정수 중 동적 자세 결정을 위한 알고리즘 개발

1.interDyMFoam

2.interDyMDAUFoam

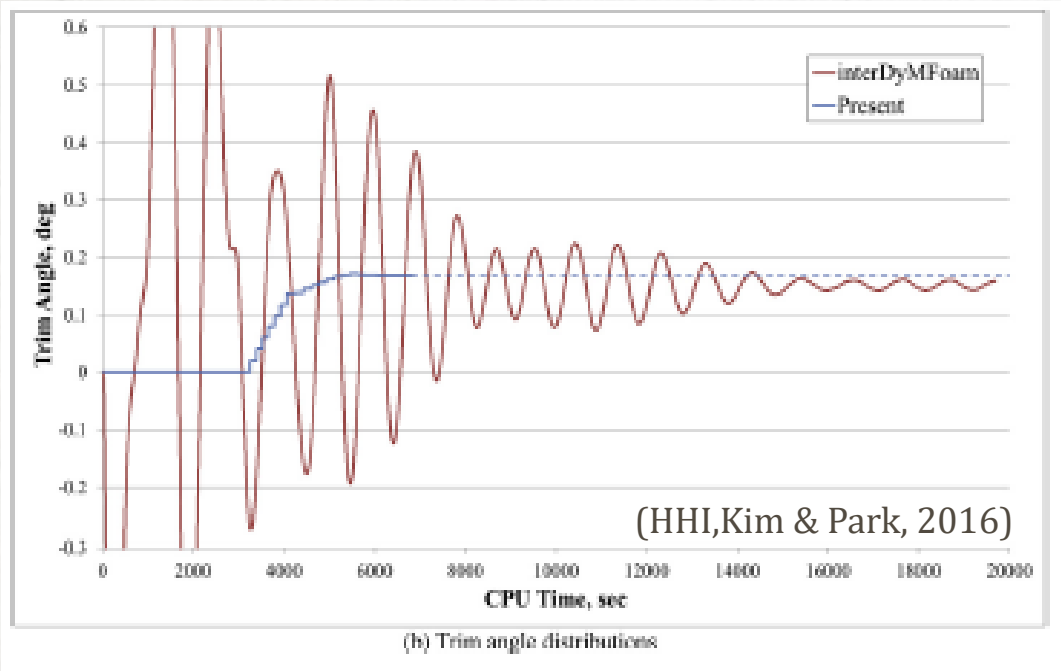
동아대학교 DONG-A UNIVERSITY

김연주

목차

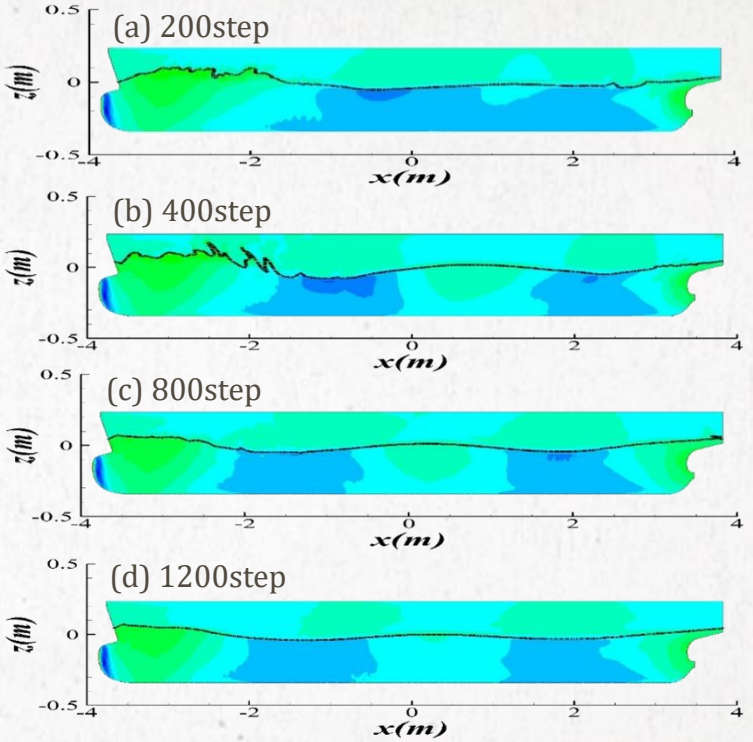
1. 연구 배경

정수 중 시간에 따른 선박의 자세 변화



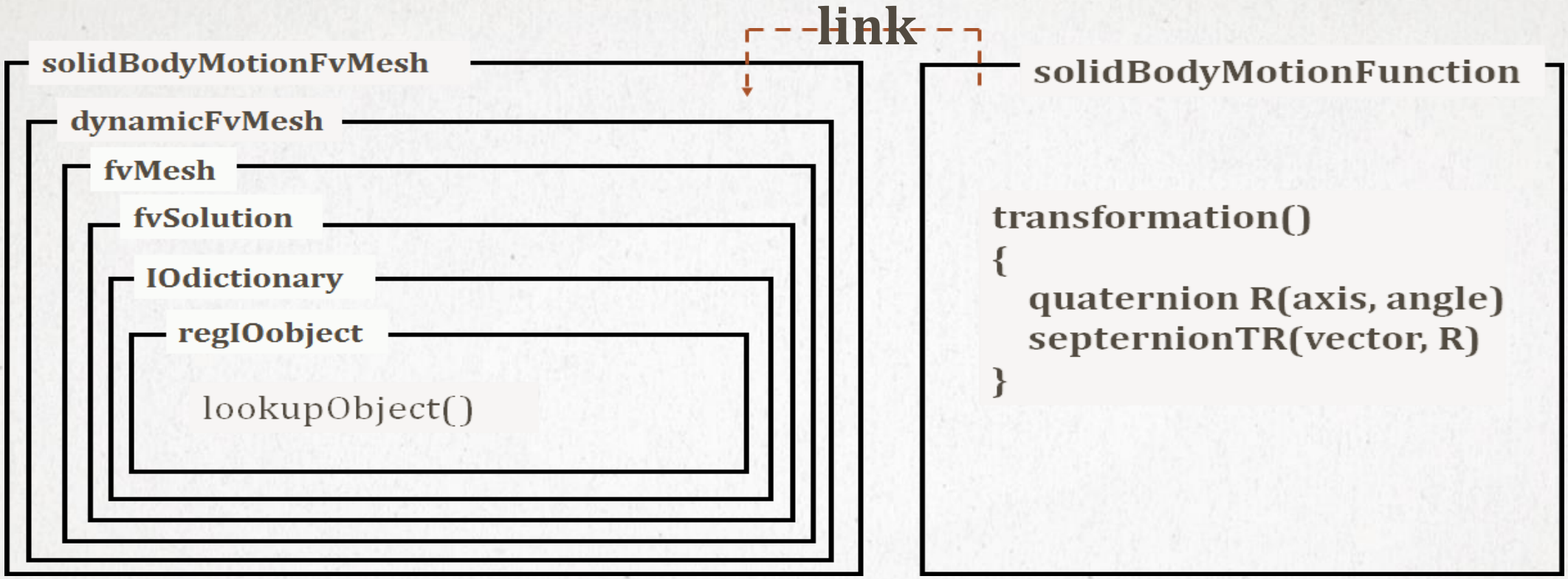
(b) Trim angle distributions

Time step별 선측 압력분포와 수면파



- interDyMFoam을 이용하여 선박의 평형상태까지 도달하는데 많은 시간 소요
- 선박의 동적 자세를 효율적으로 찾을 수 있는 interDyMDAUFoam 개발

2. interDyMFoam



- solidBodyMotion을 이용한 격자 변환은 angle, R이 주어진 상태에서만 가능
- solidBodyMotionFunction은 마찰력,압력을 이용한 격자 변환 불가능

2. interDyMFoam

solidBodyMotionFunction

RotationMotion

quaternion R(w, v)

$w = \cos(0.5 \times \text{angle})$

$v = \sin(0.5 \times \text{angle}) \div |\text{axis}| \times \text{axis}$

septernion TR(t,r)

$t = -v \cdot x_0 \times v + w \times (-w \times x_0 + v \times x_0) + (-w \times x_0 + v \times x_0) \times v + x_0$

$r = \text{quaternion}(\frac{w}{\sqrt{|w|^2+|v|^2}}, \frac{v}{\sqrt{|w|^2+|v|^2}})$

axisRotationMotion

quaternion R(w, v)

