods.c.*

- **main.c** – função principal que invoca a execução de todos os métodos e também responsável por salvar os resultados num .csv para facilitação da transposição dos dados para uma tabela (por exemplo ***Google Sheets ou LibreOffice***). Vale ressaltar que tal algoritmo não é necessário para a realização dos cálculos das raízes, só foi feito para facilitar a leitura dos dados.

A função implementada dentro da main.c responsável pela escrita dos dados no formato CSV chama concatena_linha e basicamente separa os dados com um “;”. Também em sua implementação existe um controle para que ele pare de printar iterações caso todos os resultados extraídos de um determinado método já foram printados.

- **equations.c** – contém a definição das equações 1, 2 e 3 apresentado no enunciado. Cada função contém dois parâmetros, a variável x e a ordem de sua derivada, variando assim qual equação ele retornará.

O controle da escolha da ordem da derivada é realizado por meio de um switch/case.

Parâmetros	Entrada
Variável da equação	x
Ordem da derivada	0 = Função original
	1 = Primeira derivada
	2 = Segunda derivada

$$f(x) = x - \cos(x)$$
$$f'(x) = 1 + \sin(x)$$
$$f''(x) = \cos(x)$$

```
double f(double x, int i){
    switch (i){
        case 0:
          return (x - cos(x));

        case 1:
          return (1 + sin(x));

        case 2:
          return (cos(x));

        default:
          return -1;
    }
}
```

$$g(x) = x^3 - 9x^2 + 27x - 27;$$
$$g'(x) = 3x^2 - 18x + 27;$$
$$g''(x) = 6x - 18;$$

```
double g(double x, int i){
    switch (i){
        case 0:
          return (x*x*x - 9*x*x + 27*x -27);

        case 1:
          return (3*x*x-18*x+27);

        case 2:
          return (6*x-18);

        default:
          return -1;
    }
}
```

$$h(x) = e^x - \cos(x);$$
$$h'(x) = e^x - \cos(x);$$
$$h''(x) = e^x - \cos(x);$$

```
double h(double x, int i){
    switch (i){
        case 0:
          return (exp(x) - cos(x));

        case 1:
          return (exp(x) + sin(x));

        case 2:
          return (exp(x) + cos(x));

        default:
          return -1;

}
```

- **methods.c** – contém a implementação dos métodos para aproximação de raízes: bissecção, secante, Newton e Halley. Cada método recebe como parâmetro os valores iniciais e dependendo de qual método pode receber dois valores iniciais -bissecção e secante.

Também recebe como um parâmetro um vetor para armazenar todos os resultados a fim de possibilitar a utilização desses dados posteriormente. Por fim, a função de cada implementação dos métodos retorna à quantidade de iterações que foram necessárias para cumprir os critérios de tolerância absoluta determinado no enunciado da questão. A fim de cumprir tal critério foi determinado uma condição de parada implementada na condição de uma rotina do...while.

2.4. Uso das Implementações

Para compilar o programa basta abrir o terminal na pasta do projeto e executar o seguinte comando para compilar os arquivos.c.

```
gcc main.c equations.c methods.c -o main
```

Para rodar o programa execute o seguinte comando no terminal aberto na pasta do projeto.

```
./main
```

Após a execução do programa será criado três arquivos .csv na pasta do projeto, sendo eles resultados1.csv, resultados2.csv e resultados3.csv. Cada um contendo os resultados dos quatro métodos. Resultado1 são os dados da primeira equação, resultado2 são os dados da segunda equação assim por diante.

A função responsável por converter os resultados já calculados e armazenados em um vetor para um arquivo .csv é opcional e foi somente uma forma de facilitar a visualização dos dados e não faz parte da implementação dos cálculos e dos métodos em si.

Foi discutido se isso influenciaria no desempenho do método numérico e chegou-se à conclusão que o registro dos dados não significa que o método realizou cálculos desnecessários para a obtenção das raízes.

