**Report : Calculation of Added mass of a submerged body in Heave mode**

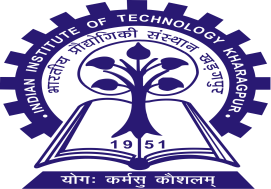
****

Under the guidance of Guidance of

***Mr. M Palaniappan***

Scientist-F

National Institute of Ocean Technology(NIOT), Chennai



By: Shivam Verma

3rd year Undergraduate Student,

Ocean Engineering and Naval Architecture

Indian Institute of Technology(IIT), Kharagpur

Date : 22/10/20

**Contents**

1. Introduction
2. Mathematical Formulation

2.1 Gauss Divergence Theorem

2.2 Reynolds Transport Theorem

1. Hydrodynamic Forces
2. Force on a Moving Body in an Unbounded Fluid
3. Added Mass

5.1 Boundry Integral Equation

5.2 Computation in Python and Output

1. Acknowledgement
2. Bibliography
3. **Introduction**

In case of freely floating body, or seakeeping problem, or for accelerating underwater vehicle, accurate calculation of added mass is very important to get the accurate prediction of the radiation force. Ideally, the radiation force may be calculated directly using Navier – Stokes (N-S) equation. However, potential theory based method is mostly used for the calculation of the added mass. In this project, lower order Rankine panel method is developed for the calculation of the heave added mass for a submerged rectangular box.

1. **Mathematical Formulation**

The general equation of continuity and equation of motion may be given as:

 (1)

 (2)

 (3)

 (4)

The above set of equations are called N-S Equations. The equation 1 represents the mass conservation and equation 2-4 represent momentum conservation and. The above set of equations can help in modelling any fluid flow problem if boundary conditions are rightly specified. However, these non-linear partial differential equations can only be solved using some numerical techniques as the known analytical techniques cannot handle these complex equations. The solutions are however greatly deviated from the actual flow phenomenon, due to cumulative errors of the applied numerical technique.

Here, however we are simplifying our problem by taking some assumptions coupled with some other equation, helps in arriving at equations which can we solved both analytically and numerically. The assumptions comprise the fluid is incompressible, inviscid, homogeneous and the flow is ir-rotational. Based on the above assumption, we can introduce a velocity potential  such that the velocity of the fluid particle can be written as . Also, if the fluid is assumed to be homogeneous, incompressible, then equation (1) takes the form

 (5) Substituting  in equation (5), we get

 (6)

In Cartesian co-ordinate system, the equation (6) can be written in the form:

 (7)

Equation (7) is called the Laplace Equation.Using the similar assumptions, the equation of motion given by (2-4) can be further modified as

 (8)

* 1. **Gauss Divergence Theorem**

The Gauss Divergence theorem relates the flux of a vector field through a closed surface to the divergence of the field in the volume enclosed.

 (9)

* 1. **Renyolds Transport Theorem**

Reynolds transport theorem helps in transferring a control mass system into a control volume system.

 (10)

is the rate of change of the system’s extensive property  . For example, if, we obtain the rate of change of momentum.

 is the rate of change of amount of the property  in the control volume.

 is the rate at which property N is exiting the surface of the control volume. The term  computes the rate of mass transfer leaving across control surface area element 

For example:

 (11)

**3 Hydrodynamic Pressure Forces**

One of the primary reasons for studying the fluid motion past a body is our desire to predict the forces and moments on the body due to dynamic pressure of the fluid. Thus, we wish to consider the six components of the force and moment vectors, which are represented by the integrals of the pressure over the body surface, or

 (12)

 (13)