$w = \cos(0.5 \times |\text{omega}|)$

$v = \sin(0.5 \times |\text{omega}|) \div |\frac{\text{omega}}{|\text{omega}|}| \times \frac{\text{omega}}{|\text{omega}|}$

septernion TR(t,r)

$t = -v \cdot x_0 \times v + w \times (-w \times x_0 + v \times x_0) + (-w \times x_0 + v \times x_0) \times v + x_0$

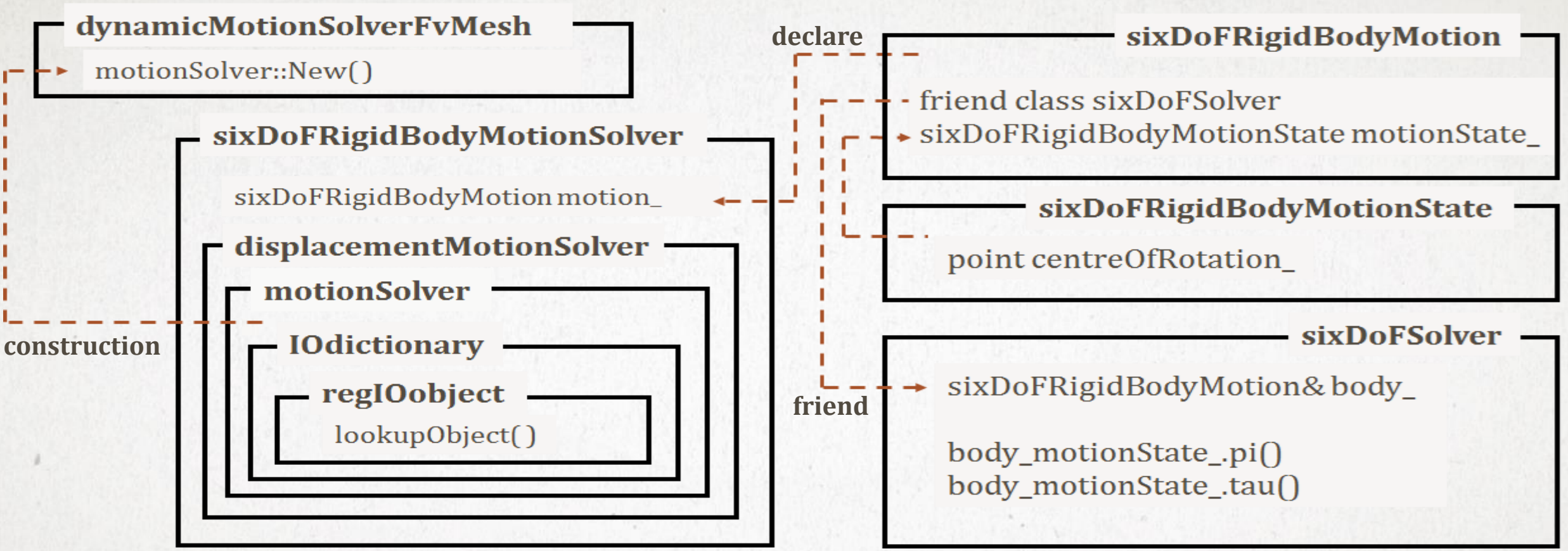
$r = \text{quaternion}(\frac{w}{\sqrt{|w|^2+|v|^2}}, \frac{v}{\sqrt{|w|^2+|v|^2}})$

translation & rotation

quaternion R
= quaternion(axis, angle)

