# **Algorithm**

DynamicSimulation\_Of\_AUVs():

Input:

Current, Temperature, Fluid\_density, Position, Velocity, Mass, Simulation\_time\_total, Simulation\_step, Position\_obstacle, Collision\_flag

Output:

AUV trajectory, Collision detection

Declaration:

Current Float (m/sec) x1

Temperature Float (°C) x2

Fluid\_density Float (kg/m3) ρ

Position Float (m) (dx, dy, dz)

Velocity Float (m/sec) v1

Mass Float (kg) m

Simulation\_time\_total Integer (sec) t1

Simulation\_step Integer (sec) t2

Position\_obstacle Float (m) (px, py, pz)

Collision\_flag Integer i

Procedure:

For t2=0: t1; t2<=t1; t2++

Update x1

Translation motion:

Fdrag ​= ½⋅C​⋅ρ⋅A⋅v1

Fdrag= drag force

C= drag coefficient

A= cross-sectional area perpendicular to the flow

Fbuoyancy​ = ρ​⋅g⋅Vdisplaced

​Fbuoyancy= buoyancy force

g= gravitational acceleration

Vdisplaced= volume of fluid displaced

Fgravity ​= m⋅g

Fgravity= gravity force

g= gravitational acceleration

F = Fdrag+Fbouyancy+Fgravity

F= total force on AUV

a = F/m

a= acceleration of AUV

v1 = v1 + a.t2

dx = dx + v1.t2

dy= dy + v1.t2

dz= dz + v1.t2

Rotation motion:

Ix = (y2 + z2).dm

Iy = (x2 + z2).dm

Iz = (y2 + x2).dm

x, y, z= distances from roll axis to mass elements

Ix = Lnet

Iy = Mnet

Iz = Nnet

Ix​= moment of inertia of the AUV about the roll axis

Iy​= moment of inertia of the AUV about the pitch axis

Iz= moment of inertia of the AUV about the yaw axis

ϕ= roll angle

θ= pitch angle

ψ= yaw angle

Lnet= net torque acting on the AUV about the roll axis

Mnet= net torque acting on the AUV about the pitch axis

Nnet= net torque acting on the AUV about the yaw axis

Collision detection:

If dx = px:

i++

If dy = py:

i++

If dz = pz:

i++

End for

Result:

If the value of the variable Collision\_flag, i, increases, collision between the obstacle and the underwater vehicle will be there. This also gives the trajectory of the vehicle.