UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 68993 ANDRE HEINECK
HAHN 19FCN107 - 1 apos 09/04, 50% _____ / / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

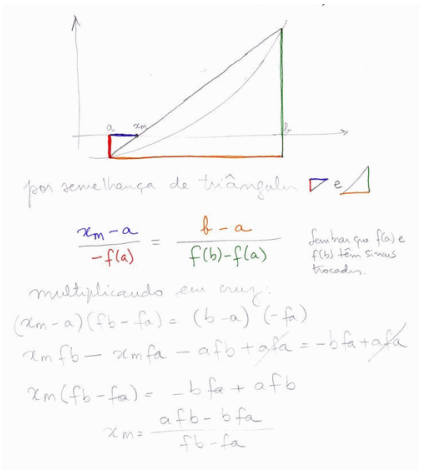
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

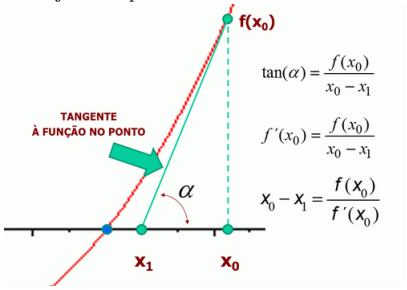
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[20]{7943}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 3 * sin(x) + 5 * cos(x) - 1.785532 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((6 * x)/10) * e^{((7*x)/10)} - 1803.358578 = 0$ no intervalo entre 8 e 9 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 6 * x^5 + 3 * x^2 - 323.737920 = 0$ no intervalo entre 2 e 3 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69789 BRUNO V
GUIMARARES 19FCN107 - 2 apos 09/04, 50% / _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

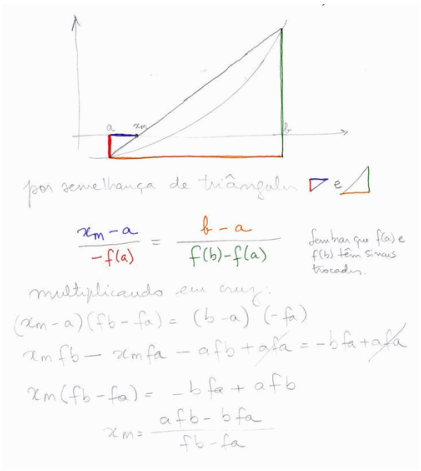
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

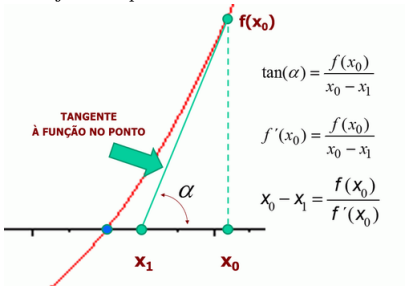
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[7]{8728}$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 8 * sin(x) + 2 * cos(x) - 3.057152 = 0$ no intervalo entre 8 e 9 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((6 * x)/10) * e^{((2*x)/10)} - 1.011605 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 7 * x^5 + 3 * x^2 - 3713.281250 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69008 EDUARDO SCARANTE DEMCZUK 19FCN107 - 3 apos 09/04, 50% _____ / _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

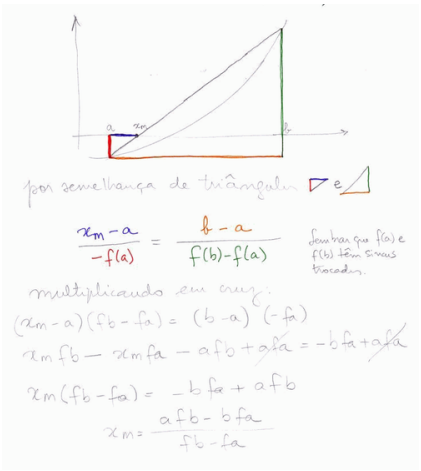
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