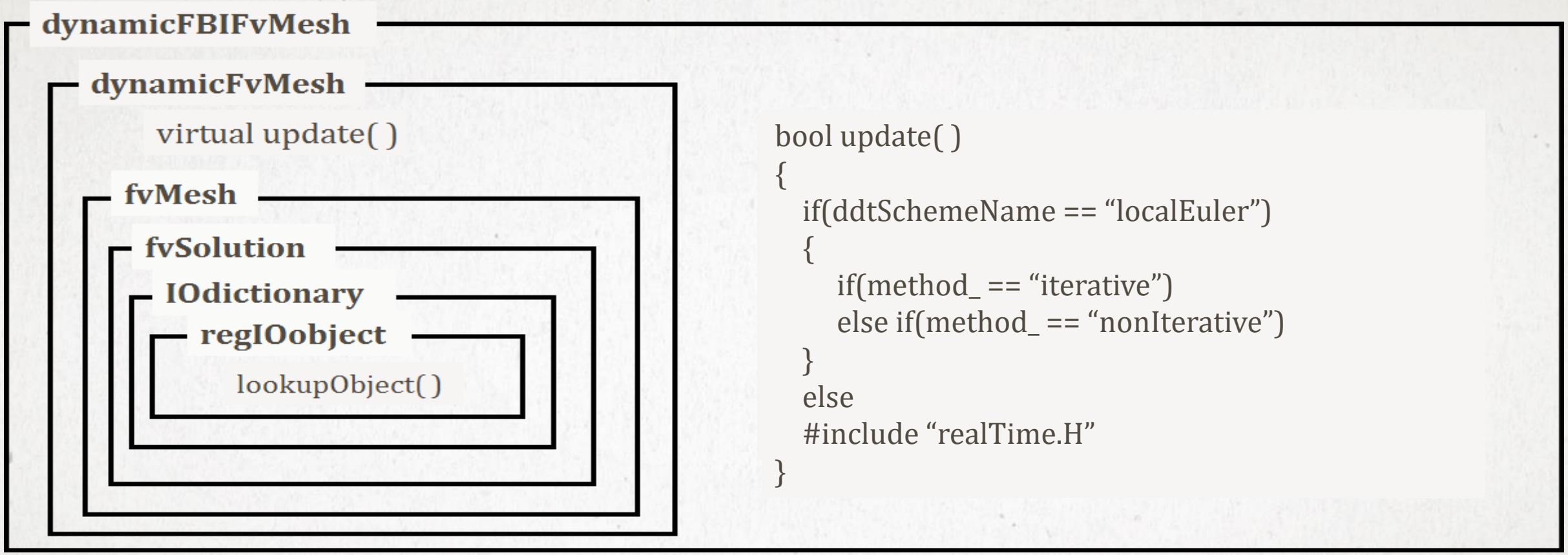
septernion TR
= septernion(-CofR-sinkage) × **R** × septernion(CofR)

2. interDyMFoam



- sixDoFRigidBodyMotionSolver는 마찰력,압력을 이용한 격자 변환 가능
- 선체 주변 격자 모양 변함

3. interDyMDAUFoam




```
bool update( )
{
    if(ddtSchemeName == “localEuler”)
    {
        if(method_ == “iterative”)
        else if(method_ == “nonIterative”)
    }
    else
    #include “realTime.H”
}
```

- interDyMDAUFoam은 localEuler의 시간 차분법에 한해 코드 개발

3. interDyMDAUFoam

```
if(method_==“nonIterative”)
{
   if(timeIndex == 1)
   {
      waveElevation.update( );
      sinkage, angle 결정
      moved = this->move(sinkage,angle,axis,CoB_);
   }
   else if(timeIndex == Trelease_)
   else if(timeIndex == Trelease_+1000)
   else if(timeIndex == Trelease_+2000)
   {
      waveElevation.update( );
      T_r+a_1S_k+b_1F_z=c_1
      T_r+a_2S_k+b_2M_y=c_2
      sinkage,angle 최종 결정
      moved = this->move(sinkage,angle,axis,CoB_);
}
```

```
bool move(sinkage, angle, axis, CofR)
{
   quaternion R(axis, angle.y( ));
   septernion TR(septernion(-CofR-sinkage)*R*septernion(CofR));
   this->movePoints(transformPoints(TR,this->points()));
}
```

```
if(moved) VOFreset(waveElevation, 0)
```

```
bool VOFreset(sampledIsoSurface& waveElevation)
{
   if(keyword == “none”)
   else if(keyword == “farField”)
   else if(keyword == “previousWave”)
   else if(keyword == “zeroLevel”)
}
```