Caso for desejado abrir os arquivos em qualquer plataforma algumas observações são importantes na hora da visualização dos dados. É importante certificar que o separador decimal esteja definido como “.” (ponto). Outro aspecto importante também na visualização é a precisão, é necessário selecionar todas as células e configurá-los como formato de número com 16 casas decimais.

Porém, ainda é possível abrir os arquivos em qualquer editor de texto, sendo possível visualizar os resultados sem qualquer formatação ou configuração.

2.4.1. Equação 1

i	Bissection	Secant	Newton	Halley
	a=0; b=2	x0=0; x1=2	x0=1	x0=1
0	1,0000000000000000	0,0000000000000000	1,0000000000000000	1,0000000000000000
1	0,5000000000000000	2,0000000000000000	0,7503638678402430	0,7408739950803430
2	0,7500000000000000	0,5854549279332180	0,7391128909113610	0,7390851338775810
3	0,6250000000000000	0,7171348682551960	0,7390851333852840	0,7390851332151600
4	0,6875000000000000	0,7399007654901230	0,7390851332151600	0,7390851332151600
5	0,7187500000000000	0,7390811360542050	0,7390851332151600	
6	0,7343750000000000	0,7390851324955940		
7	0,7421875000000000	0,7390851332151610		
8	0,7382812500000000	0,7390851332151600		
9	0,7402343750000000			
10	0,7392578125000000			
11	0,7387695312500000			
12	0,7390136718750000			
13	0,7391357421875000			
14	0,7390747070312500			
15	0,7391052246093750			
16	0,7390899658203120			
17	0,7390823364257810			
18	0,7390861511230460			
19	0,7390842437744140			
20	0,7390851974487300			
21	0,7390847206115720			
22	0,7390849590301510			
23	0,7390850782394400			
24	0,7390851378440850			
25	0,7390851080417630			
26	0,7390851229429240			
27	0,7390851303935050			
28	0,7390851341187950			
29	0,7390851322561500			
30	0,7390851331874720			
31	0,7390851336531340			
32	0,7390851334203030			
33	0,7390851333038880			
34	0,7390851332456800			
35	0,7390851332165760			
36	0,7390851332020240			
37	0,7390851332093000			
38	0,7390851332129380			
39	0,7390851332147570			

40	0,7390851332156670			
41	0,7390851332152120			
42	0,7390851332149850			
43	0,7390851332150980			
44	0,7390851332151550			
45	0,7390851332151840			
46	0,7390851332151690			
47	0,7390851332151620			
48	0,7390851332151590			
49	0,7390851332151600			
50	0,7390851332151600			

Para escolha dos valores iniciais, foi escolhido um valor que fosse próximo o suficiente da raiz, no caso da bisseção era importante que a raiz estivesse dentro do intervalo [a, b]. Não há muito o que se comentar sobre essa equação, o único detalhe e que todos os métodos acabaram convergindo para a mesma raiz, o que não ocorreu nas outras equações.