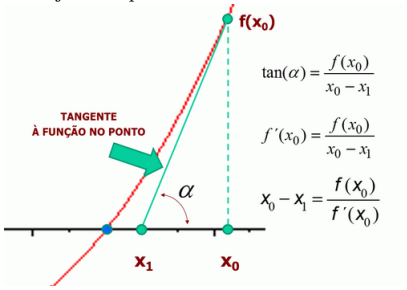
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[25]{7982}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 8 * sin(x) + 7 * cos(x) - 7.884688 = 0$ no intervalo entre 6 e 7 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((3 * x)/10) * e^{((6*x)/10)} - .638481 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 7 * x^6 + 3 * x^2 - 193855.234375 = 0$ no intervalo entre 5 e 6 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69015 FELIPE DE ROSSI REZENDE 19FCN107 - 4 apos 09/04, 50%
/ _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

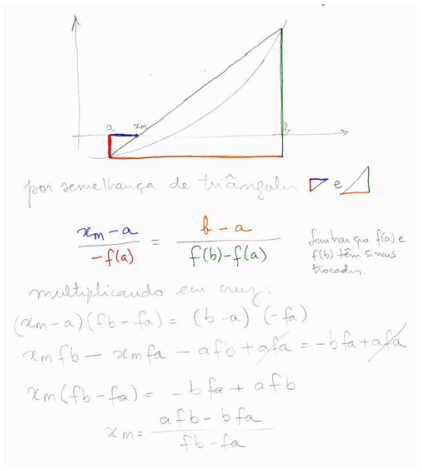
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
   -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

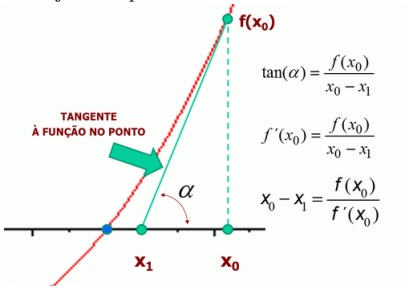
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
   -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[14]{9683}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 6 * sin(x) + 5 * cos(x) - 7.404023 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((4 * x)/10) * e^{((2*x)/10)} - 18.028268 = 0$ no intervalo entre 8 e 9 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 6 * x^5 + 4 * x^3 - 60546.380580 = 0$ no intervalo entre 6 e 7 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69022 GIOVANA STOPANOVSKI BECKER 19FCN107 - 5 apos 09/04, 50% _____ / _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

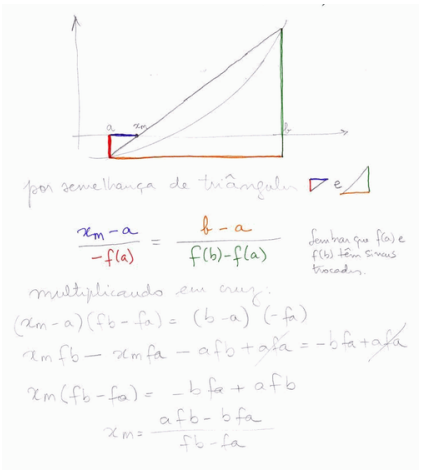
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

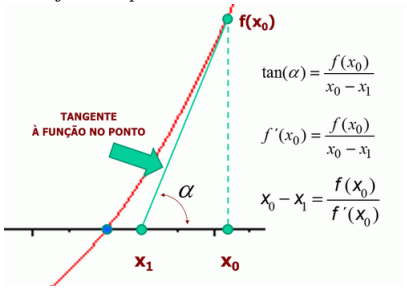
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[26]{7960}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 2*sin(x)+3*cos(x)+.361359 = 0$ no intervalo entre 5 e 6 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((6 * x)/10) * e^{((7*x)/10)} - 6073.614239 = 0$ no intervalo entre 9 e 10 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 6 * x^5 + 3 * x^2 - 4797.430080 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69208 IAN DO AMARAL PIMENTA 19FCN107 - 6 apos 09/04, 50%
/ _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

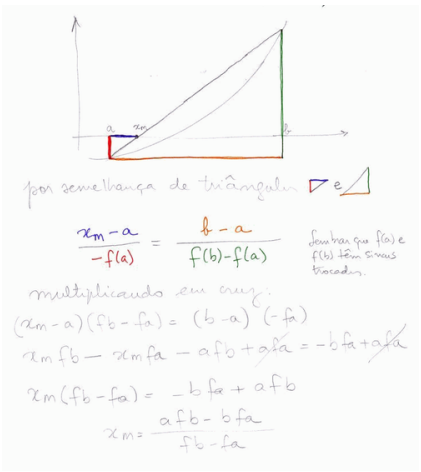
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