**4 Force on a Moving Body in an Unbounded Fluid**

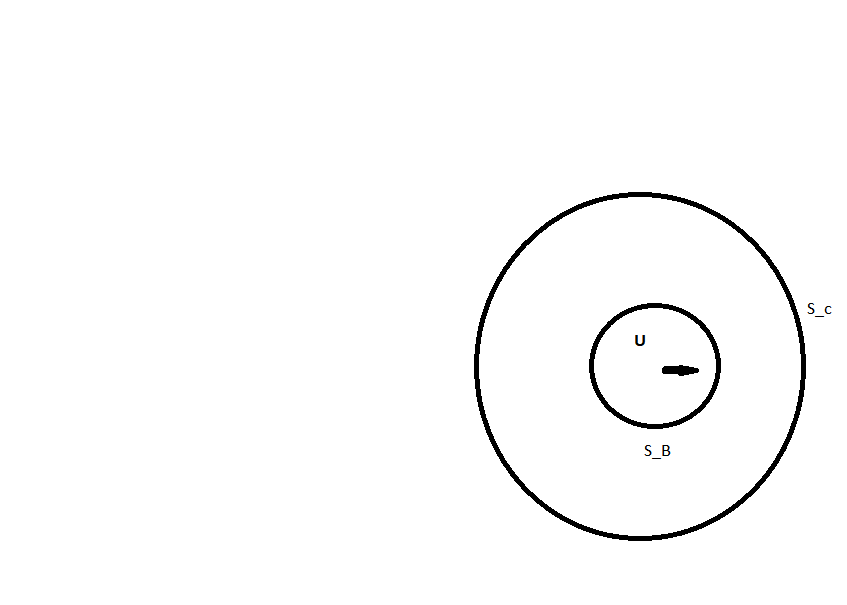
Under pure translation, we get equation (14) on expanding equation (12)

 (14)

Upon careful observation of ϕj , we can conclude that it tells the about the force in the ith direction, if the body is moving in the jth direction. Thus if the body is moving in the x direction and we are interested to know the force in the y direction, then equation (14) takes the form

 (15)

An alternative form of equation (15) can be written as

 (16) 

Now, Applying Gauss divergent theorem and Raynolds transport theorem, we can get



, hence



Omitting the quadratic component gives the following expressions, we get:

 (17)

 (18)

Suppose that the body has the translation velocity U, then the velocity must satisfy the following boundry condition

 (20)

The boundary condition suggests that the total potential may be expressed as the sum

 (21)

 (22)

, which finally gives

 (23)

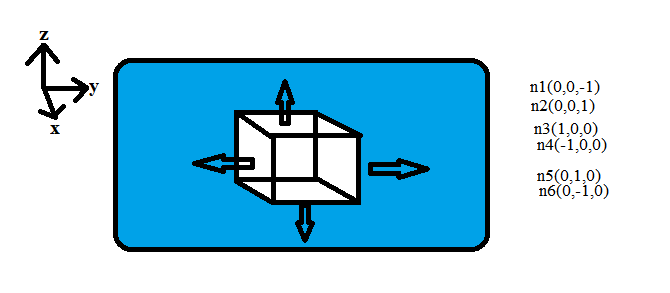
 Force in the i-th direction due to the motion in the j-th direction.

Velocity potential due to unit velocity in the i-th direction.

**5 Added Mass**

In order to understand the added mass qualitatively we can intuitively understand that any body/vessel that moves in any fluid say water, will continuously try to push the fluid surrounded by the body. This extra work done by the body onto the fluid is called the added mass effect of the body. Added mass is a common issue because the body and surrounding fluid cannot occupy the same physical space simultaneously. For simplicity this can be modeled as some volume of fluid moving with the object, though in reality "all" the fluid will be accelerated, to various degrees. The added mass is a second-order tensor, relating the fluid acceleration vector to the resulting force vector on the body

There are many methods to quantify the added mass. Here, I have used **Panel method** to calculate the added mass.



To numerically compute the added mass, I have used the **Boundry Element Integral Equation method**. The procedure is :

1. Formulate the Boundary value Problem ( for ϕ)
2. Find Proper Green’s Function
3. Derive the appropriate Integral Equation
4. Discretize the boundary by small segments called panels
5. Distribute ϕ over the boundary
6. Apply initial/boundary conditions
7. Convert Integral Equation into algebraic Equation in the form [A]{ϕ}={b}
8. Solve for ϕ

**5.1 Boundry Integral Equation**

Solution of Boundry Integral Equation gives the distribution of phi(s) across all the faces of the submerged body.is the potential of source point Q, located on the face. This ϕ(q) is used to calculate the added mass on the body, as well as forces on the body. This **ϕ(q)** is called the **Radiation Potential**.