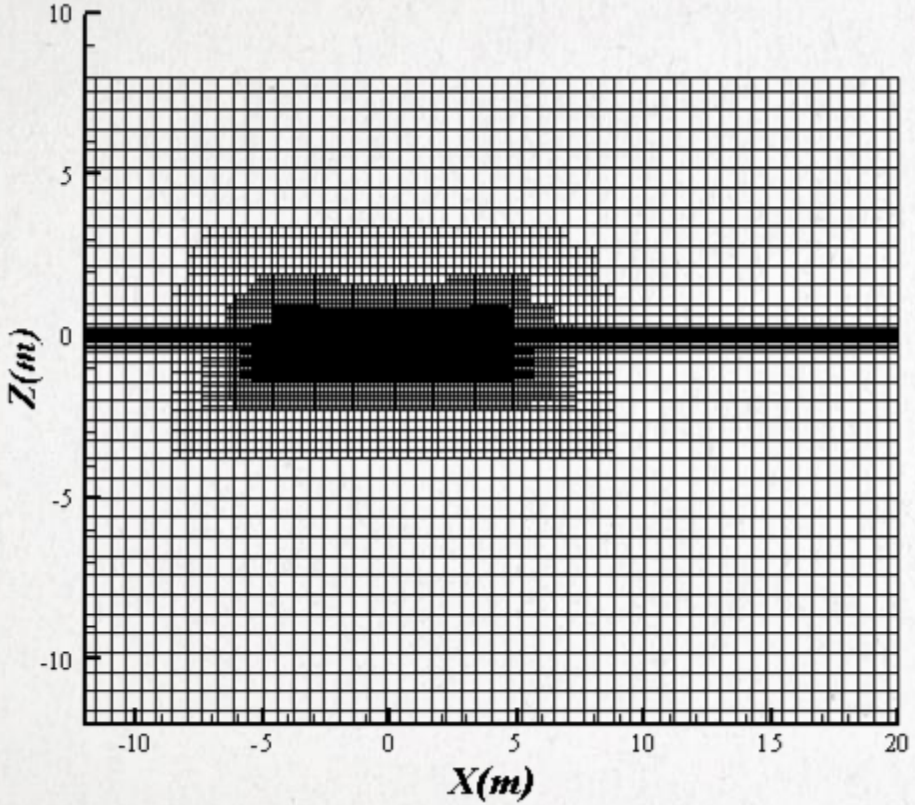
3. interDyMDAUFoam

VOFreset

none
farField
previousWave
zeroLevel

```
if(keyword == "previousWave")
{
    forAll(alphaWater,CellI)
    {
        if(keyword == "zeroLevel" || init)
        {
            if(keyword=="none")
            {
                forAll(alphaWater,CellI)
                {
                    reset = false;
                    if(this->cellCentres()[CellI].z() > 0) alphaWater[CellI] = 0.0;
                    else alphaWater[CellI] = 1.0;
                }
            }
            else if(keyword=="farField")
            {
                forAll(alphaWater,CellI)
                {
                    if(this->cellCentres()[CellI].z() > 0) alphaWater[CellI] = 0.0;
                    else alphaWater[CellI] = 1.0;
                }
            }
            else if(keyword=="previousWave")
            {
                forAll(waveElevation.points(),surfI)
                {
                    if(sqrt(sqr(this->cellCentres()[CellI].x())+sqr(this->cellCentres()[CellI].y()))>4)
                    {
                        const point& sP = waveElevation.points()[surfI];
                        if(sqrt((cC.x()-sP.x())*(cC.x()-sP.x())+(cC.y()-sP.y())*(cC.y()-sP.y()))<dist)
                        {
                            dist=sqrt((cC.x()-sP.x())*(cC.x()-sP.x())+(cC.y()-sP.y())*(cC.y()-sP.y()));
                            waveHeight=sP.z();
                        }
                    }
                }
                alphaWater[CellI]=0.0;
                if(cC.z()<waveHeight) alphaWater[CellI]=1.0;
            }
        }
    }
}
```