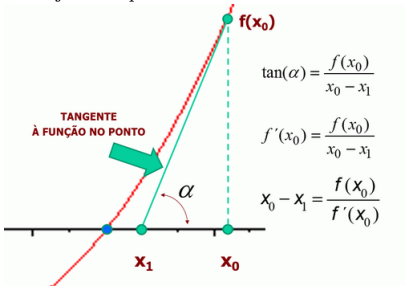
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[18]{6167}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 8 * sin(x) + 6 * cos(x) + 5.799955 = 0$ no intervalo entre 9 e 10 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((2 * x)/10) * e^{((3*x)/10)} - 29.071711 = 0$ no intervalo entre 9 e 10 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 5 * x^4 + 3 * x^2 - .310500 = 0$ no intervalo entre 0 e 1 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69039 KHAREN INGRID VIDAL BELO 19FCN107 - 7 apos 09/04, 50% / _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

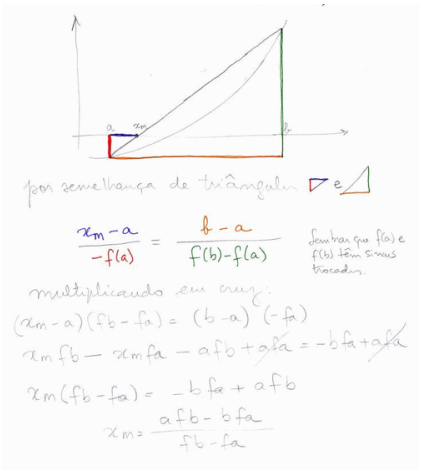
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

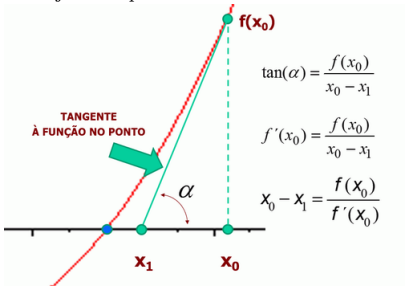
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[23]{9811}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 7*sin(x)+8*cos(x)-10.579157 = 0$ no intervalo entre 7 e 8 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((7 * x)/10) * e^{((2*x)/10)} - 25.142142 = 0$ no intervalo entre 7 e 8 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 7 * x^6 + 5 * x^2 - 183.412983 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69046 MARCELL ANDREY MACHADO PINTO 19FCN107 - 8 apos 09/04, 50% _____ / _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

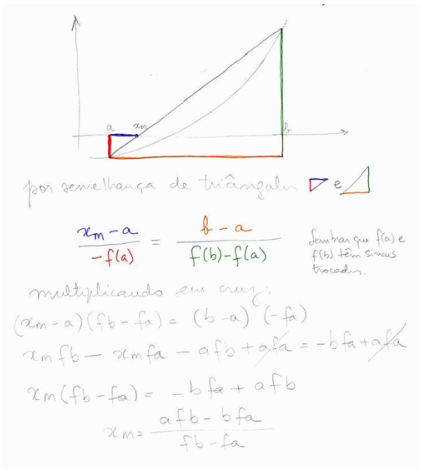
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

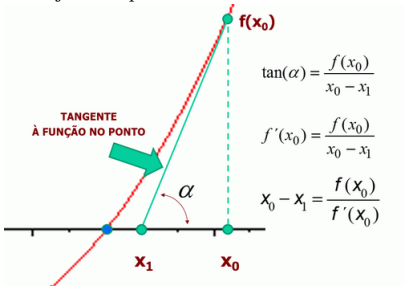
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
      {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[17]{9849}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 6 * sin(x) + 5 * cos(x) + 6.367238 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((4 * x)/10) * e^{((6*x)/10)} - 1.297166 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 7 * x^6 + 5 * x^4 - 1261669.921875 = 0$ no intervalo entre 7 e 8 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69060 MELISSA SCHELLIN BECKER 19FCN107 - 9 apos 09/04, 50%
/ _____ / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

