Name: **Undergrad/Grad**

EMCH 585 (Spring 2018): Homework # 4

Due date: March 8 (Thursday) before the start of the class

APOGEE students may email HW solution (in pdf format) to professor

General Instructions:

- All course policies apply (e.g. submitting your original work, honor code etc.)
- Submit your HW as a well-organized report (e.g. write your name clearly, leave space/margin for grading/comment, label plots, use proper units etc.)
- Please present your work such that it clearly demonstrates your understanding of the key concepts.
- Do not re-derive any expressions/equations which have already been derived in the class unless specifically asked to do so. No partial credit will be given for simply copy/pasting equations from class note unless you attempt to further advance towards a solution.
- Think in the context of contents covered this class (often thinking too general may lead to a lengthy path to solution or create confusion).
- Do not start working on it at the last minute before the deadline, instead start early and seek help as needed.

Note on summation indices m, n in Navier solution:

As we discussed in the class, for equations in Navier method m, n are summation indices and they are not cosine or sine as in CLT. Also, some expressions may have alpha and beta which are again functions of summation indices m, n. They are not thermal/moisture coefficients as in CLT.

$\alpha(m)=\frac{m\pi}{a} \quad \beta(n)=\frac{n\pi}{b}$ Any equation having $\alpha(m)$ and $\beta(n)$ becomes function of m and n. For example, while writing in a program, Qmn is written as a function $Q\text{mn}(m,n)$ *and* $\hat{C}_{11}$ is written as a function $\text{C11}hat(m,n)$. Please be aware of these to avoid mistakes. One way to avoid mistake is to create separate CLPT Navier method code instead of adding lines to ABD/CLT code. This may be inconvenient for some people but will help eliminate errors.

Note on attaching CLPT Navier Method code:

Not required. If you could not get an answer out of your code or doubt that your answer may be incorrect, then you may attach your code. I will try to help debugging.

Note on convergence check:

Example Table for convergence and its stability check

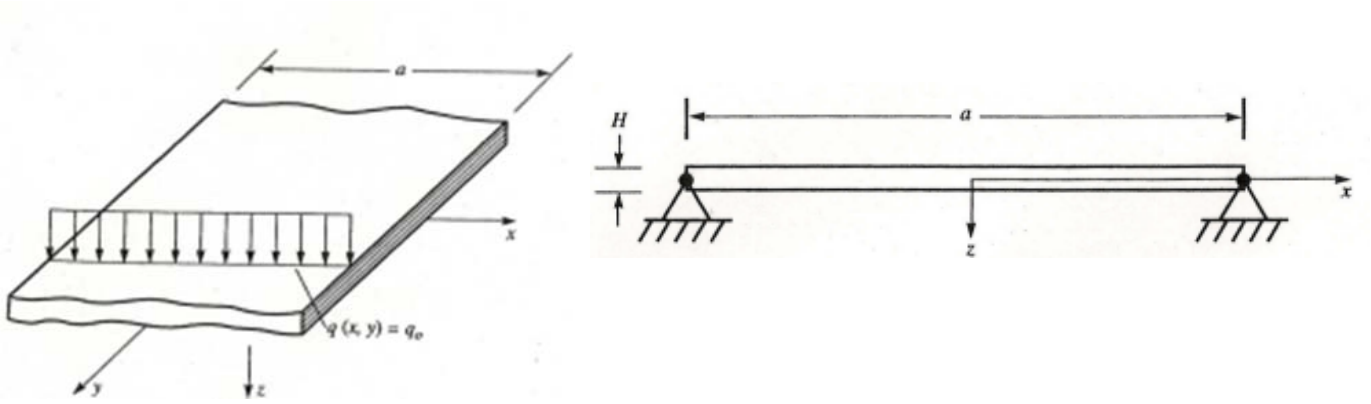
For summation index m, n choose two consecutive values to show convergence to desired accuracy	
For example, for 6 terms, you find displacement	**w=1.543267 meter**
Then for 7 terms, you find displacement again	**w=1.543578 meter (this shows 3 decimal accuracy)**
After achieve convergence, try higher term (say 20 terms), to check is convergence is stable	**w=1.543787 meter (this shows 3 decimal convergence achieved earlier is stable)**

Put your results in tabular format as above. you can customize it to show more or less number of convergence terms for needed accuracy.

General Note: This type of computational solution is very common in the real life. Please take it seriously and try to learn different aspects of it through this exercise. I am confident you will be greatly benefitted.