 (24)

Where the surfacewill be the combination of body, bottom and surface at infinity, however, for the present problem, the non trivial contribution on the integral is only coming from the body, therefore, the (24) may be re-written as:

 (25)

is the potential of the field point P. If P is on the surface, value of  is 2п, if P is outside the body the value of is 4п and if P is inside the body then the value of  is 0. Now define

Velocity potential

 where (26)

Field point

 Source point

To get the computable form of the equation (25) we need to compute the value of  as follows:



Now

, which gives

, which gives

, similarly

 and 

Thus, finally we get

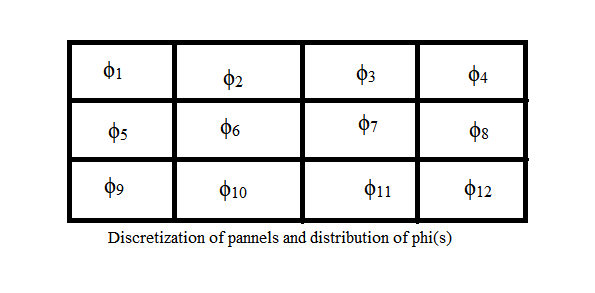
 (27)

The discretized form of equation (25) can be obtained as follows:





 (28)



From equation (28) we can see that the field point is kept fixed and the source point moves.

In order to solve the equation (28) we assume that the field point is fixed on the 1st source point, i.e.,  and hence the value of is 2п. And r is taken out for all the moving source points() and fixed field points(). Then the 1st part of the RHS of equation (28) is assumed to be



Assuming  and , we can re write the above equation as

 (29)

Similarly, 



Assuming  and , we get

 (30)

Iterating on all the panels we end up with n number of equations, one corresponding to each panel. We arrange all the equations obtained as equations (29 and 39) into a matrix form. Thus the final matrix obtained is

 (31)

 (32)

We solve for matrix {ϕ}. Then this {ϕ} is used for evaluating the equation (24). We proceed as



Once we get the value of added mass, we can get the forces using equation (23).

**5.2 Computation in Python and Output**

Computing the matrix using the approach mentioned in equation (31).

import numpy as np

import sys

def matrix(npanels,face\_coordn,face\_norml,mode,face\_ar):

a\_Fc=np.zeros((npanels,npanels))

b\_Fc=np.zeros(npanels)

for i in range(0,npanels):

r\_fp=face\_coordn[i]

b\_j=0

for j in range(0,npanels):

if j==i:

a\_Fc[i,j]=2\*3.1414

continue;

r\_sp=face\_coordn[j]

r=r\_fp-r\_sp

n=face\_norml[j]

a\_Fc[i,j]=(1\*r[mode]\*n[mode]\*face\_ar)/((sum(r\*\*2)\*\*1.5)\*(sum(n\*\*2)\*\*0.5))

b\_j=b\_j+((n[mode]\*face\_ar)/(sum(r\*\*2)\*\*0.5))

b\_Fc[i]=1\*b\_j

return(a\_Fc,b\_Fc)

if \_\_name\_\_=="\_\_main\_\_":

#extracting the no. in between

a=int(input('Enter Value of length : '))

b=int(input('Enter value of breadth : '))

c=int(input("Enter value of height : "))

print("Density of water used : 1.025tonnes/m^3");

print("....NOTE....")

print("X is along front and back face")

print("Y is along left and right face")

print("Z is along bottom and top face")

print("values of modes, heave=2, sway=1.surge=0");

mode\_k = int(input('Enter the mode : '))

if mode\_k >= 3:

print("Wrong input of mode !, exiting")

exit()

x=np.linspace(0,1\*a, (2\*a+1))

y=np.linspace(0,1\*b, (2\*b+1))

z=np.linspace(0,1\*c, (2\*c+1))

xx=np.array([x[i] for i in range(0,len(x)) if(i%2!=0)])

yy=np.array([y[i] for i in range(0,len(y)) if(i%2!=0)])

zz=np.array([z[i] for i in range(0,len(z)) if(i%2!=0)])

###panels centers coordinates storing

Fc1=[] #bottom

Fc2=[] #top

Fc3=[] #front

Fc4=[] #back

Fc5=[] #right

Fc6=[] #left

for i in range(len(xx)):

for j in range(len(yy)):

Fc1.append(tuple((xx[i],yy[j],0\*c))) #bottom

Fc2.append(tuple((xx[i],yy[j],1\*c))) #top

for i in range(len(yy)):

for j in range(len(zz)):