3. interDyMDAUFoam

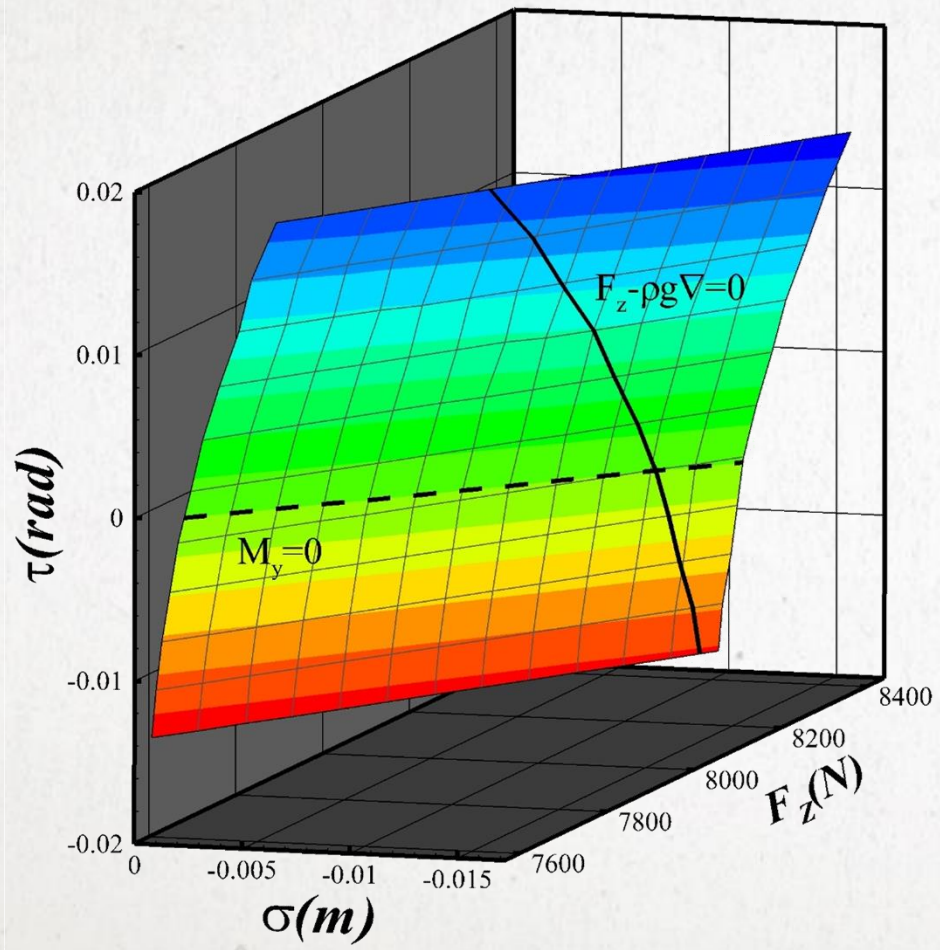


Details of KCS

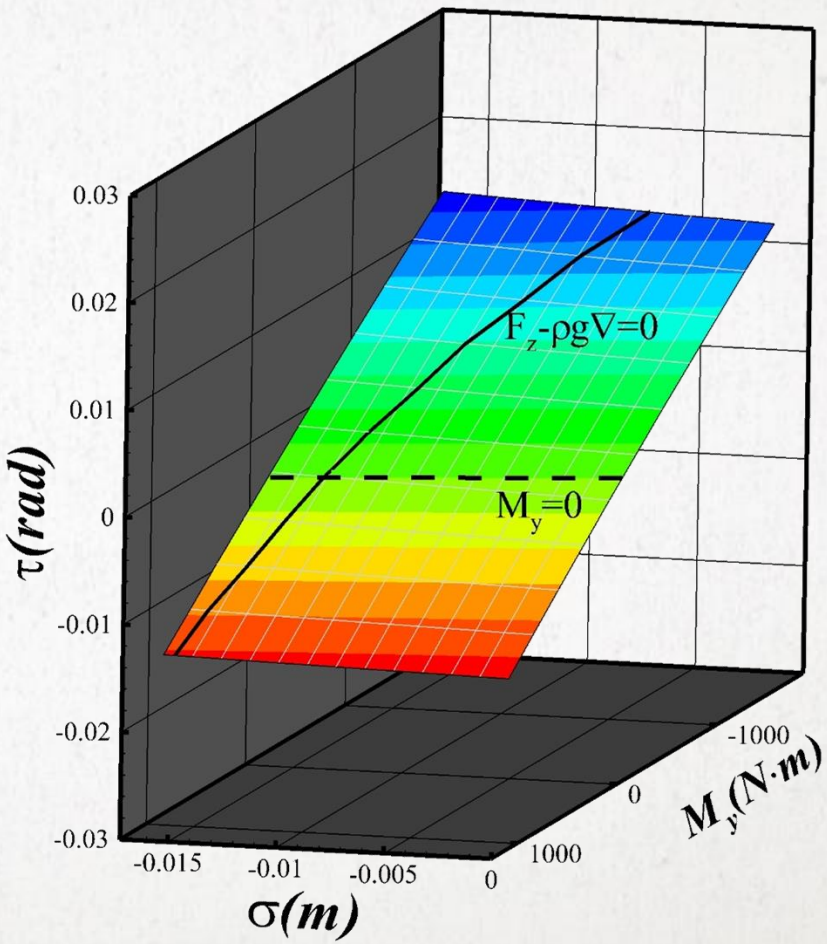
Lpp	230m
Design draught	10.8m
Displacement	52030m^3
Fn	0.260
Scale ratio	31.6
격자 수	1.5×10^6

