**Trabalho Computacional 1**

**Entrega:** 29/08/2019

**Parte 1 - Bissecção e Método de Héron**

**Códigos - Anexos I e II**

**1 -** *Use o método da bissecção para encontrar uma aproximação da raiz quadrada de 2. Use uma precisão de 10-6. Explique qual conceito de erro você utilizou. Considere que você não conhece o valor de raiz de 2.*

**R:** Como critérios de parada dos programas, foram utilizadas as relações de diferença entre o valor atual e seu anterior, bem como o valor da função no ponto médio do intervalo. A linguagem implementada foi em Python.

**2 -** *Use o método de Héron para encontrar uma aproximação da raiz quadrada de 2 utilizando a mesma precisão e o mesmo conceito de erro que você utilizou no exercício 1 e estimativa inicial igual a 1.*

**R:** Utilizando o código e o protótipo do método de Héron, temos que o valor estimado 1 fará o **k** da sequência "rodar" 4 vezes, de forma que o código apontará o resultado como 1,4142135623746899, como a aproximação da raiz de 2.

**3 -** *Utilize no exercício anterior estimativa inicial igual a 2. Compare o número de iterações para obtenção de uma aproximação com o mesmo erro e precisão.*

**R:** Utilizando a estimativa inicial igual à 2, temos um **k** determinado igual à 4, da mesma maneira com a que ocorreu com a estimativa inicial igual à 1, pois o valor da raiz em uma incerteza (precisão) igual à 10^-6 trabalha com casas determinadas de raíz de 2. Portanto, a incerteza não é suficiente para uma precisão maior entre um inteiro e outro.

**4 -** *Considerando agora o valor de raiz de dois da linguagem/computador que você está usando, calcule o erro absoluto obtido em cada exercício anterior.*

**R:** Utilizando a linguagem Python, temos a raíz de 2 igual a 1,4142135623730951. Em comparação com o método de Héron, temos que a divergência de valores se inicia na décima primeira (11ª) casa decimal de aproximação da raíz de 2. Em comparação com o método da bissecção, temos que a divergência de valores se inicia na sexta (6ª) casa decimal, como é predito que aconteça com os valores da bissecção com uma aproximação da 6ª casa decimal. Portanto, o erro absoluto com relação ao número obtido pelo método de Héron e o valor dado pelo Python acerca da raíz de 2 é dado pelo valor 0,0000000000015948. O valor do erro absoluto com relação ao número obtido pelo método da bissecção e o valor dado pelo Python acerca da raíz de 2 é dado pelo valor 0,0000018602904611. O valor do erro absoluto com relação ao número obtido pelo método da bissecção e o número obtido pelo método de Héron é dado pelo valor 0,0000001397079441.

**Parte 2 - Método de Newton**

**Código - Anexo III**

**1 -** *Use o método de Newton para encontrar uma aproximação da raiz quadrada de 2. Use uma precisão de 10-6 e x0=1. Explique qual conceito de erro você utilizou. Considere que você não conhece o valor de raiz de 2.*

**R:** Foi adotado ε igual a 10-6 e, como critério de parada do programa, foi utilizada a relação de diferença entre o valor atual e seu anterior.

**2 -** *Utilize no exercício anterior estimativa inicial igual a 2. Compare o número de iterações para obtenção de uma aproximação com o mesmo erro e precisão.*

**R:** Utilizando o código e o protótipo do método de Newton, temos que o valor estimado 2 fará o **k** da sequência "rodar" 4 vezes, de forma que o próprio código apontará o resultado como 1,4142135623730951, como a aproximação da raíz de 2. Enquanto no primeiro valor estimado (valor 1) temos 4 iterações, no segundo valor estimado (valor 2) temos também 4 iterações. É contemplado o valor de 4 iterações, pois a aproximação do valor 1,41… corresponde à parte da metade da eficácia do método de Newton para aproximações.