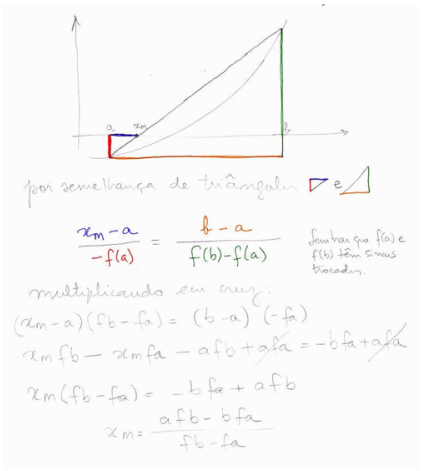
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

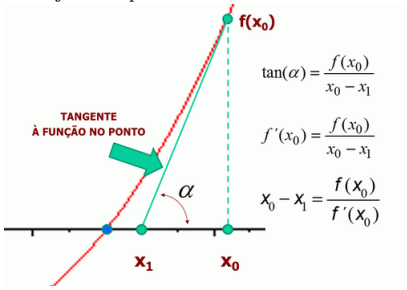
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
      {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   ----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   ----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[11]{5644}$ no intervalo entre 2 e 3 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 4 * sin(x) + 5 * cos(x) + 6.276099 = 0$ no intervalo entre 9 e 10 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((3 * x)/10) * e^{((7*x)/10)} - 2.155195 = 0$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 7 * x^6 + 4 * x^2 - 975405.846528 = 0$ no intervalo entre 7 e 8 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69077 TAYANE CORDEIRO FARIA 19FCN107 - 10 apos 09/04, 50% _____ / / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

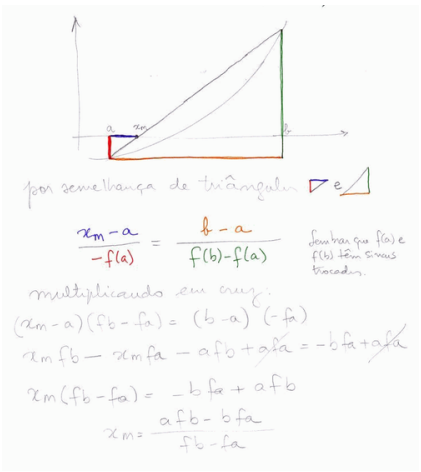
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

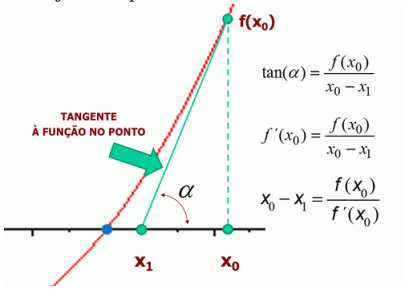
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
       {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[23]{7593}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 4 * sin(x) + 6 * cos(x) - 7.192838 = 0$ no intervalo entre 6 e 7 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((6 * x)/10) * e^{((5*x)/10)} - 217.107952 = 0$ no intervalo entre 7 e 8 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 6 * x^5 + 4 * x^3 - 2144.337920 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

UFPR - UP - UTFPR - PUCPr – 16/03/2019 - 11:19:55.4
Prof Dr P Kantek (pkantek@gmail.com)
Cálculo Numérico - Solução de equações
VIVO734a, V: 1.55 69796 THONNY G FERRAZ
19FCN107 - 11 apos 09/04, 50% _____ / / _____

Cálculo Numérico - solução de equações

Suponha que precisamos usar o computador para resolver equações. Já paramos para entender o que é resolver uma equação ? Conhece-se a forma geral de uma equação $y = f(x)$. Resolver esta equação vem a ser descobrir quais os valores de x que fazer $f(x)$ ser igual a zero. Se olharmos para o gráfico, veremos que busca-se o valor da abcissa x que determina o cruzamento da curva de $f(x)$ no eixo dos x.

Por exemplo, suponha-se precisar descobrir a raiz 11 do número 1739 ? Ou seja, qual o número que multiplicado por ele mesmo 11 vezes dá como resultado 1739 ?

Dizendo isso na forma matemática, pode-se escrever $x^{11} = 1739$, ou melhor ainda $x^{11} - 1739 = 0$. Com isso, chegamos ao ponto.