Problem #1 (For ALL students credit): 25 Points

Consider a $[90/0_2/90/0_2]_T$ laminated plate which is much longer in the y-direction compared to x-direction (see sketch below) and the plate is subjected to bending due to an applied uniformly distributed load, q(N/m^2). Hence long plate assumption may be considered to be valid.



Boundary Conditions: 1) at $x = \pm a/2, u^0 = 0$ 2) at $x = \pm a/2$, and at $y = \pm b/2, v^0 = 0$ (everywhere)

3) at $x = \pm a/2, w^0 = 0$ 4) at $x = +a/2, M_x = M_x^+ = 0$ *and* at $x = -a/2, M_x = M_x^- = 0$

Starting with relevant governing equations as derived in the class and given boundary conditions, show that (must include all steps] the displacements $u^0(x)$ *and* $w^0(x)$ of the reference-surface can be written as below.

$$u^o(x) = \frac{B_{11}}{D_{11}} \frac{q_o a^3}{24 A_{11}^R} \left\{ 4\left(\frac{x}{a}\right)^3 - \left(\frac{x}{a}\right) \right\}$$

$$v^o(x) = 0$$

$$w^o(x) = \frac{q_o a^4}{384 D_{11}^R} \left\{ 16\left(\frac{x}{a}\right)^4 - 8\left(3 - 2\frac{B_{11}^2}{A_{11}D_{11}}\right)\left(\frac{x}{a}\right)^2 + \left(5 - 4\frac{B_{11}^2}{A_{11}D_{11}}\right) \right\}$$

$$A_{11}^R = A_{11}\left(1 - \frac{B_{11}^2}{A_{11}D_{11}}\right)$$

$$D_{11}^R = D_{11}\left(1 - \frac{B_{11}^2}{A_{11}D_{11}}\right)$$

Please note that the results are already discussed in class note, and therefore, no credit will be given for only copying those from notes. For full credit, you must show all steps (for example, finding each integration constants) in your derivation. Also, do not try to use any computer or substitute any property values (like ABD terms or plate dimensions) as we are interested in general equations. This should be purely analytical derivation done by hand.

Problem #2 (For UNDERGRADUATE STUDENT credit): 25 Points

A specially orthotropic laminated plate (a=b=1.5 m) is subjected to uniformly distributed load (q0=1N/m2) and simply supported on all four edges. Find maximum displacement (w) due to bending at the center of the plate using Navier method. Please ensure that you achieve displacement result (expressed in meter) with 5 decimal places accuracy.

Use the following data extracted from [ABD] matrix of the material:

$$D11:= 2.48 \quad D22:= 0.54 \quad D12:= 0.05 \quad D66:= 0.08 \ \text{Nm}$$

Please show iterations showing convergence check and its stability confirmation in a template shown.
If you wish (particularly if you have doubts about answer), you may attach your code but not required.

Problem#2 (For GRADUATE STUDENT credit): 25 Points

For a square (a=b=1.5 m) laminated plate (4 layers of 0.150 mm thick lamina) is subjected to uniformly distributed load (q0=1N/m2). Find maximum displacement (w) due to bending at the center of the plate using appropriate Navier method. [ABD] matrix is given as below (units are [A]= N/m,[B]= N and [D]=N.m):

$$ABD = \begin{pmatrix} 5.037\times 10^{7} & 1.809\times 10^{6} & 0 & 3.231\times 10^{3} & 0 & 0 \\ 1.809\times 10^{6} & 5.037\times 10^{7} & 0 & 0 & -3.231\times 10^{3} & 0 \\ 0 & 0 & 2.64\times 10^{6} & 0 & 0 & 0 \\ 3.231\times 10^{3} & 0 & 0 & 1.511 & 0.054 & 0 \\ 0 & -3.231\times 10^{3} & 0 & 0.054 & 1.511 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.079 \end{pmatrix}$$

Please ensure that you achieve displacement result (expressed in meter) with 5 decimal places accuracy.
Please show iterations showing convergence check and its stability confirmation in a template shown.
If you wish (particularly if you have doubts about answer), you may attach your code but not required.

****

Relevant Problem for PRACTICE HW (Do not return for grading but may help you for future tests)

Using the same problems as above, can you find maximum bending moment and maximum stresses?

Hint: In the class, we have equations for Stresses and Moments in Navier solution format. You use those once you have the rest of parameters already defined and set in your program in above HW problem. You can also pick any properties from your CLT stress analysis examples.