2.4.2. Equação 2

i	Bissection	Secant	Newton	Halley
	a=1; b=6	x0=1; x1=5	x0 = 2	x0 = 2
0	3,5000000000000000	1,0000000000000000	2,0000000000000000	2,0000000000000000
1	2,2500000000000000	5,0000000000000000	2,3333333333333300	2,5000000000000000
2	2,8750000000000000	3,0000000000000000	2,5555555555555400	2,7500000000000000
3	3,1875000000000000	3,0000000000000000	2,7037037037037000	2,8750000000000000
4	3,0312500000000000		2,8024691358024200	2,9375000000000000
5	2,9531250000000000		2,8683127572015900	2,9687500000000000
6	2,9921875000000000		2,9122085048007800	2,9843750000000000
7	3,0117187500000000		2,9414723365340000	2,9921875000000000
8	3,0019531250000000		2,9609815576891500	2,9960937500000000
9	2,9970703125000000		2,9739877051238300	2,9980468750000000
10	2,9995117187500000		2,9826584700833100	2,9990234375000000
11	3,0007324218750000		2,9884389800626700	2,9995117187500000
12	3,0001220703125000		2,9922926533883500	2,9997558593750000
13	2,9998168945312500		2,9948617689313700	2,9998779296875000
14	2,9999694824218700		2,9965745124777700	2,9999389648437500
15	3,0000457763671800		2,9977163416524500	2,9999694824218700
16	3,0000076293945300		2,9984775617417700	2,9999847412109300
17	3,0000267028808500		2,9989850408280400	2,9999847412109300
18	3,0000362396240200		2,9993233596561500	
19	3,0000410079956000		2,9995489080095900	
20	3,0000433921813900		2,9996992685912700	
21	3,0000445842742900		2,9997995183422600	
22	3,0000451803207300		2,9998663423813000	
23	3,0000454783439600		2,9999108896202700	
24	3,0000456273555700		2,9999401202681300	
25	3,0000457018613800		2,9999599369244200	
26	3,0000457391142800		2,9999746933148600	
27	3,0000457577407300		2,9999894864331000	
28	3,0000457670539600		2,9999466323512500	
29	3,0000457717105700		2,9999640958743200	
30	3,0000457740388800		2,9999751196829900	
31	3,0000457752030300		2,9999827718755300	
32	3,0000457757851100		2,9999987315932100	
33	3,0000457760761400		2,9970553909015200	
34	3,0000457762216600		2,9980369272834500	
35	3,0000457762944200		2,9986912844494500	
36	3,0000457763308000		2,9991275244408700	
37	3,0000457763489900		2,9994183482817500	
38	3,0000457763580900		2,9996122333164700	
39	3,0000457763626400		2,9997414919257300	

40	3,0000457763649100		2,9998276164802600	
41	3,0000457763660500		2,9998850030092600	
42	3,0000457763666100		2,9999231513463000	
43	3,0000457763669000		2,9999488184297900	
44	3,0000457763670400		2,9999650931690800	
45	3,0000457763671100		2,9999767558607200	
46	3,0000457763671500		2,9999767558607200	
47	3,0000457763671600			
48	3,0000457763671700			
49	3,0000457763671800			
50	3,0000457763671800			
51	3,0000457763671800			

Semelhante a equação anterior também foi escolhido valores próximos a raiz, algo que quis-se salientar é que nesse método, apesar de não ser garantido o critério de convergência, a escolha de x_0 e x_1 no método da secante ou de a e b no método da bisseção é importante, já que dependendo do valor a busca passa a ser trivial, no caso da bisseção se a e b fossem iguais a x_0 e x_1 escolhidos no método da secante na primeira iteração ($i = 0$) o valor encontrado já seria 3, do mesmo modo o valor escolhido mostra que a reta secante intercepta o eixo das abcissas diretamente no valor 3, achando assim a raiz de forma trivial.

Além disso fizemos testes com outros valores e no método da secante nem sempre se convergia para um valor, as vezes retornando NaN ou infinito. Algo que podemos perceber ao analisar a equação 3 em um gráfico é que existe um intervalo grandes de pontos na região próximos da interceptação do eixo x, podendo induzir os métodos a convergirem a valores ligeiramente diferentes

Assim, mesmo seguindo a tolerância desejada, o resultado pode convergir em valores ligeiramente diferentes da raiz real, o que não ocorreu nas outras equações.