Fc3.append(tuple((1\*a,yy[i],zz[j]))) #front

Fc4.append(tuple((0\*a,yy[i],zz[j]))) #back

for i in range(len(xx)):

for j in range(len(zz)):

Fc5.append(tuple((xx[i],1\*b,zz[j]))) #right

Fc6.append(tuple((xx[i],0\*b,zz[j]))) #left

Fc1=np.array(Fc1) #bottom

Fc2=np.array(Fc2) #top

Fc3=np.array(Fc3) #front

Fc4=np.array(Fc4) #back

Fc5=np.array(Fc5) #right

Fc6=np.array(Fc6) #left

##panels centers end

#panels normals making

n\_Fc1=np.array([(0,0,-1) for i in range(0,len(Fc1))]) #bottom

n\_Fc2=np.array([(0,0,1) for i in range(0,len(Fc2))]) #top

n\_Fc3=np.array([(1,0,0) for i in range(0,len(Fc3))]) #front

n\_Fc4=np.array([(-1,0,0) for i in range(0,len(Fc4))]) #back

n\_Fc5=np.array([(0,1,0) for i in range(0,len(Fc5))]) #right

n\_Fc6=np.array([(0,-1,0) for i in range(0,len(Fc6))]) #left

#panels normals end

mode=mode\_k # 0 for surge,1 for sway, 2 for heave;

m=len(Fc1) # no. of pannels m is same for all the faces

dA=1;

# for Face 1(bottom)

a\_Fc1,b\_Fc1=matrix(len(Fc1), Fc1, n\_Fc1, mode, dA)

#for face 2(top)

a\_Fc2,b\_Fc2=matrix(len(Fc2), Fc2, n\_Fc2, mode, dA)

#for face 3(front)

a\_Fc3,b\_Fc3=matrix(len(Fc3), Fc3, n\_Fc3, mode, dA)

#for face 4(back)

a\_Fc4,b\_Fc4=matrix(len(Fc4), Fc4, n\_Fc4, mode, dA)

#for face 5(right)

a\_Fc5,b\_Fc5=matrix(len(Fc5), Fc5, n\_Fc5, mode, dA)

# for face 6(left)

a\_Fc6,b\_Fc6=matrix(len(Fc6), Fc6, n\_Fc6, mode, dA)

# finding phi

phi\_Fc1=np.dot(np.linalg.inv(a\_Fc1),b\_Fc1)

phi\_Fc2=np.dot(np.linalg.inv(a\_Fc2),b\_Fc2)

phi\_Fc3=np.dot(np.linalg.inv(a\_Fc3),b\_Fc3)

phi\_Fc4=np.dot(np.linalg.inv(a\_Fc4),b\_Fc4)

phi\_Fc5=np.dot(np.linalg.inv(a\_Fc5),b\_Fc5)

phi\_Fc6=np.dot(np.linalg.inv(a\_Fc6),b\_Fc6)

print("Phi of bottom Face(face\_1) : ",phi\_Fc1)

print('\r')

print("Phi of Top Face(face\_2) : ",phi\_Fc2)

print('\r')

print("Phi of Front Face(face\_3) : ",phi\_Fc3)

print('\r')

print("Phi of Back Face(face\_4) : ",phi\_Fc4)

print('\r')

print("Phi of Right Face(face\_5): ",phi\_Fc5)

print('\r')

print("Phi of Left Face(face\_6) : ",phi\_Fc6)

print('\r')

print('Added Mass........')

m\_1\_1=0 #bottom face added mass

m\_2\_2=0 #top face added mass

m\_3\_3=0 #front

m\_4\_4=0 #back

m\_5\_5=0 #right

m\_6\_6=0 #left

rho=1.025

if (mode==2): #heave (top-bottom)

for i in range(len(phi\_Fc1)):

m\_1\_1=m\_1\_1+phi\_Fc1[i]\*(n\_Fc1[i][mode])

for i in range(len(phi\_Fc2)):

m\_2\_2=m\_2\_2+phi\_Fc2[i]\*(n\_Fc2[i][mode])

m\_2\_2=dA\*rho\*m\_2\_2

m\_1\_1=dA\*rho\*m\_1\_1

print('Heave Added mass : ',(m\_1\_1+m\_2\_2))

elif (mode==1): #sway (left-right)

for i in range(len(phi\_Fc5)):