(2) Navier Solution SS1 Condition since A16=A26=B16=B26=D16=D26=0

Find w at center of plate => (x, y) = (a/2, b/2)

W convergence series

size: 1 precision: 0 w: 1.0000000000
size: 2 precision: 0 w: 0.0279348928
size: 3 precision: 5 w: 0.0279348928
size: 4 precision: 3 w: 0.0274652377
size: 5 precision: 5 w: 0.0274652377
size: 6 precision: 4 w: 0.0274967142
size: 7 precision: 5 w: 0.0274967142
size: 8 precision: 5 w: 0.0274905793
size: 9 precision: 5 w: 0.0274905793
size: 10 precision: 5 w: 0.0274922814
size: 11 precision: 17 w: 0.0274922814
size: 12 precision: 6 w: 0.0274916469
size: 13 precision: 17 w: 0.0274916469
size: 14 precision: 6 w: 0.0274919190
size: 15 precision: 5 w: 0.0274919190
size: 16 precision: 6 w: 0.0274917848
size: 17 precision: 5 w: 0.0274917848
size: 18 precision: 7 w: 0.0274918561
size: 19 precision: 17 w: 0.0274918561
size: 20 precision: 7 w: 0.0274918150
size: 21 precision: 5 w: 0.0274918150
size: 22 precision: 7 w: 0.0274918398
size: 23 precision: 17 w: 0.0274918398
size: 24 precision: 7 w: 0.0274918240
size: 25 precision: 17 w: 0.0274918240
size: 26 precision: 7 w: 0.0274918344
size: 27 precision: 17 w: 0.0274918344
size: 28 precision: 8 w: 0.0274918273
size: 29 precision: 5 w: 0.0274918273

Convergence to 5 decimal places occurs around m,n size of 7 => w = .02749

```
import numpy as np

np.set_printoptions(precision=3)

ABD = np.array([
    [  5.037e+07,  1.809e+06,  0        ,  3.231e+03,  0        ,  0        ],
    [  1.809e+06,  5.037e+07,  0        ,  0        , -3.231e+03,  0        ],
    [  0        ,  0        ,  2.640e+06,  0        ,  0        ,  0        ],
    [  3.231e+03,  0        ,  0        ,  1.511    ,  5.400e-02,  0        ],
    [  0        , -3.231e+03,  0        ,  5.400e-02,  1.511    ,  0        ],
    [  0        ,  0        ,  0        ,  0        ,  0        ,  7.900e-02]])

a = 1.5 #m
b = 1.5 #m

q0 = 1 #N/m/m

def Qmn(m, n): #uniform loading
    return (16*q0)/(np.pi**2 * m * n)

def mn(m_range, n_range):
    m, n = np.meshgrid( np.arange(m_range)+1, np.arange(n_range)+1 )
    return m, n

def a_mn(m, n, Nx=0, Ny=0): #Navier SS1 case
    A = (m*np.pi)/a
    B = (n*np.pi)/b

    c11 = ABD[0,0]*(A**2) + ABD[2,2]*(B**2)
    c12 = (ABD[0,1] + ABD[2,2])*A*B
    c13 = -ABD[0,3]*(A**3) - (ABD[0,4] + 2*ABD[2,5])*A*(B**2)
    c22 = ABD[2,2]*(A**2) + ABD[1,1]*B**2
    c23 = -ABD[1,4]*(B**3) - (ABD[0,4] + 2*ABD[2,5])*(A**2)*B
    c33 = ABD[3,3]*(A**4) + 2*(ABD[3,4] + 2*ABD[5,5])*(A**2)*(B**2) +
ABD[4,4]*(B**4)

    s33 = Nx*A**2 + Ny*B**2

    a0 = c11*c22 - c12*c12
    a1 = c12*c23 - c13*c22
    a2 = c13*c12 - c11*c23

    amn = c33 + (c13*a1 + c23*a2)*a0**-1

    return amn

def Wmn(m, n):
    return Qmn(m, n) * a_mn(m, n)**-1
```

```
def w_o(x, y, precision=3):
    W = 0
    W_new = 1

    size = 1
    while size < 30:
        p = np.floor( -np.log10(np.abs(W_new-W)) )
        if p == np.inf: p = precision

        print('size: %d \tprecision: %d\tw: %.10f' %(size, p, W_new))

        W = W_new

        m, n = mn(size, size)

        W_new = np.sum( Wmn(m, n) * np.sin(m*np.pi*x*a**-1) *
np.sin(n*np.pi*y*b**-1) )

        size += 1

np.seterr(divide='ignore')
w_o(a/2, b/2, precision=5)
```