Método da Bissecção Se olharmos o gráfico de uma função, ver-se-á que no entorno da raiz, de um lado a função é positiva, e do outro lado é negativa ou vice versa. Este método começa pela seleção de 2 pontos x_a e x_b sendo que a função $y = f(x)$ têm sinais opostos nesses dois pontos. Graças a essa constatação, pode-se afirmar que a raiz procurada está entre x_a e x_b. Dizendo de outra maneira. O valor de x procurado, está entre a e b. O Método da bissecção vai dividindo o intervalo a, b pela metade, questionando a seguir em qual das metades o x está. Descoberto isso, despreza-se a metade que não contém x. Levando este processo ao limite, descobre-se qual o valor de x. Em resumo, eis as etapas do método:

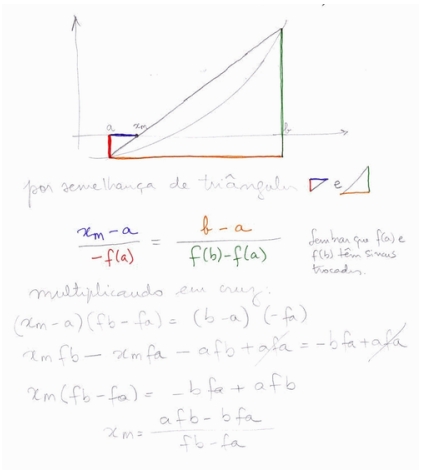
1. Arbitrar intervalo em $[a, b]$ que compreenda a raiz.
2. Arbitrar um erro permitido ϵ
3. Calcular $x_m = (a + b)/2$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $b - a > \epsilon$ voltar ao passo 3.

Seja um exemplo, calcular $f = e^x - sin(x) - 2 = 0$ nos limites $a = 1$ e $b = 1.25$ com $\epsilon = 0.01$. No primeira etapa tem-se

```
Limite Superior: 1.25
Limite Inferior: 1
Tolerancia: 0.01
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.1250  1.0000  1.1250  0.1779  0.5414 -0.1232  0.1250
  1.0625  1.0000  1.0625  0.0200  0.1779 -0.1232  0.0625
  1.0625  1.0312  1.0312 -0.0534  0.0200 -0.1232  0.0312
  1.0625  1.0469  1.0469 -0.0171  0.0200 -0.0534  0.0156
  1.0547  1.0469  1.0547  0.0013  0.0200 -0.0171  0.0078
Raiz:  1.05078125
Erro:  -0.0078125
Eis o programa Python que resolve este caso
import numpy as np
def bissec():
  a=float(input("Limite Superior: "))
  b=float(input("Limite Inferior: "))
  tol=float(input("Tolerancia: "))
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while np.abs(b-a)>tol:
    xm=(a+b)/2
    fxm=np.exp(xm)-np.sin(xm)-2
    fa=np.exp(a)-np.sin(a)-2
    fb=np.exp(b)-np.sin(b)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
   {:8.4f}{:8.4f}'.format(a,b,xm,fxm,fa,fb,np.abs(b-a)))
  xm=(a+b)/2
  print("Raiz: ",xm)
  print("Erro: ",b-a)

bissec()
```

Cordas ou Falsa Posição



O método começa com uma análise do gráfico da função nas proximidades da raiz. Por semelhança de triângulos, pode-se obter a formulação proposta. Eis os passos do algoritmo:

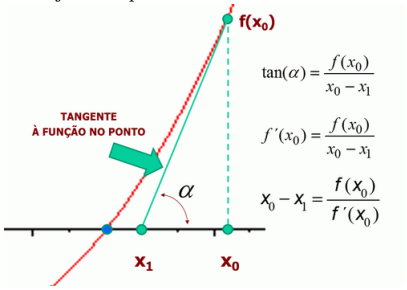
1. Arbitrar um intervalo $[a, b]$ que compreende a raiz
2. Arbitrar ϵ
3. Calcular $x_m = \frac{a.f(b) - b.f(a)}{f(b) - f(a)}$
4. Se $f(a).f(x_m) < 0$ o novo intervalo será $[a, x_m]$
5. Se $f(a).f(x_m) > 0$ o novo intervalo será $[x_m, b]$
6. Enquanto o erro $f(x_m) > \epsilon$ voltar ao passo 3

