Advanced Mechanics of Solids Term Project-Spring Semester(2022)

Term Project Report on

Buckling of Simply Supported Rectangular Plate Under Uni-axial compression

Prepared by

Group No - 2

Pankaj Suhas Pandit(21ME63R08)

Putluru Rukmangadareddy(21ME63R16)

Apurva Sunil Rangari(21ME63R17)

Dhiraj Dharmadip Raut(21ME63R31)

Index

Introduction

The Föppl–von Kármán equations are named after August Föppl and Theodore von Kármán are a set of nonlinear partial differential equations describing the large deflections of thin flat plates.With applications ranging from the design of submarine hulls to the mechanical properties of cell wall. In our problem statememnt we have considered Buckling of Uniformely compresssed Rectangular plate simply supported along two opposite sides perpendicular in the direction of compression and having various edge conditions along the other two sides.

Prerequisites

The first Föppl–von Kármán equation is

$$N_x \frac{\partial^2 w}{\partial x^2} + 2N_{xy}\frac{\partial^2 w}{\partial x \partial y} + N_y \frac{\partial^2 w}{\partial y^2} - D\nabla^4 w = -q$$

$where$

$$\nabla^4 w = \frac{\partial^4 w}{\partial x^4} + \frac{\partial^4 w}{\partial^2 x \partial^2 y} + \frac{\partial^4 w}{\partial y^4}$$

Useful representations

$(1) D = \frac{Eh^3}{12(1-\nu^2)}$
$(2) M_x = -D(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2})$
$(3) M_y = -D(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2})$
$(4) M_{xy} = -D(\frac{\partial^2 w}{\partial x \partial y})$
$(5) Q_x = \frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y}$
$(6) Q_y = \frac{\partial M_{xy}}{\partial x} + \frac{\partial M_y}{\partial y}$

$(7) N_x = \frac{\partial F^2}{\partial y^2}$
$(8) N_y = \frac{\partial F^2}{\partial x^2}$
$(9) N_{xy} = -\frac{\partial F^2}{\partial y \partial x}$
Where F satisfies the equation,
$\nabla^4 F = Eh[(\frac{\partial w^2}{\partial x \partial y})^2 - \frac{\partial^2 w}{\partial x^2}\frac{\partial^2 w}{\partial y^2}]$
The above equation represents 2^{nd} Föppl–von Kármán equation.