2.4.3. Equação 3

i	Bissection	Secant	Newton	Halley
	a=-1; b=2	x0=1; x1=2	x0=1	x0=1
0	0,5000000000000000	1,0000000000000000	1,0000000000000000	1,0000000000000000
1	-0,2500000000000000	2,0000000000000000	0,3881655168657400	0,1501880136433960
2	0,1250000000000000	0,6129566283785490	0,0920303459456630	0,0020076165148970
3	-0,0625000000000000	0,4025794026714920	0,0073463293261812	0,0000000067107334
4	0,0312500000000000	0,1348442424705910	0,0000533159208848	-0,0000000000000001
5	-0,0156250000000000	0,0375409545988820	0,0000000028423348	0,0000000000000001
6	0,0078125000000000	0,0044265015837714	0,0000000000000000	
7	-0,0039062500000000	0,0001605566619041	0,0000000000000000	
8	0,0019531250000000	0,0000007079976337		
9	-0,0009765625000000	0,0000000001136586		
10	0,0004882812500000	0,0000000000000000		
11	-0,0002441406250000	0,0000000000000000		
12	0,0001220703125000			
13	-0,0000610351562500			
14	0,0000305175781250			
15	-0,0000152587890625			
16	0,0000076293945313			
17	-0,0000038146972656			
18	0,0000019073486328			
19	-0,0000009536743164			
20	0,0000004768371582			
21	-0,0000002384185791			
22	0,0000001192092896			
23	-0,0000000596046448			
24	0,0000000298023224			
25	-0,0000000149011612			
26	0,0000000074505806			
27	-0,0000000037252903			
28	0,0000000018626451			
29	-0,0000000009313226			
30	0,0000000004656613			
31	-0,0000000002328306			
32	0,0000000001164153			
33	-0,0000000000582077			
34	0,0000000000291038			
35	-0,0000000000145519			
36	0,0000000000072760			
37	-0,0000000000036380			
38	0,0000000000018190			
39	-0,0000000000009095			

40	0,0000000000004547			
41	-0,0000000000002274			
42	0,0000000000001137			
43	-0,0000000000000568			
44	0,0000000000000284			
45	-0,0000000000000142			
46	0,0000000000000071			
47	-0,0000000000000036			
48	0,0000000000000018			
49	-0,0000000000000009			
50	0,0000000000000004			
51	-0,0000000000000002			

A equação 3 teve um bom comportamento para o cálculo das raízes, a única coisa que teve de ser levada em conta foi a escolha dos valores iniciais para que o algoritmo pudesse convergir para a raiz correta.

2.5. Estimativa da ordem de convergência

Será utilizado a seguinte fórmula para estimar a ordem de convergência

$$\frac{log\frac{\epsilon_{k+1}}{\epsilon_k}}{log\frac{\epsilon_k}{\epsilon_{k-1}}} \approx p \quad (6)$$

Sendo $\epsilon_k = |x_k - x^*|$ e $x^* = raiz\ da\ função$.

Será utilizado uma planilha .ods (OpenDocument Spreadsheet) para o cálculo da ordem de convergência de todos os métodos de todas as equações (aproveitando os arquivos anteriormente criados). O documento pode ser aberto em qualquer software, no nosso caso foi utilizado a ferramenta online Google Sheets.

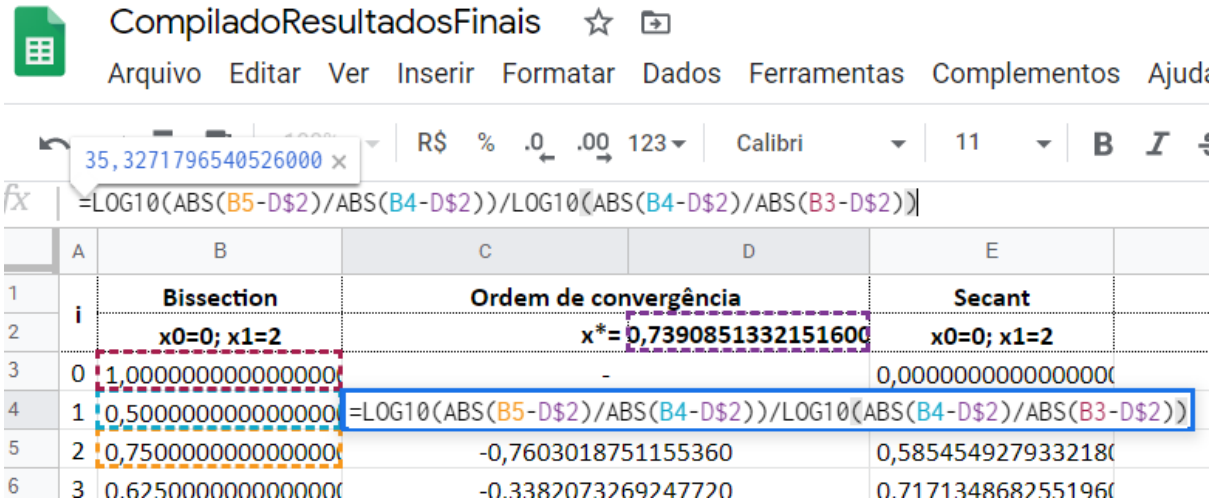

Figura 1 Exemplo de implementação do cálculo da Ordem de Convergência na equação 1

```
=LOG10(ABS(B5-D$2)/ABS(B4-D$2))/LOG10(ABS(B4-D$2)/ABS(B3-D$2))
```

