#  Authored by Christopher Allred

#  A02233404

#  Numerical Methods

#  Spring 2020

1. Linear algebraic equations can arise in the solution

of differential equations. For example, the following heat

equation describes the equilibrium temperature

T = T(x)(oC)

at a point x (in meters m) along a long thin **rod**,



d2T/dx2 = h′(T − Ta), (1)

where Ta = 10oC denotes the temperature of the surrounding

air, and h′ = 0.03 (m−2) is a heat transfer coefficient. Assume

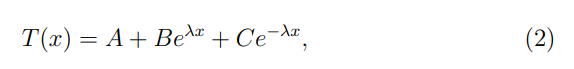
that the rod is 10 meters long (i.e. 0 ≤ x ≤ 10) and has

boundary conditions imposed at its ends given by T(0) = 20oC

and T(10) = 100oC.

a) Using standard ODE methods, which you do not need to repeat here,

the general form of an analytic solution to (1) can be derived as



T(x)=A+Beλx +Ce−λx, (2)

where A, B, C, and λ are constants. Plug the solution of type (2)

into both sides of equation (1). This should give you an equation

that must be satisfied for all values of x, for 0 ≤ x ≤ 10, for some

fixed constants A, B, C, and λ. Analyze this conclusion to determine

what the values of **A and λ** must be.

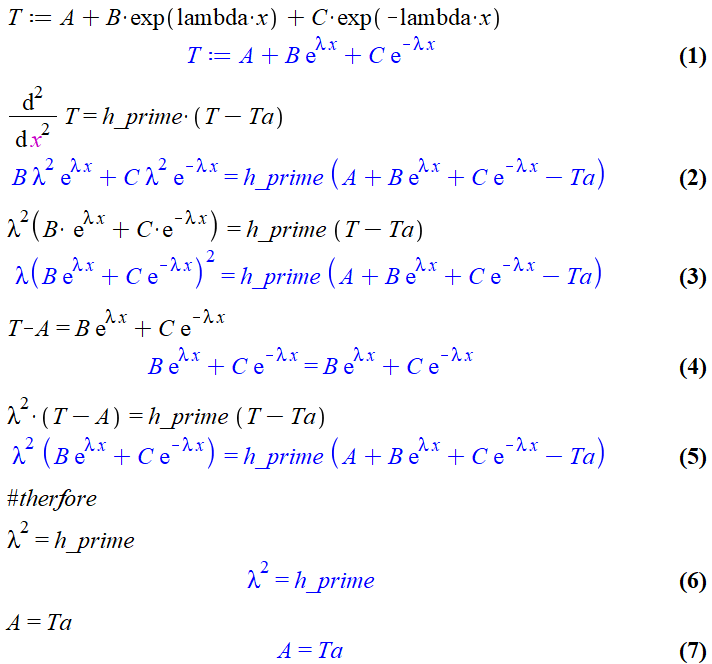
''' NO COMPUTER CODE REQUIRED

1m 5 m 10 m

| | |

^ ^

20oC 100oC



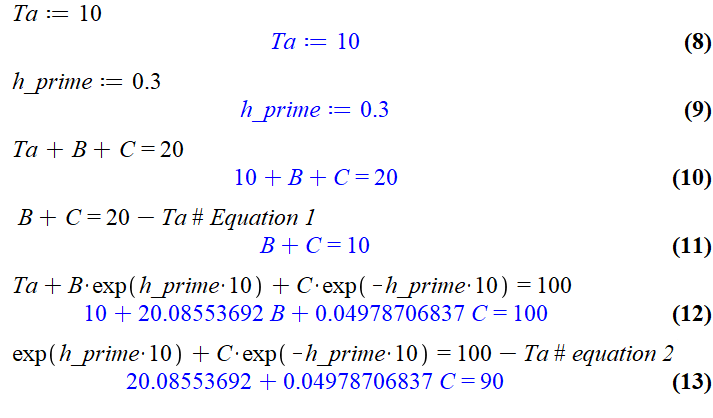
'''

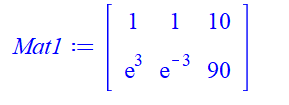
(b) Next, impose the boundary conditions T (0) = 20 oC and

    T (10) = 100 oC to derive a system of 2 linear algebraic equations

    for B and C. Provide the system of two equations you have derived.