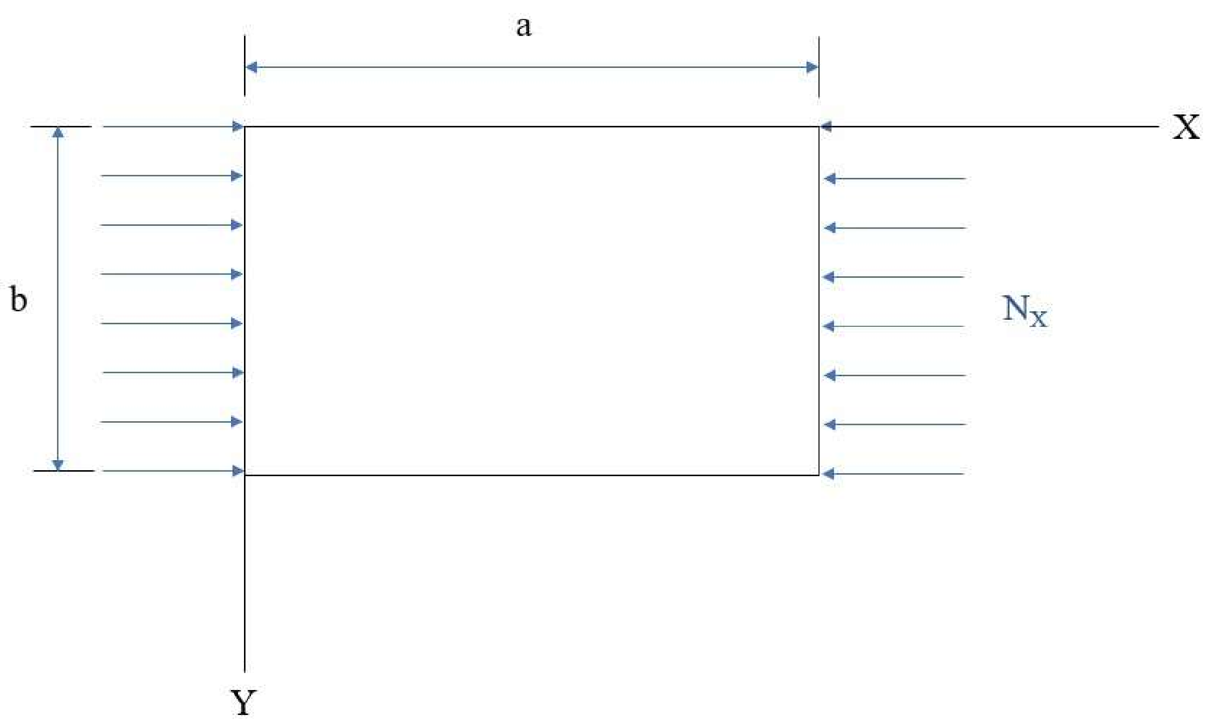
Problem statement

Here we are considering rectangular plate with dimensions a and b along x and y directions respectively.

The Reactangular plate is simply supported at the edges $x = 0$ and $x = a$.

The plate is subjected to a compressive load along the x - direction and the magnitude of the load is N_x.

The special condition that we have considered is that at $y = 0$ the plate is simply supported and at $y = b$ it is free.



Solution for Our Consideration:

The Föppl-von Kármán equation is given by:
As we are loading in x- direction, only N_x term will be there.
Final Governing equation is,

$$D(\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial x^4}) = -N_x \frac{\partial^2 w}{\partial x^2}$$

Since it is simply suported on the sides perpendicular to x-axis, boundary conditions on the sides $x = 0$ and $x = a$ are:

$$w = 0 \quad \text{and} \quad \frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2} \quad \text{at} \quad x = 0 \quad \text{and} \quad x = a$$

Similiarly the edge $y = 0$ is also simply supported,

$$w = 0 \quad \text{and} \quad \frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2} \quad \text{at} \quad y = 0$$

Since the edge $y = b$ is free end Bending moment and shear force terms are zero,

$$\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2} \quad \text{and} \quad \frac{\partial^3 w}{\partial y^3} + (2-\nu)\frac{\partial^3 w}{\partial x^2 \partial y} \quad \text{at} \quad y = b$$

Shear force term is ,

$$Q_y - \frac{\partial M_{xy}}{\partial y}$$

In [1]:

```
import sympy as sp
```

In [2]:

```
x, y = sp.symbols('x, y')
a,b,Nx,D,w,m,nu=sp.symbols('a,b,N_x,D,w,m,nu',positive=True)
f=sp.Function('f')(y)
```

As discussed before simply suported on the sides perpendicular to x-axis, boundary conditions on the sides $x = 0$ and $x = a$ are:

$$w = 0 \quad \text{and} \quad \frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2} \quad \text{for} \quad x = 0 \quad \text{and} \quad x = a$$

So we are considering an expression for w which satisfies the above two boundary conditions.
The term $sin(\frac{\pi m x}{a})$ satisfies the above two boundary condtions. But w is a function of both x,y.
w must be defined in such a way that it should depend on y also.

In [3]:

```
from IPython.display import Math, Latex
```

In [4]:

```
w=sp.sin(m*sp.pi*x/a)*f
display(Math(r'w = {}'.format(sp.latex(w))))
```

$$w = f(y)\sin\left(\frac{\pi m x}{a}\right)$$

In [5]:

```
def laplacian(f):
    return sp.diff(f,x,2)+sp.diff(f,y,2)
def biharmonic(f):
    return laplacian(laplacian(f))
```

In [6]:

```
print('Governing differential equation : ')
eq=(sp.Eq(D*biharmonic(w) + Nx*sp.diff(w,(x,2)),0)).simplify()
display(eq)
```

Governing differential equation :

$$\frac{\left(Da^4\frac{d^4}{dy^4}f(y) - 2\pi^2 Da^2m^2\frac{d^2}{dy^2}f(y) + \pi^4 Dm^4 f(y) - \pi^2 N_x a^2 m^2 f(y)\right)\sin\left(\frac{\pi m x}{a}\right)}{a^4} = 0$$

From the above expression,for non trivial soultion we take the coefficient of $sin(\frac{\pi m x}{a})$ is equal to zero.