A fórmula acima foi desenvolvida e copiada para o restante da coluna, variando as referências de ϵ_k e fixando o valor da raiz x^* em “D$2”.

Na planilha anexada no projeto estão os resultados e os dados de convergência, separado em três abas, uma aba para cada equação. Vale ressaltar que tanto o para o primeiro termo e o último não foi realizado os cálculos por não haver um valor anterior ou seguinte. Foi substituído por um “- “.

2.5.1. Equação 1

Para a primeira equação foi necessário realizar uma aproximação da raiz e dessa forma foi considerado o resultado obtido pelo método de Halley.

$$x^* = 0.7390851332151600$$

i	Bissection	Secant	Newton	Halley
	Ordem de convergência Equação 1			
0	-	-	-	-
1	35.3271796540526000	-3.9407044967344100	1.9123320892035700	2.9721231406170000
2	-0.7603018751155360	0.9243364495108540	1.9980312779063500	#NÚM!
3	-0.3382073269247720	1.6921692522046500	#NÚM!	#DIV/0!
4	1.1728234984235400	1.6152664457897700	#DIV/0!	-
5	1.5712320797384200	1.6212545741323400	-	
6	0.2854787463506430	1.5641967456617200		
7	3.2342562317409200	#NÚM!		
8	-0.2646520020048970	-		
9	-5.3033075759701700			
10	-0.3181591727875350			
11	-2.4630388644038500			
12	0.2322910327677060			
13	4.5787872417449000			
14	-0.4152219884817950			
15	-2.1722078301105200			
16	0.3838243553770440			
17	1.8480482345088300			
18	0.1334805485067590			
19	19.4798866591744000			
20	-0.7077298192016950			
21	-0.4636490450974510			
22	1.3372767117611300			
23	2.1457772989369300			
24	-0.6843472498045120			
25	-0.5292958830935000			
26	1.4415475438617500			
27	0.8812280882299900			
28	-0.0522336528878256			
29	-59.6022272440126000			
30	-0.7789102267271520			
31	-0.2746858857590060			
32	1.1050552876445000			
33	1.2732995701031200			
34	2.8772337169341800			
35	-0.7254499646446900			
36	-0.3623792855887260			
37	1.2013394499902700			

38	1.7604745172383400			
39	-0.1345032622220400			
40	-9.9211957330976600			
41	-0.5329677280194320			
42	-0.8536612199299890			
43	2.4307739501100500			
44	-0.6225933804216380			
45	-0.6252832460836960			
46	1.5334752630571200			
47	0.4608454206183700			
48	#NÚM!			
49	#DIV/0!			
50	-			

Obs. Para iterações próximos da raiz (últimas iterações do algoritmo) é possível ver erros como #NÚM! E #DIV/0.