''' NO COMPUTER CODE REQUIRED





'''

(c) Use one of the numerical algorithms you developed for

    homework 3 (Gauss elimination or LU decomposition) to solve the

    algebraic system you de- rived in question 2(b) above, and obtain an

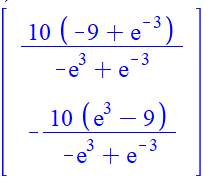
    analytic solution to (1) of the form (2). By analytic solution we mean

    an explicit solution to equation (1) which is valid for each x in

    the interval [0, 10].

'''

|  |
| --- |
| Gauss elimination |
| #  Authored by Christopher Allred  #  A02233404  #  Numerical Methods  #  Spring 2020  import numpy as np  import math  TOLERANCE = 0.0001  def substitute(a, b):      n = len(a)      aCopy = a[:]      x = [None]\*(n)      # x[n] = b[n]/aCopy[n][n]      for i in range(n-1,-1,-1):          # sum = b[i]          x[i] =aCopy[i][n]/aCopy[i][i]          for j in range( i-1,-1,-1 ):              # sum = sum + aCopy[i][j]\*x[j]              aCopy[j][n] -= aCopy[j][i] \*x[i]        return x    def guss2(a, b):      print(np.matrix(a))      print("Solution:...")      n = len(a)-1      m = len(a[1])-1      er = 0      # get the Max val in each row      s = [None] \* (n+1)      for i in range(0,n+1):          s[i] = abs(a[i][0])          for j in range( 1,n+1):              if abs(a[i][j]) > s[i]:                  s[i] = abs(a[i][j])            #Elimination:      for k in range(0,n+1):          # print("k:" +str(k))          #Pivot:          p = k          big = abs(a[k][k]/s[k])          for ii in range(k+1,n+1):              dummy = abs(a[ii][k]/s[ii])              if dummy > big:                  big = dummy                  p = ii            # Pivot          if p != k:              for jj in range( k,n+1):                  dummy = a[p][jj]                  a[p][jj] = a[k][jj]                  a[k][jj] = dummy                dummy = b[p]              b[p] = b[k]              b[k] = dummy              dummy = s[p]              s[p] = s[k]              s[k] = dummy            if abs(a[k][k]/s[k]) < 0:              er =- 1              break #EXIT FOR            for i in range(k+1,n+1):              factor =- a[i][k]/a[k][k]              for j in range (k,m+1):                  a[i][j] = a[i][j]+factor\*a[k][j]              b[i] = b[i]+factor\*b[k]      if abs(a[n][n]/s[n]) < 0:          er = -1      print(np.matrix(a))      print("Coefficients: " + str(np.matrix(b)))      # Elimination      if er != -1:            #Substitute:          vals = substitute(a,b)          print("Solution Values: "+ str(vals))          return vals    if \_\_name\_\_ == "\_\_main\_\_":      A1 =[[1,  1, 10],          [math.exp(3) , math.exp(-3),  90]]      bConsts1 = [10,90]      guss2(A1,bConsts1)  Terminal Output |



Solution Values: [4.467121518528577, 5.532878481471423]  
A = 4.467121518528577

B = 5.532878481471423

'''

(d) Next we will discuss how to obtain a numerical solution to (1).

    That is, we will seek to obtain an approximate solution to (1) which

    describes the value of T at 9 intermediate points inside the interval

    [0,10]. More precisely, the equation (1) can be transformed into a

    linear algebraic system for the temperature at 9 interior points

T1 = T(1),

 T2 = T(2),

 T3 = T(3),

 T4 =T(4),

 T5 =T(5),

 T6 =T(6),

 T7 =T(7),

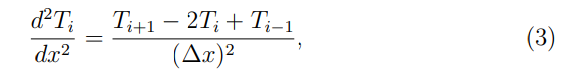
 T8 =T(8),

 T9 =T(9),

    by

    using the following **finite difference approximation**for the second

    derivative at the ith interior point,



d2Ti/ dx2 = (Ti+1 −2Ti +Ti−1)/ (∆x)\*\*2,  (3)

    where

1≤i≤9,

T0 =T(0) =20oC,

T10=T(10)=100oC,

    and **∆x is the equal spacing** between consecutive interior points

    (i.e. with 9 equally spaced interior points inside [0,10] it holds

    that ∆x = 1). Use (3) to rewrite (1) as a system of 9 linear algebraic

    equations for the unknowns T1, T2, T3, T4, T5, T6, T7, T8, and T9.

    Provide the system of 9 equations you have derived.

''' NO COMPUTER CODE REQUIRED