m\_5\_5=m\_5\_5+phi\_Fc5[i]\*(n\_Fc5[i][mode])

for i in range(len(phi\_Fc6)):

m\_6\_6=m\_6\_6+phi\_Fc6[i]\*(n\_Fc6[i][mode])

m\_6\_6=dA\*rho\*m\_6\_6

m\_5\_5=dA\*rho\*m\_5\_5

print('Sway Added mass : ',(m\_6\_6+m\_5\_5))

elif (mode==0): #surge (front-back)

for i in range(len(phi\_Fc3)):

m\_3\_3=m\_3\_3+phi\_Fc3[i]\*(n\_Fc3[i][mode])

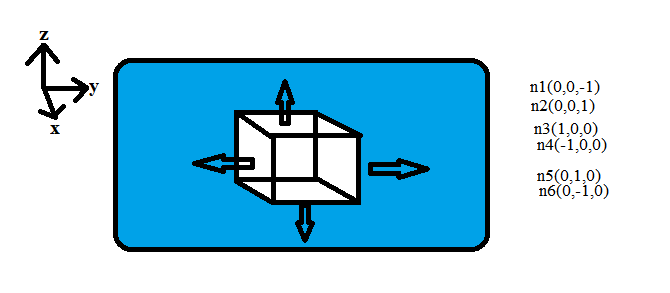
for i in range(len(phi\_Fc4)):

m\_4\_4=m\_4\_4+phi\_Fc4[i]\*(n\_Fc4[i][mode])

m\_4\_4=dA\*rho\*m\_4\_4

m\_3\_3=dA\*rho\*m\_3\_3

print('Surge Added mass : ',(m\_3\_3+m\_4\_4))

****

**Output**

**Output : 1 [Heave added mass calculation]**

Added Mass Calculation of a submerged rectangular body

Enter Value of length : 5

Enter value of breadth : 4

Enter value of height : 6

Density of water used : 1.025tonnes/m^3

....NOTE....

X is along front and back face

Y is along left and right face

Z is along bottom and top face

values of modes, heave=2, sway=1.surge=0

Enter the mode : 2

Phi of bottom Face(face\_1) : [-1.26707112 -1.48201833 -1.48201833 -1.26707112 -1.51447765 -1.78486883

-1.78486883 -1.51447765 -1.58061116 -1.86522179 -1.86522179 -1.58061116

-1.51447765 -1.78486883 -1.78486883 -1.51447765 -1.26707112 -1.48201833

-1.48201833 -1.26707112]

Phi of Top Face(face\_2) : [1.26707112 1.48201833 1.48201833 1.26707112 1.51447765 1.78486883

1.78486883 1.51447765 1.58061116 1.86522179 1.86522179 1.58061116

1.51447765 1.78486883 1.78486883 1.51447765 1.26707112 1.48201833

1.48201833 1.26707112]

Phi of Front Face(face\_3) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Back Face(face\_4) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Right Face(face\_5): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.]

Phi of Left Face(face\_6) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.]

Added Mass........

Heave Added mass : 63.72508966381723

Surge Added mass : 0

Sway Added mass : 0

**Output : 2 [Sway added mass calculation]**

Added Mass Calculation of a submerged rectangular body

Enter Value of length : 10

Enter value of breadth : 7

Enter value of height : 5

Density of water used : 1.025tonnes/m^3

....NOTE....

X is along front and back face

Y is along left and right face

Z is along bottom and top face

values of modes, heave=2, sway=1.surge=0

Enter the mode : 1

Phi of bottom Face(face\_1) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Top Face(face\_2) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Front Face(face\_3) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Back Face(face\_4) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Right Face(face\_5): [1.99802035 2.27247534 2.34818786 2.27247534 1.99802035 2.34438375

2.691366 2.78744044 2.691366 2.34438375 2.53579153 2.91641729

3.02394787 2.91641729 2.53579153 2.64492014 3.04127793 3.15447377

3.04127793 2.64492014 2.69521513 3.09794507 3.213459 3.09794507

2.69521513 2.69521513 3.09794507 3.213459 3.09794507 2.69521513

2.64492014 3.04127793 3.15447377 3.04127793 2.64492014 2.53579153

2.91641729 3.02394787 2.91641729 2.53579153 2.34438375 2.691366

2.78744044 2.691366 2.34438375 1.99802035 2.27247534 2.34818786

2.27247534 1.99802035]