3. interDyMDAUFoam

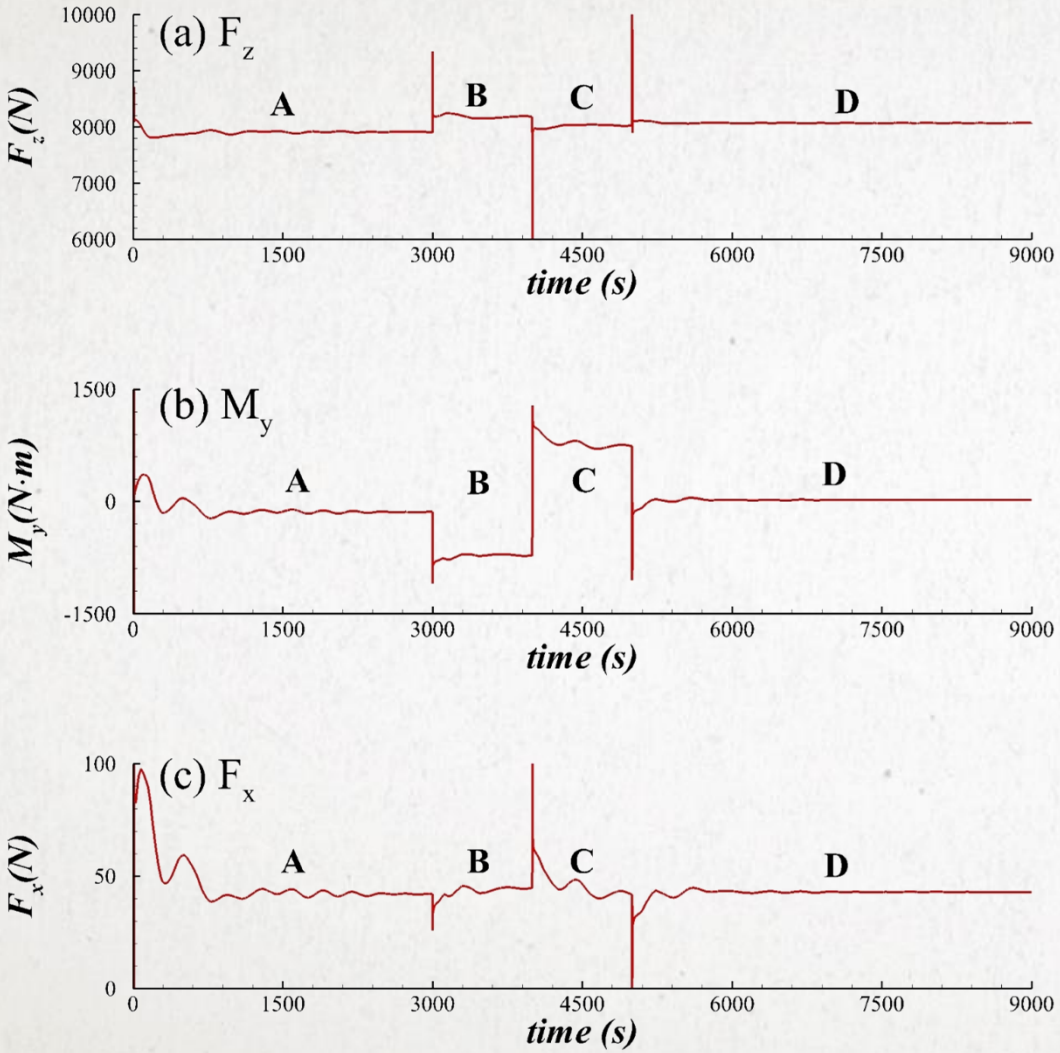
$$T_r+a_1S_k+b_1F_z=c_1$$

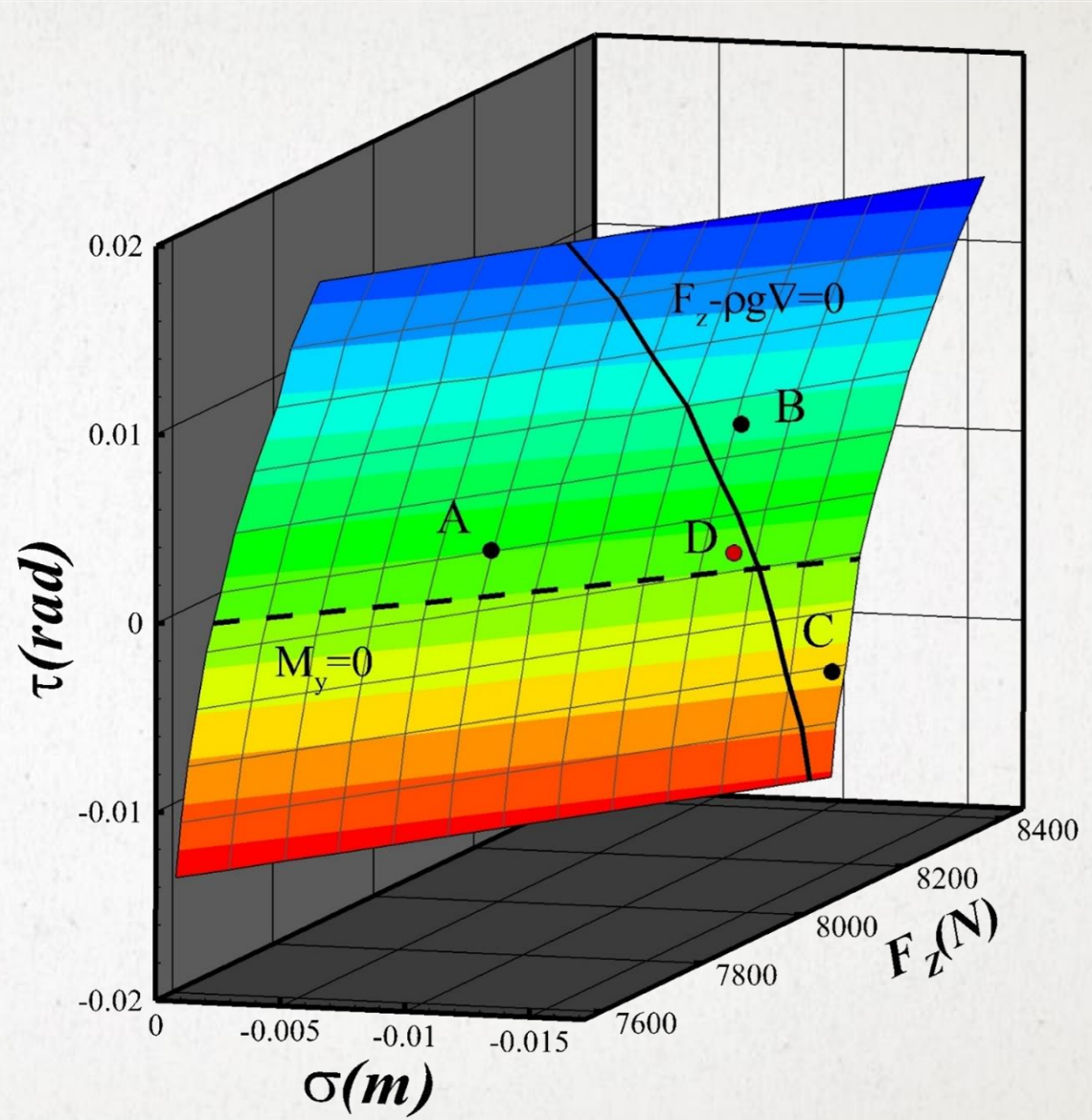


$$T_r+a_2S_k+b_2M_y=c_2$$



3. interDyMDAUFoam





4. 결론

전체 code구조 이해 어려움

기능 모듈 설계 어려움

class와 member function의 사용 이해 부족

감사합니다.