2.5.2. Equação 2

$$x^* = 3$$

i	Bissection	Secant	Newton	Halley
	Ordem de convergência Equação 2			
0	-	-	-	-
1	-4.4190225827029100	#NÚM!	0.9999999999999380	1.0000000000000000
2	-0.2262943855309170	#DIV/0!	1.0000000000001300	1.0000000000000000
3	-4.4190225827029100	-	0.9999999999993560	1.0000000000000000
4	-0.2262943855309170		1.0000000000001500	1.0000000000000000
5	-4.4190225827029100		0.9999999999925740	1.0000000000000000
6	-0.2262943855309170		1.0000000000140400	1.0000000000000000
7	-4.4190225827029100		0.9999999999822320	1.0000000000000000
8	-0.2262943855309170		0.9999999997963920	1.0000000000000000
9	-4.4190225827029100		1.0000000003227800	1.0000000000000000
10	-0.2262943855309170		1.0000000014134900	1.0000000000000000
11	-4.4190225827029100		1.0000000027145800	1.0000000000000000
12	-0.2262943855309170		0.9999999985498340	1.0000000000000000
13	-4.4190225823081300		0.9999998941532340	1.0000000000000000
14	-0.2262943853697510		1.0000001037129200	0.9999999997690660
15	-4.4190225862688200		1.0000010363497900	0.9999999997480720
16	-0.6991803251659060		0.9999981534856200	0.0000000000000000
17	0.2437665045381570		0.9999975441082000	-
18	0.4047851467671570		1.0000118610484400	
19	0.4571669097733370		0.9999633970631890	
20	0.4795765335751450		1.0001011820285900	
21	0.4900170410254230		0.9998681549653520	
22	0.4950635576823780		0.9999119692621970	
23	0.4975452479459930		0.9805925690777790	
24	0.4987760123783460		1.0109303807925500	
25	0.4993887478674540		1.1430852480754800	
26	0.4996946111679380		1.9121199147505800	
27	0.4998476463495160		-1.8494243688435800	
28	0.4999231686663780		-0.2439785763848240	
29	0.4999628987513570		0.9253762228004520	
30	0.4999786369761810		1.0020683413974800	
31	0.4999950406220260		7.0980722110260800	
32	0.4999776842774290		-2.9707238850226000	
33	0.5000151638689940		-0.0523182838027634	
34	0.4999988078652640		0.9999992147263710	

35	0.5000024557697100		1.0000049337576500	
36	0.4999935984279980		0.9999901266044980	
37	0.5002684058661890		1.0000128836491200	
38	0.5000243265697690		1.0000219015641300	
39	0.4989263723112910		0.9993298482503610	
40	0.5021517817948900		0.9990390841190840	
41	0.4912348947381800		0.9956825037256630	
42	0.5178429828559040		1.0084115621936700	
43	0.4823889634851090		0.9415402252614370	
44	0.4984127230149820		1.0625365286603700	
45	0.5796178550088800		0.0000000000000000	
46	0.2417583054097620		-	
47	1.0454538361051900			
48	0.9565223881386050			
49	0.0000000000000000			
50	#DIV/0!			
51	-			

2.5.3. Equação 3

$$x^* = 0$$

i	Bissection	Secant	Newton	Halley
	Ordem de convergência Equação 3			
0	-	-	-	-
1	1.0000000000000000	-1.7061430996611400	1.5209529924777600	2.2759713325120200
2	1.0000000000000000	0.3554869661130510	1.7563356752591000	2.9221150958202300
3	1.0000000000000000	2.6017298739611900	1.9485291293477500	1.4293092617930700
4	1.0000000000000000	1.1690625607810200	1.9975480360091300	0.0000000000000000
5	1.0000000000000000	1.6718879635985500	#NÚM!	-
6	1.0000000000000000	1.5514465348317800	#DIV/0!	
7	1.0000000000000000	1.6353399839838200	-	
8	1.0000000000000000	1.6108146981451700		
9	1.0000000000000000	#NÚM!		
10	1.0000000000000000	#DIV/0!		
11	1.0000000000000000	-		
12	1.0000000000000000			
13	1.0000000000000000			
14	1.0000000000000000			
15	0.9999999999905450			
16	1.0000000000283600			
17	0.9999999999810900			
18	1.0000000000000000			
19	1.0000000000000000			
20	1.0000000000000000			
21	0.9999999993948900			
22	1.0000000006051100			
23	1.0000000000000000			
24	1.0000000000000000			
25	1.0000000000000000			
26	1.0000000000000000			
27	1.0000000387270500			
28	0.9999998838188540			
29	1.0000000774541100			
30	1.0000003098164300			
31	0.9999996901836700			
32	0.9999987607346960			
33	1.0000037178015800			
34	0.9999975214744710			
35	0.9999900858988600			