Now the expression yields to

$$\frac{d^4 f}{dy^4} - \frac{2m^2\pi^2}{a^2}\frac{d^2 f}{dy^2} + (\frac{m^4\pi^4}{a^4} - \frac{N_x}{D}\frac{m^2\pi^2}{a^2})f = 0$$

By solving the above equation

In [7]:

```
print('On solving upon above differential equation, we get : ')
y_soln = sp.dsolve(eq).simplify()
display(y_soln)
```

On solving upon above differential equation, we get :

$$f(y) = C_1 e^{-\frac{\sqrt{\pi}\sqrt{my}\sqrt{\pi\sqrt{D}m-\sqrt{N_x}a}}{\sqrt[4]{D}a}} + C_2 e^{\frac{\sqrt{\pi}\sqrt{my}\sqrt{\pi\sqrt{D}m-\sqrt{N_x}a}}{\sqrt[4]{D}a}} + C_3 e^{-\frac{\sqrt{\pi}\sqrt{my}\sqrt{\pi\sqrt{D}m+\sqrt{N_x}a}}{\sqrt[4]{D}a}} + C_4 e^{\frac{\sqrt{\pi}\sqrt{my}\sqrt{\pi\sqrt{D}}}{\sqrt[4]{D}a}}$$

Useful substituions

Now lets

$$\alpha = \frac{\sqrt{\pi}\sqrt{m}\sqrt{\pi\sqrt{D}m - \sqrt{N_x}a}}{\sqrt[4]{D}a} = \sqrt{\frac{m^2\pi^2}{a^2} + \sqrt{\frac{N_x}{D}\frac{m^2\pi^2}{a^2}}}$$

$$\beta = \frac{\sqrt{\pi}\sqrt{m}\sqrt{\pi\sqrt{D}m + \sqrt{N_x}a}}{\sqrt[4]{D}a} = \sqrt{-\frac{m^2\pi^2}{a^2} + \sqrt{\frac{N_x}{D}\frac{m^2\pi^2}{a^2}}}$$

In [8]:

```
alpha, beta,A1,A2,A3,A4 = sp.symbols('alpha, beta,A_1,A_2,A_3,A_4,')
f1 = sp.Function('f')(y)
eqn = sp.Eq(f1,A1*sp.exp(-alpha*y) + A2*sp.exp(alpha*y) + A3*sp.cos(beta*y) + A4*sp.sin(bet
display(eqn)
```

$$f(y) = A_1 e^{-\alpha y} + A_2 e^{\alpha y} + A_3 \cos(\beta y) + A_4 \sin(\beta y)$$

In [9]:

```
print('Boundary conditions 1 : ')
bc1_lhs = eqn.rhs.subs(y,0)
bc1 = sp.Eq(bc1_lhs,0).simplify() # deflection at y=0 is 0
display(bc1)
```

```
Boundary conditions 1 :
```

$$A_1 = -A_2 - A_3$$

In [10]:

```
print('Boundary conditions 2 : ')
bc2_lhs = (sp.diff(eqn.rhs,(y,2)) + nu*sp.diff(eqn.rhs,(x,2))).subs(y,0)
bc2 = sp.Eq(bc2_lhs,0) # My at y=0 is 0
display(bc2)
```

```
Boundary conditions 2 :
```

$$A_1\alpha^2 + A_2\alpha^2 - A_3\beta^2 = 0$$

From above two equations,

$A_1 = -A_2 - A_3 - - - - > (1)$

$A_1\alpha^2 + A_2\alpha^2 - A_3\beta^2 = 0 - - - - - - - > (2)$

$(A_1 + A_2)\alpha^2 - A_3\beta^2 = 0$

from equation 1, substituting $A_1 + A_2 = -A_3$ in equation 2

$-A_3\alpha^2 - A_3\beta^2 = 0$

$-A_3(\alpha^2 + \beta^2) = 0$

$A_3 = 0$

we can conclude that $A_3 = 0$ and $A_1 = -A_2$ then $f(y)$ yields to :

$f(y) = A_1 e^{-\alpha y} - A_1 e^{\alpha y} + A_4 sin(\beta y)$

Using $Sinh(x)$ = $\dfrac{e^x - e^{-x}}{2}$

On further simplifaction $f(y)$ can be written as,

$f(y) = B_1 sinh(\alpha y) + B_2 sin(\beta y)$

In [11]:

```
B1,B2 = sp.symbols('B_1,B_2')
f2 = sp.Function('f')(y)
eqn2 = sp.Eq(f2,B1*sp.sinh(alpha*y) + B2*sp.sin(beta*y))
display(eqn2)
```

$$f(y) = B_1 \sinh(\alpha y) + B_2 \sin(\beta y)$$

In [12]:

```
w_soln1 = ((eqn2.rhs)*(sp.sin(m*sp.pi*x/a))).simplify()
print('w =')
display(w_soln1)
```

w =

$$\left(B_1 \sinh(\alpha y) + B_2 \sin(\beta y)\right) \sin\left(\frac{\pi m x}{a}\right)$$