**3 -** *Compare o desempenho do método de newton com os métodos da bissecção e de Heron.*

**R:** Contemplando os valores anteriores aos correspondentes ao método de Newton, obtemos o valor de 1,4142135623746899 para o método de Héron para aproximações e o valor de 1,4142136573791504 para o método da bissecção já dados dos exercícios anteriores. Comparando com o valor 1,4142135623730951 advindo do método de Newton, temos determinados valores de erros absolutos. O erro absoluto advindo da comparação do método da bissecção e do método de Newton é dado pelo valor 0,0000000950060553 enquanto o erro absoluto advindo da comparação do método de Héron e do método de Newton é dado pelo valor 0,0000000000015948. Tendo em vista que o valor obtido utilizando a linguagem de programação Python para a raíz de 2 é dado pelo número 1,142135623730951 e ele corresponde exatamente com o valor advindo do método de Newton, temos que seu erro absoluto é 0,0000000000000000.

**Parte 3 - Método da Secante**

**Código - Anexo IV**

**1 -** *Use o método da secante para encontrar uma aproximação da raiz quadrada de 2. Use uma precisão de 10^(-6) e estimativas iniciais 0.5 e 1. Explique qual conceito de erro você utilizou. Considere que você não conhece o valor de raiz de 2.*

**R:** Foi adotado ε igual a 10-6 e, como critério de parada do programa, foi utilizada a relação do valor da função quando menor que ε.

**2 -** *Utilize no exercício anterior estimativas iniciais igual a 1 e 2. Compare o número de iterações para obtenção de uma aproximação com o mesmo erro e precisão.*

**R:** Utilizando como estimativas iniciais os valores de 0,5 e 1 há 6 iterações, obtendo o valor de 1,4142135622302456. Já para os valores iniciais de 1 e 2 há 5 iterações, obtendo o valor de 1,4142135626888699. Como é possível perceber no último caso, mesmo havendo menor número de iterações, o valor da raíz de dois fica mais próximo do valor obtido pelo próprio Python, que é de 1,4142135623730951.

**3 -** *Compare o desempenho do método da secante com os métodos da bissecção, de Heron e de Newton.*

**R:** Contemplando os valores anteriores aos correspondentes ao método de Newton, obtemos o valor de 1,4142135623746899 para o método de Héron para aproximações, valor de 1,4142136573791504 para o método da bissecção e o valor 1,4142135623730951 do método de Newton já dados dos exercícios anteriores. Comparando com o valor 1,4142135626888699 advindo do método das secantes, temos determinados valores de erros absolutos. O erro absoluto advindo da comparação do método da bissecção e do método das secantes é dado pelo valor 0,0000000946902805, o erro absoluto advindo da comparação do método de Héron e do método das secantes é dado pelo valor 0,0000000003141800 e o erro absoluto advindo da comparação do método de Newton e do método das secantes é dado pelo valor 0,0000000003157748. Tendo em vista que o valor advindo da linguagem de programação Python para a raíz de 2 é dado pelo número 1,4142135623730951 e ele corresponde exatamente com o valor advindo do método de Newton, temos que seu erro absoluto é 0,0000000003157748.

**ANEXOS**

**Anexo I - Código escrito para o Método da Bissecção:**

| import numpy as np  import math  def func(x):  return 2 - (x\*\*2)  a = 0.0  b = 2.0  while abs(func((a+b)/2)) >= (10\*\*(-6)) and abs(a-b) >= (10\*\*(-6)):  if func((a+b)/2) < 0:  b = (a+b)/2  if func((a+b)/2) > 0:  a = (a+b)/2    print((a+b)/2) |
| --- |

**Anexo II - Código escrito para o Método de Héron:**

| k = 0.0  a = 1.0  while abs((-(a\*\*2)+2)) >= 10\*\*(-6):  a = (a+(2/a))/2  k = k + 1  print(a)  print(k)  print(2\*\*(1/2)) |
| --- |

**Anexo III - Código escrito para o Método de Newton:**

| x0 = 1.0  x1 = 0.0  k = 0.0  for i in range(1000):  x1 = x0- (((x0\*\*2)-2)/(2\*x0))    if abs(x1-x0) < 10\*\*(-6):  break  k = k + 1  x0 = x1    print(x1)  print(k) |
| --- |

**Anexo IV - Código escrito para o Método da Secante:**

| def func(x):  return 2 - (x\*\*2)    x2 = 0.0  x1 = 1.0  x0 = 2.0  cont = 0  while abs(func(x2)) >= (10\*\*(-6)):  x2 = ((x0 \* func(x1))-(x1\*func(x0)))/(func(x1)-func(x0))  cont += 1  x0 = x1  x1 = x2  print(x2)  print(cont) |
| --- |