Veja a seguir a solução da mesma equação, $(e^x - sen(x) - 2 = 0)$ com os mesmos intervalos do método anterior.

```
Limite inferior: 1
Limite superior: 1.25
Tolerância: 0.001
  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro
  1.0463  1.2500  1.0463 -0.0184 -0.1232  0.5414
  1.0530  1.2500  1.0530 -0.0026 -0.0184  0.5414
  1.0540  1.2500  1.0540 -0.0004 -0.0026  0.5414
Raiz:  1.0539728656764054
Erro:  -0.0003663936835978099

Eis o programa Python que resolve este caso
from numpy import *
def cordas():
  a=float(input("Limite inferior: "))
  b=float(input("Limite superior: "))
  tol=float(input("Tolerância: "))
  fxm=1
  print("  -----a  -----b  ----xm  ---fxm  ----fa  ----fb  --erro")
  while abs(fxm)>tol:
    fa=exp(a)-sin(a)-2
    fb=exp(b)-sin(b)-2
    xm=((a*fb)-(b*fa))/(fb-fa)
    fxm=exp(xm)-sin(xm)-2
    if fa*fxm>0:
      a=xm
    else:
      b=xm
    print('{:8.4f}{:8.4f}{:8.4f}{:8.4f}{:8.4f}
      {:8.4f}'.format(a,b,xm,fxm,fa,fb))
  print("Raiz: ",xm)
  print("Erro: ",fxm)
cordas()
```

Método de Newton Este método começa com a interpretação geométrica da derivada. Lembrando, Newton é um monstro da ciência. Entre outras coisas, ele criou o cálculo diferencial e integral, a teoria da luz, a gravitação, as leis do movimento, além de, no fim da vida ter – como funcionário público – dirigido a casa da moeda inglesa, pendurando na forca alguns falsários delinquentes. Voltando ao cálculo diferencial, uma das interpretações da derivada de uma função em um ponto é a tangente da curva da função naquele ponto. Veja-se o que se disse neste desenho



Eis as etapas do método:

1. Obter a derivada da função a resolver
2. Arbitrar um ponto inicial x_0 para a função
3. Arbitrar o erro ϵ
4. Calcular $f(x_0)$ e $df(x_0)$
5. Calcular o "novo" x, $x = x_0 - fx/dfx$
6. Se $|fx| > \epsilon$ retornar ao passo 4

Por exemplo, seja calcular as raízes da mesma equação acima que é $e^x - sen(x) - 2 = 0$. A derivada é $\frac{\partial}{\partial x} = e^x - cos(x)$. Obteve-se a seguinte tabela

```
   --------x   -----------------fx   -------dfx
   1.0000000 -0.12318915634885141    2.1779795
   1.0565612  0.00579321190120519    2.3845934
   1.0541318  0.00001105001756940    2.3754999
   1.0541271  0.00000000004045120    2.3754825
Raiz:  1.0541271241082415
Erro:  4.0451197946822504e-11
Eis o programa Python que resolve este caso
from numpy import *
def newton1():
  x=1
  tol=0.00001
  fx=exp(x)-sin(x)-2
  dfx=exp(x)-cos(x)
  print("   --------x   -----------------fx   -------dfx")
  print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  while abs(fx)>tol:
    x=x-fx/dfx
    fx=exp(x)-sin(x)-2
    dfx=exp(x)-cos(x)
    print("{:12.7f} {:20.17f} {:12.7f}".format(x,fx,dfx))
  print("Raiz: ",x)
  print("Erro: ",fx)
newton1()
```

☞ Para você fazer

1. Resolva pelos 3 métodos, para poder comparar, a seguinte equação $x = \sqrt[21]{7131}$ no intervalo entre 1 e 2 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

2. Idem $y = 3 * sin(x) + 7 * cos(x) + 7.144825 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

3. Idem $y = ((8 * x)/10) * e^{((4*x)/10)} - 9.882632 = 0$ no intervalo entre 3 e 4 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	

4. Idem $y = 6 * x^4 + 3 * x^2 - 183.774600 = 0$ no intervalo entre 2 e 3 e responda a seguir os 3 valores encontrados

bissecção	
cordas	
Newton	