In [13]:

```
print('Boundary conditions 3 : ')
bc3_lhs = (sp.diff(w_soln1,(y,3)) + (2-nu)*sp.diff(sp.diff(w_soln1,(x,2)),y)).subs(y,b)
bc3 = sp.Eq(bc3_lhs,0).simplify() # dMns/ds at y=b is 0
display(bc3)
```

Boundary conditions 3 :

$$\frac{\left(a^2 \left(B_1 \alpha^3 \cosh(\alpha b) - B_2 \beta^3 \cos(b\beta)\right) + \pi^2 m^2 (\nu - 2)\left(B_1 \alpha \cosh(\alpha b) + B_2 \beta \cos(b\beta)\right)\right) \text{si}}{a^2}$$

In [14]:

```
print('Boundary conditions 4 : ')
bc4_lhs = (sp.diff(w_soln1,(y,2)) + nu*sp.diff(w_soln1,(x,2))).subs(y,b)
bc4 = sp.Eq(bc4_lhs,0).simplify() # My at y=b is 0
display(bc4)
```

Boundary conditions 4 :

$$\frac{\left(a^2 \left(B_1 \alpha^2 \sinh(\alpha b) - B_2 \beta^2 \sin(b\beta)\right) - \pi^2 m^2 \nu \left(B_1 \sinh(\alpha b) + B_2 \sin(b\beta)\right)\right) \sin\left(\frac{\pi m x}{a}\right)}{a^2} =$$

When we try to solve the equations obtained by using boundary conditions is 3 and 4 , we get the arbitary constants will be zero. So again for non trivial solution, taking the determinant of coefficient matrix is zero

In [15]:

```
term1 = bc3.lhs.expand().coeff(B1).simplify()/sp.sin(m*sp.pi*x/a)
display(term1)
```

$$\frac{\alpha\left(a^2\alpha^2+\pi^2m^2\nu-2\pi^2m^2\right)\cosh(\alpha b)}{a^2}$$

In [16]:

```
term2 = bc3.lhs.expand().coeff(B2).simplify()/sp.sin(m*sp.pi*x/a)
display(term2)
```

$$\frac{\beta\left(-a^2\beta^2+\pi^2m^2\nu-2\pi^2m^2\right)\cos(b\beta)}{a^2}$$

In [17]:

```
term3 = bc4.lhs.expand().coeff(B1).simplify()/sp.sin(m*sp.pi*x/a)
display(term3)
```

$$\frac{\left(a^2\alpha^2-\pi^2m^2\nu\right)\sinh(\alpha b)}{a^2}$$

In [18]:

```
term4 = bc4.lhs.expand().coeff(B2).simplify()/sp.sin(m*sp.pi*x/a)
display(term4)
```

$$-\frac{\left(a^2\beta^2+\pi^2m^2\nu\right)\sin(b\beta)}{a^2}$$

In [19]:

```
M = sp.Matrix([[term1, term2], [term3, term4]])
display(M)
```

$$\begin{bmatrix}\frac{\alpha\left(a^2\alpha^2+\pi^2m^2\nu-2\pi^2m^2\right)\cosh(\alpha b)}{a^2} & \frac{\beta\left(-a^2\beta^2+\pi^2m^2\nu-2\pi^2m^2\right)\cos(b\beta)}{a^2}\\ \frac{\left(a^2\alpha^2-\pi^2m^2\nu\right)\sinh(\alpha b)}{a^2} & -\frac{\left(a^2\beta^2+\pi^2m^2\nu\right)\sin(b\beta)}{a^2}\end{bmatrix}$$