Phi of Left Face(face\_6) : [-1.99802035 -2.27247534 -2.34818786 -2.27247534 -1.99802035 -2.34438375

-2.691366 -2.78744044 -2.691366 -2.34438375 -2.53579153 -2.91641729

-3.02394787 -2.91641729 -2.53579153 -2.64492014 -3.04127793 -3.15447377

-3.04127793 -2.64492014 -2.69521513 -3.09794507 -3.213459 -3.09794507

-2.69521513 -2.69521513 -3.09794507 -3.213459 -3.09794507 -2.69521513

-2.64492014 -3.04127793 -3.15447377 -3.04127793 -2.64492014 -2.53579153

-2.91641729 -3.02394787 -2.91641729 -2.53579153 -2.34438375 -2.691366

-2.78744044 -2.691366 -2.34438375 -1.99802035 -2.27247534 -2.34818786

-2.27247534 -1.99802035]

Added Mass........

Sway Added mass : 274.71284945186056

Heave Added mass : 0

Surge Added mass : 0

**Output : 3 [Surge added mass calculation]**

Added Mass Calculation of a submerged rectangular body

Enter Value of length : 8

Enter value of breadth : 8

Enter value of height : 8

Density of water used : 1.025tonnes/m^3

....NOTE....

X is along front and back face

Y is along left and right face

Z is along bottom and top face

values of modes, heave=2, sway=1.surge=0

Enter the mode : 0

Phi of bottom Face(face\_1) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Top Face(face\_2) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Front Face(face\_3) : [2.36299746 2.71511943 2.89841583 2.98107006 2.98107006 2.89841583

2.71511943 2.36299746 2.71511943 3.15084733 3.37455871 3.47393365

3.47393365 3.37455871 3.15084733 2.71511943 2.89841583 3.37455871

3.62296877 3.73358398 3.73358398 3.62296877 3.37455871 2.89841583

2.98107006 3.47393365 3.73358398 3.84963092 3.84963092 3.73358398

3.47393365 2.98107006 2.98107006 3.47393365 3.73358398 3.84963092

3.84963092 3.73358398 3.47393365 2.98107006 2.89841583 3.37455871

3.62296877 3.73358398 3.73358398 3.62296877 3.37455871 2.89841583

2.71511943 3.15084733 3.37455871 3.47393365 3.47393365 3.37455871

3.15084733 2.71511943 2.36299746 2.71511943 2.89841583 2.98107006

2.98107006 2.89841583 2.71511943 2.36299746]

Phi of Back Face(face\_4) : [-2.36299746 -2.71511943 -2.89841583 -2.98107006 -2.98107006 -2.89841583

-2.71511943 -2.36299746 -2.71511943 -3.15084733 -3.37455871 -3.47393365

-3.47393365 -3.37455871 -3.15084733 -2.71511943 -2.89841583 -3.37455871

-3.62296877 -3.73358398 -3.73358398 -3.62296877 -3.37455871 -2.89841583

-2.98107006 -3.47393365 -3.73358398 -3.84963092 -3.84963092 -3.73358398

-3.47393365 -2.98107006 -2.98107006 -3.47393365 -3.73358398 -3.84963092

-3.84963092 -3.73358398 -3.47393365 -2.98107006 -2.89841583 -3.37455871

-3.62296877 -3.73358398 -3.73358398 -3.62296877 -3.37455871 -2.89841583

-2.71511943 -3.15084733 -3.37455871 -3.47393365 -3.47393365 -3.37455871

-3.15084733 -2.71511943 -2.36299746 -2.71511943 -2.89841583 -2.98107006

-2.98107006 -2.89841583 -2.71511943 -2.36299746]

Phi of Right Face(face\_5): [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Phi of Left Face(face\_6) : [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

Added Mass........

Surge Added mass : 420.9864239929449

Heave Added mass : 0

Sway Added mass : 0

**Acknowledgement**

A course named **Numerical Ship and Offshore Hydrodynamics** taught in our department by eminent

**Dr. Ranadev Datta** , was instrumental in solving this problem.

**Bibliography**

1. Marine Hydrodynamics, J. N. Newman
2. <https://en.wikipedia.org/wiki/Added_mass>

**Thank you**