36	1.0000099141994300			
37	1.0000000000000000			
38	1.0000000000000000			
39	1.0001586337943100			
40	0.9995241915372850			
41	1.0003173508273000			
42	1.0012694193875700			
43	0.9987321899950340			
44	1.0000000000000000			
45	1.0000000000000000			
46	0.9798221180623700			
47	1.0205934134019500			
48	1.0000000000000000			
49	1.1699250014423100			
50	0.8547556456757270			
51	-			

A equação 3 apesar de ser a única equação que possuía múltiplas raízes foi a que apresentou o melhor resultado para o cálculo da ordem de convergência.

Obs. Para iterações próximos da raiz (últimas iterações do algoritmo) é possível ver erros como #NÚM! E #DIV/0.

- **#NÚM!** – A planilha tenta calcular $\log(0)$, isso deve-se ao fato que o valor convergido é igual ao à raiz x^*. Quando $x_{k+1} = x^*$, temos que $e_{k+1} = 0$, assim a planilha tenta calcular $\frac{log\left(\frac{0}{\dots}\right)}{log(\dots)}$.
- **#DIV/0!** – A planilha tenta realizar uma divisão por 0, isso ocorre quando x_k e x_{k+1} são iguais à raiz x^*. Assim $e_{k+1} = e_k = 0$ ocorrendo uma operação $log\left(\frac{0}{0}\right)$, não sendo possível realizar o cálculo.
- **Última iteração da convergência resultar em 0** – Acontece quando ocorre dois termos calculados iguais em sequência e ambos são diferentes da raiz. Quando $x_k = x_{k+1}$ e ao mesmo tempo que $x_k \neq x^*$, temos que $e_{k+1} = e_k \neq 0$. Portanto, ao calcular $log\left(\frac{e_{k+1}}{e_k}\right)$ temos $log(1) = 0$, e temos portanto algo como $\frac{0}{log(\dots)} = 0$.

3. Conclusão

Em suma, podemos observar resultados indesejados para todos os métodos na equação 2 e especificamente problemas de convergência no método da bissecção da equação 1. Os valores esperados para cada ordem de convergência para os métodos da bisseção, secante, Newton e Halley eram:

Bissecção	Secante	Newton	Halley
1	~1.618	2	3

O valor estimado da ordem de convergência para o **método da bisseção na equação 1** não foi o esperado, provavelmente isso se deve ao fato de não termos a raiz exata o que leva a aproximações nos cálculos e na discrepância encontrada na tabela.

Os **resultados da convergência da equação 2** ocorreram pois um dos critérios de convergência - como pode ser visto nas notas de aula - é que $f'(x)$ seja diferente de 0, entretanto para equação 2 dado por $f(x) = x^3 - 9x^2 + 27x - 27)$, possui $f'(x) = 0$ já que:

$$f'(x^*) = 3(x^*)^2 - 18x^* + 27 = 3(3)^2 - 18 * 3 + 27 = 0$$

Assim, como o critério não é atendido as convergências esperadas não são garantidas, além disso alguns valores de valores iniciais na equação 2 mostravam também que para alguns métodos não era possível achar a raiz, já que o critério de convergência não é atendido

Para o **método da secante na equação 3** foi necessário escolher x_0 e x_1 à direita da maior raiz, já que não há raízes para $x > 0$, assim não haveria risco do algoritmo convergir para alguma outra raiz que não fosse a maior no caso $x = 0$, desde que os valores iniciais fossem próximos o suficiente da raiz esperada

Para os outros métodos para k próximo às iterações finais podemos ver que todos os valores foram próximos do esperado, particularmente na equação tal resultado é o mais evidente entre as três equações.

Por fim, os métodos numéricos conseguiram de fato uma ótima aproximação e até mesmo em alguns casos a raiz exata das funções, porém existe casos específicos em que é necessário tomar alguns cuidado para obter a raiz correta, seja na escolha dos valores iniciais ou mesmo comparar a os resultados com outros métodos, uma vez que o resultado calculado por divergir ligeiramente por causa do comportamento da função.