In [20]:

```
eqn3= sp.Eq(M.det(),0).expand()
display(eqn3)
```

$$-\alpha^3\beta^2\sin(b\beta)\cosh(\alpha b)+\alpha^2\beta^3\cos(b\beta)\sinh(\alpha b)-\frac{\pi^2\alpha^3m^2\nu\sin(b\beta)\cosh(\alpha b)}{a^2}-\frac{\pi^2\alpha^2}{}$$
$$-\frac{\pi^2\alpha\beta^2m^2\nu\sin(b\beta)\cosh(\alpha b)}{a^2}+\frac{2\pi^2\alpha\beta^2m^2\sin(b\beta)\cosh(\alpha b)}{a^2}-\frac{\pi^2\beta^3m^2\nu\cos(b\beta)\sinh}{a^2}$$
$$+\frac{2\pi^4\alpha m^4\nu\sin(b\beta)\cosh(\alpha b)}{a^4}+\frac{\pi^4\beta m^4\nu^2\cos(b\beta)\sinh(\alpha b)}{a^4}-\frac{2\pi^4\beta m^4\nu\cos(b\beta)\sinh(\alpha}{a^4}$$

On simplifying the above equation,

In [21]:

```
eqn4 = sp.Eq(beta*(alpha**2-nu*m**2*sp.pi**2/a**2)**2*sp.tanh(alpha*b),alpha*(beta**2+nu*m*
display(eqn4)
```

$$\beta\left(\alpha^2 - \frac{\pi^2 m^2 \nu}{a^2}\right)^2 \tanh(\alpha b) = \alpha\left(\beta^2 + \frac{\pi^2 m^2 \nu}{a^2}\right)^2 \tan(b\beta)$$

Now substituting thevalues of α and β in the above expresssion

$$\alpha = \sqrt{\frac{m^2\pi^2}{a^2} + \sqrt{\frac{N_x}{D}\frac{m^2\pi^2}{a^2}}}$$

$$\beta = \sqrt{-\frac{m^2\pi^2}{a^2} + \sqrt{\frac{N_x}{D}\frac{m^2\pi^2}{a^2}}}$$

In [22]:

```
eqn5 = eqn4.subs([(alpha,sp.sqrt((m**2)*(sp.pi**2)/(a**2) + sp.sqrt(Nx*(m**2)*(sp.pi**2)/(D
eqn6 = eqn5.subs([(beta,sp.sqrt(-(m**2)*(sp.pi**2)/(a**2) + sp.sqrt(Nx*(m**2)*(sp.pi**2)/(D
display(eqn6)
```

$$\sqrt{-\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}}\left(-\frac{\pi^2 m^2 \nu}{a^2} + \frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}\right)^2 \tanh\left(b\sqrt{\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N}}{\sqrt{D}}}\right.$$

$$= \sqrt{\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}}\left(\frac{\pi^2 m^2 \nu}{a^2} - \frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}\right)^2 \tan\left(b\sqrt{-\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N}}{\sqrt{D}}}\right.$$

In the above expression N_x is there inside trignometric and Hyperbolc functions.Instead of obtaining an expression for N_x, numerical solution can be found easily.

Numerical Solution for N_x:

In [23]:

```
lhs=sp.sqrt(-sp.pi**2*m**2/a**2+sp.pi*sp.sqrt(Nx)*m/(sp.sqrt(D)*a))*(-sp.pi**2*m**2*nu/a**2
display(lhs)
```

$$\sqrt{-\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}}\left(-\frac{\pi^2 m^2 \nu}{a^2} + \frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{Da}}\right)^2 \tanh\left(b\sqrt{\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N}}{\sqrt{D}}}\right.$$

In [24]:

```
rhs=sp.sqrt(sp.pi**2*m**2/a**2+sp.pi*sp.sqrt(Nx)*m/(sp.sqrt(D)*a))*(sp.pi**2*m**2*nu/a**2-s
display(rhs)
```

$$\sqrt{\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{D}a}}\left(\frac{\pi^2 m^2 \nu}{a^2} - \frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}m}{\sqrt{D}a}\right)^2 \tanh\left(b\sqrt{-\frac{\pi^2 m^2}{a^2} + \frac{\pi\sqrt{N_x}}{\sqrt{D}}}\right.$$

In [25]:

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
```

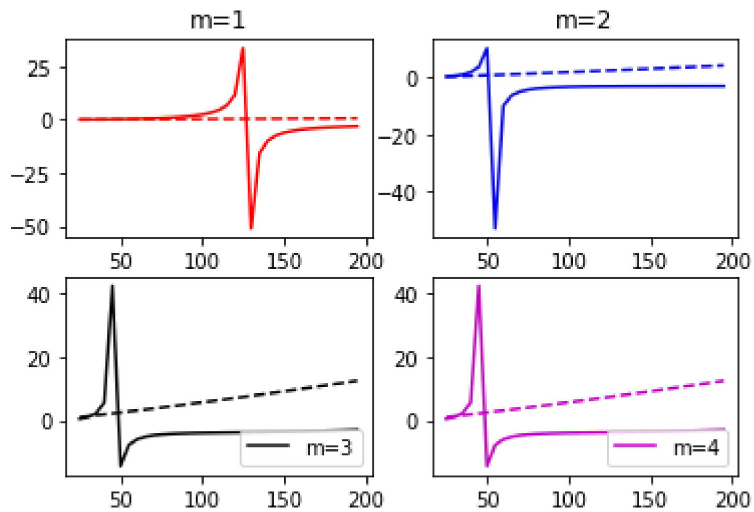
In [26]:

```
a=12
b=2
h=0.01
E = 210000000
nu =0.25
D= E*h**3/(12*(1-nu**2))




Nx_Values = np.arange(25,200,5)
m=1
lhs1 = np.sqrt(-np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(-np.pi**2*m*
rhs1 =  np.sqrt(np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(np.pi**2*m**

m=2
lhs2 = np.sqrt(-np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(-np.pi**2*m*
rhs2 =  np.sqrt(np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(np.pi**2*m**
m=3
lhs3 = np.sqrt(-np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(-np.pi**2*m*
rhs3 =  np.sqrt(np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(np.pi**2*m**

m=4
lhs4 = np.sqrt(-np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(-np.pi**2*m*
rhs4 =  np.sqrt(np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values)*m/(np.sqrt(D)*a))*(np.pi**2*m**
```

In [27]:

```
fig = plt.figure();
fig, ax = plt.subplots(2,2) ;
ax[0][0].plot(Nx_Values,lhs1,'r--');
ax[0][0].plot(Nx_Values,rhs1,'r');
ax[0][0].title.set_text('m=1')
ax[0][1].plot(Nx_Values,lhs2,'b--');
ax[0][1].plot(Nx_Values,rhs2,'b');
ax[0][1].title.set_text('m=2')
ax[1][0].plot(Nx_Values,lhs3,'k--');
ax[1][0].plot(Nx_Values,rhs3,'k',label="m=3");
ax[1][0].legend(loc='lower right');
ax[1][1].plot(Nx_Values,lhs3,'m--');
ax[1][1].plot(Nx_Values,rhs3,'m',label="m=4");
ax[1][1].legend(loc='lower right');
```

```
<Figure size 432x288 with 0 Axes>
```



In [28]:

```
Nx_Values1 = np.arange(127.9,128,0.05)
m=1
lhs11 = np.sqrt(-np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values1)*m/(np.sqrt(D)*a))*(-np.pi**2*
rhs11 =  np.sqrt(np.pi**2*m**2/a**2+np.pi*np.sqrt(Nx_Values1)*m/(np.sqrt(D)*a))*(np.pi**2*m
check1=lhs11-rhs11
display(check1)
```

```
array([-2601.01829992,  9580.05598663])
```

Since α and β contain N_x , calculation of the critical value of N_x if the dimensions of the plate and the elastic constants of the material are known. The smallest value of N_x can be obtained by taking $m = 1$ i.e., by asssuming that the buckled plate has only one half-wave. The magnitude of the corresponding critial N_x can be represented .

From the set of plots drawn, $N_{x(cr)}$ the first plot is drawn for $m = 1$ and we are getting crictical value around 127.95 as the LHS and RHS expressions are equal at that point.

Critical Buckling coefficient(k_{cr}):

From Theory of Elastic Stability by Timoshenko and Gere,
The general expression for $(N_x)_{cr}$ is,

$$(N_x)_{cr} = k_{cr}\frac{\pi^2 D}{b^2 h}$$

where ,

$$k_{cr} = (0.456 + \frac{b^2}{a^2})$$

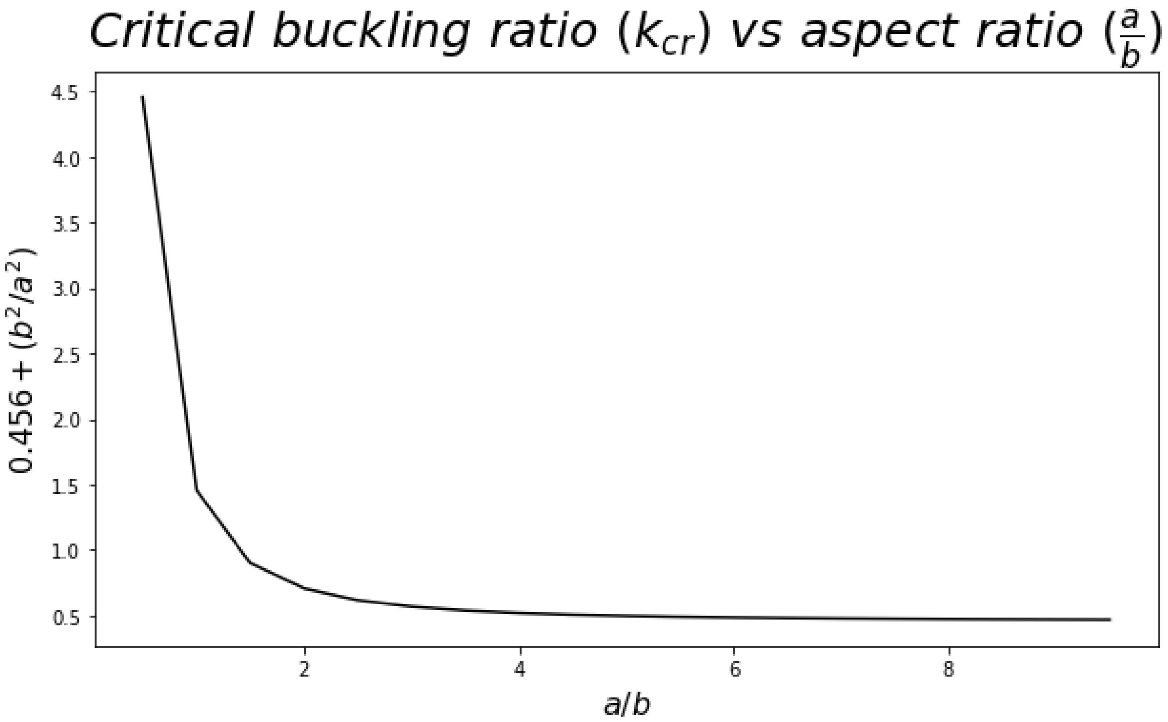
In [29]:

```
a_by_b = np.arange(0.5,10,0.5)
b_by_a=1/a_by_b
k = (0.456 + (b_by_a)**2)
import tableprint as tp
data = np.column_stack((a_by_b,k))
headers = ['a/b', 'k']
tp.table(data, headers)
```

a/b	k
0.5	4.456
1	1.456
1.5	0.90044
2	0.706
2.5	0.616
3	0.56711
3.5	0.53763
4	0.5185
4.5	0.50538
5	0.496
5.5	0.48906
6	0.48378
6.5	0.47967
7	0.47641
7.5	0.47378
8	0.47163
8.5	0.46984
9	0.46835
9.5	0.46708

In [30]:

```
fig = plt.figure(figsize=(10,10/1.8))
plt.plot(a_by_b,k,'k-');
plt.title(r'$Critical$ $buckling$ $ratio$ $(k_{cr})$ $vs$ $aspect$ $ratio$ $(\frac{a}{b})$'
plt.xlabel(r'$a/b$',fontsize=16);
plt.ylabel(r'$0.456 + (b^2/a^2)$',fontsize=16);
```



Terms used:

a = Length of the plate along the x-direction
b = width of the plate along the y -direction
h = thickness of the plate
ν = Poisson's ratio
E = Modulus of Elasticity
D = Bending rigidity
w = deflection of the plate in z - direction

References:

1.Solid Mechanics – A Variational Approach by Clive L. Dym and Irving H. Shames

2.Theory of Elastic Stability by Timoshenko and Gere – (Second Edition)

3.Research paper by Sappati Padmaja Vani on Plate Buckling due to combined bending and Compression

4.Analytical solution for the buckling of rectangular plates under uni-axial compression with variable thickness and elasticity modulus in the y - direction by N Saeidifar,S N Sadeghi, and M R Saviz

Work Distribution:

Primary idea of the project - Putluru Rukmangadareddy

Programming - Dhiraj Dharmadip Raut , Pankaj Suhas Pandit

Research and Paper work - Putluru Rukmangadareddy , Apurva Sunil Rangari

Documentation - Putluru Rukmangadareddy .